

# 5<sup>€</sup> AMIGA

Markt & Technik

DM 16,-  
ÖS 120,- / Sfr 16,-  
Lit 16000 / hfl 21,- / dkr 75,-

## STONEDER

### Das beste aus aller Welt

- Spitzen-Programme fast umsonst

### Das Grafikprojekt

- Schritt für Schritt zur perfekten 3D-Animation

**PD-Diskette zum Aktionspreis**

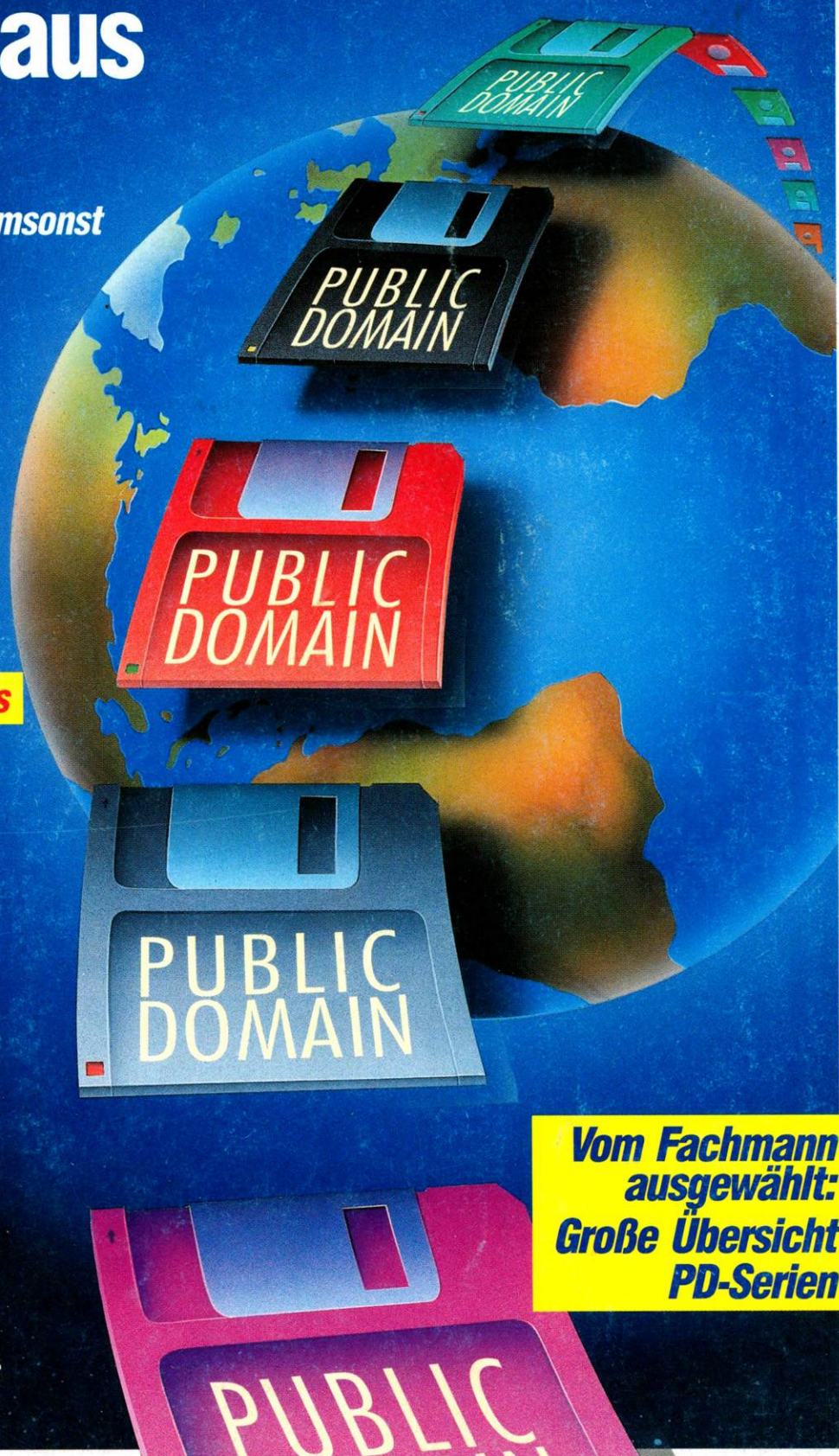
### Utilities

- Turbo-Backup: komfortabel kopieren
- VirusX: keine Chance für Viren

### Workshops

- Modula: Der Weg zum Erfolg
- DME: Brandaktuelle Version des programmierbaren Editors

**Vom Fachmann ausgewählt:  
Große Übersicht  
PD-Serien**





# AMIGA

Markt&Technik

## SOFTWARE

SOFTWARE  
**EXTRA**

**NEU: Amiga Extra Nr. 6:  
AudioWorx**

Ihr privates Sampling-Studio. Das Digitalisieren von Sounds, sei es vom CD-Spieler, Tonband usw. ist mit AudioWorx ein Kinderspiel. Die Möglichkeit, aufgenommene Tracks zu verändern, sind vielfältig: Mit Hilfe zahlreicher Funktionen können Sie interessante Effekte erzeugen.

**Bestell-Nr. 38748**  
**DM 49,-\*** (sFr 45,-\*/öS 490,-\*)

**NEU: Amiga Extra Nr. 9: Sonix**  
Licht aus - woommm. Den Amiga an den Hifi-Verstärker. Sonix in den Amiga. Monitor-Spot an - yeah! Die Klangdimensionen des Amiga haben Grenzen, aber die sind weit gesteckt. Wie weit, das erleben Sie, wenn Sie die Musik auf dieser Diskette abspielen lassen. 13 Stücke sind enthalten.

**Bestell-Nr. 38753**  
**DM 49,-\*** (sFr 45,-\*/öS 490,-\*)

**Amiga Extra Nr. 5: Spiele**

Spannende Unterhaltung mit vier Super-Spielen! Breaking out: Actionspiel mit schneller Grafik und tollem Sound. Decoder: Verwandeln Sie Ihren Amiga in eine Morsestation. Megamind: Anregende Unterhaltung für kluge Köpfe. Wikinger II: Spannendes Strategiespiel, angesiedelt im Mittelalter.

**Bestell-Nr. 38752**  
**DM 49,-\*** (sFr 44,-\*/öS 490,-\*)

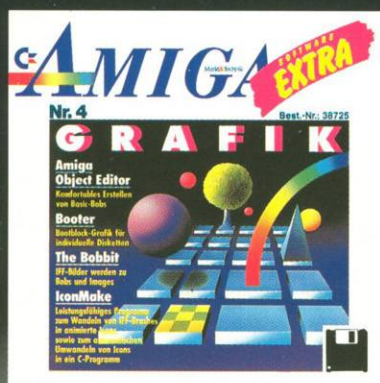


**Amiga Extra Nr. 1: Grafik**  
CADos 3-D: Konstruktion und Rotation dreidimensionaler Körper. Funktionsplotter: Grafische Auswertung komplexer Funktionen. Fractal Construction Kit: Bilder einer fremden Welt.

**Bestell-Nr. 38708**  
**DM 49,-\*** (sFr 44,-\*/öS 490,-\*)

**Amiga Extra Nr. 4: Grafik**  
Amiga Object Editor: The Bobbit. Iconmake. Booter.

**Bestell-Nr. 38725**  
**DM 49,-\*** (sFr 44,-\*/öS 490,-\*)



**Amiga Extra Nr. 3: Spiele**

Bliff: Eine ausgeklügelte Variante des Billards. Quadriga: Ein Spiel für Denker, angelehnt an »Vier Gewinnt«. Wikinger I: Ein Strategiespiel, angesiedelt im 10. Jahrhundert.

**Bestell-Nr. 38724**  
**DM 49,-\*** (sFr 44,-\*/öS 490,-\*)

**Amiga Extra Nr. 2: Disk Utilities**

DiskEd, Select Copy, DCopyII, Check, Bootgirl Plus.

**Bestell-Nr. 38726**  
**DM 49,-\*** (sFr 45,-\*/öS 490,-\*)

\* Unverbindliche Preisempfehlung

Markt&Technik-Produkte erhalten Sie im  
Computerfachgeschäft, in den Fachabteilungen  
der Warenhäuser, im Versandhandel  
und in Ihrer Buchhandlung.

  
**Markt&Technik**  
Zeitschriften · Bücher  
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2,  
8013 Haar bei München, Telefon (089) 4613-0.

SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656.

ÖSTERREICH: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0;

Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526;

Ueberreuter Media Verlagsges.mBH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.



Fragen Sie Ihren  
Fachhändler nach unserem  
kostenlosen Gesamtverzeichnis  
mit über 500 aktuellen  
Computerbüchern und Software.  
Oder fordern Sie es direkt  
beim Verlag an!



## Die Spreu vom Weizen getrennt

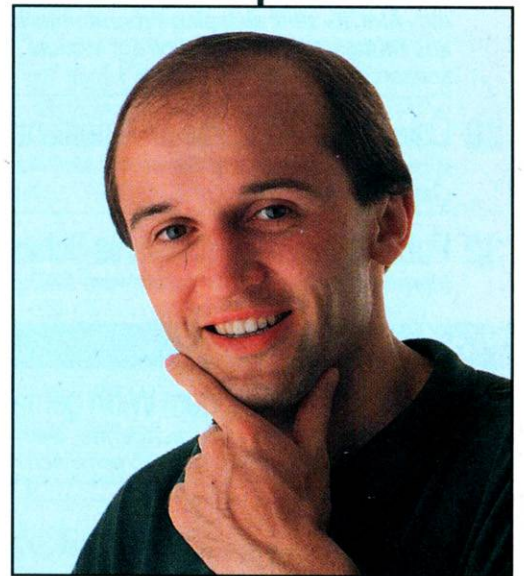
● Public Domain ist ein unerschöpfliches Software-Reservoir. Über 2000 Disketten umfaßt inzwischen der PD-Pool. Allein die populären »Fish«-Disks sind mit Redaktionsschluß dieser Ausgabe auf die stolze Zahl von 200 angewachsen.

● Neben dieser Reihe existieren eigenständige Serien, die sich teilweise Spezialgebieten verschrieben haben oder die eine etwas andere Auswahl bieten. Wer kann diese wahre Sintflut noch überblicken? Wir würden uns das nicht anmaßen. Dieses Heft ist keineswegs als vollständiger »PD-Katalog« gedacht, in dem Sie alle interessanten Gebiete nachschlagen können.

● Wir möchten Ihnen eine Hilfe geben, um die Spreu vom Weizen zu trennen. Sie finden auf den folgenden Seiten eine Auswahl der PD-Programme, die unserer Meinung nach in keinem Diskettenschrank fehlen dürfen. Natürlich kann diese Auswahl nie vollständig sein. Jeder benötigt für seine speziellen Anwendungen ein anderes Programm. Wir geben Ihnen Anregungen, indem wir einige Programme vorstellen, die sich bei uns im »Dauerbetrieb« bewährt haben. Wir hätten diese gerne gesammelt auf die Programmservice-Diskette kopiert.

Dies war – trotz Packen fast aller Files – nicht möglich. Sie finden aber alle Programme, die im Inhalt mit einem Diskettensymbol versehen sind, auf dieser Diskette.

● Neben täglich benötigten Utilities finden Sie einige Anwendungen und jede Menge Tools für Programmierer. Aus letzteren haben sich die Anfänge der PD-Szene entwickelt. Inzwischen gibt es eine bunte Vielfalt, angefangen von einfachen Routinen über kleine aber ständig benötigte Utilities und Spezialanwendungen. Zusätzlich findet der PD-Fan Demo-Versionen kommerzieller Programme, die den Anwender auf diese aufmerksam machen sollen. Bei dieser Fülle an Programmen verliert man leicht die Übersicht. Schnell ist viel Geld investiert in einen Berg von Disketten, die nie wieder aus dem Schrank geholt werden. Mit den Programmen aus diesem Heft wird Ihr »Fish«-Zug keine Tauchfahrt ins Ungewisse.



Ihr  
Klaus Sonnenleiter  
(Redakteur)

*K. Sonnenleiter*



## Grafikprojekt

### 6 Der Weg zur 3D-Animation

*Dreidimensionale Animationen vom Feinsten — mit »DBW-Render«*



## Szene

### 20 PD für die Welt

*Mit Public Domain hat sich eine faszinierende Idee über die ganze Welt verbreitet*

### 22 Bunte PD-Szene in Deutschland

*Wir sprachen mit einem der ersten deutschen PD-Anbieter*

### 24 Amiga läuft Amok

*Mit »AMOK« stellt sich eine Programmiererguppe aus Stuttgart vor, die sich voll auf Modula konzentriert*

### 28 Österreich: PD-Szene im Alpenland

*Eine kleine, aber aktive Szene hat sich inzwischen in Österreich etabliert*

### 32 Public Domain — Blickpunkt Schweiz

*Interview mit Beppo Ctvrtnicek vom CAC Zürich*

## Workshop

### 35 Die Rechnung mit dem Wirth gemacht

*Niklaus Wirth gilt als »Sprachgenie«. Sein neuester Streich — Modula — ist auf PD erhältlich.*

### 61 Das Chamäleon

*Der »DME« ist ein Editor, den Sie optimal der jeweiligen Umgebung anpassen*



### 68 Der Blick ins Innere

*»XOper« öffnet Ihnen die Augen. Was Sie schon immer über Multitaskingsysteme wissen wollten, aber noch nie herausfanden.*



## Grundlagen

### 72 Public Domain — was steckt dahinter?

*Die Geschichte einer begeisternden Idee*

### 74 Frisieren Sie die Workbench

*Ersetzen Sie den »Datenschrott« Ihrer Workbench mit leistungsfähigen PD-Varianten*

### 76 CLI-Starthilfe

*Das CLI ist für den Workbench-verwöhnten Anwender die reine Provokation. Wir helfen über Startschwierigkeiten hinweg.*

### 80 Große Public Domain-Übersicht

*Den gesamten Bereich der PD-Software zu überblicken, ist schlicht unmöglich. In dieser Übersicht finden Sie alle bisher erhältlichen »Fish«- und »RPD«-Disketten*



### 120 In letzter Minute

*Kurz vor Redaktionsschluß erreichten uns die neuesten Exemplare der Fish-Disks*

DBW-Render kitzelt das äußerste aus den Grafikeigenschaften des Amiga. Lassen Sie sich von Ray-Tracing mit 4096 Farben gleichzeitig beeindrucken.

**SEITE 6**



## Anwendungen

### 98 Kraftprotz mit rauher Oberfläche

*Tabellenkalkulationen gehören zur hohen Kunst des Programmierens. Mit der sehr leistungsfähigen Variante »AnalytiCalc« sollten Sie rechnen.*

### 102 Der Datenspediteur

*Das Terminalprogramm »AZComm« befördert Ihre Daten per Telefon an den Bestimmungsort*

### 105 Der Ton macht die Musik

*Der Amiga besitzt fantastische Sound-Fähigkeiten. Mit den hier vorgestellten PD-Programmen nutzen Sie diese voll aus.*

## Programmieren

### 109 Künstliche Intelligenz

*Kaum ein Begriff wird häufiger mißverstanden als die sogenannte »Künstliche Intelligenz«. Überzeugen Sie sich am Beispiel von »XLisp«, daß auch in diesem Bereich nur mit Wasser gekocht wird.*

### 113 Der Public Domain-Assembler

*Der A 68 K ist ein »Schnupper«-Assembler. Auch das Abtippen der meisten Listings ist problemlos möglich.*

### 116 PDC — ein totgeborenes Kind?

*C-Compiler sind teuer und sehr umfangreich. Kann dieses PD-Programm ein kommerzielles Paket ersetzen?*

### 117 Von Wanzen und ähnlichem Ungeziefer ...

*Programmierer fürchten sie wie der Teufel das Weihwasser: die verhängnisvollen Bugs. Entsprechend hoch ist das Ansehen, das sogenannte »Debugging«-Hilfen genießen.*

## Bücher

### 144 Aus unserem Bücher-Regal

*Zum Thema Public Domain gibt es einige interessante Bücher. Die Standardwerke zu diesem Bereich haben wir für Sie unter die Lupe genommen.*





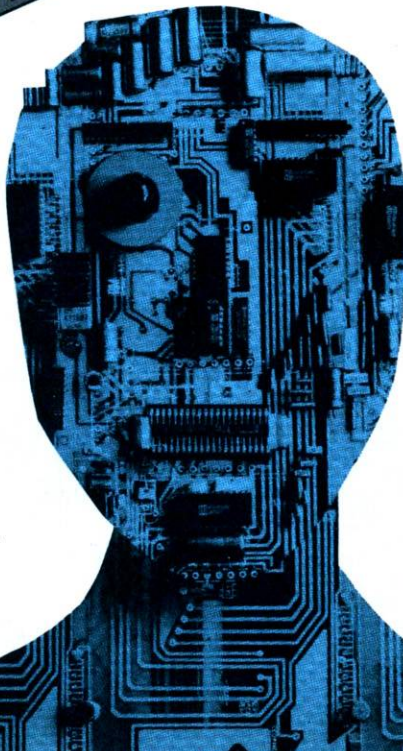
VirusX säubert Ihre Disketten von jedem bisher bekannten Virus. Schaffen Sie sich dieses »Datenunkraut« vom Hals.

**SEITE 141**



Lisp ist die Sprache der Künstlichen Intelligenz. Lassen Sie sich von faszinierender Sprache beeindruckten.

**SEITE 109**



## Utilities

- 125 Jetzt können Sie einpacken** +  
 Mit »Zoo« schaffen Sie Platz auf übervollen Disketten. Noch wichtiger: Mit diesem Programm wird die Programmservice-Diskette entpackt.
- 
- 127 Glasnost für Ihre Disketten** +  
 »DiskX« bringt die Transparenz für AmigaDOS, die Sie schon immer gewünscht haben
- 
- 129 Kopieren? Aber Sicher!** +  
 Disketten kopieren in weniger als zwei Minuten? Kein Problem mit »Turbobackup«.
- 
- 131 Amiga mit Überschall** +  
 Mit »Mach II« richten Sie Ihre Bedienungsoberfläche voll nach Ihrem Geschmack ein
- 
- 135 Rettung für Ihre Daten**  
 Wer keine Festplatte hat, dem bleibt die Qual der Wahl: Zwischen der sicheren Diskette und der schnellen RAM-Disk. Beide Vorteile vereint die resetfeste ASDG-RAM-Disk.
- 
- 137 Hintergründiges im CLI**  
 Mit »run« gestartete Programme verhindern das Schließen des CLI-Fensters. »Runback« beseitigt dieses Manko
- 
- 138 Sicher ist sicher** +  
 Wer wichtige Daten auf seiner Festplatte hat, sollte sie von Zeit zu Zeit auf Disketten sichern. »MRBackup« ist komfortabel und zuverlässig.
- 
- 146 Flottes Auskunftsbüro**  
 Eine Uhr und noch viel mehr haben Sie mit diesem Utility ständig parat. Lassen Sie sich überraschen.

## Viren

- 141 Der stille Wächter** +  
 Viren sind die widerwärtigste Art von Programmen. »VirusX« verhindert, daß Sie der Infektion schutzlos ausgeliefert sind.
- 
- 142 Kleine Viruskunde**  
 Mit dem vermehrten Auftreten von Viren wird es – leider – immer wichtiger, über diese Abart von Programmen informiert zu sein. Unsere kleine Viruskunde enthält alle derzeit bekannten Formen.

## Sonstiges

- 3 Editorial**
- 
- 139 Mitmachkarte**
- 
- 145 Programmservice**
- 
- 146 Impressum**
- 

Alle Programme aus Artikeln mit einem +-Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft





**V**or drei Jahren brachte der Amerikaner David B. Wecker die erste Version des DBW-Render als Shareware auf den Markt. Damals gab es noch Einschränkungen, wie fehlende PAL-Auflösung, einen mäßigen Objektparser und mehrere versteckte Fehler (bugs), die das Programm zum Absturz brachten. Vor einem Jahr brachte der Autor die völlig überarbeitete und erweiterte Version 2.0 in Umlauf. Mit dieser lassen sich Bilder in jeder beliebigen Auflösung berechnen und bis zu 16 Millionen Farben für den voraussichtlich noch in diesem Jahr erhältlichen Framebuffer

erzeugen. In Hinblick auf Computeranimationen und der Erstellung aufwendigerer Objekte als Kugeln, Dreiecke und Quadrate, wurde diese Version mit einer speziellen Eingabesprache ausgestattet.

Leider werden in der mitgelieferten Anleitung nur die Parameter zu den Befehlen beschrieben. Wichtige Informationen, die jedem Anfänger willkommen wären, werden verschwiegen. Außerdem ist die Anleitung mit inhaltlichen Fehlern behaftet. Damit ist zeitraubendes Herumprobieren angesagt, da die Berechnung, wie es für einen Ray-Tracer üblich ist, mehrere

Stunden oder sogar Tage dauert. Bis alle Objekte so aussehen, wie man es sich vorstellt, die Szene richtig ausgeleuchtet ist und Farben und Texturen nicht wie mit einem Bleichmittel gewaschene Wäsche aussehen, vergehen somit mehrere Tage, wenn nicht Wochen. Viele Anwender schmeißen das Programm dann wütend in die Ecke und lassen es verstauben. Sie erfahren nicht, wie leistungsfähig der DBW-Render eigentlich ist. Es war ja nur ein Public Domain Programm...  
Damit es Ihnen nicht wie beschrieben ergeht, stellen wir unser Grafikprojekt vor, in dem Sie alle wichtigen

# Der Weg zur 3D-Animation ■ Der Weg zur



**Hervorragende Grafiken mit 4096 Farben, realistische Spiegelungen und 3D-Animation - das alles kann der »DBW Render«. Das Ray-Tracing-Programm steht den kommerziellen Programmen Sculpt 3D und Turbo Silver in nichts nach, ja es übertrifft sie in mancher Hinsicht sogar. Lernen Sie das PD-Programm Schritt für Schritt kennen.**

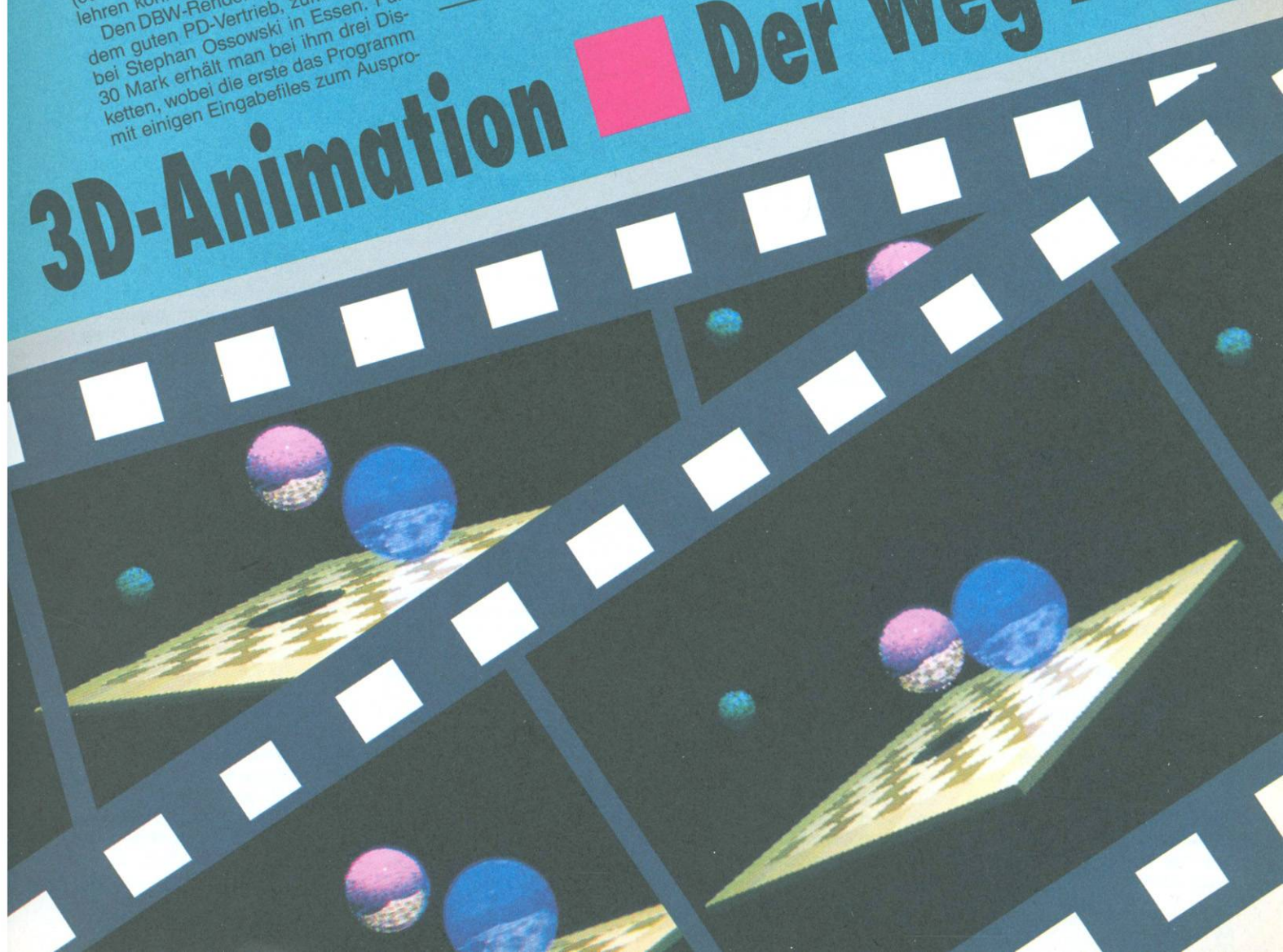
Funktionen kennenlernen werden. Sie erweitern so Schritt für Schritt Ihr Wissen um den Ray-Tracer und erhalten Anregungen zu weiterführenden Experimenten. Als besonderes Bonbon zeigen wir Ihnen am Schluß, wie man sogar eigene Animationen erstellen kann, die das vielgelobte Sculpt 3D (oder auch Sculpt 4D) das Fürchten lehren könnten.

Den DBW-Render erhält man bei jedem guten PD-Vertrieb, zum Beispiel bei Stephan Ossowski in Essen. Für 30 Mark erhält man bei ihm drei Disketten, wobei die erste das Programm mit einigen Eingabefiles zum Auspro-

bieren enthält, die zweite verschiedene Bilder von den Programmautoren und die dritte die deutsche Anleitung mit mehreren Bildern von Marco Vitolini und Christian Zerbes inklusive der Eingabefiles. Wie fast alle PD-Disketten sind keine der drei bootfähigen. Sie sind also zuerst eine Arbeitsdiskette an. Dazu kopieren Sie eine Original-Workbench und löschen anschließend auf der Kopie alle Fonts, Demos und sonstige nicht mehr gebrauchten Files. Anschließend sind von der Programmdiskette die Files »Ray«, »Ray0«, »Ray1«, »Ray2«, »Ray3«, »Ray.key« und »Raykey« ins c-

Von Marco Vitolini-Naldini

## 3D-Animation ■ Der Weg zur 3D





Directory unserer Programm- diskette zu kopieren. Wer will, kann sich noch allerlei Zusätze wie Conman, Screenblanker oder eine Uhr einbinden. Dem Komfort sind beim Amiga bekanntlich kaum Grenzen gesetzt. Wir empfehlen eine Installation, die für zwei Laufwerke ausgelegt ist. Das hat den Vorteil, daß man ohne Angst vor dem »Disk is Full«-Requester arbeiten kann, da die Rechen- daten eines Bildes je nach Auf- lösung über 200 KByte lang werden können. Die Startup- Sequence kann auch zusam- mengekürzt werden. Lediglich muß mit dem Befehl »Stack 30000« ein genügend großer Stapelspeicher geschaffen werden, da DBW-Render eine

DBW-Render eine Paßwortab- frage hinein. Und dazu noch eine ganz ausgefuchste. Es handelt sich hierbei um eine zeitbe- zogene Abfrage, die, nachdem eine gewisse Zeitspanne abge- laufen ist, jeden Umgang mit dem Programm verwehrt. Bei einem Public Domain Pro-

## Paßwortabfrage im PD-Programm

gramm ist das völlig unver- ständlich. Viele Amiga-Anwen- der besitzen keine Hardware- Uhr – die Aktivierung des Paß- wortes wird dann eher zufällig gesteuert (je nachdem die Uhr des Amigas nach dem Ein- schalten vom erstgebooteten

hängigen Zutritt zum Ray- Tracer gewährt. Beim folgen- den Booten wird dieses File auf unsere Arbeitsdiskette im Lauf- werk »df1:« kopiert. Es muß sich immer dort befinden, von wo aus man den Ray-Tracer startet. Fangen wir jetzt endlich mit dem interessanten Teil an.

Anhand verschiedener Bil- der, die es zu berechnen gilt, möchten wir Sie Schritt für Schritt in die Eingabesprache des DBW-Render einführen. Beginnen wir mit etwas ganz Einfachem: einer auf dem Bo- den liegenden Kugel.

Jedes Programm dieser Art hat Bilder mit Kugeln und Flä- chen als Demos parat. Nun, Kugeln sind sehr einfach zu be- rechnen und machen doch im-

sprung in der Monitormitte), sind alle X-Werte, die rechts vom Ursprung liegen positiv, alle links davon negativ. Bei den Y-Werten sind alle oberen Punkte positiv, alle unteren negativ. Und nun in die dritte Di- mension – die Z-Achse. Sie ver- läuft »aus dem Monitor hinaus«, also sind alle Punkte, die zwi- schen dem Ursprung und dem Betrachter liegen positiv, die »hinter dem Monitor« befindlichen negativ. Um einfacher im dreidimensionalen Raum den- ken zu können, ist es ratsam, sich die gedachte Kamera, aus der das zu berechnende Bild betrachtet wird, immer auf der Z-Achse vom Positiven zum Negativen schauen zu lassen.

Im Schemabild (Bild 1) wurde die Kamera auf den Punkt  $x=0$ ,  $y=30$  und  $z=100$  gesetzt. Das bedeutet, die Kamera schwebt 30 Einheiten über dem Boden und liegt 100 Einheiten vom Achsenzentrum entfernt. Nun müssen wir noch die Blickrichtung angeben. Diese wird als Vektor bestimmt. Aus der Schulgeometrie wissen Sie vielleicht noch, wie die Vektor- addition funktioniert, im Schemabild ist es aufgezeichnet.

Man zeichnet von der Kamera einen Vektor weg, hängt den nächsten an diesen an und so weiter. Zur Addition aller Vektoren verbindet man nun den Anfangspunkt des ersten Vektors mit dem Endpunkt des letzten. In unserem ersten Beispiel blickt die Kamera leicht nach unten in den Monitor hinein. Aber jetzt bin ich schon bei der Erstellung des Eingabefiles.

Dazu kommen wir aber erst später. Zur Kamerapositionierung möchte ich noch auf zwei Eigenheiten des Programms aufmerksam machen. Der Blickpunkt läßt sich bis zu einer Entfernung von maximal 250 Einheiten setzen. Größere Werte werden ignoriert. Weiterhin läßt sich durch die Länge des Z-Vektors ein Tele- oder Weitwinkelobjektiv simulieren. Die benötigte Länge hängt von der Größe der Szene ab. Bei einer Szene, die für X und Y rund 200 Einheiten einnimmt, sollte ein Z-Vektor von 100 bis 150 genommen werden, wenn die Kamera 200 Einheiten vom Nullpunkt entfernt ist. Genauso wie die Kamera werden auch alle anderen Objekte im Raum beschrieben. Es ist daher sehr ratsam, sich die Szene vorher auf Millimeterpapier aufzuzeichnen, um die Werte sofort zur Hand zu haben. Machen wir uns aber jetzt endlich ans Ein- gabefile.

Schemazeichnung zu Bild 1:

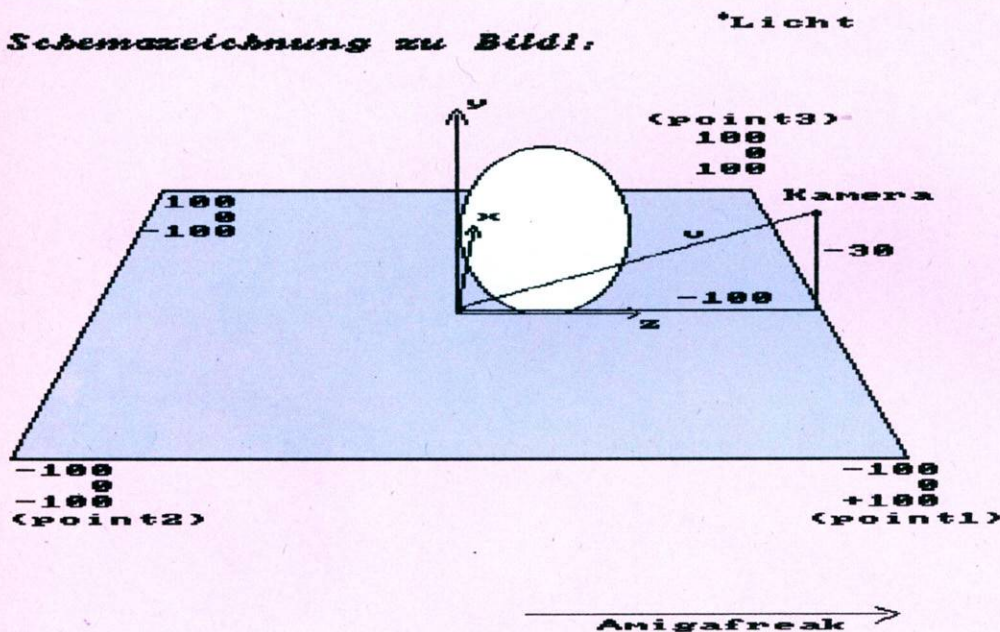


Bild 1. Das Schemabild zeigt das dreidimensionale Koordinatensystem

Menge zu sortieren hat. Des weiteren ist es nötig, bei jedem Booten das File »ray.key« mit

```
copy df0: c/ray.key to
df1:
```

auf die externe Arbeitsdiskette zu kopieren. Schreiben Sie diesen Befehl ebenfalls in Ihre Startup-Sequence. Welche Bewandnis es damit hat, erkläre ich gleich. Ebenso ist die Befehlsfolge »cd df1:« am Schluß der Startup-Sequence ratsam. Wenn alles installiert ist und eine frisch formatierte Diskette im Laufwerk df1: bereitliegt, kann es endlich losgehen. Reset, booten, kurz warten und schon befinden wir uns in unserer zukünftigen Arbeitsumgebung auf »df1:«. Doch halt! Die Programmierer haben sich eine kleine Gemeinheit ausgedacht. Wie bei vielen kommerziellen Programmen, mußte leider in

Programme gesetzt wird). Also machen wir uns erst einmal daran, uns Eintritt in das PD-Programm zu verschaffen. Setzen wir uns erst mit

```
cd df0:c
```

in das C-Directory der Bootdiskette. Nun starten wir das Programm »raykey«, welches ein Wort aus der englischen Anleitung wünscht. Ein Beispiel:

```
LOOK, Line 5, Word 5
```

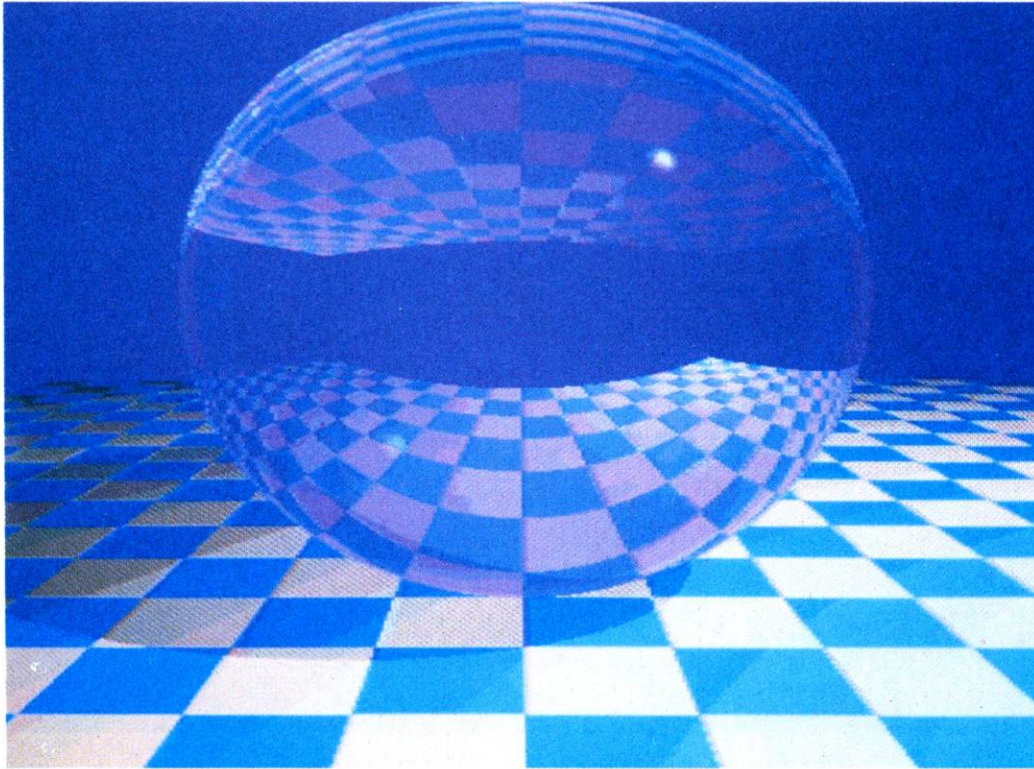
Ein Blick in die Anleitung unter dem Befehl »LOOK« (Seite 18), 5. Zeile, zählen wir bis zum 5. Wort – ein »the«. Wenn Sie nun das vom DBW-Render geforderte Wort eingegeben haben, und das Laufwerk zum Stillstand gekommen ist, müssen Sie den Amiga noch einmal rebooten. Es wurde nämlich gerade ein File namens »ray.key« erzeugt, welches den zeitab-

mer wieder großen Eindruck. Der Programmierer muß sich dabei auch nicht überarbeiten. Damit wir aber gleich beim ersten Bild etwas zum Vorzeigen haben, stellen wir eine Glaskugel her, die auf einem Boden mit Schachbrettmuster liegt. Sehen Sie sich dazu einmal das Schemabild 1 an. Da haben wir einmal die drei Raumachsen x, y und z, die Kugel, den Boden, die Kamera und die Lichtquelle. Prägen Sie sich die Raumachsen gut ein, falls Sie im Mathe-Unterricht gefehlt,

## Ab in die 3. Dimension

oder es wieder vergessen haben. Diese stellen den Schlüssel zu allen Objekterstellungen dar. Vom Anwender auf den Monitor gesehen (Koordinatenur-





**Bild 2. Das erste Projekt – eine halbdurchlässige Kugel auf Schachbrettmuster**

Ein für den Ray-Tracer verständliches Eingabefile muß im ASCII-Code vorliegen und die Endung ».inp« besitzen. Verwenden kann man jeden Editor oder sogar eine Textverarbeitung. Achten Sie aber darauf, daß sich keine speziellen Kommandos des Editors im Textfile befinden. Ein jedes Eingabe- oder auch Inputfile gliedert sich in sechs Teile, die in folgender

## Eingabe-Files im ASCII-Code

Reihenfolge gegliedert sein müssen: die Makros, die Parameter, die Texturdefinitionen, die Lichter, die Oberflächeneigenschaften (auch Attribute genannt) und die Objekte. Verwenden Sie dazu das rohe Inputfile in Listing 1a, dann kann nichts passieren. Auf die Makros gehen wir in Listing 2 ein. Die Parameter beschreiben für das Programm wichtige Grundwerte, wie Kameraposition, Hintergrundfarbe, Bildgröße und einige andere. Die Texturdefinitionen enthalten die Grundvoraussetzungen für die spätere Verwendung in den Oberflächeneigenschaften. Es werden beliebig viele Lichtquellen verschiedener Art definiert. Die Attribute legen Farbe, Transparenz, Textur, Spiegelung und Brechungsindex der einzelnen Objekte fest. Zu guter Letzt werden die Objekte definiert. Im Laufe dieses Workshops werden wir nun nach und nach alle

Befehle besprechen und das nötige Hintergrundwissen dazu liefern. Beginnen wir also mit unserem ersten Projekt (Bild 2). Tippen Sie das dazugehörige Datenfile (Listing 1b) ein und starten es mit dem Befehl

```
Ray bild1.inp
```

Der Pre-Prozessor übersetzt diese in einen für den Ray-Tracer verständlichen Code und speichert es als »bild1.dat«. Danach wird automatisch das Hauptprogramm gestartet, das nun das »dat«-File einliest und die Berechnung startet. Nach jeder gerechneten Bildschirmzeile werden die Daten in ein temporäres File (»name.tmp«) geschrieben. Ist das Bild fertig gerechnet, wird ein weiteres Programm aufgerufen. Dieses liest das temporäre File ein, berechnet die Grundfarbpalette und gibt ein IFF-Bild (»name.ilbm«) aus.

So wie hier abgedruckt, dauert es rund 17 Stunden, bis das Bild fertig ist. Planen Sie deshalb gut, wann Sie den Computer für andere Zwecke benutzen wollen. Das beste ist, die Berechnung vor dem Schlafengehen zu starten, am nächsten Morgen nachzusehen, ob noch alles läuft und dann der gewohnten Tätigkeit am Tage nachzugehen. Am Nachmittag ist das Bild dann im allgemeinen fertig. Nun aber zur Beschreibung der einzelnen Befehle:

```
/* ... */
```

Zwischen diesen beiden Zeichenfolgen lassen sich, genauso wie in der Programmiersprache C, beliebige Kommentare setzen.

## SCAN from x to y rows y1 columns z mode 0 bpp 4

Dieser Befehl beschreibt die Bildgröße und die maximale Farbenanzahl. Ein Amigabild hat im HAM-Interlace ein Pixelformat von 320 x 512 Punkten. Sie können hier auch andere Werte angeben. DBW-Render unterstützt nämlich auch das Overscan-Format mit maximal 352 x 564 Punkten. Das sind rund 20 Prozent mehr Pixel, deshalb dauert es auch dementsprechend länger, das Bild zu berechnen. Bei den Probeberechnungen während der Erstellung eines Bildes verwendet man üblicherweise viel kleinere Formate, damit man sich das Ergebnis nach wenigen Minuten bereits ansehen kann. So steigert man langsam die Bildgröße, bis die Grafik im endgültigen Format perfekt ist.

Die Parameter im einzelnen: x ist die horizontale Bildschirmzeile, bei der die Berechnung beginnen soll; üblicherweise immer Null. Bei welcher Scanline mit der Berechnung aufgehört werden soll, beschreibt der Parameter y. Die Einstellung von x und y eignet sich auch dazu, das Bild nicht in einem, sondern in mehreren Abschnitten zu berechnen. Dabei sollte sich der x-Wert des

```

/*****
 *
 * Rohes Eingabefile *
 *
 *****/

Macros:
  Macro Name
  /* Hier kommen die Macrodefinitionen hinein */
  done Name

parameters:
  scan from 0 to ? rows ? columns ? mode 0 bpp 4
  /* Hier kommen weitere Parameterdefinitionen hinein */

textures:
  /* Hier werden die einzelnen Texturen definiert */

lights:
  /* Hier werden die Lichtquellen eingegeben */

attributes:
  attribute name
  texture ...
  /* Hier werden die Objekteigenschaften definiert */

objects:
  /* Hier kommen Macronamen oder Objektdefinitionen hin */

message "done"

```

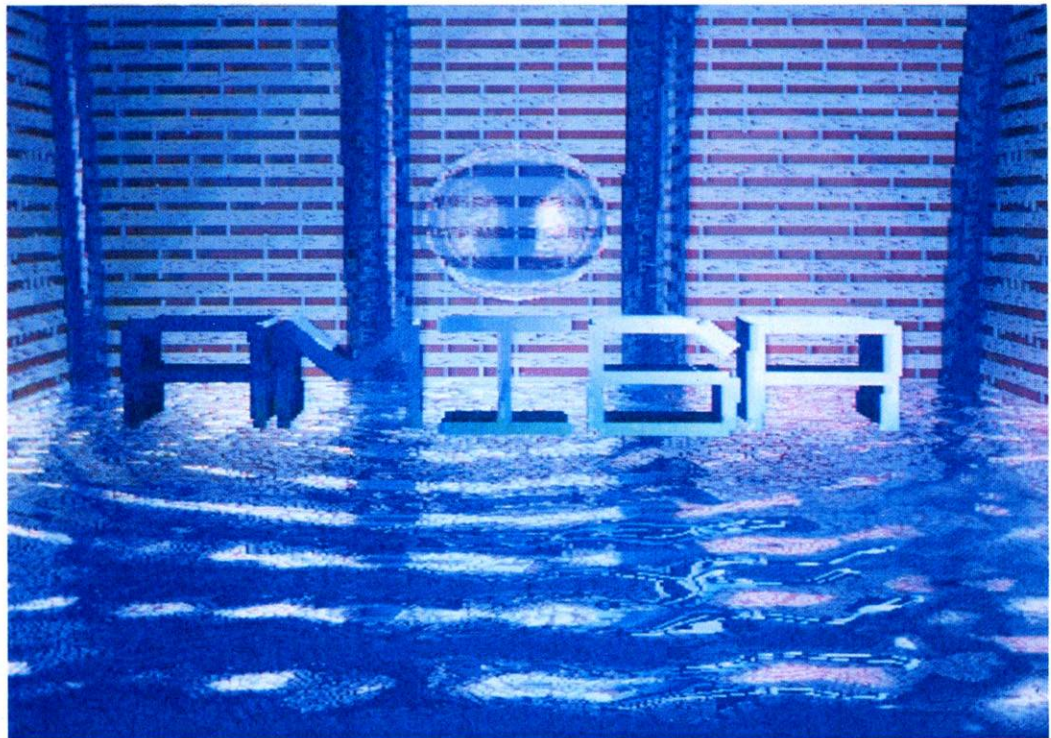
**Listing 1a. Diese Vorgaben erleichtern das Erstellen eines Eingabefiles**



neuen Teils immer um einige Einheiten mit dem y-Wert des alten Teils überlappen. Andernfalls könnten am Schluß schwarze Linien im Bild sein. Dazu noch eine Bemerkung: Vergessen Sie nie, dem nächsten Eingabefile einen anderen Namen zu geben, da sonst die schon berechneten Daten überschrieben werden. Sind alle Teile eines Bildes komplett, werden sie zusammengelenkt.

Die Parameter y1 und z geben das eigentliche Bildformat in Pixeln an. y1 bestimmt die vertikale Auflösung, z die horizontale. Dabei muß z eine durch 8 teilbare Zahl sein. Bei genauerem Hinsehen werden Sie feststellen, daß sich die y- und y1-Parameter in den Eingabefiles nicht decken, was ja bedeutet, daß wir nicht das ganze Bildformat berechnen.

Leider ist das Programm für die NTSC-Norm geschrieben. Diese hat in der vertikalen Auflösung 400 statt 512 Zeilen. Dies zeigt sich oft an einem leeren Bildstreifen im unteren Teil des Bildschirms. Damit Kugeln auch rund erscheinen, muß man das Format für europäische Verhältnisse entsprechend ändern. Addieren Sie daher dem y1-Parameter 28 Prozent des y-Parameters hinzu.



**Bild 2a. Texte, Hintergrund und Wasser: alles null Problemo**

Das Verhältnis z zu y sollte immer 1 zu 1,6 betragen.

Es gibt noch zwei weitere Parameter im Scan-Befehl. Mit »mode« gibt man an, in welcher Weise das Bild berechnet wer-

den soll. 0 bedeutet volles Ray-Tracing, 1 die einfarbige Darstellung von Objekten und 2 eine noch gröbere Darstellung als beim Wert 1. Mit dem letztgenannten lassen sich besonders schnell Bilder berechnen, um die Positionierung der Objekte zu überprüfen. Der Parameter bpp muß immer auf 4 gesetzt sein. Dies stellt die maximale Farbanzahl auf 4096 ein. Wenn im Herbst der Framebuffer endlich erhältlich ist, kann man für bpp den Wert 8 nehmen, der setzt die Farbanzahl auf satte 16 Millionen! Inwieweit man den Framebuffer aber damit ansteuern kann, sei dahingestellt. Ebenso verkräftet DBW-Render eine maximale Bildschirmauflösung von 1024 x 1024 Punkten. Nur ansehen kann man sie sich (noch?) nicht.

**LOOK from x1 x2 x3 to y1 y2 y3 up 0 1 0**

Hiermit wird die Kameraposition und der Blickvektor bestimmt. Die Parameter x1, x2 und x3 geben die Position der Kamera im Raum an. y1, y2 und y3 bestimmen die Blickrichtung und das mm-Objektiv (also Tele, Weitwinkel etc.). Die Bestimmung des Blickvektors wurde schon weiter oben erläutert.

Die letzten drei Werte (eigentlich immer 0, 1, 0) geben die Achse an, die senkrecht im Raum steht, also gewöhnlich die Y-Achse. Wer seinen Monitor hochkant aufgestellt hat, muß die X-Achse anwählen (also 1, 0, 0).

**BACKGROUND r g b**

Hier wird lediglich die Hintergrundfarbe gesetzt. Die »r g b«-Werte reichen von 0 (keine Farbe) bis 1 (volle Intensität). Der Amiga verarbeitet pro Farbe 16 Helligkeitsstufen. Dunkelorange hat beispielsweise die Werte 1, 0,44, 0.

**AMBIENT p**

Dieser Parameter bestimmt die Hintergrundbeleuchtung. Normalerweise geben Ray-Tracer Objekte, die im Schatten anderer Objekte liegen, vollkommen schwarz wieder. Da das in der Realität aber nie passiert,

**Auch im Schatten gut beleuchtet**

nimmt man einen gewissen Prozentsatz der eigentlichen Farbe des Objektes, stellt es also abgedunkelt dar. Im DBW-Render kann man zwar jedes einzelne Objekt einer »Ambienzfärbung« zuordnen, aber dieser Befehl ist einfacher. Er setzt übrigens alle definierten Objektambienzfärbungen außer Kraft. Für p haben sich Werte zwischen 0,1 und 0,25 als brauchbar erwiesen.

**ANTIALIAS a 1**

Das ewige Leid in der Computergrafik ist, daß Linien, Kanten oder Kreisränder immer mit einem Treppcheneffekt dargestellt werden. Im Englischen nennt man das »aliasing«. Nun kann man sich aber eines Tricks bedienen, dem Antialiasing. Hier werden an den Stellen, an

```

/*****
 *
 * Bild1.inp      -   by Marco Vitolini-Naldini 1989
 *                  (Maillingerstr. 20, 8 Muenchen 19)
 *
 *****/

parameters:
scan from 0 to 512 rows 656 columns 320 mode 0 bpp 4
look from 0 30 100 to 0 -25 -100 up 0 1 0
background .4 .1 .6
ambient .15
antialias 2 .1

textures:
check .3 .5 .7 10 0 10

lights:
spherical color 1 1 1 100 150 100 210 1

attributes:
attribute boden
texture 50 0 0 0 0 0 0 0 0 0 0 .9 .6 .3
attribute glas
texture 0 0 .35 1.3 .7 .7 .7 0 0 0 0 0 0

objects:
quad boden point -100 0 100
point -100 0 -100
point 100 0 100
sphere glas rad 1 25
trans 0 25 25

```

**Listing 1b. Das Inputfile zur Gestaltung des ersten Bildes**



dem diese Treppchen auftreten, Zwischenfarben gesetzt, die aus der Hintergrundfarbe und der Objektfarbe gemischt sind. Beim Ray-Tracing geht man dabei den Weg, mehrere Strahlen pro Pixel zu berechnen und dann deren Mittelwert zu nehmen. Dies stellt der Parameter *a* ein. Die Zahl kann 2, 4, 8, 16 und so fort annehmen. Mit der Größe des Wertes steigt die Qualität, aber auch die Rechenzeit an. Bei vier Rays pro Pixel bedeutet das auch die vierfache Rechenzeit! Der zweite Parameter unterteilt die einzelnen Strahlen noch einmal, wenn der Farbunterschied zwischen benachbarten Strahlen einen gewissen Wert überschreitet, und zwar um noch mal so viele, wie in Parameter *a* angegeben. Das dadurch erzielte Ergebnis ist mit der immensen Rechenzeit aber nicht mehr vereinbar. Deshalb empfehle ich den Wert bei 1 zu belassen. Wer für das erste Bild statt 17 nur 8 Stunden rechnen will, der kann den Anti-Alias-Befehl weglassen.

## CHECK r g b x z y

Nun ein Befehl zur Texturdefinition. CHECK überzieht ein Objekt mit einem Schachbrettmuster. Die rgb-Werte stellen die eine Farbe des Musters dar (die zweite Farbe wird am Objekt selbst definiert), die x-y- und z-Werte stellen die Karogröße im dreidimensionalen Raum ein. In unserem Beispiel verwenden wir zehn Einheiten auf der x-z Fläche. Der y-Wert muß Null sein. Dieser Textur wird automatisch die Nummer 50 für die spätere Objektdefinition zugewiesen. Man kann bis zu zehn verschiedene Schachbrettmuster definieren, die den Nummern 50 bis 59 zugewiesen werden.

## SPHERICAL r g b x z y a b

Damit unsere Bilder auch ins rechte Licht gerückt werden, benötigen wir eine Lichtquelle. Mit Spherical definiert man eine punktförmige Lichtquelle mit abnehmender Intensität. Die rgb-Werte beherbergen wieder die Farbe, die x-, y- und z-Werte stellen den Standpunkt der Lichtquelle im Raum dar. Mit *a* legt man die Reichweite dieses Lichts fest. Das bedeutet, daß alle Objekte, die weniger als *a* Einheiten von der Lichtquelle entfernt sind, etwas überbelichtet werden. Wenn sie weiter entfernt sind, werden die Objekte etwas zu dunkel. Bei 200 Einheiten als Reichweite kann man davon ausgehen, daß Objekte von 150 bis 250 Einheiten Entfernung gut ausgeleuchtet werden. Diese Lichtquelle hat den

Vorteil, besonders räumlich wirkende Bilder zu produzieren. Der Parameter *b* kann aus dem punktförmigen Licht auch eine unsichtbare lichtstrahlende Kugel machen. Er definiert den Radius der Lichtquelle, von der das Licht ausgeht. Mehr Rechenzeit ist erforderlich, wenn *b* nicht 1 ist.

## Bilder ins rechte Licht gerückt

Als Alternative gibt es das direkte Licht, das hier auch beschrieben werden soll:

## DIRECTIONAL r g b x y z

Die Farbwerte sind wie immer mit *r*, *g* und *b* zu definieren. *x*, *y* und *z* können Werte zwischen 0 und 1 annehmen. Man stellt sich am besten einen Pfeil vor, der mit der Spitze auf den Koordinatenursprung zeigt. Wählt man nun für *x* den Wert 1 und für die anderen 0, legen wir diesen Pfeil damit auf den positiven Bereich der x-Achse. Das Licht kommt nun also exakt von rechts. Setzen Sie für alle Werte 1 ein, kommt das Licht von schräg rechts vorne. Die Szene wird dann gut ausgeleuchtet. Der Unterschied der Lichtquellen besteht darin, daß Spherical wie eine Lampe, Directional wie Sonnenlicht wirkt. Bei der Einstellung Directional wird schneller gerechnet.

## ATTRIBUTE name texture n f m i t1 t2 t3 a1 a2 a3 r g b

Mit dem Befehl Attribute definiert man man Oberflächeneigenschaften, die später mit Namen einem Objekt zugewiesen werden. Für name setzen Sie ein beliebiges Wort ein, um die Eigenschaften einem Objekt übertragen zu können. Der Parameter *n* bestimmt die Textur, die vorher definiert sein muß (zum Beispiel unser Schachbrettmuster). Die Dokumentation der einzelnen Nummern finden Sie im Anhang Ihrer Anleitung für das Programm. Einige Beispiele:

Wert	Struktur
20	Marmor
40	Holz
50	unser Schachbrett
60	eine fraktale Oberfläche (dazu später mehr)
70	eine Farbblende
80	Schnee
90	Kieselstein

Es gibt noch einige zusätzliche, fest integrierte Texturen. Für diese muß keine vorherige Definition erfolgen (z.B. die Nummer 6 – eine Mauertextur). Die Textur 1 sollte einen vordefinierten

Farbverlauf auf das Objekt übertragen. Aus irgendeinem Grund funktionierte das bisher bei uns allerdings nicht.

Der nächste Parameter stellt einen Rauigkeitsfaktor dar. Um Objekte nicht zu perfekt aussehen zu lassen, kann man hier für *f* Werte zwischen 0 und 1 einsetzen, wobei 1 eine Bimssteinoberfläche erzeugt. Im HAM-Modus des Amiga leidet die Qualität dieser Funktion ein wenig, es sieht dabei eher nach einem Fehler in der Berechnung aus. Aber die Geschmäcker sind bekanntlich verschieden.

Nun zu dem sehr interessanten Parameter *m*, der Spiegelung. Hierfür lassen sich wieder Werte zwischen 0 und 1 einsetzen. Hundertprozentige Refle-

xion erreichen Sie mit 1, keine Spiegelung mit 0. Je höher der Wert gesetzt wird, desto länger dürfen Sie auch auf das Bild warten, da Reflexionen natürlich sehr rechenaufwendig sind. Mit diesem Parameter erreicht man auch das bekannte Highlight, also eine punktförmige weiße Spiegelung auf dem Objekt. Bei durchsichtigen Objekten wie unserer Glaskugel, soll sich das Licht, wie in der

Realität auch, brechen. Dafür sorgt der nächste Parameter, der Brechungsindex *i*. Soll das Objekt undurchsichtig sein, so lassen Sie ihn getrost auf 0. Keine Brechung wird bei einem Wert von 1 durchgeführt. Bei einer Glaskugel setzt man Werte um 1.2 bis 1.6 ein, für Wasseroberflächen 1.1 bis 1.3 und für Diamanten oder einige Phantasiobjekte 2.0 oder höher. Die Werte *t1* bis *t3* stellen die rgb-Farben für das Licht dar, das durch das Objekt hindurchgehen soll – die Transparenzfarben. Danach kommen die Ambientfarben. Dies sind die Farben, die das Objekt im Schatten erhalten soll. Diese Werte werden ignoriert, wenn, wie oben schon erwähnt, der Ambient-Befehl gesetzt wurde.

## Marco Vitolini-Naldini



Marco Vitolini-Naldini, 25, wurde in München geboren und wuchs in Prien am Chiemsee auf. Nach 13 Schuljahren besuchte er einen Computerkurs am Control Data Institut. Schon 1984 besaß er einen C64, den er in Basic programmierte. 1986 war es dann soweit: Marco erwarb einen der ersten Amiga 1000. Von Anfang an begeisterte unser Autor sich für die fantastische Grafik des Amiga. Schnell entwickelte sich daraus eine Leidenschaft, die ihn bis heute nicht losläßt. Über einfachere Grafiken, Fraktale, Ray-Tracing-Programme und Animationen reicht das Spektrum. Mit sei-

nem Freund Christian Zerbes entstand in sieben Monaten eine Ray-Tracing-Animation von 90 Sekunden Länge. Dieser beeindruckende Film diente Marco als Bewerbungsgrundlage bei verschiedenen Firmen – mit toller Resonanz. Seit 1988 arbeitet er nun als Junior Technical Director bei Steiner-Film. Bei dieser auf Computeranimationen für Werbefilme spezialisierten Grafik-Schmiede wechselt Marco demnächst in die Abteilung Video-Animation.

Dem Amiga hält er trotz der Arbeit an erheblich teureren und leistungsfähigeren Supercomputern die Treue. Mit DBW-Render V1.0 und V2.0 stellte er viele faszinierende Grafiken und Animationen her. Einen Ausschnitt aus einer kurzen Sequenz zeigen die Bilder auf den ersten beiden Seiten dieses Artikels. Auch die weiteren Bilder zeigen: Die dreijährige Arbeit mit dem PD-Programm hat deutliche Spuren hinterlassen. Profitieren Sie von Marco Vitolinis Erfahrungen, erleben Sie die wunderbare Welt der Grafik aus einer neuen Perspektive. Vielleicht wird Ihr Hobby mit diesen Kenntnissen auch bei Ihnen zum Beruf.

Schließlich ordnet man dem Objekt mit den Parametern *r*, *g*, *b* seine eigentlichen Farben zu. Auch hier sind es immer Werte zwischen 0 und 1. **QUAD name point x1 y1 z1 point x2 y2 z2 point x3 y3 z3** Kommen wir nun zur Erstellung unseres ersten Objektes. Mit dem Befehl »QUAD« kann man ein beliebiges flaches Rechteck definieren. Dazu müssen wir lediglich drei der



vier Eckpunkte als Raumkoordinaten angeben.

Der erste Punkt muß dabei immer zwischen den anderen beiden liegen. Sehen Sie sich dazu das Schemabild an. Es wird zuerst der Punkt links vorne angegeben, danach die Punkte, die vom ersten ausgehen, also links hinten und rechts vorne. Die beiden letzten Punkte kann man dabei ohne weiteres vertauschen.

Unser Schachboden liegt planar auf der x-z-Ebene. Es lassen sich aber natürlich auch

nach oben und um 25 Einheiten nach vorne. Dazu muß gesagt werden, daß eine Kugel unmitelbar nach ihrer Definition ihren Mittelpunkt im Koordinatenursprung hat. Damit sie nicht mitten im Boden liegt, haben wir sie um die Größe ihres Radius nach oben bewegt. Nun liegt sie genau auf der Fläche.

Soweit die Beschreibung der Befehle zu unserem ersten Projekt. Wenn das Programm das Bild fertig berechnet hat, gibt es einige Werte auf den Bildschirm aus. Da danach aber

Die Umwandlung ins IFF-Format läßt sich dann mit dem Aufruf

```
ray bild1.tmp
```

erreichen. Das Programm erkennt an den Endungen, was es mit den Files anstellen soll. Um ein fertiges Bild anzusehen, gibt man einfach

```
ray bild1.ilbm
```

ein. Doch nun zu den vom DBW-Render für unser erstes Bild ausgegebenen Werten. Mit »Average distribution Count«

IFF-Bild wird die Anzahl der verwendeten Farben angezeigt. In unserem ersten Bild sind es 295 verschiedene. Viele davon entstanden durch die Antialias-Funktion. Wenn Sie sich das Bild nun mit

```
ray bild1.ilbm
```

betrachten wollen, sehen Sie erst einmal gar nichts. Wo ist aber nun unser Bild? Die Display-Routine mag Auflösungen über 400 Pixel (vertikal) nicht. Wenden Sie den folgenden Trick an: Fahren Sie mit dem nun unsichtbaren Mausfeil so weit nach oben, bis Sie meinen, daß er am oberen Bildschirmrand anstößt. Drücken Sie dann die linke Maustaste und ziehen Sie den Screen nach unten und dann wieder nach oben. In dem Moment, in dem Sie den Screen bewegen, erscheint dann auch das Bild. Und wie gefällt Ihnen Ihr erstes Werk?

Um wieder zurück ins CLI zu kommen, müssen Sie die Return-Taste drücken. Sollte sich nun wider Erwarten nichts tun, so liegt das daran, daß Ihr CLI-Fenster nicht mehr aktiviert ist. Dazu müssen Sie den Screen mit dem Bild hinunterziehen, auf Ihr CLI-Fenster klicken und nochmals die Return-Taste drücken. Nun befinden Sie sich wieder in Ihrer gewohnten Arbeitsumgebung.

Experimentieren Sie ruhig mit verschiedenen Werten und Bildschirmgrößen. Oder wollen Sie gleich in die vollen? Dann tippen Sie mal das Eingabefile zu unserem zweiten Bild ab und beginnen die Berechnung. Wir beschreiben Ihnen derweil die neu hinzugekommenen Befehle:

#### MACRO Name /.../ done Name

Eines der leistungsfähigsten Merkmale des DBW-Render ist sicherlich die Macro-Funktion. Mit dieser lassen sich verschiedene Objektdefinitionen zusammenfassen und als ein Ganzes behandeln. Somit kann man ein kompliziertes Objekt, das aus mehreren Flächen besteht, mühelos in der Größe verändern, rotieren oder im Raum bewegen. Weiterhin lassen sich die Macros ineinanderschachteln, womit man mit wenigen Objektdefinitionen recht komplizierte Körper erstellen kann. Ein Beispiel dazu gibt es in Bild 3. Doch zurück zur jetzigen Berechnung. Mit

```
Macro <Macroname>
```

definiert man den Beginn eines Macros. Danach folgen die einzelnen Objektdefinitionen, wie wir sie schon anhand der Kugel



Bild 3. Ein Beispiel dafür, wie mit wenigen Objektdefinitionen komplizierte Körper entstehen

quer im Raum liegende Flächen erstellen.

Der Parameter name teilt dem Programm die dazugehörige Materialeigenschaft mit. In unserem Fall »boden«, welches im Attribute-Abschnitt definiert wurde.

#### SPHERE name rad 1 r

Hier wird eine Kugel mit einem Radius von r Einheiten definiert. Auch hier weist der Parameter name auf die dazugehörige Materialeigenschaft hin. Die 1 nach dem rad-Befehl besagt, daß der nachfolgende Wert einen äußeren Radius beschreibt. Dies ist sinnvoll, da es auch ringförmige Objekte gibt, die einen inneren Radius besitzen – nämlich einem Loch in der Mitte. Dem Sphere-Befehl folgt in unserem Eingabefile noch ein weiterer, der sich auf das zuletzt genannte Objekt bezieht. Mit trans lassen sich die Objekte im Raum bewegen. In unserem Fall um 25 Einheiten

gleich das Umwandlungsprogramm, das die berechneten Daten in ein IFF-Bild konvertiert, gestartet wird, und diese ebenfalls mehrere Meldungen auf den Bildschirm ausgibt, kann man die vorigen nur dann lesen, wenn man bei der Ausgabe vor dem Computer sitzt oder den Aufruf der Wandlung nach der Berechnung unterdrückt. Das geht folgendermaßen: Mit »ray -o...« gibt man vor dem Eingabefile das File an,

#### Der DBW-Render erstattet Meldung

das als letztes erzeugt werden soll. Damit werden weitere automatische Aufrufe gestoppt. Da in diesem Fall das temporäre File unser letztes sein soll, schreiben wir als Aufruf im CLI-Fenster:

```
ray -o bild1.tmp  
bild1.inp
```

gibt das Programm den Mittelwert aller gerechneten Strahlen pro Bildpunkt an. Mit unserem Antialias daher 2 pro Pixel. Für das ganze Bild wurden 163840 Bildpunkte berechnet. Das entspricht unserer Auflösung von 320 x 512 Punkten. Die komplette Rechenzeit betrug bei uns mit Fast-RAM 58756 Sekunden, also rund 16,5 Stunden. Mit »Intersect Run« ist die Anzahl der Schnittpunktberechnungen des Lichtstrahls mit den Objekten gemeint: fast 2 Millionen für unser Bild. Die Schnittpunktberechnung ist die eigentliche Ursache für die erhebliche Dauer einer Berechnung. Interessant ist auch der benötigte Stack für die Sortierungen. Sollte ein Bild plötzlich ab einer Stelle keine Texturen mehr aufweisen, so liegt das am vorher zu klein definierten Stackspeicher. Deshalb sollte man ihn großzügig bemessen (siehe oben). Bei der Umwandlung der Daten in ein



und der Fläche kennengelernt haben. Unser erstes Macro definiert drei Flächen, die so angeordnet sind, daß sie drei Wände eines Raumes darstellen. Als Macronamen verwenden wir daher sinnigerweise »Raum«. Damit der Parser auch weiß, wo das Macro aufhört, schließen wir dessen Eingabe mit

```
done <Macroname>
```

## Leistung durch Macros

Das nächste Macro definiert eine Wasserfläche, das dritte ein Stahlrohr und das vierte eine Glaskugel. Das Stahlrohr ist mit einem neuen Objektbefehl erstellt:

```
OBJ name point x y z (...)  
sides n arc g close f f
```

Mit diesem Befehl lassen sich auf recht einfache Weise Rotationskörper erstellen. Als Name geben wir wie immer den Materialnamen an. Mit den point-Kommandos beschreiben wir die Umrisse des Objektes am linken Rand der Y-Achse. Der z-Wert bleibt deshalb immer auf null.

Das einfachste Objekt ist ein Rohr, das nur zwei point-Kommandos benötigt, nämlich den unteren und den oberen Punkt an der Y-Achse. Den Radius beschreibt man mit der Entfernung auf der X-Achse zum Nullpunkt. In unserem Beispiel also ein Rohr mit gleichbleibendem Radius von 7 und einer Höhe von 200 Einheiten. Das sides-Kommando teilt mit, wie viele Unterteilungen das Objekt haben soll. Da wir es recht klein sehen werden, reicht eine Unterteilung von 32 Flächen, um eine relativ runde Oberfläche zu bekommen.

Der arc-Befehl gibt an, wie weit die Linie um die Y-Achse gedreht werden soll. Da wir eine volle Drehung, also 360 Grad wünschen, können wir diesen Befehl weglassen: 360 ist der default-Wert von arc. Wenn Sie die Enden des Rohres geschlossen haben wollen, können Sie dies mit

```
close 1 1
```

erreichen. Damit berechnet das Programm auch noch die Flächen für die Deckel unten und oben am Rohr. Wir sehen die Enden nicht, deshalb können wir uns auch das sparen.

Eine weitere Möglichkeit des OBJ-Befehls, sowie ein komplizierteres Rotationsobjekt, werden Sie weiter unten in Bild 5 zu sehen bekommen.

## WAVE x y z l a d p

Dieser Befehl definiert Wellen, die irgendwo im Raum liegen. Man braucht sie für verschiedene Texturen, wie Marmor oder Holz und natürlich für das Aussehen von in Bewegung befindlichem Wasser. Die x-, y- und z-Werte definieren den Wellenursprung, l die Wellenlänge, a die Amplitude – also die Höhe der Welle. d definiert das sogenannte damping (ein Faktor, der die Welle langsam ausklingen läßt) und p die Propagation, auf Deutsch, wie weit die Welle schon »gewellt« hat. Wenn man Texturen einsetzt, die von Wellen definiert werden, oder wenn man realistisches Wasser erzeugen möchte, sollte man mehrere davon definieren. Fünf Stück haben sich als praktisch erwiesen, maximal zehn sind zulässig. Dabei sollten die Wellenursprünge möglichst gleichmäßig über die Szene verteilt werden. Die Wellenlänge sollte so gewählt werden, daß sie sich innerhalb des Bildes maximal dreißigmal wiederholt. Wenn sie zu eng gewählt wird, sehen Spiegelungen und Lichtreflexe nicht mehr glatt und ineinander übergehend aus, sondern werden eher zerstückelt. Aber probieren Sie ruhig verschiedene Werte aus.

Die Amplitude muß immer im Bereich von 0 und 1 liegen. Auch sollte man beachten, daß kleinere Wellenlängen auch kleinere Amplituden haben sollten und umgekehrt.

Mit dem Dampingwert legt man das Ausklingen der Welle fest. Es sollte immer knapp den Wert 1 erreichen, es sei denn, Sie wollen eine Überlagerung der Wellen verhindern, um zum Beispiel ins Wasser gefallene Regentropfen zu simulieren.

## Wassertropfen im Monitor

Für die Animation ist der Propagate-Faktor wichtig, auf den wir am Ende des Artikels eingehen werden. Setzen Sie hier einen beliebigen Wert ein.

## BLEND r g b s r

Möchten Sie ein Objekt mit einem Farbverlauf erstellen? Mit Blend ist das kein Problem. Hiermit lassen sich weiche Farbübergänge von unten nach oben realisieren. Die untere Farbe wird aus den definierten Attributen des Objekts genommen. Die Farbe, in die das Objekt wechseln soll, geben Sie über die rgb-Parameter ein. Der Parameter s gibt den Y-Wert an, ab dem der Farbübergang be-

```

/*****
 *
 * Bild2.inp - by Marco Vitolini-Naldini 1989
 * (Maillingerstr.20, 8 Muenchen 19)
 *
 *****/
Macros:
Macro Raum
quad Mauer point -100 -100 -100
           point -100 150 -100
           point 100 -100 -100
quad Mauer point -100 -100 -100
           point -100 150 -100
           point -100 -100 200
quad Mauer point 100 -100 -100
           point 100 150 -100
           point 100 -100 200
done Raum

Macro See
quad Wasser point -100 0 -100
            point -100 0 200
            point 100 0 -100
done See

Macro Rohr
obj Metall point -7 -100 0
            point -7 100 0
            sides 32
done Rohr

Macro Kugel
sphere Glas rad 1 20
done Kugel

parameters:
scan from 0 to 512 rows 650 columns 320 mode 0 bpp 4
look from 0 40 200 to 0 -20 -100 up 0 1 0
background .1 .1 .6
ambient .15

textures:
wave -70 -30 50 7 .3 .98 0
wave 80 20 20 16 .4 .97 0
wave 29 -50 -40 15 .3 .96 0
wave -30 40 -5 8 .4 .95 0
wave -10 30 -30 19 .7 .98 0
blend .4 .4 .7 10 10

lights:
spherical color 1 1 1 50 50 50 100 1
spherical color .8 .8 .3 -60 -25 -60 55 1
spherical color .3 .8 .8 0 -25 -60 55 1
spherical color .8 .3 .8 60 -25 -60 55 1

attributes:
attribute Wasser
texture 4 0 .6 1.2 .3 .3 .3 0 0 0 0 0 .2
attribute Mauer
texture 5 0 0 1 0 0 0 0 0 0 .8 .3 .3
attribute Metall
texture 0 0 .3 1 0 0 0 0 0 0 .2 .2 .3
attribute Glas
texture 0 0 .6 1.3 .7 .7 .7 0 0 0 0 0 0
attribute Farbe
texture 70 0 0 1 0 0 0 0 0 0 .2 .6 .2

objects:
Raum
See
Rohr trans -30 0 -100
Rohr trans -100 0 -100
Rohr trans 100 0 -100
Rohr trans 30 0 -100
Kugel trans 0 40 -100
text "AMIGA" Farbe
scale 20 20 20
trans -50 10 30

message "done"

```

Listing 2. Makros im DBW-Render erleichtern die Arbeit



ginnen soll, der r-Wert, wie weit der Verlauf stattfinden soll, bis das Objekt die neue Farbe angenommen hat. In unserem Beispiel beginnen wir bei  $y=10$  mit dem Farbverlauf, der bei  $y=20$  (also 10 Einheiten höher) beendet sein wird. Sie können auch hier wieder bis zu zehn verschiedene Farbverläufe pro Szene definieren.

Bereits bekannt ist das »spherical light«. In unserem Bild wurde eine Lichtquelle für die Beleuchtung des Raumes verwendet und drei farbige Lichtquellen unter das Wasser gelegt, um ein paar schöne Effekte zu bekommen.

Bei den Attributen sei auf die Wasserdefinition hingewiesen. Sie hat einen Brechungsindex von 1,3 und eine Transparenz von 30 Prozent. Die Reflexion ist auf 60 Prozent gesetzt und die Texturnummer (siehe Anhang B in der Anleitung) auf 4. Bei den Farben wurde ein sehr dunkles Blau definiert, da die Farbe des Wassers stark davon abhängt, was sich darin spiegelt. Deswegen wurde die Hintergrundfarbe auch auf Blau gesetzt, obwohl man im Bild nichts davon sieht, sich der Hintergrund aber im Wasser spiegelt. Eine neue Textur lernen

wir in der zweiten Attributdefinition kennen. Sie nennt sich Brick (auf Deutsch Ziegelstein) und ist im Programm bereits vordefiniert. Mit der Texturnummer 5 kann man sie auf die gewünschten Objekte übertragen. Bei den nun folgenden Objektdefinitionen können Sie die Vorteile der Macros schon erkennen. Sie brauchen nur noch den Macronamen anzugeben, um ein Objekt in die Szene zu setzen. Bei unserem Stahlrohr erweist es sich als besonders nützlich, da wir insgesamt vier Rohre in die Szene setzen wollen und nur eines definieren mußten. Mit dem trans-Befehl setzen wir die Objekte an die Stellen, die wir wünschen: in die beiden Ecken und links und rechts der Glaskugel.

### Text in der Glaskugel

Mit dem jetzigen Wissen müßten Sie schon in der Lage sein, einiges aus DBW-Render herauszuholen. Aber es gibt noch mehr:

#### TEXT »String« name

Hiermit läßt sich zwei- oder dreidimensionaler Text in die Szene setzen. »String« enthält

in unserem Beispiel – wie könnte es anders sein – die Buchstaben AMIGA. Für name setzen wir wieder den Namen des definierten Attributes ein.

Der Befehl TEXT erzeugt dabei die Objekte für die Buchstaben, beginnend im Koordinatenursprung mit der Größe  $1 \times 1 \times 1$  Einheiten pro Buchstabe. Der Text AMIGA ist demnach fünf Einheiten lang und eine Einheit hoch wie tief. Für unsere Szene ist das natürlich zu klein, also müssen wir ihn vergrößern. Dies geht mit dem »scale«-Befehl.

Die anschließenden Werte multiplizieren die Größe in der x-, y- oder z-Richtung um den genannten Wert. In unserem Beispiel wird das Objekt insgesamt zwanzigmal vergrößert. Mit diesem Befehl lassen sich Objekte natürlich auch beliebig dehnen und strecken. Soll in einer Richtung nicht vergrößert werden, setzen Sie den Wert 1 ein. Werte zwischen 0 und 1 verkleinern den Körper, der Wert 0 läßt ihn verschwinden (oder den Amiga abstürzen). Zum Abschluß unseres Bildes noch ein nicht unwichtiger Befehl:

#### MESSAGE »String«

Damit können Sie sich wäh-

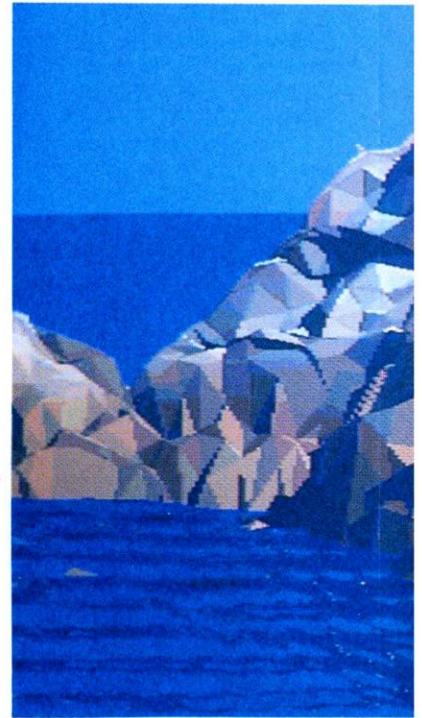
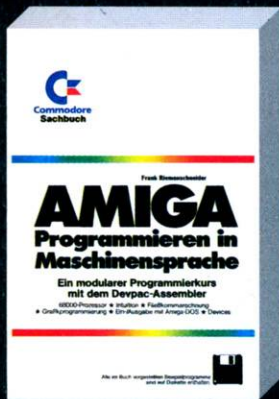


Bild 4. Fraktales Gebirge mit Schnee

rend der Übersetzung und beim Einlesen der Daten in den Ray-Tracer Meldungen auf dem Bildschirm ausgeben lassen. Das kann nützlich sein, wenn der Computer während des

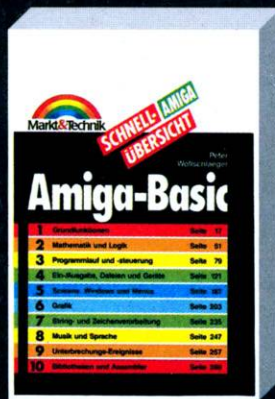
## Aktuelle Bücher und Bookware für den Amiga:

# STERN



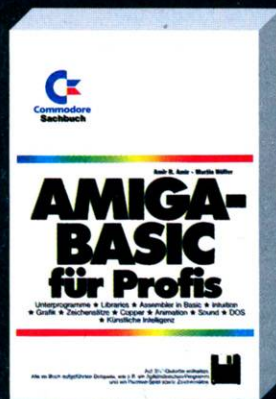
F. Riemenschneider  
Amiga: Programmieren in Maschinensprache

Durch diesen ausführlichen Assemblerkurs wird die Programmierung auch für Anfänger und Nur-Basic-Programmierer ein Kinderspiel. Der Clou dieses Buches besteht darin, daß das Betriebssystem des Amiga in ausführlichen Super-Beispielen vorgestellt wird.  
1989, 432 Seiten, inkl. Diskette  
Bestell-Nr. 90712, ISBN 3-89090-712-1  
DM 59,- (sFr 54,30,- / oS 460,-)



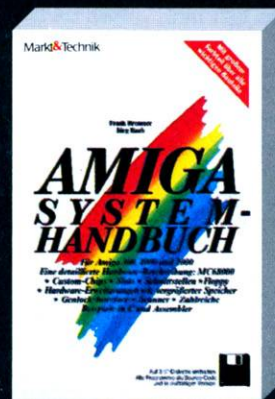
P. Wollschlaeger  
Schnellübersicht Amiga-Basic

Schnelle Antworten auf die Fragen, die bei der täglichen Arbeit mit dem Programm auftreten.  
1989, 299 Seiten,  
Bestell-Nr. 90736, ISBN 3-89090-736-9  
DM 39,- (sFr 35,90 / oS 304,-)



A. Amir/M. Höfler  
Amiga-Basic für Profis

Das vorliegende Buch umfaßt Gebiete wie die gesamte Intuition, Grafik, 4096 Farben, Darstellung von dreidimensionalen Objekten, Fractals, Animation, Amiga-DOS, digitalisierter Sound, künstliche Intelligenz und viele andere.  
Lieferbar 2. Quartal 1989, 565 Seiten, inkl. 2 Disketten  
Bestell-Nr. 90710, ISBN 3-89090-710-5  
DM 79,- (sFr 72,70 / oS 616,-)



J. Kremser/F. Koch  
Amiga-Systemhandbuch

Systemhandbuch für engagierte Amiga-User und Hobby-Bastler! Mit zahlreichen Beispielen in C und Assembler für maschinennahes Programmieren. Ausführliche Erläuterung über die Möglichkeiten der Amiga-Custom-Chips und Hardware-Erweiterungen.  
1988, 421 Seiten, inkl. Diskette  
Bestell-Nr. 90550,  
ISBN 3-89090-550-1  
DM 79,- (sFr 72,70 / oS 616,-)



I. Krüger  
Amiga: Programmieren mit Modula 2

Leichtverständlicher Modula-2-Kurs! 1988, 350 Seiten, inkl. Diskette  
Bestell-Nr. 90554, ISBN 3-89090-554-4  
DM 69,- (sFr 63,50 / oS 538,-)

Markt&Technik

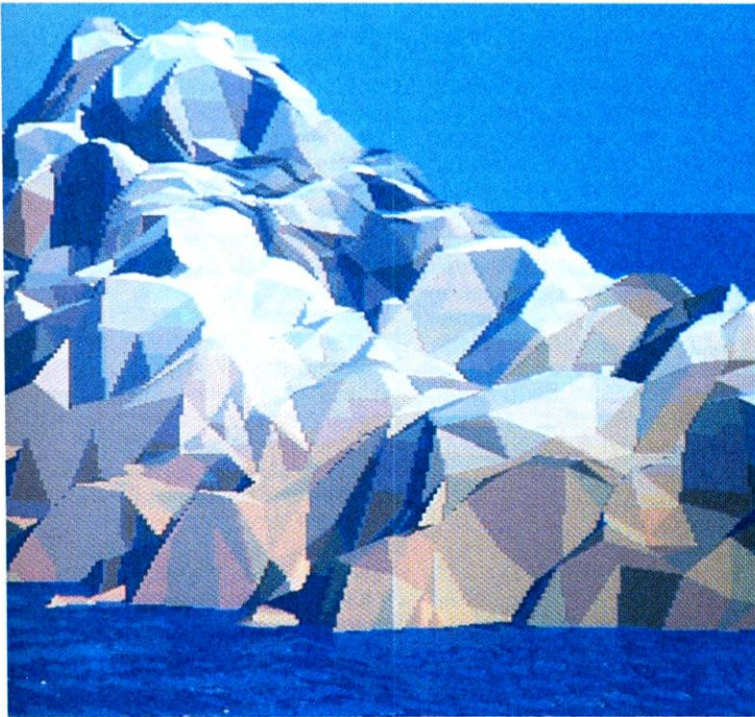
Zeitschriften · Bücher

Software · Schulung

AMIGA-SONDERHEFT 5

Markt & Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0.  
Bestellungen im Ausland bitte an: SCHWEIZ: Markt & Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440550,  
ÖSTERREICH: Markt & Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0,  
Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526,





- Dreiecke bilden das Gerüst

Einlesens der Daten abstürzt oder einfach stehenbleibt. Man kann hiermit die Zeile des Eingabefiles eingrenzen, die den Fehler verursacht. Sonderbarerweise muß des öfteren bei

der Verwendung von Macros am Ende der Eingabefiles eine Message stehen, da der Parser sonst verrückt spielt. Gewöhnlich Sie sich gleich an, eine abschließende Message einzufü-

gen. Wie Sie sicher bemerkt haben, ist die Rechenzeit nicht mehr dieselbe wie im vorigen Beispiel. Es dauert rund eineinhalb Tage, bis das Bild durchgerechnet ist. Wenn Sie Ihren Amiga nun andersweitig gebrauchen möchten, können Sie den Rechenvorgang mit CTRL-C abbrechen. Stellen Sie sicher, daß das CLI-Window aktiv ist, in dem der DBW-Render gestartet wurde, und drücken Sie die Tastenkombination. Jetzt müssen Sie allerdings noch etwas warten, bis die aktive Scanline durchgerechnet ist. Dabei kann es sich um bis zu vier Minuten handeln. Danach bricht das Programm ab und der Amiga meldet sich wieder mit dem Prompt. Zählen Sie nun ab, bis zu welcher Scanline das Bild gerechnet wurde und merken Sie sich die Zahl. Wenn Sie sich ansehen wollen, was bisher gerechnet wurde, tippen Sie

```
ray bild2.tmp
ein und warten Sie die Umwandlung des Bildes ab. Mit
ray bild2.ilbm
```

können Sie es sich danach anschauen. Wenn Sie weiterrechnen wollen, wechseln Sie zuerst in Ihren Editor, laden das Eingabefile zu Bild 2 und än-

dern die erste Zahl im SCAN-Befehl auf die Scanzeile, in der weitergerechnet werden soll. Angenommen, DBW-Render hat bis zur Zeile 312 gerechnet, geben Sie 305 als Startzeile ein. Somit vermeiden Sie eventuelle schwarze Linien im Bild.

## Kurze Unterbrechung

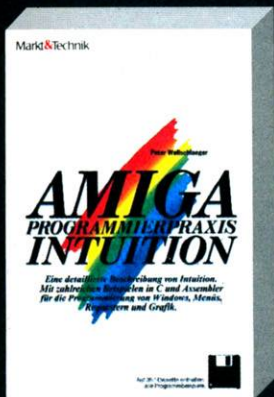
Aber das haben wir schon erklärt. Speichern Sie nun das neue Eingabefile als »bild2-1.inp« und beginnen Sie die Berechnung mit

```
ray bild2-1.inp
```

Sie können Ihr Bild in bis zu zehn Teile auf diese Weise aufsplitten. Wenn Sie zufällig zehn Amigas zu Hause rumstehen haben, können Sie somit eine beachtliche Geschwindigkeitssteigerung von rund 1000 Prozent erzielen.

Am Rande sei noch gesagt, daß DBW-Render kaum auf Turbokarten anspricht. Für das Geld kaufen Sie sich lieber noch ein oder zwei Amiga 500 und schon haben Sie drei Bilder in der Zeit von einem fertig. Wenn Sie nun alle Teile des Bildes durchgerechnet haben,

# STUNDEN



**P. Wollschlaeger**  
**Amiga: Programmierpraxis Intuition**  
 Eine detaillierte Beschreibung von Intuition! Neben der Programmierung von Fenstern, Menüs und Grafiken behandelt der Autor auch wichtige Randgebiete, wie die Ein- und Ausgabe von Texten oder Zugriff auf die Diskette.  
 1988, 330 Seiten, inkl. Diskette  
 Bestell-Nr. 90593, ISBN 3-89090-593-5  
 DM 69,- (sFr 63,50/öS 538,-)



**J.-P. Laub/J. Wenzl**  
**Amiga Public-Domain-Dokumentation**  
 Mit diesem Buch erhalten Sie einen detaillierten Überblick über die Vielfalt der Public-Domain-Programme.  
 1989, 295 Seiten,  
 Bestell-Nr. 90675, ISBN 3-89090-675-3  
 DM 49,- (sFr 45,10/  
 öS 382,-)

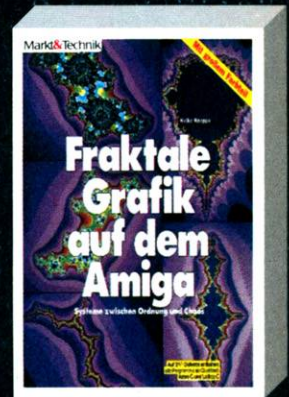


**F. Kremser**  
**Das Amiga-Programmier-Handbuch Teil 2**  
 1989, 208 Seiten, inkl. Diskette  
 Bestell-Nr. 90758, ISBN 3-89090-758-X  
 DM 69,- (sFr 63,50/öS 538,-)

**F. Kremser/J. Koch**  
**Amiga-Programmier-Handbuch**  
 1987, 387 Seiten, inkl. Diskette  
 Bestell-Nr. 90491,  
 ISBN 3-89090-491-2  
 DM 69,- (sFr 63,50/öS 538,-)



**H. R. Henning**  
**Grafik mit Amiga-Basic**  
 Dieses Buch ist speziell der Grafik-Programmierung auf dem Amiga gewidmet. Der erste Teil stellt für den Anfänger alle bekannten Grafik-Befehle des Amiga-Basic vor. Mit Beginn des zweiten Teiles werden die Routinen des Betriebssystems zur Grafik-Programmierung herangezogen.  
 1989, 488 Seiten, inkl. Diskette  
 Bestell-Nr. 90669, ISBN 3-89090-669-9  
 DM 59,- (sFr 54,30/öS 460,-)



**H. Knappe**  
**Fraktale Grafik auf dem Amiga**  
 Ein Buch für Forscher, die an einer revolutionären Entwicklung in den Naturwissenschaften teilnehmen wollen und bereit sind, auf Entdeckungsreise zu gehen. Reisen Sie mit!  
 1988, 272 Seiten, inkl. Diskette  
 Bestell-Nr. 90600, ISBN 3-89090-600-1  
 DM 79,- (sFr 72,70/öS 616,-)

\* Unverbindliche Preisempfehlung

Markt & Technik-Produkte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler

AMIGA-SONDERHEFT 5

Fragen Sie Ihren Fachhändler nach unserem kostenlosen Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!



setzen Sie es mit folgendem Befehl zusammen:

```
ray bild2.tmp bild2-1.tmp
mp ... bild2-9.tmp
```

Nacheinander werden nun alle »tmp«-Files zusammengefügt und das Bild unter dem Namen des als erstes angegebenen Files gespeichert. In unserem Fall also unter »bild2.ilbm«. Na, wie gefällt es Ihnen? Unser Bild hat 1566 verschiedene Farben. Hier zeigt sich schon sehr deutlich die Leistungsfähigkeit dieses Ray-Tracers gegenüber kommerziellen Programmen. Es kann sogar noch mehr! Wissen Sie, was ein fraktales Gebirge ist? Es handelt sich hierbei um ein Dreieck, das wiederum in mehrere Dreiecke unterteilt wird, deren Ecken um einen bestimmten Wert im Raum bewegt werden. Diese werden wieder in kleinere Dreiecke unterteilt, deren Ecken wiederum... und so fort. Je häufiger man das Grunddreieck unterteilt, desto realistischer wirkt das Objekt. Auch diese Art von Körpern ist mit DBW-Render möglich. Beispiel dafür ist unser drittes Bild (Bild 4). Bevor Sie aber die Berechnung starten, lesen Sie erst mal weiter.

**FRACT name point x y z  
point x y z  
point x y z**

Mit diesem Objektbefehl erstellen Sie das Grunddreieck, woraus später das eigentliche fraktale Gebirge errechnet wird. Mit den drei Koordinatenpunkten geben Sie die Ecken des Dreiecks an. Der Parameter name steht wie immer für das Attribut, welches sich etwas kompliziert aufbaut. Dazu aber gleich mehr. Wie Sie an unseren Koordinatenwerten erkennen können, liegt das Dreieck ziemlich schräg in unserer Szene. Man muß einige Versuche starten, bis der Berg auch nach einem Berg aussieht. Die Struktur hängt erheblich von der Lage des Dreiecks und den definierten Wellenzentren ab. Im Ray-Tracer geschieht nichts per Zufall. Wenn aber quasi »zufällige« Werte benötigt werden, bedienen wir uns der Wellendefinitionen. Das hat den Vorteil, daß man das Bild in mehreren Abschnitten berechnen lassen kann, ohne daß sich die Form des Gebirges dauernd verändert.

**SNOW s r f t**

Eine weitere vorhandene Textur ist der Schnee. Sie wird wie die Blend-Funktion mit einem Start (s) und einem Range (r) definiert. Zusätzlich läßt sich mit f die ansteigende Schneedichte im definierten Bereich

verändern. Mit f=0 bekommt man schon beim mit »Start« definierten Y-Wert die volle Schneedichte. Bei f=1 entsteht eine gleichmäßig ansteigende und bei f größer 1 eine erst langsam, dann immer schneller ansteigende Schneedecke.

Der Parameter t bestimmt den Winkel einer Fläche, ab dem diese mit Schnee bedeckt sein soll. Gar keinen Schnee erzielt man mit 0 und eine rein weiße Landschaft mit 1. In unserem Beispiel benutzen wir den Wert 0.4. Dieser bestimmt, daß bei einer weniger als 40 Grad abfallender Fläche

Schnee liegen soll. Der »Schnee« wird übrigens dargestellt, indem die Farbe des Objektes einfach auf Weiß gesetzt wird. Kommen wir nun zum etwas komplizierten Teil der Fractal-Attribut-Definition.

**FRACTAL recursion r scale x y z finalatt name**

Dieses Kommando im Attribut-Teil unsere Eingabefiles definiert den Aufbau des Fractals und überträgt es auf eine andere mit name definierte Objekteigenschaft. Diese andere Objekteigenschaft ist bei uns der Schnee (mit dem Namen Rock).

Der Rekursionsparameter r definiert, wie oft unser Grundobjekt unterteilt werden soll. Der Wert 7 schafft 4096 einzelne Objekte im Speicher. Und das war ein wichtiges Stichwort. Diese Anzahl ist leider nur mit mindestens 1,5 MByte Speicher erreichbar. Mit 512 KByte läßt sich leider nur eine Rekursionstiefe von 5 einstellen. Es wäre sicher interessant, das Objekt einmal mit 9 MByte Speicher berechnen zu lassen. Es müßte sich sicher ein sehr realitätsnaher Felsen erstellen lassen.

Der scale-Parameter gibt den Ablenkungsfaktor der Eckpunkte aller Dreiecke im Raum an. Damit kann man den Berg in beliebige Richtungen »wachsen« lassen. Mit 0 wird keine Ablenkung erreicht. In unserem Beispiel sind mit den Werten 0.1, 0.3 und 0.1 schon recht große Ablenkungen entstanden,

## Fraktale mit Ablenkungen

daher sollte man hierbei etwas vorsichtig umgehen. Zu guter Letzt muß diese kombinierte Objekteigenschaft noch in eine dritte, nämlich die unseres Berges übertragen werden. Sie sehen das beim Attribut »Gebirge«. Die Texturnummer muß dabei 60 sein, da damit das Fraktal angesprochen wird. Wenn Sie nun die Berechnung des Bildes starten, werden Sie ein Weilchen warten müssen, bis der Ray-Tracer loslegt. Die einzelnen Dreiecke des Fractals müssen zuerst ausgerechnet werden. Bei einer Rekursion von 7 kann das schon mehrere Minuten in Anspruch nehmen. Die reine Rechenzeit des Bildes läßt sich in einem Tag leicht unterbringen. Falls Sie mit der Version 1.0 von DBW-Render schon einmal gearbeitet haben, so sei darauf hingewiesen, daß dort die Berechnung dieses Bildes fast zwei Wochen (!) in Anspruch nahm. Seien wir deshalb mit einem Tag zufrieden.

Wenn Sie mit dem Fraktal-Bild ein wenig experimentieren wollen, so vermindern Sie während der Testrechnungen die Rekursion auf den Wert 3. Dieser zeigt schon die ungefähre Form des Fractals, die Berechnung dauert aber nicht so lange.

Wenden wir uns nun einem Beispiel zu, das im Vergleich zu den vorherigen recht komplex ist. Vorher sollten Sie aber alle Files mit den Endungen »tmp« und »dat« auf Ihrer Arbeitsdis-

```

/*****
 *
 * Bild3.inp - by Marco Vitolini-Naldini 1989
 * (Maillingerstr.20, 8 Muenchen 19)
 *
 *****/

Macros:
Macro Berg
  fract Gebirge point 0 100 0
                point 120 0 150
                point -220 0 0
done Berg

Macro Meer
  quad Wasser point -1000 2 -1000
               point 1000 2 -1000
               point -1000 2 1000
done Meer

Macro Bild
  Berg trans 0 0 -50
  Meer
  done Bild

parameters:
scan from 0 to 512 rows 650 columns 320 mode 0 bpp 4
look from 0 100 500 to 0 0 -100 up 0 1 0
background .2 .3 .7
ambient 0.17

textures:
wave -40 60 -200 20 .4 1 0
wave -100 -200 87 35 .2 1 0
wave 203 -300 187 17 .3 1 0
wave 406 98 -132 6 .3 1 0
wave -207 154 -85 15 .2 1 0
wave -88 -33 23 25 .3 1 0
snow 20 50 1.0 0.6

lights:
directional 1 1 1 1 1 1

attributes:
attribute rock
  texture 80 0 0 1 0 0 0 0 0 .5 .3 .1
  fractal recursion 7 scale .1 .3 .1 finalatt rock
attribute Gebirge
  texture 60 0 0 0 0 0 0 0 0 0 0 0
attribute Wasser
  texture 4 0 .5 1.2 0 0 0 0 0 0 0 .2

objects:
Bild

message "done"

```

**Listing 3. Auch fraktale Landschaften sind mit DBW-Render realisierbar**



kette löschen, da der Platz für unser letztes Bild sonst nicht mehr ausreicht. Beginnen wir mit der Beschreibung der einzelnen Macros. Picture, unser erstes Macro, beschreibt eine einzelne Fläche, die schon fertig in unserer Szene positioniert ist. Dort wird später ein Bild »draufgemappt«.

Das zweite Macro erzeugt das weiter oben versprochene komplexere Rotationsobjekt (das Weinglas in Bild 5). Wie dessen Macroname schon sagt, soll es ein Weinglas werden. In Schemabild 2 (Bild 6) sind die Umrisse einer Seite dieses Glases eingezeichnet. Das soll Ihnen die Koordinatenübertragung veranschaulichen. Die 19 einzelnen Punkte werden also definiert und dann mit »sides 64« in 1216 einzelne Polygone zerlegt und um 360 Grad rotiert. Da ein Weinglas normalerweise oben offen und unten geschlossen ist, lassen wir uns dies mit »close 0 1« berechnen. Die zwei nächsten Macros »Boden« und »Wand« stellen wieder nur je eine Fläche dar, die unseren Raum mit einer Wand und einem Boden abschließt.

Als weiteres Objekt wollen wir einen dreieckigen Tisch in die Szene stellen. Bevor wir aber alle drei Tischbeine einzeln entwerfen, definieren wir uns nur eines davon in einem Macro und rufen es danach dreimal in unserem Tisch-Macro auf. Natürlich müssen die Beine noch an die richtige Position gebracht werden. Dies erledigt wieder der »trans«-Befehl. Dieses ist ein Beispiel für die Verschachtelung von Macros ineinander. Für die Tischfläche benutzen wir einen neuen Objektbefehl:

```
TRI name      point x y z
               point x y z
               point x y z
```

»Tri« steht für Triangle, also Dreieck. Er funktioniert eigentlich genauso wie das schon kennengelernte fraktale Grunddreieck. Nur bleibt uns hier die Fläche als Ganzes erhalten. Die drei point-Befehle definieren die jeweiligen Eckpunkte im Raum. name ist, wie sollte es anders sein, mal wieder die Attributbezeichnung.

In unserem Tisch-Macro erstellen wir mit dem ersten TRI-Befehl die Tischoberfläche, mit dem zweiten die Unterseite des Tisches. Um nun nicht zwei einzelne Dreiecke im Raum »rumhängen« zu lassen, werden wir einen Tischrand mit drei Flächen erzeugen. Als Eckpunkte verwendet man natürlich die Koordinaten der beiden Drei-

ecke. Danach rufen wir dreimal das Bein-Macro auf und verschieben unsere Tischbeine an realistische Positionen.

Werfen Sie mal einen Blick auf die scan-Parameter. Richtig, dieses Bild lassen wir in Overscan-Auflösung berechnen. Das bedeutet, daß nun auch der Rand mitberechnet wird. Natürlich dauert das wieder etwas länger. Bei diesem Bild müssen Sie mit rund 40 Stunden rechnen. Sie können es aber auch in mehreren Teilen bearbeiten. Eine weitere Lösung des Zeitproblem ist, es bei einem Freund mit den Worten

oder es würde öfter wiederholt werden. Zum Ausprobieren können Sie ja mal die Texturen der Wand mit denen des unter Picture bezeichneten Bildes tauschen.

## MARBLE rgb x t s

Mit diesem Befehl läßt sich eine, leider sehr rechenaufwendige, Marmortextur erstellen. Dazu benötigt man einige im Raum definierte Wellen. Die rgb-Parameter definieren die Marmorfarbstreifen, die in das Objekt eingelegt werden. Mit »x« definiert man den Abstand von gedachten, vertikal auf der X-Achse stehenden Linien,

ne Holztextur erzeugen. Auch hier müssen wieder einige Wellen vorher definiert werden. Die rgb- und xts-Parameter entsprechen denen der Marmortextur. Der d-Parameter beeinflusst zusätzlich die Stärke der Holzmaserung. Werte unter 1 schwächen sie ab, über 1 werden sie verstärkt.

Beim Boden nehmen wir eine neue, bereits im Programm integrierte Textur: das sogenannte Mottled Diffuse. Diese Textur ist mit der Nummer 3 abrufbar. Das Ergebnis können Sie im entsprechenden Bild erkennen: der Boden ist damit ge-

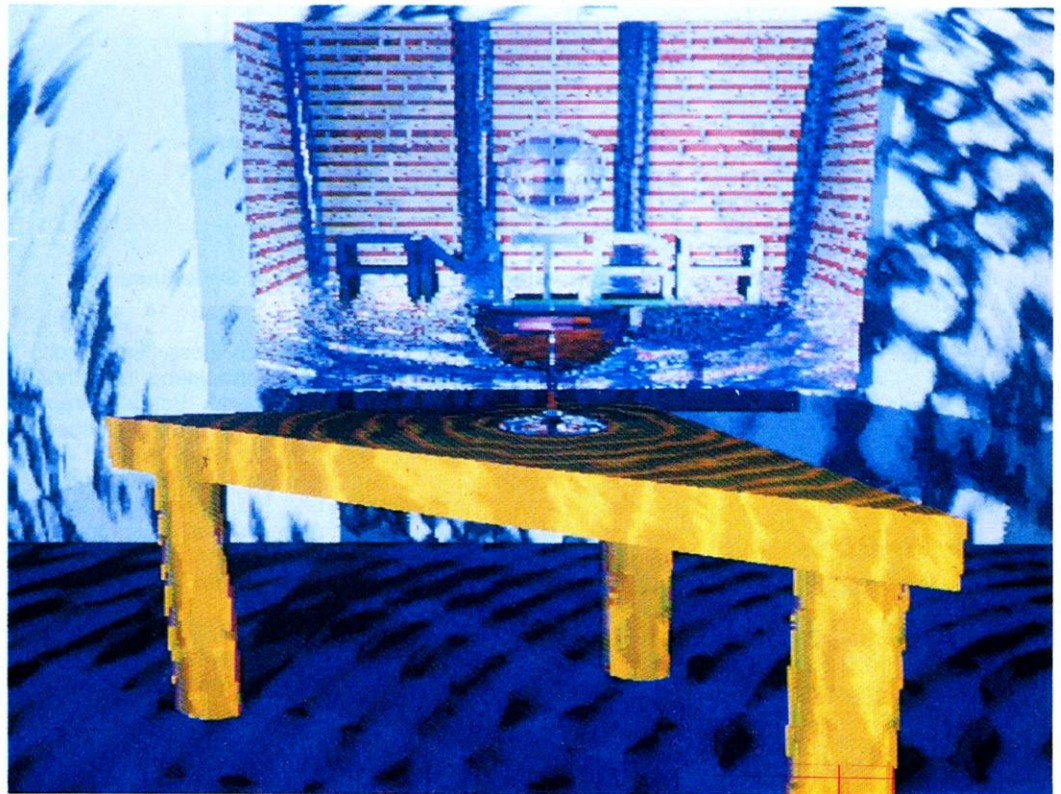


Bild 5. Schon etwas komplizierter: das Rotationsobjekt, hier ein Weinglas

»Ich habe hier ein total interessantes Bild, schau Dir das mal an« berechnen zu lassen. Dieses ist zwar die effektivste Lösung für Sie, aber nicht gerade die feine Art...

```
IFF size x y edge x y filename
"Bild.ilbm"
```

Hiermit läßt sich ein beliebiges IFF-Bild als Textur in die Szene laden. Der Size-Parameter bestimmt dabei die Ausmaße, die diese Texturemap haben soll. Mit dem Edge-Parameter bestimmt man die linke untere Ecke des Bildes, um es korrekt im Raum zu positionieren. Da wir das Bild komplett auf unsere Picture-Fläche mappen wollen, geben wir bei size die Ausmaße und bei edge die linke untere Ecke unserer Fläche an. Andernfalls bekämen wir nur einen Teil des Bildes zu sehen,

durch die später die Marmorturbulenz mehr oder weniger stark erfolgt. Der Parameter t definiert nun die Stärke der Interferenz an den mit x gesetzten Linien. »s« schließlich beeinflusst die Dicke der Interferenzlinien, welche beim Wert 2 halb so dick

## Sehr edel: Marmortextur

werden, bei 3 ein Viertel so dick und so fort. Richtwerte für x sind laut Anleitung 0.1 bis 1; für t 1 bis 3 und für s ebenfalls 1 bis 3. Die Erfahrung zeigt, daß sich mit völlig anderen Werten ebenfalls sehr interessante Marmorstrukturen erzeugen lassen.

## WOOD rgb d x t s

Ähnlich dem Marmor, läßt sich mit diesem Kommando ei-

staltet. Auch hier muß man vorher verschiedene Wellen definieren.

Bei den Objekten wird mit trans das Weinglas auf dem Tisch positioniert. Aus Rechenzeitgründen nehmen wir für das Weinglas nicht durchsichtiges Glas, sondern rötliches Metall. Auch der Tisch wird verändert. Er wird mit scale auf 70 Prozent seiner X-Achsenausdehnung und auf 50 Prozent seiner Z-Achsenausdehnung gestaucht. Anschließend wird er noch um 30 Grad gedreht. Die Message »Ende« beschließt dieses Eingabefile. Wir hatten oben schon gesehen, daß diese abschließende Message sinnvoll ist.

Falls der Speicherplatz nicht ausreicht, kürzen Sie die Anzahl der bei Rotationsobjekten angegebenen sides.



Sie haben nun eine Vielzahl der Befehle des Ray-Tracers erlernt und wissen sie hoffentlich auch zu gebrauchen. Dennoch möchte ich auch auf die restlichen Kommandos eingehen:

#### DIFFUSE s d o

In der Realität werden Objekte nicht nur von Lichtquellen beleuchtet, sondern auch von anderen mit Licht angestrahlten Objekten. Die Objekte reflektieren einen Teil des Lichts, mit dem Sie selbst angestrahlt werden. Daher ist ein Körper, der im Schatten einer Lichtquelle liegt, nie völlig im Dunkeln. Diese sogenannte diffuse Beleuchtung ist in fast keinem Ray-Tracer vorzufinden, da die Rechenzeit mit dieser Option immens ansteigt. Es müßten nun nicht nur Schnittpunktberechnungen zwischen den Objekten mit den Lichtquellen berechnet werden, sondern auch die Objekte untereinander in ihrer Wechselwirkung. Das ist weit aufwendiger als eine Reflexion, da bei der Spiegelung genau bestimmt werden kann, in welche Richtung der Lichtstrahl weiterläuft. Der DBW-Render hat diese Möglichkeit jedoch implementiert.

Leider ist sie so zeitaufwendig, daß man einen Amiga schon mehrere Jahre mit einem Bild beschäftigen könnte. Mit

## Objekte strahlen sich an

dem Parameter s wird die Anzahl der Lichtstrahlen pro Pixel eingestellt. Im Gegensatz zum Antialiasing, wo es nur zwei oder vier sind, sollte man hier um die hundert Strahlen angeben.

Die beiden anderen Parameter beschreiben, in wie viele Lichtstrahlen sich ein einzelner aufsplitten soll, wenn dieser ein Objekt getroffen hat, und wie oft sich dieser Vorgang wiederholen soll. Man bekommt dabei pro einzelнем Bildschirmpunkt gut und gerne ein paar Millionen Lichtstrahlen zusammen. Vielleicht wird man mit einem Hyper-Turbo-Amiga diese Funktion einmal vernünftig nutzen können.

#### FOCUS a d

Einer normalen Kamera nach, müßten bei einer gewissen Entfernungseinstellung näher und entfernter liegende Objekte zunehmend unschärfer erscheinen. Mit dem Befehl Focus kann man diesen Tiefenschärfe-Effekt nachvollziehen, wobei a die Blendenöffnung symbolisiert (Werte von 0,5 bis 2 sind o.k.) und d die Entfernung,

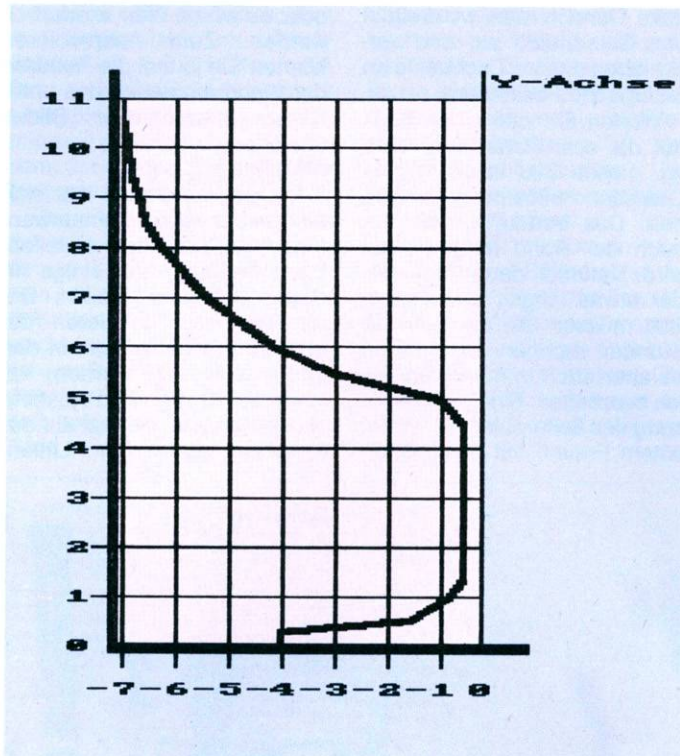


Bild 6. Schematische Darstellung einer Seite unseres Rotationsobjektes

bei der die Objekte scharf dargestellt werden. Auch dieser Parameter verschlingt immens Rechenzeit, da er nur mit einem Antialiasing von 20 bis 30 Rays pro Bildschirmpunkt zusammenarbeitet. Auch hier heißt es warten auf den Hyper-Amiga.

#### HAZE r g b d

Mit diesem Kommando läßt sich eine komplette Szene in Nebel hüllen. Man definiert mit »rgb« eine Farbe, die mit zunehmender Entfernung von der Kamera immer dominanter wird, bis alle Objekte darin verschwinden. Der Parameter d gibt die Entfernung von der Kamera an, ab der im Nebel nichts mehr zu sehen ist. Erfahrungsgemäß sollte dieser Wert immer etwas großzügiger gewählt werden, da sonst das Bild zu blaß wirkt.

#### INDEX r

Wie bei Wasser oder Glas, bei denen Lichtbrechungen auftreten, kann man mit »Index« das komplette Bild damit beeinflussen. Der Brechungsindex der Luft ist bekanntlich eins. Doch wie würde eine Szene unter Wasser aussehen? Dieser globale Brechungsindex verschafft Ihnen völlig neue Einsichten.

#### PEBBLE s z

Mit diesem Befehl ist angeblich Kieselstein zu simulieren. Allerdings sieht das Ergebnis nicht gerade danach aus. Daher weisen die Autoren des Ray-Tracers auch daraufhin,

daß sich diese Textur noch in der Entwicklung befindet. Es soll dem Schneeparameter ähnlich sein und winzige Teile des Objektes zufällig »heranzoomen«. Für s und z sollten 1.0 und 0.5 eingesetzt werden. Aber das steht auch in der Anleitung.

#### POLY name point x y z ... point x y z

Lernen wir noch ein weiteres Objektkommando kennen. Mit diesem kann man ein beliebiges Vieleck definieren. Die einzelnen Punkte geben dabei die entsprechenden Ecken des Polygons an. Da der Parser dieses Objekt für den Ray-Tracer in Dreiecke zerlegt, darf das Polygon nicht konkav sein. Ansonsten werden nach innen gewölbte Linien ausgefüllt gezeichnet. name gibt wie immer die Objekteigenschaft an.

#### PRODUCE histogramm

Anstatt ein Bild ausrechnen zu lassen, ist es mit diesem Be-

## Animation leichtgemacht

fehl möglich, bei gesetztem Antialiasing die Rechenzeit pro Pixel grafisch darzustellen. Anstatt eines Bildes erhalten Sie ein in grau gehaltenes 2D-Bild. Je heller die Punkte sind, desto mehr wird an dieser Stelle am Bild gerechnet.

#### RING name rad 0 r1 rad 1 r2

Mit diesem Objektbefehl sind

Ringe zu definieren. Der erste Radius kennzeichnet den inneren Radius, also die Größe des Lochs im Ring. Der zweite Radius bestimmt den äußeren Rand des Objektes. Die olympischen Ringe sind mit diesem Befehl keine Schwierigkeit mehr.

#### SUPERQ name scale x y z trans x y z eps e1 e2

Dieser etwas komplizierte und rechenaufwendige Objektbefehl läßt Körper nahezu beliebiger Form entstehen. Diese werden lediglich durch die zwei eps-Parameter (Epsilon) definiert, die eine Überlagerung zweier mathematischer Kurven beschreibt. Die genauen Formeln dieser Parameter können Sie der Anleitung des DBW-Renderers entnehmen. Des Weiteren enthält Ihre Ray-Tracing Diskette ein Bild mit verschiedenen Beispielen dieser Objekte.

Damit sind wir schon fast am Ende dieses Workshops. Zum Schluß möchte ich noch ein paar Tips für Animationen geben. Eine Animation ist ja nichts anderes, als eine Aneinanderreihung von mehreren Bildern, die sich relativ wenig voneinander unterscheiden. Für den Amiga gibt es schon verschiedene Programme, die aus einzelnen Bildern eine Animation erstellen können. Als Beispiel sei hier der Pageflipper Plus FX genannt, oder als Public Domain-Programm das »pilm« auf der Sculpt-Animate-Demodiskette. Die einzige Schwierigkeit bei all diesen Programmen ist, daß für die komplette Animation jedes Bild die gleichen 16 Grundfarben im HAM-Modus haben muß. DBW-Render läßt sich diesbezüglich aber nicht beeinflussen. Also müssen vorher alle Bilder auf die Grundpalette des ersten Bildes konvertiert werden. Das besorgen HAM-Malprogramme wie zum Beispiel PhotonPaint II. Leider kann nicht jedes Malprogramm die Bilder des DBW-Render lesen. Deluxe Photolab oder Butcher geben nur einen Fehler im Bildformat an. Hat man diese Hürde aber hinter sich gebracht, lassen sich sehr eindrucksvolle Animationen herstellen. Vorher sollte man sich überlegen, was bewegt werden soll. Das Beste ist eine zyklische Bewegung wie eine Drehung oder Rotation. Weiter sollten Sie darauf achten, daß die Bildunterschiede von Bild zu Bild nicht zu groß werden, da sonst die Animation nicht mehr als fließende Bewegung dargestellt wird.



Um den Text »Amiga« aus Bild X um seine eigene Achse rotieren zu lassen, fügen Sie nach dem trans-Kommando im Text-Befehl die Zeile

```
rot y 20
```

ein und speichern Sie das Eingabefile als »Bild2-1.inp«. Lassen Sie den Schriftzug in den weiteren Bildern um jeweils weitere 20 Grad rotieren. Mit 18 Bildern ist eine komplette Umdrehung des Textes erfolgt. Der Ray-Tracer eignet sich hervorragend für die realistische Bewegung von Wasser. Jetzt kommt endlich der Propagate-

Wert im Wave-Kommando zur Geltung. Ist der Wert auf 4 gesetzt, schwappt die Welle um genau eine Wellenlänge. Für unsere 18 Bilder müssen wir diesen Parameter um jeweils 4/18, also 0.22 verändern. In der Anleitung zum DBW-Render ist an dieser Stelle übrigens ein Fehler. Dort steht für eine Wellenlänge der Wert 1. Durch langes Herumprobieren sind wir endlich auf den richtigen Wert gekommen.

Ebenso ist die Richtung der Welle falsch bezeichnet. Mit den Eingaben von 0 nach 4 lassen wir die Welle rückwärts lau-

fen. Um die Welt wieder in Ordnung zu bringen, muß daher das erste Bild den Wert 4 bekommen, das zweite 3.78, das dritte 3.56 und so fort. Als weitere Möglichkeiten könnte man einen sich verändernden Brechungsindex einer Glaskugel animieren.

Der eigenen Fantasie sind kaum Grenzen gesetzt. Eine davon ist, daß sich bei bewegenden Wellen keine weiteren Texturen im Bild befinden dürfen, die auf den Wellendefinitionen aufbauen (Mottled diffuse, Marmor, Holz und ein fractales Gebirge), da sie sich sonst mit-

bewegen würde. Es sei denn, man beabsichtigt es.

Auf jeden Fall sollten Sie für die Erstellung einer Animation Geduld mitbringen. Die 18 Bilder der oben vorgeschlagenen Animation beschäftigen den Amiga schon einen Monat. Mit diesen rosigen Aussichten möchte ich nun diesen Workshop in der Hoffnung abschließen, daß er Ihnen einen leichten Einstieg in die Welt des Ray-Tracing ermöglicht hat.

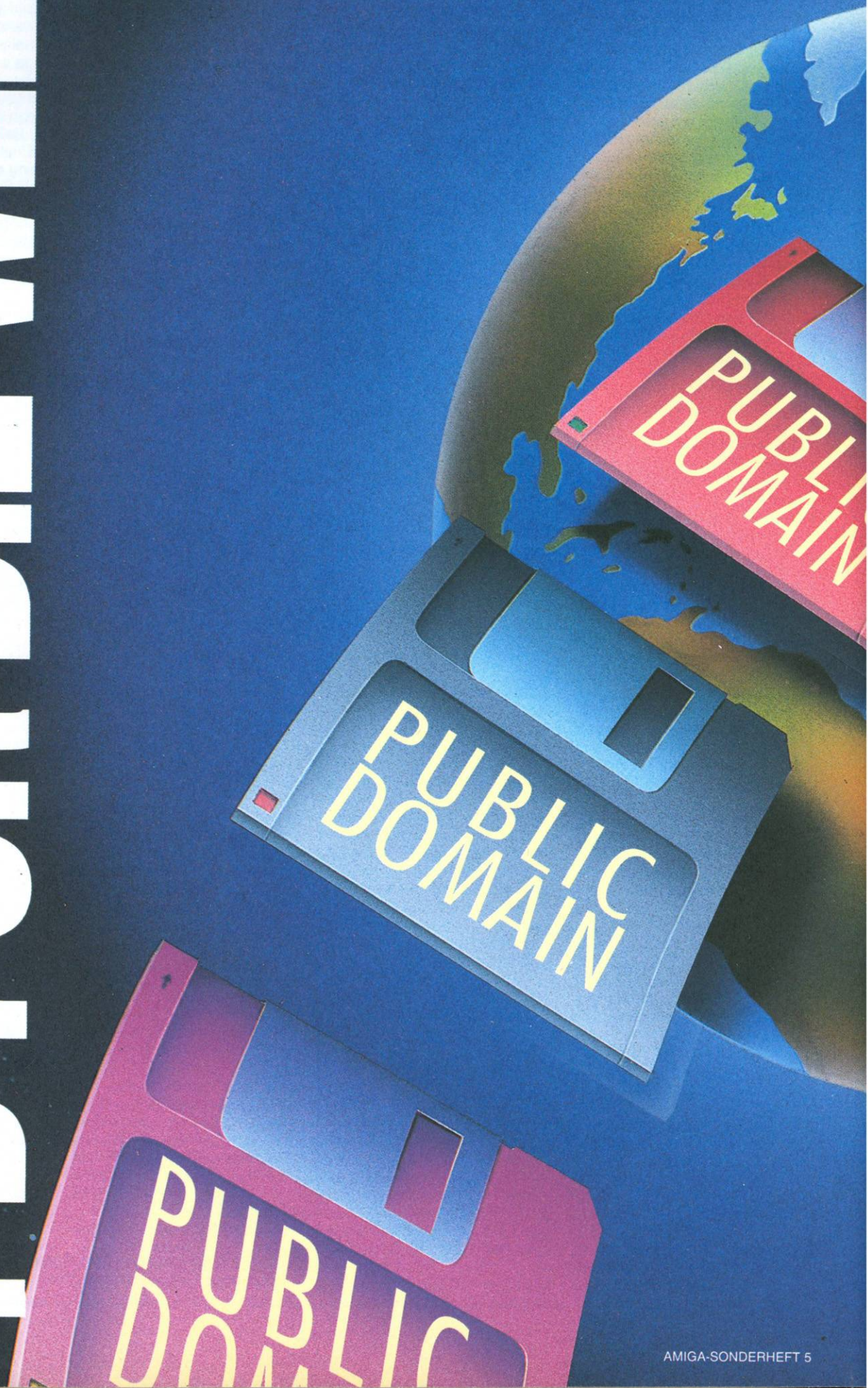
Viel Spaß, und sorgen Sie bei einem Amiga 500 für die nötige Kühlung,  
Ihr Marco Vitolini-Naldini. rs

```
/******  
* Bild4.inp - by Marco Vitolini-Naldini 1989 *  
* (Maillingerstr.20, 8 Muenchen 19) *  
* *****  
*/  
  
Macros:  
Macro Picture  
quad Bild point -35 25 -50  
point -35 75 -50  
point 35 25 -50  
done Picture  
  
Macro Weinglas  
obj Glas point -7 11.0 0  
point -7 10.5 0  
point -6.9 10.0 0  
point -6.85 9.5 0  
point -6.7 9.0 0  
point -6.5 8.5 0  
point -6.2 8.0 0  
point -5.85 7.5 0  
point -5.4 7.0 0  
point -4.9 6.5 0  
point -4.2 6.0 0  
point -3.0 5.5 0  
point -1.0 5.0 0  
point -0.5 4.5 0  
point -0.5 1.5 0  
point -.75 1.0 0  
point -1.5 0.5 0  
point -4.0 0.2 0  
point -4.2 0.0 0  
sides 64  
close 0 1  
done Weinglas  
  
Macro Boden  
quad Boden point -500 0 -500  
point 500 0 -500  
point -500 0 500  
done Boden  
  
Macro Wand  
quad Marmor point -500 0 -60  
point 500 0 -60  
point -500 200 -60  
done Wand  
  
Macro Bein  
obj Holz point -5 0 0  
point -5 25 0  
sides 32  
done Bein  
  
Macro Tisch  
tri Holz point 0 30 -30  
point 50 30 30  
point -50 30 30  
tri Holz point 0 25 -30  
point 50 25 30  
  
quad Holz point -50 25 30  
point 0 30 -30  
point 0 25 -30  
quad Holz point 50 30 30  
point 0 30 -30  
point 0 25 -30  
quad Holz point -50 30 30  
point -50 25 30  
point 50 30 30  
Bein trans -40 0 25  
Bein trans 40 0 25  
Bein trans 0 0 -30  
done Tisch  
  
Parameters:  
scan from 0 to 564 rows 720 columns 352 mode 0 bpp 4  
look from 0 50 100 to 0 -20 -150 up 0 1 0  
background .3 .5 .6  
ambient .15  
  
textures:  
wave -70 -30 50 9 .3 1 0  
wave 80 20 20 6 .4 1 0  
wave 29 -50 -40 11 .3 1 0  
wave -30 40 -5 8 .4 1 0  
wave -10 30 -30 12 .5 1 0  
iff size 70 50  
edge -35 25  
filename "df1:bild2.ilbm"  
marble .1 .1 .1 .1 3 2  
wood .8 .4 0 .7 2 3 2  
  
lights:  
spherical 1 1 1 100 100 100 150 1  
spherical 1 1 1 -50 50 150 200 1  
spherical .7 .3 .6 10 35 0 15 1  
  
attributes:  
attribute Bild  
texture 6 0 0 1 0 0 0 0 0 0 0 0  
attribute Boden  
texture 3 0 .3 1 0 0 0 0 0 0 .3 .1 .6  
attribute Marmor  
texture 20 0 0 1 0 0 0 0 0 0 .7 .7 .7  
attribute Holz  
texture 40 0 0 1 0 0 0 0 0 0 .4 .25 0  
attribute Glas  
texture 0 0 .4 1.3 .7 .7 .7 0 0 0 0 .2 .3  
  
objects:  
Picture  
Weinglas trans 10 30 0  
Tisch scale .7 1 .5  
rot y 30  
  
Boden  
Wand  
  
message "Ende"
```

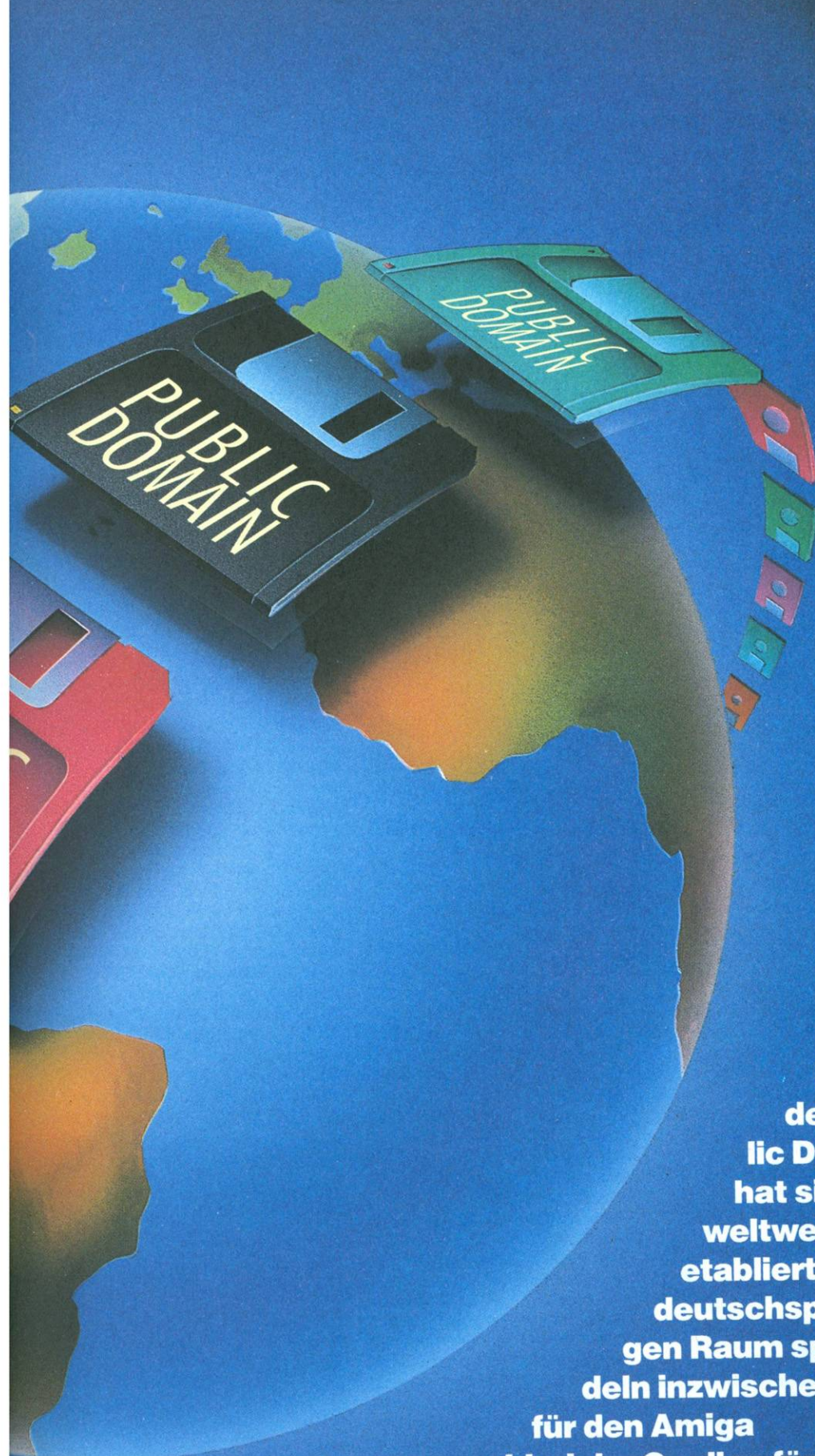
Listing 4. Selbst komplexe Szenen sind einfach und schnell zu programmieren



# PD FÜR DIE WELT







**D**ie Bewegung der freikopierbaren Software stammt aus den Vereinigten Staaten. Sie wurde in der Zeit geboren, als jede Art von kommerziellen Programmen noch sündhaft teuer war. Aus diesem Grund entstand vor allem an den Universitäten eine Idee: selbstgeschriebene Programme sollten allen Interessenten zur Verfügung gestellt werden. Dieser Grundsatz hat bis heute Bestand. Inzwischen hat sich auch im deutschsprachigen Raum eine rege Szene etabliert. Eigene Serien mit teilweise sehr speziellem Angebot sind entstanden.

Ein Blick in den Anzeigenteil der Fachzeitschriften zeigt die Fülle des Angebots. Das hat natürlich auch Nachteile: Die Auswahl einer geeigneten Diskette fällt schwerer denn je. Auf verschiedenen Serien sind Programme mehrfach vorhanden, Korrekturen (sogenannte Updates) verwirren zusätzlich. Außerdem gilt es vor der Bestellung genau die Preise zu vergleichen. Auch wenn PD-Software keinen kommerziellen Vorteil für den Vertreiber bringen darf – das legen die Vermerke der Programmautoren eindeutig fest – gibt es Preisunterschiede bei der Bestellung. Den Anbietern entstehen natürlich Kosten, die vom Käufer gedeckt werden müssen. Dazu gehört der Materialpreis der Diskette, die Arbeitszeit beim Kopieren, Verpackung und Versandkosten. Die Kalkulationen der Verreiber sind dabei sehr unterschiedlich. Ein niedriger Preis bringt allerdings nicht immer nur Vorteile für die Kunden. Über alle angesprochenen Details geben die Gespräche mit Anbietern in den folgenden Artikeln ausführlich Auskunft.

Eine Hoffnung verbinden wir mit diesen Szene-Berichten: Stellen auch Sie gute Programme dem PD-Pool in irgendeiner Form zur Verfügung. Die Vorteile liegen auf beiden Seiten. Sie erfahren, wie gut das selbstgeschriebene Programm wirklich ist. Der Test durch eine große Zahl von Anwendern wird das schnell zeigen. Rückmeldungen der aufgetretenen Fehler helfen Ihnen, Programme zukünftig noch besser zu gestalten. Software-Häuser könnten auf Ihre Produkte aufmerksam werden, vielleicht ist das der Anfang einer steilen Karriere.

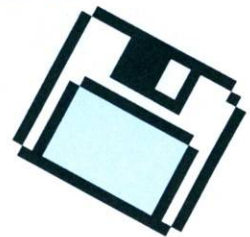
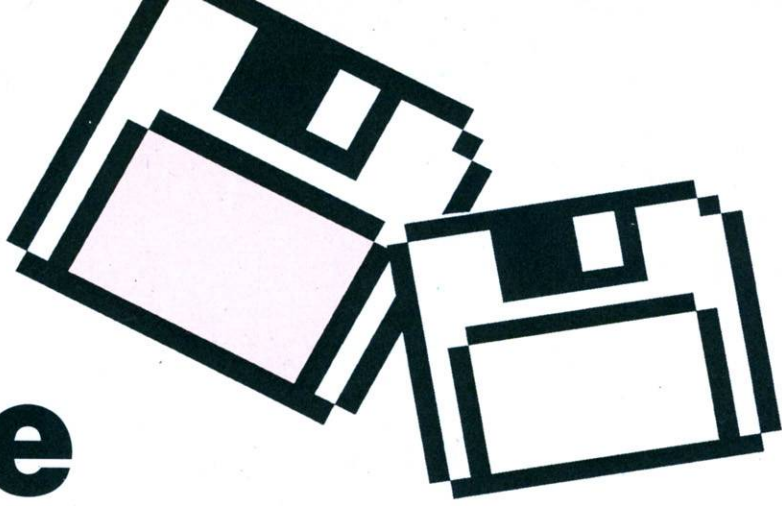
Die Vorteile für die Anwender sind ebenso deutlich zu erkennen: Ein immer größeres Angebot hilft beim Aufbau einer wahren Schatzkiste. Der Umgang mit dem Amiga wird damit immer komfortabler – und das ohne sprunghaft steigende Kosten für kommerzielle Software.

rs

**Die Idee der Public Domain hat sich weltweit etabliert. Im deutschsprachigen Raum sprudeln inzwischen für den Amiga zahlreiche Quellen für PD-Software. Anbieter und Serien aus der Bundesrepublik, Österreich und der Schweiz stellen wir Ihnen hier vor.**



# Bunte PD-Szene in Deutschland



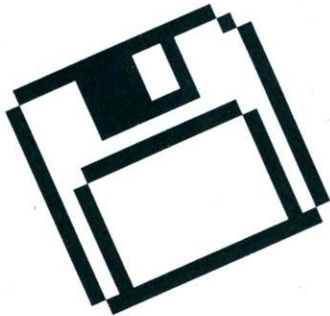
**P**ublic Domain-Software gibt es für die AMIGA Familie mittlerweile wie Sand am Meer. Ebenso zahlreich scheint in Deutschland die Flut an PD-Anbietern zu sein, die monatlich den Anzeigenmarkt überschwemmen. Welche Motive sind es, die einen derartigen Boom aufkommen lassen? Wie kommt es zu Preisdifferenzen unter den einzelnen Vertreibern, die ja eigentlich nur kostendeckend arbeiten sollten? Wer steht hinter den SOFT-Pseudonymen, die mit auf dieser Welle des »öffentlichen Besitzes« (Public Domain wörtlich übersetzt) schwimmen? Diesen und anderen Fragen sind wir nachgegangen. Einem PD-Vertreiber boten wir zusätzlich Gelegenheit, sich dem breiten Amiga-Publikum einmal vorzustellen.

Wir sprachen mit Markus Scheer, in Insiderkreisen besser bekannt unter seinem Firmennamen »RUHRSOFT«. Er war in Deutschland einer der ersten, die den Sprung in den PD-Pool wagten. Markus ist 19 Jahre jung und kann sich trotz seines Alters heute zu den etablierten Anbietern zählen. Durch seriösen und kundenfreundlichen Vertrieb hat er sich in der Szene einen Namen gemacht. Angefangen hat alles vor zirka zwei Jahren, als Markus, wie viele andere über VC 20 und C64 zum Amiga wechselte. Seine schulische Laufbahn hat er mit dem Fachabitur abgeschlossen. Der damit verbundene finanzielle Engpaß zwang ihn zunächst, auf Programme aus dem Low-Cost-

**Die deutsche PD-Szene für den Amiga schillert in allen Farben. Rosa Zeiten, goldene Aussichten und schwarze Schafe. Wir sprachen mit einem der etablierten deutschen PD-Anbieter, Markus Scheer von RUHRSOFT.**

Von Thorsten Froese

Budget zurückzugreifen. So entstand durch Kaufen und Tauschen eine Programmsammlung, die bald alle erhält-

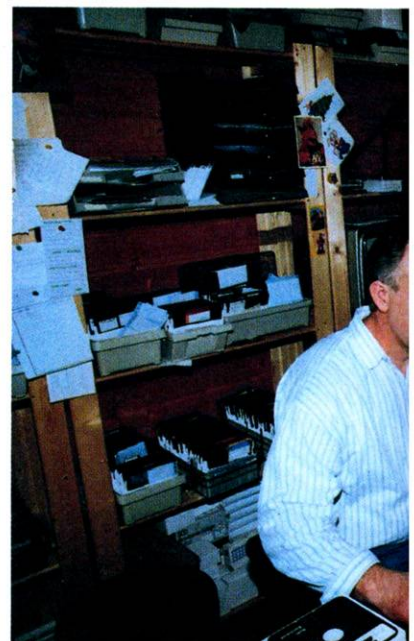


lichen PD-Serien und -Disketten umfaßte. Markus hatte die Absicht, ein Studium der BWL oder der Informatik aufzunehmen. Aus dem Gedanken heraus, das Hobby mit dem späteren Beruf zu kombinieren, entstand die Idee, einen PD-Vertrieb zu gründen. Zur Werbewirksamkeit fehlte natürlich noch ein Firmennamen. Obwohl Markus Scheer zu seiner Heimatstadt Bochum steht, wurde das ursprüngliche BOCHUM-SOFT kurzerhand zu RUHRSOFT umgetauft. Die Amiga-

User nahmen die Bereitstellung der günstigen Programmpakete aber in einem derartigen Ausmaß an, daß er allein die Kundenaufträge eines einzelnen Monats nicht bewältigen konnte. Aus diesem Grunde kann man heute von einem Familienunternehmen sprechen, da sich sowohl die Eltern als auch die Schwester im Unternehmen engagieren (Bild 1). Jedem von ihnen kommt dabei eine bestimmte Tätigkeit zu, denn schließlich geht eine solche Aufgabe über das Kopieren von Disketten hinaus. So übernimmt Mutter Scheer beispielsweise das Kopieren am Vormittag, während Vater Scheer sich dann gegen Abend um das Verpacken und den Versand kümmert. Markus Schwester ist damit beschäftigt, die drei Katalogdisketten auf den neuesten Stand zu bringen, die RUHRSOFT dem Kunden inzwischen zur Auswahl seiner Programmwünsche anbieten kann. Nach eigenen Angaben sind es bis zu 50 neue Disketten im Monat, die sortiert, archiviert und katalogisiert werden müssen (Bild

2). In der Regel bedeutet dies für jeden der Familie Scheer täglich zirka drei Stunden Freizeit in den Dienst der Kundschaft zu stellen. Eine Menge Arbeit also, die übrigens auch während der Ferienzeit nicht zum Stillstand kommen kann, da bei einer derartigen Nachfrage ein Pausieren anschließend das absolute Chaos bedeuten würde. Die sonnigen Urlaubstage im Jahr werden somit meistens getrennt verbracht.

Es gibt allerdings noch einige undankbare Nebeneffekte, womit wir bei den Preisen angefangen wären. Immer mehr PD-Anbieter versuchen, sich ge-



**Bild 1.** Bei der von Markus Scheer aus Bochum hilft die ganze Familie



gegenseitig zu unterbieten. Bei RUHRSOFT werden 3 1/2-Zoll-Disketten zum Preis von DM 4,50 Mark und 5 1/4-Zoll-Disketten für 3,50 Mark pro Stück angeboten. Diese Preise können bei größeren Abnahmemengen und durch Abos auf 4 beziehungsweise 3 Mark gesenkt werden. Seine Verkaufszahlen bezifferte Markus Scheer in unserem Gespräch auf 3000 bis 4000 Disketten pro Monat.

Bei dieser Gelegenheit ließ der RUHRSOFT-Gründer auch erkennen, daß der Verkauf von PD-Software nicht als überaus lohnenswerte Tätigkeit angesehen werden sollte. Nicht selten kommt es vor, daß Computerbesitzer angesichts dieser Verkaufszahlen darin einen finanziellen Nebenerwerb sehen. Laut Aussage von RUHRSOFT sind Fälle von unvorbereiteten Anbietern bekannt, die nach einigen Wochen nicht mehr in der Lage waren, alle Bestellungen zu bearbeiten. Die Konsequenz ist, daß Kunden bei schlechter oder sogar ausbleibender Lieferung verunsichert werden. Weitere Bestellungen überlegen sich diese potentiellen Interessenten dann natürlich gründlich. Durch rückläufige Zahlen werden die PD-Händler dann wiederum gezwungen, die Preise anzuheben. Trotzdem werden von einigen Anbietern Preise von weniger als 3 Mark pro Diskette angeboten.

Wie können solche Preise eigentlich realisiert werden? Markus Scheer kann sich diese sagenhaft knapp kalkulierten Angebote nicht erklären. Qualität hat nunmal ihren Preis. Der

RUHRSOFT-Gründer verwendet nur doppelseitige Disketten, die beim Kopieren mit zweimaligem Verify dupliziert werden. Nur so kann eine möglichst hohe Datensicherheit gewährleistet werden.

Weiter werden firmen- und programmbezogene Aufkleber benötigt, die zusammen mit der oder den Disketten und einem

sein. Diese Weisheit versucht auch RUHRSOFT gerecht zu werden. Um möglichst günstig ausländische Serien zu erwerben, wechselt man sich bei Einkäufen mit Partnern ab. Nicht selten kommt es vor, daß beispielsweise für eine neu erschienene Diskette aus der englischen Faug-Serie ein Preis von vier Pfund bezahlt wird. Diese



**Bild 2. Die Disketten stapeln sich im provisorischen Lager. Bis zu 50 neue werden monatlich ins Angebot genommen.**

Lieferschein in einer Versandtasche landen. Diese wird noch am gleichen Tage des Auftrages an den Kunden zurückgeschickt. Weiterhin gibt es unter Tausenden von Disketten natürlich auch einige Ausfälle, die im nachhinein zur Zufriedenstellung der Kunden ersetzt werden müssen.

Einige schwarze Schafe, die sich unter den Kunden tummeln, ärgern die Familie Scheer am meisten. So ist es bereits vorgekommen, daß ein Stammkunde ausnahmsweise eine Lieferung von 100 Disketten per Rechnung erhielt, was normalerweise per Nachnahme geschieht. Auf die ausstehende Überweisung wartet RUHRSOFT allerdings heute noch. Ein weiteres, immer wieder auftretendes Ärgernis, ist, daß fehlerfreie Disketten mit dem Vermerk »defekt« zurückgeschickt werden. Der Vertreter wird gleichzeitig aufgefordert, doch eine andere als die ursprünglich bestellte Diskette zurückzusenden. Sicherlich kann man diese Tatsache nur als ziemlich dreiste Form der Bereicherung bezeichnen. All dies muß natürlich bei der Kalkulation des Preises berücksichtigt werden. RUHRSOFT rechtfertigt deshalb seine Preise mit der Qualität aller erbrachten Leistungen.

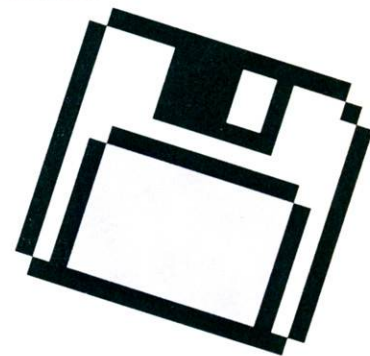
Wer konkurrenzfähig sein will, der muß auch »up to date«

Kosten lassen sich natürlich nur tragen, wenn sich möglichst viele Interessenten zusammenschließen - oder wenn eben abwechselnd eingekauft wird. Eine Alternative hierzu ist die Produktion von eigenen Serien. Als Beispiel sei die RUHR- (Grafik- und Soundutilities) und RPD-Serie genannt, die von RUHRSOFT selbst zusammengestellt wurden. Völlig neu hingegen wird eine neue Serie sein, die RUHRSOFT in den nächsten Wochen unter dem Label DDD (Die Deutsche Disk) veröffentlichen wird. Bei dieser Serie sollen alle Programme und Textfiles komplett in deutscher Sprache veröffentlicht werden. Sicherlich eine erfreuliche und begrüßenswerte Nachricht für alle PD-Sammler in den deutschsprachigen Ländern.

Zusammenfassend könnte man den Eindruck gewinnen, daß der Vertrieb von PD-Software für den Verbraucher zwar eine erfreuliche, für den Anbieter aber nicht gerade lohnenswerte Tätigkeit darstellt. Daß dem natürlich nicht so ist, dürfte klar sein. Auch für den Anbieter bleibt der Kauf und Verkauf beziehungsweise Tausch selbstverständlich nur interessant, wenn man selbst auch ein wenig davon profitieren kann. In Anbetracht der monatlichen Verkaufsstückzahlen gestaltet sich der Vertrieb auch für den Händler bestimmt noch positiv.

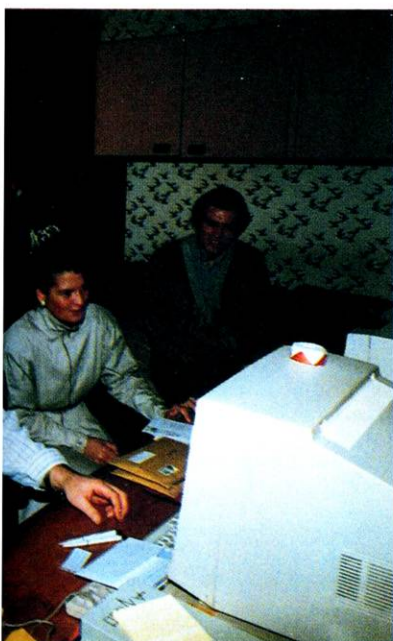
Profit muß aber eben nicht nur Kommerz bedeuten. Am Beispiel von RUHRSOFT wird dies besonders deutlich. Für Markus Scheer stand von Anfang an das ideelle Ziel im Vordergrund, das er mit der Unterstützung seiner Familie bis heute kontinuierlich verfolgt hat.

Die Erfahrungen, die er dabei in den letzten 2 Jahren gesammelt hat, erlauben ihm heute einen Schritt weiterzugehen. STARBYTE heißt ein neues Projekt, das Markus zusammen mit zwei Freunden ins Leben gerufen hat. Durch STARBYTE werden so bekannte Produkte, wie »MINIGOLF PLUS« oder »XORRON 2081« bereits vertrieben. Der internationale Durchbruch wird wohl nicht mehr lange auf sich warten lassen, denn durch STARBYTE wird mit »LEONARDO« seit kurzem ein hervorragend programmiertes Geschicklichkeitsspiel vertrieben, das durch sehr gute Grafiken und tollen Sound überzeugt. Mit »AUNT ARTIC«, einem komplexen »Jump and Run«-Spiel, wird STARBYTE die Palette seiner Produkte um ein weiteres gutes Programm aufwerten. Dieses Spiel befindet sich zwar noch in der Testphase, hinterläßt aber schon jetzt einen vielversprechenden Eindruck.



Wir werden die Entwicklung dieses Newcomers natürlich interessiert verfolgen. Mit RUHRSOFT stellte sich uns ein PD-Vertreiber vor, der in bezug auf seine Leistungen Vorbildliches leistet. Von diesem PD-Anbieter kann man sich nur wünschen, daß er weiterhin seine Linie beibehält. Unsere Amigas werden so auch in Zukunft mit preisgünstiger Software versorgt werden. Wir wünschen Markus Scheer für seine weitere Laufbahn viel Erfolg und hoffen, daß er uns mit RUHRSOFT und STARBYTE in der nächsten Zeit noch viele interessante Projekte beschert. rs

Adresse: RUHRSOFT, M. Scheer, Kapellenweg 42, 4630 Bochum 5



**gegründeten Firma RUHRSOFT eifrig mit**



**W**ie würden Sie reagieren, wenn Ihnen Ihr Software-Händler mit dem neuesten Compiler gleich die Adressen der letzten fünf Käufer mitliefern würde? Sauer? Gleichgültig? Oder würden Sie einen Klub gründen?

Die letzte Alternative ist sicher die produktivste, wie das Beispiel von Amok zeigt. Die Gruppe von sechs Programmierern hat sich auf diesem Weg gefunden. Seitdem ist daraus eine sehr fruchtbare Zusammenarbeit entstanden.

Vielleicht ist Ihnen schon einmal eine Diskette mit dem merkwürdigen Namen »Amok # x« in die Hände gefallen. Dann dürfen Sie bemerkt haben, daß die

se Reihe zu den wenigen gehört, die Ihre Disketten komplett mit deutscher Dokumentation ausliefern. Für andere Programmierer gehört es meist zum guten Ton – auch wenn sie aus dem deutschsprachigen Raum stammen – ihre »doc«-Files nur in der (vermeintlichen) Muttersprache der Amiga-Gurus abzufassen, also in Englisch. Nicht so bei Amok. Alle Anleitungen sind in Deutsch gehalten (eine englische Anleitung ist meist zusätzlich enthalten).

Die Amok-Reihe umfaßt inzwischen 18 Disketten. Zu allen Programmen sind die sehr gut dokumentierten Sourcecodes in Modula enthalten.

Neben Toolboxes mit vielseitig verwendbaren Bibliotheken gibt es jede Menge Demos, Uti-

# Amiga lä

**Die PD-Szene wird immer vielseitiger. kleine Teilbereiche. Einer davon ist die sehr  
Jahr ist hier der »AmigaMODulaKlub«**

lities und Anwendungsprogramme, welche die Leistungsfähigkeit des M2Amiga-Compilers eindrucksvoll demonstrieren (siehe Tabelle 1).

Aber was bringt sechs junge Leute auf die Idee, sich mit so großem Engagement dem PD-Konzept zu verschreiben? Programmautoren verdienen mit PD kein Geld, es handelt sich

eher um ein Verlustgeschäft. Schließlich müssen Literatur, Disketten und Verschleißmaterial für den Computer erstanden werden. Die dabei hergestellten Programme dagegen sind kostenlos. Es sprechen also genügend »vernünftige« Gründe gegen ein Engagement für die lobenswerte Idee der Public Domain-Software.

## Das aktuelle Amok-Programm

Zu jedem Programm finden Sie hinter dem Titel den Namen des Autors und eine kurze Beschreibung der Programm-Funktionen. Wenn nur ein Kürzel angegeben ist, stammt das Programm direkt von Amok. Dabei haben die Kürzel folgende Bedeutung:

Kürzel	Name	Kürzel	Name
[bne]	Nicolas Benezan	[brk]	Bernd Kirschner
[fbs]	Fridtjof Siebert	[pit]	Pit Burkhardt
[fst]	Frank Staudte	[mif]	Michael Frieß

### Das komplette Programm:

Programm	Autor	Kurzbeschreibung
----------	-------	------------------

## Amok # 1

Fish	[fbs]	Modula-Umsetzung des bekannten Bob-Demo von Cathy J. Wagner.
CtrlIntuition	[fbs]	Schaltet Intuition ab.
DME	Matthew Dillon	Die Version 1.28 des programmierbaren Editors (siehe Seite 61).
DME-m2c	[fbs]	Eigene Tastenbelegung für den DME, um Compiler und Linker direkt aus dem Editor zu starten
Sundown	[pit]	Copperlist-Demo (muß man gesehen haben)
M2Error	[fbs]	Verbesserte Version von M2Error
DosTime	[pit]	Prozedur zum Lesen und Schreiben des System-Datums
Sprites	[pit] und [fst]	Ein Demo zum Gebrauch von Hardware-Sprites
Interrupt	[fbs]	Ein Demo zu Interrupts
VSpriteDemo	[fbs]	Umsetzung der »VSprites« aus dem Amiga-Programmier-Handbuch von Markt & Technik
GameSupport	[mif]	Eine brauchbare Joystick-Routine, die Sie in eigene Programme einbinden können.
DragNumber	[pit]	Prozeduren zur Darstellung von sogenannten 7-Segment-Zahlen.
OverScn	[fbs]	Zeigt ein Problem mit dem Overscan-Modus
Trouble	[fbs]	Erzeugt HexDump und macht diesen Vorgang wieder rückgängig.
HexDump/ReDump	[fbs]	Eine Umsetzung des Programm-Beispiels im ROM-Kernel Reference Manual (Libraries & Devices)
ShowEvents	[fbs]	Startet CLI-Batchfiles per Mausklick von der Workbench
WBExecute	[pit]	

## Amok # 2

IFFLoad	[fbs]	Ladeprozeduren für ILBM-Format (ermöglichen Color-Cycling, Overscan, und vieles mehr).
M2Code	[fbs]	Bindet Assembler-Routinen in Modula-Programme ein.
SuperLists	[bne]	Unterstützende Routinen für File- und String-Requester.
IntuiStruct	[bne]	Bietet vereinfachte Routinen zur Initialisierung von Intuitionstrukturen
halfBrite	[pit]	Demo zum ExtraHalfbrite-Modus (64-Farben-Modus)
H.A.M	[pit]	Demo zum HAM-Modus (4096 Farben gleichzeitig darstellbar).
MODIV	[fbs]	»MOD« und »DIV«-Prozeduren, die mit LONGCARDS zusammenarbeiten.
DMEError	[fbs]	Dieses Programm zeigt Fehlermeldungen von »M2Amiga« im DME an. Damit können Sie sich Ihre eigene »integrierte Oberfläche« schaffen.

## Amok # 3

IFFToCode	[pit]	Generiert Modula-Code für Image-Daten aus IFF-Brushes (die Sie vorher beispielsweise mit »DPaint« oder einem ähnlichen Programm erzeugt haben).
IFFLoad_1.1	[fbs]	Update des Programms von Amok # 2. Der Ladeteil ist jetzt in Assembler geschrieben und dadurch etwa dreimal schneller als die alte Version.
RecordInput	[fbs]	Zeichnet Tastatur- oder Mausingaben auf und macht sie per Tastendruck aufrufbar.
Gels	[pit]	Eine Art »GelTools« für Modula-Programmierer.
DosSupport	[bne]	Programmierhilfe zur einfachen Handhabung von BCPL-Pointern und Devicelists.
IntuiStruct1.3	[bne]	Update der Intuition-Unterstützung von der Amok # 2.

## Amok # 4

Checker	[pit]	Demonstriert eine neue Form der Informationsdarbietung in Kraftfahrzeugen.
---------	-------	--



## uoft Amok

**Viele Gruppen spezialisieren sich auf lebendige Modula-Szene. Seit fast einem (AMOK) aus Stuttgart sehr aktiv.**

Bei Amok sieht man diesen Sachverhalt etwas anders. Diese sechs Programmierer wollen mit ihrer Arbeit in erster Linie die Verbreitung von Modula unterstützen. Diese moderne Sprache unterstützt den Programmierer mit vielen interessanten Fähigkeiten, die in anderen Sprachen nicht oder nur unvollständig vorhanden sind

(siehe auch Seite 35). Die Amok-Disketten erfreuen den engagierten Modula-Programmierer mit allen nur erdenklichen Hilfeleistungen. Dazu gehören neben Programmierwerkzeugen (Editor, Make-Utility und Rück-Übersetzer für »obj«-Dateien) vor allem die Dokumentationen der Sourcecodes. Diese sind größtenteils

in lockerem, leicht verständlichem Stil geschrieben und enthalten alle wichtigen Informationen zu den jeweiligen Programmen. Alle verwendeten Tricks und Kniffe werden so einem großen Publikum zugänglich gemacht.

Die eigenen Programme bei möglichst vielen Leuten verbreitet zu wissen, bedeutet dieser Gruppe mehr als der kommerzielle Erfolg. Sicher ist mit kommerzieller Software-Entwicklung auch wesentlich mehr Aufwand und ein gewisses Risiko verbunden. Möglicherweise werden einzelne Mitglieder der Gruppe in Zukunft einen solchen Schritt in Angriff nehmen. Dies wird sich dann allerdings außerhalb des bisherigen Amok-Konzepts abspielen. Alle Amok-Leute hatten bereits eini-

ge Vorerfahrung mit Computern. So arbeiteten drei vor ihrer Amiga-Zeit mit einem C64, zwei programmieren auf einem PC.

Bei der bisherigen Qualität der Amok-PD-Serie darf man auf die zukünftigen Produkte dieser Software-Schmiede gespannt sein. Sie läßt für Anwender ebenso wie für Modula-Programmierer viel Interessantes erwarten.

Alle AMOK-Disketten erhalten Sie bei folgender Adresse: AMOK - Amiga Modula Klub, Bernd Kirschner, Gottlob-Grotz-Straße 24, 7120 Bietigheim-Bissingen

Fragen zu den einzelnen Programmen beantworten auch die jeweiligen Autoren: Nicolas Benezan, Postwiesenstr. 2, 7000 Stuttgart 60

Fridtjof Siebert, Nobileweg 67, 7000 Stuttgart 40

Michael Frieß, Kernerstr. 22a, 7000 Stuttgart 1  
Frank Staudte, Hasenbergstr. 91, 7000 Stuttgart 1

Pit Burkhardt, Stettiner Str. 25, 7030 Böblingen

### Amok #5

Graphics	[brk]	Bilder im IFF-Format. Mit einem Doppelklick können die einzelnen Bilder angezeigt werden.
PrinterSupport	[fbs]	Routinen, die die Programmierung des »printer.device« erleichtern.
Print	[fbs]	Sendet ein Textfile zum Drucker. Dies ersetzt den CLI-Befehl »TYPE >PRT: <Name>« durch eine leistungsfähigere Variante. (Die Perforation wird nicht bedruckt.)
MemSystem	[bne]	Leistungsfähiger Ersatz für das Standard-Modul »Heap« mit erweiterten Multitaskingfähigkeiten.
SuperLists1.2	[bne]	Update des Requester-Tools von Amok #2, in dieser Version werden Scrollbar-Requester unterstützt.
PrintIt!	[fbs]	Hardcopyprogramm für IFF-Bilder.
IconSupport	[fbs]	Bibliothek zum Erzeugen eigener Icons. Eine entsprechende Prozedur wird exportiert.
CreateAmok-Disk	[fbs]	Erzeugt Amok-Icons für Ihre Dateien, Directories und Disks.

### Amok #6

SoftScroll	[fst]	Bibliothek für Modula-Programmierer. Exportiert eine Prozedur zum Scrollen einer BitMap.
MathLib	[bne]	Mathematik-Bibliothek für Vektoren- und Matrizenrechnung.
IFFSupport	[fbs]	Bibliothek für Modula-Programmierer. Exportiert Prozeduren zum Laden und Speichern von ILBM-Bildern.
ListAll	[bne]	Erweiterung des CLI-Commandos »LIST«. Alle Subdirectories werden untersucht.

### Amok #7

AVL	[mif]	Implementiert den dynamischen Datentyp »AVL-Baum«.
List	[mif]	Implementiert den dynamischen Datentyp »Liste«.
Queue	[mif]	Implementiert den dynamischen Datentyp »Schlange«.
Stack	[mif]	Implementiert den dynamischen Datentyp »Stapel«.
ProgInfo	[fbs]	Liest die Informationen über Autor und Programm aus den AMOK-Source-Codes (oder anderen, die sich an das gleiche Format halten).

SoundTask	[fbs]	Demo, das einen neuen Task aufsetzt, um Sound abzuspielen.
M2Tests	[mif]	Hinweise auf einige Fehler in den Bibliotheksmodulen von »M2Amiga«.
PP	Achim Siebert	Utility zum Drucken von Texten auf einem Epson LQ-800.
Reminder	Achim Siebert	Ein einfacher Wecker, der Sie aus langen Programmierernächten aufschreckt.
WarpText	[fbs, Bill Kelly]	Schnelle Textausgaberoutinen.
MuchMore	[fbs]	Textanzeige mit einer sehr schönen Scroll-Routine.
NTSC	[fbs]	Bibliothek für Modula-Programmierer. Exportiert eine Prozedur, die zwischen PAL/NTSC-Screens unterscheidet.

### Amok #8

Printer-Support2.0	[mif]	Routinen, die die Programmierung des »Printer.device« erleichtern.
CreateAmok-Disk_1.1	[fbs]	Erzeugt Icons für Ihre Dateien, Directories und Disks. Verarbeitet jetzt auch verschiedene Suffixes - Update von Amok #5.
IDCMP	[bne]	Bibliothek für Modula-Programmierer. Exportiert Prozeduren zur einfachen Handhabung des »Intuition Direct Communications Message Port«.
SpaceDemo	[fbs]	Ein 3D-Demonstrationsprogramm.
IFF8SVXLoad	[fbs]	Enthält Lade- und Abspielprozeduren für gesampelte Sounds.
Blitter	[fbs]	Beispiel für die direkte Blitterprogrammierung
MemSystem 1.1e	[bne]	Leistungsfähiger Ersatz für das Standard-Modul »Heap« mit erweiterten Multitaskingfähigkeiten - Update von Amok #5.
Sample	[brk]	Gesampelter Sound eines Cock-Robin-Liedausschnitts.

### Amok #9

Graphic-Support	[fst]	Bibliothek für Modula-Programmierer. Enthält Makros für die Routinen der Graphics- und Intuition-Library.
Graphics	[brk]	IFF-Bilder. Müssen zum Betrachten einfach 2x angeklickt werden.
Sounds	[brk]	Gesampelte Sounds. Zum Starten einfach doppelklicken.

**Tabelle 1. Die vollständige »Amok«-Liste. Sie enthält eine große Zahl nützlicher Programme, die fast alle mit »M2Amiga« geschrieben wurden.**



PrintIt2.0	[fbs]	Hardcopyprogramm für IFF-Bilder. Update von Amok #5.
M2Test	[fbs]	Dokumentiert einen Fehler bei »ARRAY«s mit Index vom Typ »BOOLEAN«.
TextFont	[fst]	Beispielprogramm zum Laden eines neuen Zeichensatzes.

## Amok # 10

Amok10	[Amok]	Bitte anklicken.
FlightDemo	[bne]	Zentralprojektion in Echtzeit aus Vogelperspektive.
3D-Demo Graphics	[fbs] [brk]	3D-Grafik Demonstration in Echtzeit. Gemalte Bilder im Deluxe-Paint-Format.
Sounds	[brk]	Gesampelte Sounds.
FileRequest	[bne]	Komfortabler Dateiauswahlrequester.
SuperLists1.3	[bne]	Unterstützende Routinen für File- und String-Requester. Für »FileRequest« wird diese neue Version benötigt – Update von Amok #2 und Amok #5.
Speech	[mif]	Deutsche Sprachausgabe über die Sound-Chips des Amiga.
FeigenBaum	[fbs]	Zeichnet einen »Liegenden Feigenbaum«.

## Amok # 11

R.o.M.	Stefan Selewski	Komfortables Mathematik-Programm und Funktionsplotter.
DMErr1.1	[fbs]	Dieses Programm zeigt Fehlermeldungen von »M2Amiga« im DME an. Damit können Sie sich Ihre eigene »integrierte Oberfläche« schaffen. Kann ab dieser Version auch von der Workbench gestartet werden.
Beep	[fbs]	Prozedur, die ein akustisches Warnsignal erzeugt.
TrackDisk-Support	[fbs]	Bibliothek für Modula-Programmierer. Erleichtert die Programmierung des TrackDisk.Device.
m2druka	Rolf Kersten	Druckutility für ASCII-Files.

## Amok # 12

Fischer-technik	Norbert Klapczynski	Modul zum Ansteuern des Fischertechnik-Interfaces einschließlich einiger Beispiele.
StripText	Bernd Preusing	Ersetzt in Textfiles Spaces durch TABs.
ConvLibSym	Bernd Preusing	Erzeugt ein Library Module.
Diff	Bernd Preusing	Vergleicht zwei Dateien.
MakeInline	Bernd Preusing	Erzeugt Inline-Code.
M2Tips&Tricks	Bernd Preusing	Einige Tips zum M2Amiga-Compiler.
KwikBackUp	[fbs]	Schnelles und sicheres BackUp-Programm für die Festplatte.
Ecce	Norbert Süßdorf	Einfacher Ersatz für den Testfile-Lister »More« in Modula-II.
DisObject	Bernd Preusing	Disassembliert M2Amiga-Object-Files. Dazu eine Menge importierbarer Module von Bernd Preusing:
ArraySort	Bernd Preusing	Sortieroutine für Datenfelder
Janus	Bernd Preusing	Library-Modul für die Janus-Library (endlich wird auch PC-Karten-Besitzern etwas geboten).
TextWindow	Bernd Preusing	Unterstützt die Textausgabe in Intuition-Fenstern.
Timer2	Bernd Preusing	funktionierendes Timer-Modul.
InOut2	Bernd Preusing	Erweitertes InOut.
TwoFiles	Bernd Preusing	Modul zum gleichzeitigen Bearbeiten einer Ein- und Ausgabe-Datei.

## Amok # 13

CompControl	Ralf Kaiser	Erleichtert die Bedienung von Editor, Compiler und Linker.
NetWork	[fbs]	Vernetzt die Workbench. Äußerst hilfreiches Utility.
Rows	[bne]	Implementiert den dynamischen Datentyp Reihe. Dieser ermöglicht während der Laufzeit dimensionierbare Arrays.
FlipMouse	[fbs]	Invertiert die Richtung der Mausbewegungen.
XHair	[fbs]	Ersetzt den Mauszeiger durch ein Fadenkreuz. Dadurch wird beispielsweise das Positionieren von Icons erleichtert.
WBSshadow	[fbs]	Gibt allem, was auf der Workbench dargestellt ist (Windows, Icons, Gadgets, Text, ...), einen Schatten.
WBPic	[fbs]	Legt ein Bild hinter die Workbench, ohne deren Funktionen zu beeinträchtigen. Im WBPic-Directory befinden sich viele Beispielbilder. Die Bilder sind teilweise von Bernd Kirschner.
With	[fbs]	Ein neuer leistungsfähiger CLI-Befehl. Er ermöglicht, beliebige CLI-Befehle auf alle Files oder Directories, die zu einem Filemuster passen, auszuführen.

## Amok # 14

ARP	[fbs]	M2Amiga-Implementation der »arp.library«. »ARP« ist eine neue leistungsfähige Bibliothek, die das BCPL-DOS des Amiga durch C-Routinen ersetzt und verbessert. Hier finden Sie eine Umsetzung nach Modula. In dem Directory befinden sich zusätzlich die wichtigsten Files des original ARP1.1_Users und ARP1.1_Programmers Directories. Für die wichtigsten Befehle von ARP existieren einige Demos in Modula.
TimerSupport	Hartmut Höhn	Bibliothek für Modula-Programmierer. Sie enthält Prozeduren, die die Benutzung des Timer-Device vereinfachen. Man braucht das Device nicht mehr selbst zu öffnen, Umrechnungen in Jahre und Tage werden übernommen etc.

## Amok # 15

Kurve	Jörg Wese-mann	Leistungsfähiges Kurvendiskussionsprogramm. Zeichnet Ableitungen und Integrale. Es können Nullstellen, Extrema, Wendepunkte etc. bestimmt werden. Außerdem können Sie binäre Tabellen berechnen.
Detach	[bne]	»Endpositioniert« Icons. Sehr praktisch! Funktioniert auch mit Disketten- und Directory-Icons.
Originale	Bernd Preusing	Es handelt sich um die Rückübersetzungen einiger M2Amiga-Module (beispielsweise Arts, Terminal, Strings etc.). Sie wurden mit Hilfe des Programms »DisObj« (Amok # 12) erzeugt. Besonders sinnvoll für alle, die wissen wollen, wie diese Bibliotheken programmiert sind.
Seafarer's Manual	Reinhard Geisler	Die Programme und Lösungen der Aufgaben aus »Modula-II A Seafarer's Manual and Shipyard Guide«.
Calc	[fbs]	Ähnliches Programm wie »EVAL« von der Workbench 1.3. Berechnet das Ergebnis eines Terms. Dabei sind Klammern etc. erlaubt. Funktionen wie Sinus etc. werden unterstützt.



## Amok # 16

M2Make	[bne]	Ein »Make-Utility« für M2Amiga. Untersucht ein vorher definiertes Projekt auf geänderte Module und compiliert diese und alle davon abhängigen und linkt danach. Unverzichtbar für jeden ernsthaften Modula-Programmierer.
MemSystem V1.3	[bne]	Leistungsfähiger Ersatz des Standard-Moduls »Heap« mit erweiterten Multitaskingfähigkeiten - Update von Amok #5 und #8. Neu sind die Module »TaskMemory« und »ErrorReq«. Unterstützt jetzt Levels für lokale Speicherrückgabe.
AVL V1.2	[mif],[bne]	Implementiert den dynamischen Datentyp Baum. Update des AVL-Baum von Amok #7. Diese Version wird für »M2Make« benötigt.
List V1.1	[mif],[bne]	Implementiert den dynamischen Datentyp Liste - Update des List-Moduls von Amok #7. Diese neue Version wird für »M2Make« benötigt.
Queue V1.2	[mif],[bne]	Implementiert den dynamischen Datentyp Schlange - Update des Queue-moduls von Amok #7. Diese neue Version wird für Make benötigt.
ModList	Andreas Kopp, [bne]	Druckt Modula-II-Listings formatiert mit Hervorhebung der Schlüsselwörter.
M2Test	[fbs]	Dokumentation eines Fehlers des M2Amiga-Compilers.
RotatelFF	[fbs]	Rotiert ein IFF-Bild um die horizontale Bildschirmachse.
MultiSelect	[fbs]	Ermöglicht die Aktivierung mehrerer Icons auf der Workbench, ohne die Shift-Taste drücken zu müssen. Diese wird durch den rechten Mausknopf ersetzt.

## Amok # 17

MegaWB	[fbs]	Endlich keine Probleme mehr mit einer von Fenstern überfüllten Workbench! MegaWB vergrößert die BitMap der Workbench auf eine
--------	-------	---

Personal-ausweis	Florian Unger
IFFToImage	Jochen Kupfer
ILBMHandler	Stefan Salewski
M2Test	[fbs]
GetiT-Zeiger	Jochen Kupfer

beliebige Größe (beispielsweise die standardmäßig eingestellten 1024 x 512). Damit weiterhin die gesamte Workbench benutzt werden kann, wird sie bei Mausbewegungen gescrollt. Überprüft und berechnet (!) Nummern des neuen Personalausweises. Nicht für feindliche Spione. Wandelt IFF-Bilder in einfach nachzuladende Image-Dateien um, die von Intuition problemlos verwaltet werden. Bibliotheksmodul für Modula-Programmierer. Enthält Prozeduren zum Laden und Speichern von IFF-Bildern. Außerdem enthält es Prozeduren für Colorcycling und zum Ein- und Ausblenden von Screens. Dokumentation eines Fehlers des M2Amiga-Compilers v3.1d bei der Reihenfolge der Variablendeklaration. Programm zum einfachen Verändern der Workbenchfarben und des Mauszeigers.

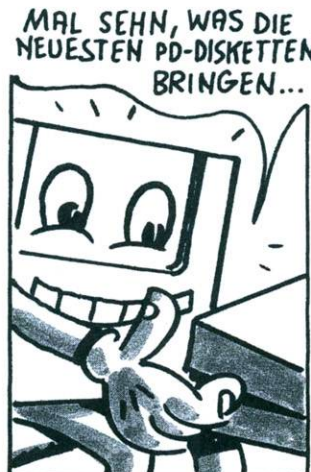
## Amok # 18

Profile	[fbs]	Erzeugt aus einem Modula-Quelltext einen neuen Text, der um Bemerkungen erweitert ist. Diese enthalten Informationen über den Wertebereich von Variablen, Anzahl der Schleifendurchläufe und der Aufrufe von Prozeduren.
PopUp-Menu	Jochen Kupfer	Modul zur Unterstützung von »PopUp«-Menüs. Dies ist eine Modula-II Umsetzung der C-PopUp-Menu-Routinen von Derek Zahn.
IFFLib	[fbs]	M2Amiga-Implementation der iff.library von Christian A. Weber. Diese Library enthält schnelle Routinen zum Laden und Speichern von IFF-Dateien. Besonders gut werden Bilder unterstützt.
Pictures		Zwei digitalisierte »HoldAndModify-Overscan-Interlace«-Bilder.

Tabelle 1. (Schluß)

# AMICUS

DIE ABENTEUER EINER FLOPPY-DISC





# Österreich: Lebendige

Von Martin Jobst

Jeder Salzburger verbindet mit dem Namen »Safe« die hiesige Elektrizitäts-Gesellschaft. Insider wissen jedoch, daß sich hinter dieser Abkürzung auch der Name einer innovativen österreichischen PD-Serie verbirgt. SAFE steht nämlich für **Salzburger Amiga FreundEskreis**. Die Reihe verspricht, mit vorwiegend neuen Programmen Leben in die PD-Szene zu bringen.

Wir haben mit Bernd Küppers, dem geistigen Vater der Serie, über seine Absichten und Erfahrungen in der PD-Szene gesprochen. Die interessantesten Disketten »seiner« Serie stellen wir Ihnen vor. Außerdem zeigen wir, was die AUSTRIA-Reihe des »m.a.r.-Computershop« in Wien zu bieten hat. Einen Überblick über alle besprochenen Programme finden Sie in Tabelle 1.

Begonnen hatte alles vor rund drei Jahren. Bernd Küppers war vom Gedanken der frei kopierbaren Software für den Amiga begeistert und wollte daran teilhaben. Aus diesem Grund kaufte er in Deutschland für eine größere Summe eine Reihe von PD-Disketten ein.

Als Neuling auf dem Amiga entstanden ihm damals aber erhebliche Probleme. Nicht nur, daß die meisten gekauften Disketten nicht bootfähig waren, es gab auch Schwierigkeiten im Umgang mit den »reinen Datendisketten«, wie er sie bezeichnet: Sie waren einfach nicht benutzerfreundlich. Auch verschiedene Anfragen beim Anbieter blieben fruchtlos – alleine im Regen zu stehen ist nicht schön.

## Eine Serie wird geboren

Nach einigen weiteren Erfahrungen mit PD-Software wurde – zusammen mit einigen Freunden – die Idee einer eigenen Serie geboren. »Die Serie entstand eigentlich aus einer Ohnmacht heraus, ich hatte einfach genug von all den benutzerunfreundlichen Serien.« Es entstand ein Label, das diesem Mißstand ein Ende bereiten sollte. Alle Disketten der SAFE-

**Die PD-Szene in Österreich ist in Bewegung.  
Eine Serie mit dem Namen SAFE macht dort von sich reden.  
Der Gründer dieses Labels gewährte uns den Blick hinter die Kulissen.  
Außerdem stellen wir die Reihe AUSTRIA vor.**

Serie sind aus diesem Grund »autobootend« und alle Programme können – sofern sinnvoll und möglich – von der Workbench gestartet werden. Bernd Küppers legt großen Wert auf eine verständliche, deutschsprachige Dokumentation. Zudem wird ein Telefonservice für auftretende Probleme mit den Disketten angeboten.

Vor zwei Jahren war es dann soweit: die erste SAFE-Diskette, voll mit diversen Druckertreibern, erblickte das Licht der Welt.

Ein besonderes Anliegen der SAFE-Reihe ist es, möglichst nur neue Programme anzubieten. »Ich sehe keinen Sinn darin, herzugehen und aus einer Anzahl Disketten die besten Programme zusammenzufügen. Niemand kann doch eine objektive Auswahl treffen.« Bernd Küppers versuchte also frischen Mutes, seiner anwenderfreundlichen Serie auf die Beine zu helfen – und hatte dabei auch gleich einige größere Hürden zu überwinden. So hat er nach wie vor Probleme, Autoren zu finden, die ihre Program-

me auf PD veröffentlichen wollen. Aufrufe bleiben meistens ohne Resonanz. »Wenn sich ein Programm kommerziell nicht vermarkten läßt, versuchen die Autoren meist bei einem Fachmagazin ihr Glück. Oft verstaubt es dann dort in irgend einer Schublade.« So finden Programme seiner Meinung nach schneller den Weg in Ablagen, als zu einer breiten Veröffentlichung auf PD. Aus diesem Grund würde er sich eine engere Zusammenarbeit der Fachzeitschriften mit den PD-Anbietern wünschen.

Bernd Küppers beschäftigt sich mit PD-Software aus reinem »Spaß an der Freude«.

## Autoren gesucht!

Hauptberuflich betreibt er ein zahntechnisches Labor in Mittersill, einer Kleinstadt im Süden des Bundeslandes Salzburg. Das große Geschäft will und kann er mit PD-Software auch gar nicht machen. »Manchmal ist es aber doch frustrie-

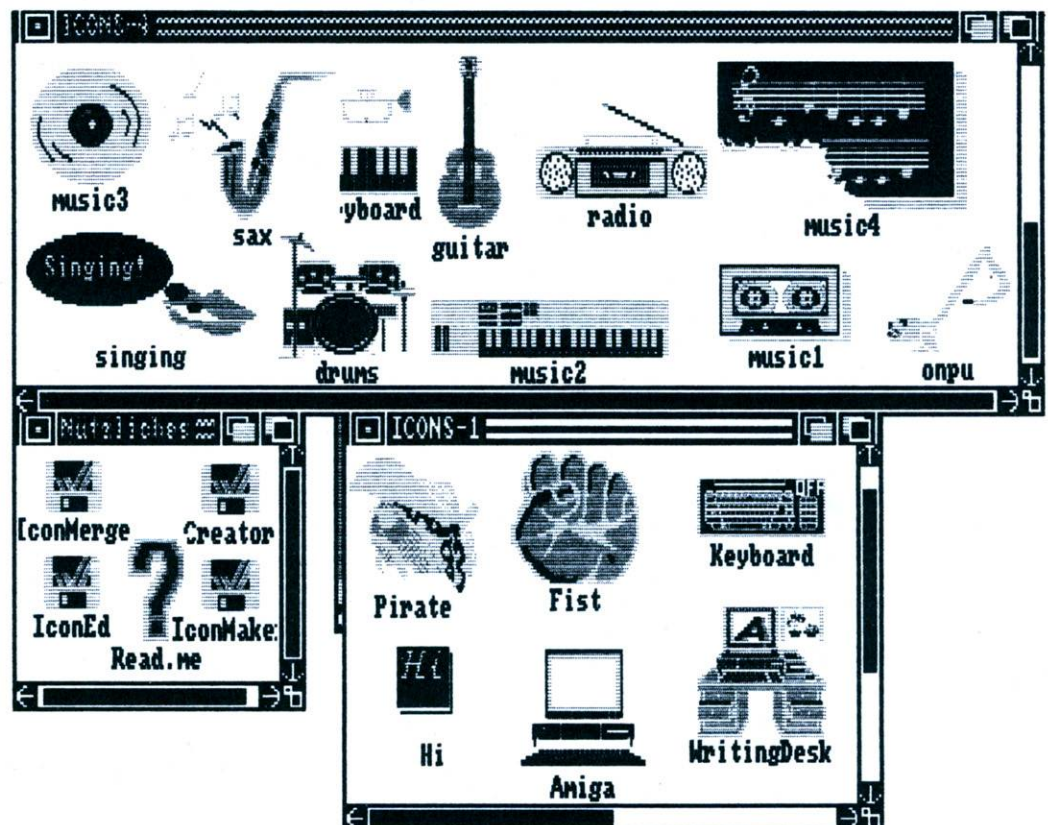


Bild 1. Icons in allen Variationen befinden sich auf SAFE 4



# PD-Szene im Alpenland

Diskette	Programm
SAFE 4	Icons Creator IconMake IconEd
SAFE 5	ShowHAM ShowDigi Showilbm PaintPrint DirUtil DiskSalv More TxEd MyCli Shell
SAFE 7	ASDG RAM-Disk Discard MicroEmacs DiskSalv Sectorama Blitz DirUtil DoTil FuncKey PopCLI II Undelete
SAFE 8	Diashow
SAFE 9	Diashow (Farbanimationen)
SAFE 13	Lyrik
SAFE 18	Zimmerverwaltung
SAFE 19	Zeichensätze CTD
SAFE 19a	Zeichensätze
SAFE 27	Guardian Virsearch VirusAlert VirusTest Diff
SAFE 29	ByteBanditKiller ASDG-RAM-DISK mit Dokumentation
SAFE 30	Berufswahl
SAFE 32	Lichtorgel Steuer Paratest Transfer
SAFE 35	Braitenberg
SAFE 36	Hotelverwaltung (1)
SAFE 36a	Hotelverwaltung (2)
A.U.S.T.R.I.A. 2 A.U.S.T.R.I.A. 4	Sprite Sprite Moire WBUCL DropTab Digi Display Split Genius
A.U.S.T.R.I.A. 5 A.U.S.T.R.I.A. 8	Matrix Quickmenu Scattr
A.U.S.T.R.I.A. 10	VCheck Guardian Virus Alert ViewBoot VirusX Z-SystemIII

Tabelle 1. Alle besprochenen Programme auf einen Blick

rend, wenn nicht einmal die hohen Unkosten, vor allem für Werbung, durch die Einnahmen vom Versand abgedeckt werden.« Gerade in Österreich, mit einem relativ kleinen Markt für PD-Software ist es eben schwer, dabei nicht in die roten Zahlen zu kommen. »Mehr als etwa 400 Disketten pro Monat sind nicht an den Mann zu bringen.«

Trotz der Probleme ist die SAFE-Serie bis heute am Leben und hat es bereits auf rund 40 Disketten gebracht. Wenn auch bei den Prinzipien einige Abstriche gemacht werden mußten. Aus den erwähnten Gründen war es nicht möglich, auf allen Disketten grundsätzlich neue Programme zu bringen. Trotzdem sind laut Bernd Küppers etwa die Hälfte aller Programme vorher noch auf keinem anderen Label erschienen. »Seit Diskette 29 gibt es aber nichts mehr, was auf irgend einer Serie bereits veröffentlicht wurde.« Das »alte« Prinzip wird also wieder konsequent fortgeführt. Bernd Küppers beabsichtigt, die Serie nur

sätzliche Sinn von PD-Software verloren.«

Kommen wir nun zu einem Überblick über die Serie. Diskette Nummer 4 der Serie - »Icon Joy« - ist nützlich für alle, die auf der Suche nach neuen Icons sind. In den verschiedenen Schubladen finden sich aber nicht nur Icons (Bild 1), sondern auch einige nützliche Utilities zur Bearbeitung dieser. Nämlich »Creator«, »IconMake«, »IconMerge« und der bekannte »IconEd«.

## Utilities en masse

Eine Sammlung von »c-Befehlen« aus allen Bereichen befindet sich auf SAFE 5. Das sind Kommandos, die wie die Befehle »Dir« oder »Copy« in den c-Ordner kopiert und vom CLI aus aufgerufen werden. Auf SAFE 5 befinden sich Kommandos zum Anzeigen von IFF-Bildern, wie »ShowHam«, »ShowDigi« und »Showilbm«. »PaintPrint« dient zum Ausdrucken von HAM-Bildern. »DirUtil« und »DiskSalv«



Bild 2. Absolut sehenswert - die Diashow von SAFE 8

weiterzuführen, wenn es ihm gelingt, neue Programme aufzutreiben. »Ich bin grundsätzlich bereit, jedes Programm, das lauffähig ist, zu veröffentlichen.«

## Die Perlen der SAFE-Serie

Wie sinnvoll und brauchbar diese dann für den einzelnen sind, muß jeder selbst entscheiden. Wenn ich eine Art Zensur ausüben würde, ginge ja der grund-

helfen beim Arbeiten im CLI und beim Reparieren von Disketten. Zum Betrachten und Bearbeiten von ASCII-Files dienen »More« und »TxEd«. Insgesamt befinden sich auf der Diskette rund 70 verschiedene Kommandos für das C-Directory. Zwei Shells »MyCli« und »Shell« samt Dokumentation runden das Angebot der Diskette ab.

Absolut empfehlenswert ist SAFE 7. Diese enthält eine wirklich gelungene Auswahl an Utilities, die jeder Amiga-Besitzer



früher oder später einmal benötigt. Darunter befindet sich etwa eine Version der resetfesten »ASDG RAM-Disk«. »Discard« löscht Dateien vollständig, also nicht nur die jeweiligen Header. Mit »MicroEmacs« erhalten Sie einen sehr brauchbaren Texteditor. »DiskSalv« repariert Disketten ebenso wie »Sectorama«, während mit »Blitz« ASCII-Files betrachtet und auch bearbeitet werden. Mit diesem Hilfspro-

gramm können beispielsweise alle »Carriage>Returns« in einem Text entfernt oder durch »Line-Feeds« ersetzt werden. Das bekannte »DirUtil« hingegen ist eine komfortable Benutzeroberfläche für CLI-Operationen. Genauso erleichtert »DoTil« den Umgang mit dem CLI und bietet zudem eine Fülle an Sonderfunktionen. »FuncKey« schließlich dient zur freien Belegung der Funktionstasten mit verschiedenen Befehlsfolgen. »Pop-CLI II« schaltet den Bildschirm nach einer bestimmten Zeit ab und ermöglicht das Öffnen eines neuen CLI-Windows mit der Kombination <CBM ESC> oder das Aufrufen anderer Funktionen. »Undelete« dient zum »Retten« von gelöschten Files auf Diskette.

Art »elektronisches Buch« und verwöhnt den Leser mit heiteren Gedichten von Eugen Roth, untermalt von dezenter Musik. Für alle, die sich für Literatur interessieren, durchaus empfehlenswert. Besitzern eines kleinen Gästehauses oder einer Pension ist mit SAFE 18 geholfen. Auf dieser befindet sich eine einfache Zimmerverwaltung für kleinere Betriebe. Neben der Ver-

waltung, kann auch eine Statistik über die Zimmerbelegung eines Jahres angezeigt werden.

Jede Menge Zeichensätze befinden sich auf den »Font Disks« 19 und 19a. Neben einigen interessanten, mit Hilfe des »Calligrapher« erstellten, Color-fonts enthält die Diskette auch normale Amiga-DOS-Fonts. Ein kleines Zusatzprogramm (»CTD«) auf SAFE 19 erlaubt es, die farbigen Zeichensätze mit DPaint zu verwenden.

Nicht nur Menschen sind durch Viren gefährdet. Auch Computer laufen immer mehr Gefahr, durch Viren außer Schach gesetzt zu werden. Auch der Amiga ist durch eine Vielzahl verschiedenartiger Viren bedroht. Ein Doktorkofferchen mit allerlei Gegenmitteln für bekannte Viren ist die SAFE-Disk 27. Sie enthält jede Menge Viruschecker, -killer und -prüfer. Neben dem bekanntesten »Guardian« befinden sich etwa die Programme »Virsearch«, »VirusAlert« oder »VirusTest« auf der Diskette. »Diff« dient zum Vergleichen von zwei Files und hilft dadurch beim Auffinden von sogenannten Link-Viren. Der »ByteBanditKiller« ist spezialisiert zum Erkennen und



Bild 3. Der Amiga hilft bei der Berufswahl (SAFE 30)

Entfernen dieser Virusgattung. Ein kurzer Text auf der Diskette gibt nicht nur allgemeine Informationen zu Computerviren, sondern auch Hilfestellung zur Heilung des Amiga von allen gängigen Typen. SAFE 29 enthält neben anderen Tools eine Version der ASDG RAM-Disk mit ausführlicher Dokumentation. Verschiedene Installationshinweise ermöglichen ein effektives Ausnutzen dieser resetfesten RAM-Disk.

## Der Amiga als Poet

Für die Freunde beeindruckender Bilder sind die Disketten SAFE 8 und 9 gedacht. Während erstere eine Diashow neuer, teilweise digitalisierter, Bilder bringt (Bild 2), wird auf SAFE 9 mit Farbanimationen gespielt - Prädikat besonders sehenswert.

Eine Neuheit stellt SAFE 13 dar. Der Amiga fungiert als eine

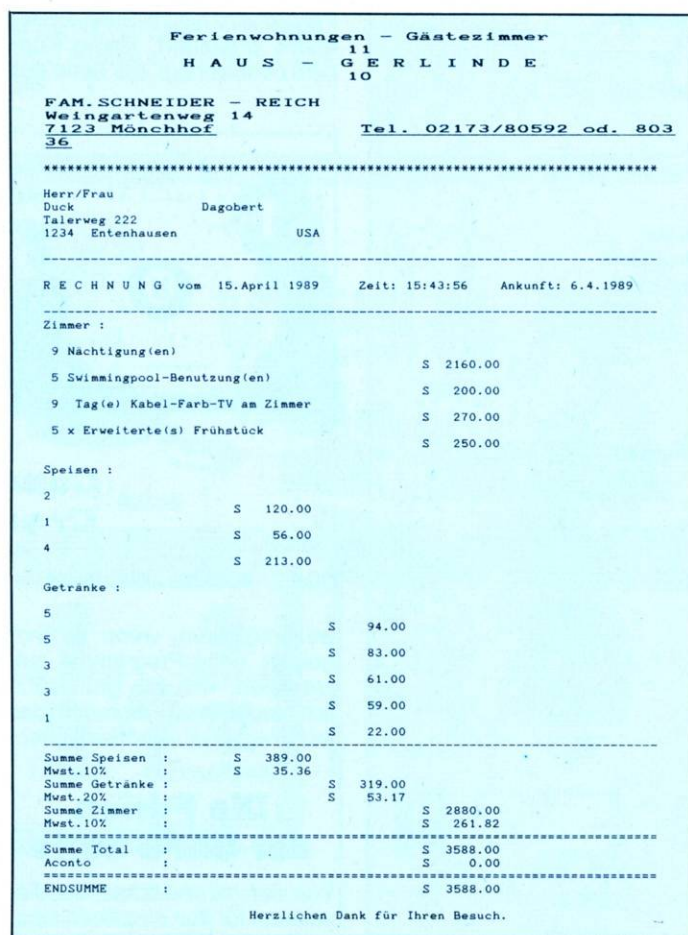


Bild 4. Eine professionelle Hotelverwaltung mit zahlreichen Finissen finden Sie auf SAFE 36/36a

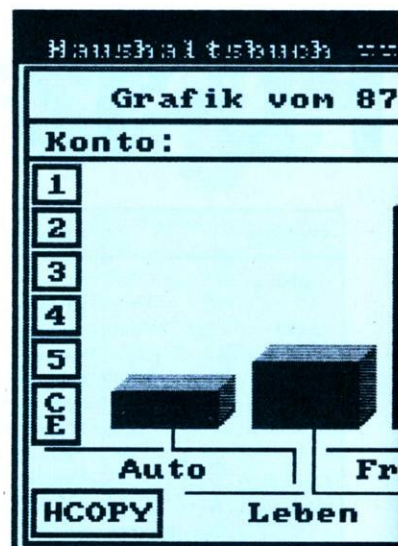


Bild 5. Grafische Darstellung der Überweisungen - das Haushalts-

Entfernen dieser Virusgattung. Ein kurzer Text auf der Diskette gibt nicht nur allgemeine Informationen zu Computerviren, sondern auch Hilfestellung zur Heilung des Amiga von allen gängigen Typen.

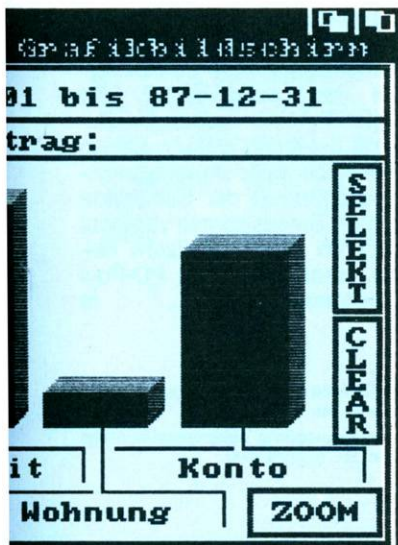
SAFE 29 enthält neben anderen Tools eine Version der ASDG RAM-Disk mit ausführlicher Dokumentation. Verschiedene Installationshinweise ermöglichen ein effektives Ausnutzen dieser resetfesten RAM-Disk.

## Licht-Show aus dem Computer

Auf psychologischen Grundlagen basiert das Programm »Berufswahl« (Bild 3) auf SAFE 30. Es soll in erster Linie jungen Menschen helfen, den richtigen Beruf zu finden. Der Test gliedert sich in drei Teile mit insgesamt 600 Fragen. Werden diese beantwortet, kann sich das Programm ein immer präziseres Bild der Eigenschaften und Interessen des Kandidaten machen. Als Endergebnis macht er dann zwei oder drei Vorschläge, welche Berufe für den Testkandidaten am besten geeignet wären.

Einige sehr interessante Programme enthält SAFE 32 (Tools). Auf dieser Diskette befindet sich ein Schaltplan für eine computergesteuerte Lichtorgel mit acht Kanälen als IFF-Bild. Mit den Programmen »Steuer« und »Paratest« kann diese effektiv gesteuert werden. Neben vorgegebenen Mustern und einer Zufallsfunktion lassen sich auch eigene Abläufe für das Aufblinker der Lampen eingeben. Ein weiteres Tool





Ausgaben für Auto, Freizeit und buch auf »A.U.S.T.R.I.A. 2«

mit dem bezeichnenden Namen »Transfer« erlaubt die Umrechnung verschiedener Einheiten. Folgende Größen können transformiert werden: Grad Celsius in Grad Fahrenheit, Atmosphären in bar, Kalorien in Joule, Meilen/h in Kilometer/h und Knoten in Kilometer/h. Die Ausgabe erfolgt wahlweise auch in Wertetabellen mit frei definierbaren Bereichen auf Bildschirm oder Drucker.

Ein äußerst komplexes Programm mit dem Namen »Braitenberg« befindet sich auf SAFE 35. Es simuliert das Leben künstlicher Wesen und ist eine Anlehnung an ein ähnliches Programm, das es seit längerem für MS-DOS-Systeme gibt.

Die vorerst letzten beiden Disketten der SAFE-Serie sind die Disketten 36 und 36a. Auf diesen befindet sich ein Verwaltungsprogramm für Hotels (Bild

4). Das Programm stellt ein vollständiges Gästebuch mit sämtlichen Funktionen dar. So werden Bestellungen von Gästen genauso verwaltet, wie Rechnungen oder ein Gästebuch ausgegeben werden können. Mit »Faktura« können speziell für gastgewerbliche Zwecke Reservierungsanfragen etc. mittels Formbriefen bearbeitet werden.

## A.U.S.T.R.I.A. transparent

Soweit zur Auswahl aus dem Angebot von SAFE. Diese ist jedoch nicht das einzige PD-Label in der Alpenrepublik. Mit der A.U.S.T.R.I.A.-Reihe findet man Österreich eine weitere – wenn auch nicht so umfangreiche – PD-Serie. Ziel von A.U.S.T.R.I.A. ist es ebenfalls, möglichst bisher unveröffentlichte Programme anzubieten. Wir wollen Ihnen diese Serie ebenfalls nicht vorenthalten und stellen Ihnen einige interessante Programme aus der bisher zehn Disketten umfassenden Reihe vor.

Ein absolut professionelles Produkt ist das Haushaltsbuch auf der »A.U.S.T.R.I.A. 2«. Mit allen nur erdenklichen Funktionen verwöhnt es seinen Benutzer und hilft, die Haushaltskosten in den Griff zu bekommen. Der Einsatz eines Filters erlaubt es beispielsweise, bestimmte Kosten – etwa für Benzin – herauszupicken. Die grafische Darstellung Ihrer Ausgaben als Balkendiagramm spricht für sich (Bild 5).

»A.U.S.T.R.I.A. 4« enthält neben einer Reihe interessanter Grafik-Demos, wie »Sprite«, »Moire« und »WBUCL« einige nützliche Utilities. So ersetzt

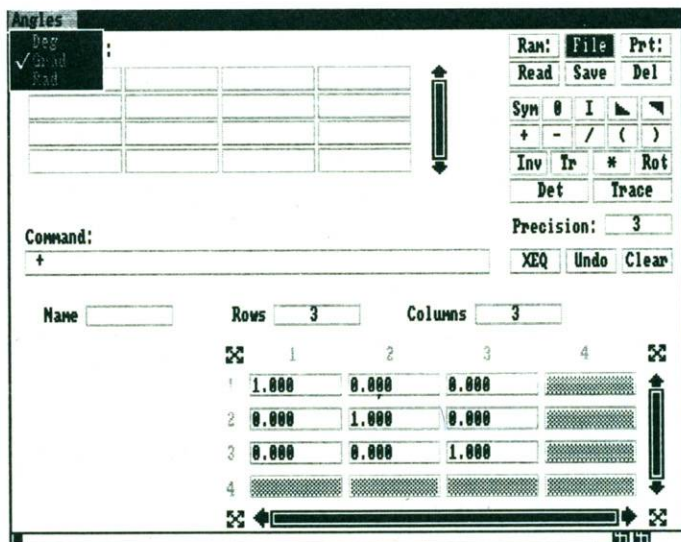


Bild 6. »Matrix« erlaubt die Verknüpfung von Zahlen in Matrizen (A.U.S.T.R.I.A. 5)

## Ein Buch für Entdecker

# Fraktale Grafik auf dem Amiga



Das Thema dieses Buches wird den meisten ungewöhnlich erscheinen, denn es führt in die Grenzbereiche des heutigen Wissens der Mathematik und Technik.

Ist die Kenntnis von mathematischen Grundlagen, wie z.B. komplexen Zahlen, notwendig, werden diese kurz erklärt. Das Buch wendet sich deshalb nicht an den Spezialisten und Mathematiker, sondern ist für jedermann geeignet, der sich für Computergrafik begeistert. Grundlegende Kenntnisse der Programmiersprache C und ihrer Anwendung auf dem Amiga werden vorausgesetzt. Für die nachträgliche Veränderung berechneter Bilder ist es sinnvoll, ein Malprogramm (z.B. Deluxe Paint) zu besitzen.

● Ein Buch für Forscher, die an einer revolutionären Entwicklung in den Naturwissenschaften teilnehmen wollen und bereit sind, auf Entdeckungsreise zu gehen. Reisen Sie mit!

1988, 278 Seiten,  
inkl. Diskette  
Bestell-Nr. 90600  
ISBN 3-89090-600-1  
DM 79,- (sFr 72,70/öS 616,-)

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Str. 2, 8013 Haar bei München, Tel. (089) 46 13-0. Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. (042) 415656. ÖSTERREICH: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Tel. (0222) 587 1393-0, Rudolf Lechner & Sohn, Heizwerkstr. 10, A-1232 Wien, Tel. (0222) 67 75 26, Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Tel. (0222) 48 15 43-0.



»DropTab« in ASCII-Texten Tabulatorsprünge durch eine entsprechende Anzahl an Leerzeichen. »Digi« dient zum Abspielen von IFF-Sounds, während »Display« IFF-Bilder anzeigt. »Split« ist ein neuer CLI-Befehl und teilt ein File in zwei Teile auf (konträr zum CLI-Befehl »Join«). »Genius«, eine grafisch ansprechende Variante von »Master-Mind«, setzt das Pünktchen auf das »i«.

Vor allem für Mathematiker interessant ist »Matrix 1.5«. Das auf »A.U.S.T.R.I.A. 5« veröffentlichte Programm dient zur Berechnung von Matrizen. Damit

können in einer Matrix angeordnete Zahlen durch verschiedene Operationen verknüpft werden. Vor allem für komplexe Gleichungssysteme mit Vektoren ist dieses sehr nützlich. Die übersichtliche Benutzeroberfläche (Bild 6) ermöglicht einfache Bedienung.

## Keine Chance den Viren

»Quickmenu« auf Diskette 8 der Serie erlaubt die Erstellung von weiteren Menüs auf der Workbench. Damit wird es

möglich, verschiedene Programme per Menü zu starten. »Scattr« (ebenfalls auf dieser Diskette) dient zum Verändern der Textattribute, Farben etc. im CLI.

Die bislang letzte Diskette der »A.U.S.T.R.I.A.-Reihe« – Nummer 10 – enthält verschiedene Programme zum Schutz Ihres Amiga vor Viren. Neben »VCheck« und »Guardian« befinden sich auf der Diskette »Virus Alert«, »ViewBoot«, verschiedene Versionen von »VirusX« und der »Z-SystemIII«-Viruskiller. Sie sehen, auch in Österreich etabliert sich eine in-

teressante PD-Szene. Die teilweise sehr gelungenen Zusammenstellungen auf den Disketten der Labels SAFE und »A.U.S.T.R.I.A.« sollten in Zukunft belebend wirken. Haben Sie Hilfs- oder Anwendungsprogramme in der Schublade liegen? Dieses könnte vielleicht anderen Amiga-Besitzern helfen, wenn Sie es im PD-Pool schwimmen lassen. *rs*

SAFE, Bernd Küppers, Felberstraße 7, A-5730 Mittersill, Tel.: 06562/282  
m.a.r-Computershop, Weldengasse 41, A-1100 Wien, Tel.: 0222/621535

# Public Domain - Blickpunkt Schweiz

Von Dietmar Inäbnit

**D**ie Schweiz ist bekanntlich ein recht kleines Land. Dadurch gibt es für den PD-hungrigen Amiga-Freak im Gegensatz zur Bundesrepublik nur wenige Software-Quellen. Natürlich gibt es in der Schweiz auch Anbieter, die PD-Software professionell vertreiben. Ebenso findet man PD-Freaks, die ihren eigenen kleinen Software-Service gegründet haben. Die größte Rolle als PD-Lieferant dürften in der Schweiz allerdings die verschiedenen Computerclubs übernommen haben. Fast jeder Club besitzt einen großen Fundus an PD-Software, der durch alle Mitglieder ständig erweitert wird. Fast alle großen PD-Serien sind darin vertreten. Einige Clubs haben auch eigene Serien, die dann meistens in die verschiedenen Anwendungsgebiete, wie Grafik, Programmiersprachen und Spiele, aufgeteilt sind. Jedes Clubmitglied kann zu sehr günstigen Konditionen die gewünschten Disketten aus dem Pool beziehen.

Der CAC (Computer Anwender Club) in Zürich ist ein bekannter Club, der für seine Mitglieder einen PD-Service anbietet. Der Club besitzt nicht nur einen großen Schatz an Amiga-PD, sondern auch PD-Software für IBM-, Atari- und

## Wie weit ist PD-Software für den Amiga in der Schweiz verbreitet? Wir führten ein Interview mit dem Ressortleiter Amiga des Computer-Anwender-Clubs in Zürich. Lassen Sie sich von den Antworten dieses kompetenten Fachmanns überraschen.

MSX-Rechner. Der Ressortleiter (Beppo Ctvrtnicek) für Amiga-PD hat sich freundlicherweise sofort bereit erklärt, mit einem Interview Licht in die Schweizer PD-Szene zu bringen.

**AMIGA:** Was kostet eine Diskette beim CAC?

**Ctvrtnicek:** Pro Diskette verlangen wir einen Betrag von fünf Franken. Darin ist der Preis für Diskette und Verpackung sowie das Porto enthalten.

**AMIGA:** Wie entsteht dieser günstige Preis für PD-Disketten?

**Ctvrtnicek:** Wir wollen mit dem Verkauf von PD-Disketten keinen Profit erzielen. Selbst Schüler können sich für diesen geringen Betrag PD-Disketten leisten. Unser Ziel ist es, die Public Domain-Software möglichst vielen Amiga-Freunden in der Schweiz näherzubringen. Gleichzeitig ist es unser Wunsch, das Raubkopieren damit einzudämmen.

**AMIGA:** Wie wird das große Angebot an PD-Software von

den Mitgliedern Ihres Clubs genutzt?

**Ctvrtnicek:** Der seit zirka einem Jahr existierende PD-Pool findet bei unseren Mitgliedern noch keine große Beachtung. In Anfragen wird zwar immer wieder unsere PD-Liste verlangt, aber es gehen nur sehr wenige Bestellungen ein. Bis jetzt wurden insgesamt nur 35 Disketten bestellt.

**AMIGA:** Wie erklären Sie sich dieses Desinteresse an PD-Software?

## Mehr Information ist notwendig

**Ctvrtnicek:** Die verschiedenen Namen, wie Fred Fish, Amicus, oder Faug, sind den meisten Amiga-Anwendern ein Begriff. Die wenigsten wissen jedoch, welche Programme auf diesen Disketten zu finden sind. Kauft nun jemand eine Diskette, auf der für ihn nichts Interessantes zu finden ist, glaubt er automatisch, daß alle

PD-Disketten nichts taugen. Der Ruf der PD-Disketten als Virus-Träger hat zudem stark dazu beigetragen, das Interesse an PD-Software zu senken.

**AMIGA:** Was wäre Ihrer Meinung nach nötig, um die Leute besser zu informieren und das Interesse an PD-Software wieder zu steigern?

**Ctvrtnicek:** Der schlechte Ruf als Virus-Träger sollte endlich einmal abgebaut werden.

In den verschiedenen Computermagazinen wird der PD-Software noch viel zu wenig Beachtung geschenkt. Es sollte nicht, wie allgemein üblich, nur eine oder zwei Seiten für PD-Software reserviert werden. Pro Ausgabe sollten mindestens 20 bis 30 Public Domain-Disketten kurz vorgestellt werden.

**AMIGA:** Es gibt doch schon einige PD-Bücher auf dem Markt. In diesen wird der Inhalt der verschiedenen Disketten genau vorgestellt.

**Ctvrtnicek:** Sie können doch von keinem Computer-Freak erwarten, daß er zuerst ein teures Buch kauft, damit er weiß, welche Diskette er haben muß! Diese Bücher sind nur etwas für Leute, die sich in der PD-Szene schon etwas auskennen.

**AMIGA:** Wir danken Ihnen für dieses Gespräch. *rs*

Anschrift: Josef Ctvrtnicek, Höschgasse 38, CH-8008 Zürich



**AMIGA**  
**SONDERHEFT**

# PROGRAMM- SERVICE

## Sonderaktion Public Domain

Die aktuelle Diskette zum Heft:

### Amiga-Sonderheft 5: Public Domain – das Beste aus aller Welt

#### DBW Render:

Hervorragende Grafiken mit 4096 Farben, realistische Spiegelungen und 3-D-Animation. Dieses Raytracing-Programm steht kommerziellen Programmen in nichts nach.

#### DME:

Die bisher unveröffentlichte Version des programmierbaren Editors von Matthew Dillon.

#### TurboBackup:

Disketten kopieren schnell und vielseitig!

#### DiskX:

Durchleuchten Sie Ihre Disketten mit dem Monitor von Steve Tibbett.



erstellt mit DBW Render

Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 5 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

**Aktionspreis:**

**Bestell-Nr. 45905**

**DM 11,90\***

(sFr 11,90\*/öS 119,-\*)

\* Unverbindliche Preisempfehlung

Weitere Angebote  
auf der Rückseite!

  
**Markt&Technik**

Zeitschriften · Bücher  
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440 550. ÖSTERREICH: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 587 1393-0; Rudolf Lechner&Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 6775 26



# AMIGA PROGRAMMSERVICE



Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp. Bei Fragen zu Bestellung und Versand der Programmservice-Disketten wählen Sie bitte: Telefon (089) 46 13-232.

Bestellungen bitte nur gegen Vorkasse an: Markt & Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 46 13-0. Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440 550. Österreich: Markt & Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5 87 13 93-0. Microcomput-ique, E. Schiller, Fasangasse 24, A-1030 Wien, Telefon (0222) 78 56 61; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 83 31 96. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt & Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar. Nur gegen Bezahlung der Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die beigeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungsscheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.



## Amiga-Sonderheft 4: C und Assembler

**Raytracer:** berechnet fantastische 3-D-Bilder. Die Grafikfähigkeiten des Amiga werden eindrucksvoll genutzt.

**Sound-Effekte-Editor:** Zeichnen Sie beliebige Geräusche mit der Maus. Der Clou: Die Sounds werden als C-Sourcecode gespeichert und sind leicht in eigene Programme zu übernehmen.

**Programmiertools:** Jeweils eine mausgesteuerte Benutzeroberfläche für Lattice- und Aztec-Compiler. Programmieren wird ebenso komfortabel wie das Arbeiten mit der Workbench. Eine Shell hilft bei der Vereinfachung umständlicher CLI-Operationen.

Weiterhin befinden sich auf der Diskette alle Programme komplett mit den jeweiligen Quelltexten, die im Inhaltsverzeichnis des Amiga-Sonderhefts 4 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

Bestell-Nr. 45904 **DM 29,90\*** sFr 24,90\*/öS 299,-\*

## Amiga-Sonderheft 3: Basic, Spiele

**Broker:** Erleben Sie die Faszination der Börse hautnah. Diese Simulation für 2 bis 4 Spieler ist einzigartig. Der Autor setzt seine fundierten Kenntnisse in spannendes Spielgeschehen um.

**Ping-Pong:** Dieses Sportspiel bringt Wettkampfstimmung ins Wohnzimmer. Dreidimensionale Darstellung, realistische Soundeffekte und rasante Ballwechsel führen zu lang anhaltendem Spielspaß.

**Anpfiff:** Als Manager in der Fußball-Bundesliga führen Sie Ihr Lieblingsteam durch die Saison. Zusätzliche Spiele im UEFA-Cup verhelfen Ihrer Mannschaft zu Ruhm und Ihnen zu vielen Managerpunkten.

**Basic-Routinen:** Die Basic-Kurse im Sonderheft 3 bieten zahlreiche, hilfreiche Routinen. Alle dort vorgestellten Programme finden Sie auch auf dieser Programmservice-Diskette.

Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 3 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

Bestell-Nr. 45903 **DM 29,90\*** sFr 24,90\*/öS 299,-\*

## Amiga-Sonderheft 2: Grafik, Anwendung

**Object-Editor:** Animierte Figuren, beispielsweise für eigene Spiele, entwickeln Sie mit diesem Editor auf komfortable Weise. Sogar mit Deluxe Paint erstellte Pinsel lassen sich einlesen.

**Haushaltsbuch:** Mit diesem hervorragenden Anwendungsprogramm verwalten Sie alle Einnahmen und Ausgaben auf übersichtliche Weise. Eine Monats- oder Jahresstatistik zeigt, in welchen Bereichen Sie zukünftig sparen können. Jetzt haben Sie Ihre Finanzen im Griff.

**Keyboard-Master:** Lernen Sie im Zehn-Finger-System zu tippen. Mit diesem didaktisch ausgereiften Programm ist dies kein Problem. Für Programmierer sind sogar Spezial-Lektionen mit wichtigen Sonderzeichen vorhanden.

**FastLoadCopy:** Dieses Tool bringt den DIR-Befehl auf Trab. Nach der »Operation« wird das Inhaltsverzeichnis einer Diskette im D-Zug-Tempo eingelesen. Zusätzlich kopiert das Programm Disketten und versieht diese mit dem schnellen Directory. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 2 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

Bestell-Nr. 45802 **DM 29,90\*** sFr 24,90\*/öS 299,-\*

## Bewegte Grafik per Farbdurchlauf - Diashow für Anspruchsvolle

Es ist vollbracht! Die besten Cycling-Grafiken können nun alle grafikbegeisterten Leser hautnah erleben. Die schönsten Bilder des »Color-Cycle«-Wettbewerbs, das im Amiga-Magazin (Ausgabe 3/88 Seite 142) gestartet wurde, präsentieren wir Ihnen auf zwei randvoll bespielten Disketten.

Die Bilder können entweder mit dem enthaltenen Diashow-Programm angesehen oder mit jedem gängigen IFF-Malprogramm (zum Beispiel Deluxe Paint II) geladen werden. Lassen Sie sich die faszinierenden Computer-Bilder nicht entgehen.

Zwei Disketten für Amiga.

Bestell-Nr. 49901 **DM 29,90\*** sFr 24,90\*/öS 299,-\*

## Die Wiederbelebung für die C64-Peripherie

Viele Amiga-Besitzer haben noch einen C64 mit Peripheriegeräten zu Hause stehen. Mit ein bißchen Hard- und Software können Sie diese zu neuem Leben erwecken und Ihre Daten so weiterbenutzen. Dabei ist die Bedienung wirklich einfach.

Der fertig aufgebaute IEC-Handler erlaubt es, alle C64-Geräte wie die Floppy 1541 oder 1571, Commodore-MPS-Drucker und natürlich auch den C64 (zur Datenübertragung) am Amiga zu betreiben.

Das Gesamtpaket besteht aus der fertig aufgebauten Platine mit Verbindungskabel, der Treibersoftware auf 3 1/2"-Diskette sowie einer entsprechenden Dokumentation.

Bestell-Nr. 39101 **DM 79,-\*** sFr 71,-\*/öS 790,-\*

Bestellungen bitte nur gegen Vorkasse bei:

Markt & Technik Verlag AG

- Buchverlag -

IEC-Handler

Hans-Pinsel-Straße 2, 8013 Haar bei München

## Amiga 3/88 Bildschirmfüllende Boot-Bilder mit allen Extras

**BootGirl:** Fantastische Bilder sofort nach dem Reset. Bis zu 32 Farben mit Color-Cycling. Die Bilder können auch bildschirmfüllend ohne Rand sein. Ein absolutes Muß für jeden Amiga-Besitzer.

**CassCover:** Selbstgedruckte Kassettenhüllen geben Ihnen den richtigen Überblick. Einfache Bedienung macht das Eingeben und Ausdrucken zur wahren Freude.

**Command:** Das Programm ermöglicht die Steuerung des Aztec-C-Compilers mit der Maus. Keine langen Eingaben per Tastatur, sondern ein einziger Mausklick startet nun die Übersetzung.

**VideoText:** Ein unentbehrliches Werkzeug für alle Video-Fans, die ihren eigenen Vorspann mit dem Amiga generieren wollen. lauffäden, verschiedene Schriften und IFF-Bilder sind nur einige Stichpunkte, die das Programm so interessant machen.

3 1/2"-Diskette für Amiga

Bestell-Nr. 48803 **DM 29,90\*** sFr 24,90\*/öS 299,-\*

\* Unverbindliche Preisempfehlung

**Übrigens:** Mit den Gutscheinen aus dem »Super-Software-Scheckheft« für DM 149,- können Sie sechs Software-Disketten Ihrer Wahl aus dem Programm-Service-Angebot der Zeitschriften

PC Magazin		Computer persönlich
PC Magazin Plus	Amiga-Magazin	64'er-Magazin
Happy-Computer	Amiga-Sonderheft	64'er-Sonderheft

bestellen - egal, ob diese DM 29,90 oder DM 34,90 kosten. Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingehafteten Zahlkarte direkt beim Verlag bestellen.

Kennwort: Software-Scheckheft, Bestell-Nr. 39100.



# DIE RECHNUNG

## MIT DEM WIRTH GEMACHT

**C**gilt aufgrund hoher Effektivität und leichtem Zugriff auf das System als die »Muttersprache« des Amiga. Wer allerdings glaubt, damit sei das letzte Wort in Sa-

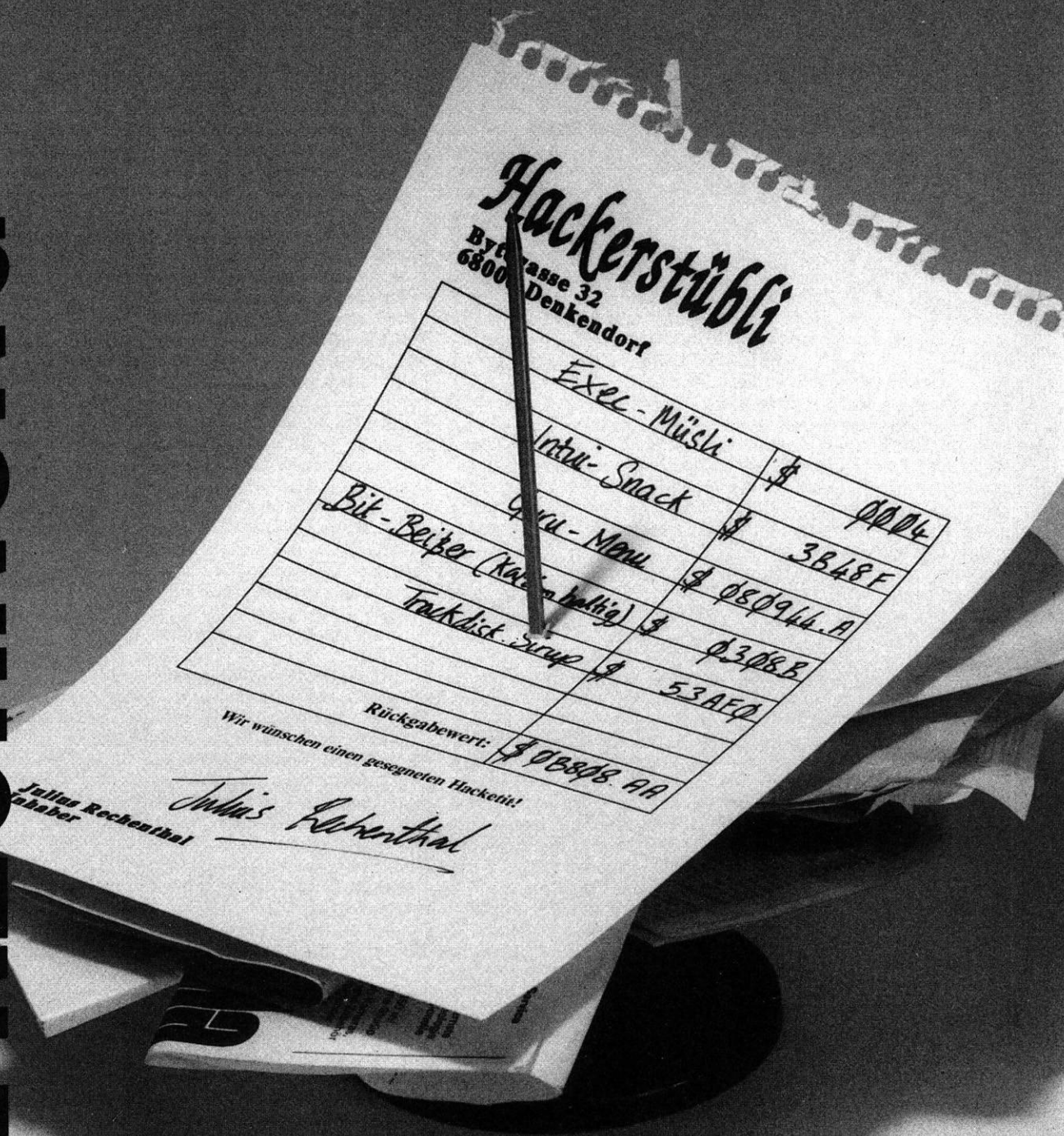
Von Klaus Sonnenleiter

chen Programmieren gesprochen, hat die Rechnung ohne den Wirt(h) gemacht.

Beim Entwickeln von Programmiersprachen gilt der

**Nichts geht mehr ohne den Wirth. Betrachtet man die Entstehung moderner Programmiersprachen, stößt man immer wieder auf den Namen Niklaus Wirth.**

**Die neueste Entwicklung dieses Schweizer Informatikers - Modula-2 - ist in einer sehr leistungsfähigen Version auf PD erhältlich.**



**Hackerstübli**  
Bytstrasse 32  
6800 Denkendorf

Exec-Müsli	\$	0004
Intui-Snack	\$	3848F
Gru-Mau	\$	080944.A
Bit-Beifer (Kaffeehaltig)	\$	0308B
Trackdisk-Sirup	\$	53AE0
Rückgabewert:	\$	08808.AA

Wir wünschen einen gesegneten Hacken!

Julius Rechenhal  
Inhaber

*Julius Rechenhal*



Schweizer neben Dennis Ritchie als »das« Genie. Getreu seinem Grundsatz »gute Sprachen entstehen in einem Kopf« hat er alle seine Programmiersprachen weitgehend im Alleingang entwickelt – im Gegensatz zu »Sprachenungetümen« wie »Ada« oder »Algol«, die in vielköpfigen Gremien und nach unzähligen Kompromissen entstanden.

Die Sprache Pascal (Wirths erster Streich) war ursprünglich als reine Lehrsprache angelegt. Folglich liegt der Schwerpunkt auf Lesbarkeit und klarer Strukturierung. Daneben bietet die strikte Typentrennung eine große Hilfe für den Programmierer: Unsauberkeiten, die bei anderen Sprachen zu langwieriger Fehlersuche führen, entdeckt der Compiler (siehe Kästen). Dieses – besonders für

## Compiler

...sind »Übersetzungsmaschinen«. Sie übersetzen sogenannten Quelltext in Befehle, die der Computer »versteht«. Der Quelltext ist in unserem Fall ein in Modula geschriebenes Programm. Dieses besteht aus Ausdrücken, die größtenteils der englischen Sprache entnommen sind. Dadurch wird der Programmablauf für den Anwender lesbar.

Diese Hochsprache hat nur den Nachteil, daß sie für den Amiga völlig unverständlich ist. Dieser »versteht« nur zwei Werte: die Null und die Eins. Nun könnte man natürlich auf die Idee kommen, diese Nullen und Einsen direkt einzugeben. Aber was lesen Sie lieber?

```
1101001001010100
oder
```

```
IMPORT Dos;
```

Nehmen Sie für die leichtere Lesbarkeit einige Sekunden Übersetzungszeit in Kauf? Der Compiler kann aber weit mehr als stur vor sich hin übersetzen. Er deckt sozusagen als Nebenprodukt die Fehler des Programmierers auf und legt eine Tabelle an, die der Editor direkt weiterverarbeitet. Er kontrolliert die externen Referenzen (Import-Anweisungen) – bei nicht vorhandenen Libraries wird ein Fehler gemeldet.

Programmieranfänger hilflos – Konzept hat dazu geführt, daß Pascal vor allem an Universitäten und unter Gelegenheitsprogrammierern weitverbreitet ist.

Modula ist eine konsequente Weiterentwicklung von Pascal. Viele Unzulänglichkeiten wurden beseitigt, so die mangelhafte Unterstützung des Biblio-

thekskonzepts. Daneben wurde der Tatsache Rechnung getragen, daß an eine moderne Programmiersprache andere Anforderungen gestellt werden als an eine reine Lehrsprache. So sollte sie weitgehend systemunabhängig sein, getrennte Compilierung zulassen, Routinen enthalten, die Multitasking unterstützen. Die Programme müssen zudem leicht zu ändern sein.

Für den Amiga gibt es einen sehr leistungsfähigen Compiler, der den vollen Modula-Standard unterstützt. »M2Amiga« wird von der Schweizer Firma A+L AG vertrieben. Alle Interessenten an dieser neuen Sprache können einen Service in Anspruch nehmen, der die Entscheidung für Modula sehr erleichtert: Die A+L AG bietet eine Demoversion von M2Amiga für 10 Mark an. Dieselbe Version ist auf der Fish-Disk 113 und einigen anderen Public-Domain-Disketten enthalten.

Mit dieser PD-Version von M2Amiga werden dem Programmierer kaum Beschränkungen auferlegt. Alle wichtigen Programme (Editor, Compiler, Linker) sind enthalten, ein Teil der Libraries (siehe unten) wird mitgeliefert und alle Programme entsprechen weitgehend denen der professionellen Version. Die einzige Beschränkung ist die Limitierung der Code-Größe auf etwa 10 KByte. Dies spielt jedoch in Modula eine untergeordnete Rolle, da sich ein Programm in einzelne Module aufteilen (splitten) läßt.

Mit M2Amiga haben Sie ebenso einfachen Zugriff auf die Routinen des Amiga-Betriebssystems wie mit C. Sie sollten sich aber darüber im klaren sein, daß Programme, die diese Fähigkeit nutzen, nicht mehr portabel sind. Denn auf einem MS-DOS- oder Unix-Rechner haben diese Routinen natürlich andere Namen und eine andere Bedeutung. M2Amiga teilt daher seine Bibliotheken in drei Teile:

**System:** Hier sind system-spezifische Definitionen und Funktionen enthalten, beispielsweise das Laufzeitsystem und die Implementation der Realzahlen.

**Interfaces:** Dies sind Schnittstellen zum Betriebssystem. Diese sollten Sie keinesfalls nutzen, wenn Sie übertragbare Programme schreiben wollen.

**Libraries:** Sind die Standard-Bibliotheken von Modula. Programme, die nur diese Bibliotheken nutzen, sollten mit relativ wenig Aufwand auf andere

Systeme übertragbar sein. Die PD-Version von M2Amiga unterstützt die Libraries nicht vollständig. Die kompletten Libraries sind selbstverständlich in der kommerziellen Version enthalten.

Wir wollen uns in diesem Workshop vorrangig mit der Programmierung des Amiga auseinandersetzen. Die hier gezeigten Programme werden also meist nicht auf andere Computer portabel sein. Wenn Sie auf übertragbare Programme Wert legen, empfehlen wir Ihnen das Buch »Programmieren in Modula-2« von Dal Cin, Lutz und Risse. Hier finden Sie viele wertvolle Hinweise.

Bevor Sie anfangen zu programmieren, sollten Sie sich Ihre »Umgebung« einrichten. Sie benötigen dafür:

- eine leere formatierte Diskette,
- Ihre gewohnte Workbench-Diskette und
- eine Diskette mit der PD-Version von M2Amiga (beispielsweise die Fish-Disk 113).

M2Amiga läuft mit 512 KByte Speicher und einem Laufwerk. Die Arbeit wird Ihnen jedoch sehr viel leichter fallen, wenn Sie mindestens 1 MByte und zwei Laufwerke zur Verfügung haben.

Auf der Fish-Disk 113 finden Sie sechs Directories:

- Dme
- C
- NolconPos
- DosDev

## Libraries

...sind im Entwicklungspaket mitgelieferte Bibliotheken. Sie enthalten keine Bücher, sondern vordefinierte Daten und Routinen.

Diese sind in kompilierter Form vorhanden und können direkt in eigene Programme eingebunden werden (mit der IMPORT-Anweisung). Dadurch müssen Sie das Rad nicht jedesmal neu erfinden – Sie können die Erfindungen anderer nutzen. Der Amiga hat einige vordefinierte Bibliotheken (die Betriebssystemschnittstellen), die von Modula angesprochen werden können. Außerdem werden die Standard-Module geliefert, die für Ein-/Ausgabe oder Speicheroperationen benötigt werden.

Die Schnittstellen zum System (Interfaces) sind in sogenannten Definitionsmodulen dokumentiert. Diese finden Sie auf der Fish-Disk 113 im Ordner »M2Amiga/Defs«. In der kommerziellen Version sind sie im Handbuch abgedruckt.

- AmiCron
- M2Amiga

Uns interessiert nur der Ordner »M2Amiga«. Sehen Sie sich diesen im CLI an. Geben Sie dazu folgende Zeilen ein:

```
cd dfx:M2Amiga
dir
```

Dabei ist »dfx:« das Laufwerk, in dem Ihre Fishdisk 113 liegt. Sie sehen die Files »Product Info«, »contents«, »DoMe« und »ReadMe«, sowie die Directories »Disk«, »Demos« und »Defs«.

## Automatische Installation

In Product Info, ReadMe und contents finden Sie interessante Informationen zu M2Amiga, die Sie mit »type« auf den Bildschirm bringen. Die Directories werden uns später noch interessieren. Vorerst genügt es zu wissen, daß im Ordner Disk alle benötigten Programme und Libraries liegen.

Starten Sie das Programm »DoMe« (mit der Eingabe von »DoMe« gefolgt von <RETURN>). Es fragt Ihre Konfiguration (ein oder zwei Laufwerke) ab und installiert eine Diskette einschließlich Startup-Sequenzen und allen benötigten Programmen. DoMe fordert Sie selbständig auf, die jeweils benötigten Disketten einzulegen. Der gesamte Vorgang dauert eine Weile, lassen Sie sich davon nicht irritieren.

Installieren Sie die neu entstandene Diskette mit

```
install dfx:
```

wobei »dfx:« das Laufwerk mit Ihrer neuen Modula-Diskette ist. Sie können jetzt die Startup-Sequenz nach Ihren Wünschen modifizieren, weitere Befehle in den C-Ordner kopieren und dann von dieser Diskette booten. Denken Sie vor allem daran, mit »setmap d« die deutsche Tastaturbelegung zu installieren.

Im weiteren Verlauf gehen wir davon aus, daß alle Programme (Editor, Compiler und Linker) im »C:«- oder im »System«-Ordner liegen und die Startup-Sequenz einen Suchpfad in dieses Directory setzt. Dies erledigt für den System-Ordner die Zeile

```
path SYS:System add
```

Der C-Ordner benötigt keinen Suchpfad. Daneben sollten Sie alle Befehle aus dem CLI oder einer Shell starten. Es empfiehlt sich also, die Zeile »endcli« aus der Startup-Sequenz zu löschen. Um mit Ihrer Arbeitsumgebung vertraut



## M2Emacs-Steuercodes

Die Funktion in der linken Spalte lösen Sie mit einem Druck auf die angegebene Funktionstaste aus, die rechte Spalte erreichen Sie durch gleichzeitiges Drücken von SHIFT und der Funktionstaste.

ohne SHIFT	mit SHIFT
<F1> Neue Datei öffnen	Aktuelle Datei speichern
<F2> Nächsten Fehler anzeigen	Fehlerliste erneut einlesen
<F3> Begriff suchen	Begriff ersetzen
<F4> Beginn d. Makro-Definition	Abschluß d. Makro-Definition
<F5> Ausführen des Makro	Abbruch der Eingabe
<F6> Gehe zur Zeile...	Informationen über die Datei
<F7> Fenster teilen	Fenster schließen
<F8> Fenster verkleinern	Fenster vergrößern
<F9> Text nach oben scrollen	Text nach unten scrollen
<F10> Aktuelle Zeile löschen	Neue Zeile einfügen

### Tastenkombinationen

Folgende Befehle erreichen Sie durch gleichzeitiges Drücken von CTRL und der angegebenen Taste:

<CTRL-@>	Setzt eine Marke
<CTRL-a>	Setzt den Cursor an den Zeilenanfang
<CTRL-b>	Bewegt den Cursor ein Zeichen zurück
<CTRL-c>	Öffnet ein CLI-Fenster
<CTRL-d>	Löscht das Zeichen unter dem Cursor
<CTRL-e>	Setzt den Cursor an das Zeilenende
<CTRL-f>	Bewegt den Cursor ein Zeichen nach rechts
<CTRL-g>	Bricht die aktuelle Eingabe ab
<CTRL-h>	Löscht das Zeichen links vom Cursor
<CTRL-i>	Setzt einen Tabulator
<CTRL-j>	Fügt ein »Linefeed« ein
<CTRL-k>	Löscht alle Zeichen bis zum nächsten »Linefeed«
<CTRL-l>	Erneuert den Bildschirminhalt
<CTRL-m>	Fügt ein »Linefeed« ein
<CTRL-n>	Setzt den Cursor eine Zeile nach unten
<CTRL-o>	Fügt eine Zeile ein
<CTRL-p>	Setzt den Cursor eine Zeile nach oben
<CTRL-q>	Initialisiert den »Quote-Mode« für die Übernahme eines Steuerzeichens
<CTRL-r>	Sucht rückwärts nach einem Begriff
<CTRL-s>	Sucht vorwärts nach einem Begriff
<CTRL-t>	Vertauscht die letzten beiden Zeichen
<CTRL-u>	Setzt die Zählvariable
<CTRL-v>	Setzt den Cursor eine Bildschirmseite nach unten
<CTRL-w>	Löscht einen vorher markierten Block
<CTRL-x>	Verzweigt zu den erweiterten Befehlen (siehe unten)
<CTRL-y>	Fügt einen Block ein
<CTRL-z>	Beendet M2Emacs mit vorherigem Speichern

### Erweiterte Befehle

Diese erreichen Sie mit der Tastenkombination <CTRL-x>. Nach dieser wartet M2Emacs auf die Eingabe eines weiteren Zeichens. Dies kann eines der folgenden sein:

<CTRL-b>	Zeigt eine Liste aller Textpuffer
<CTRL-c>	Beendet M2Emacs, ohne zu speichern
<CTRL-d>	Löscht die aktuelle Zeile
<CTRL-f>	Gibt der aktuellen Datei einen neuen Namen
<CTRL-j>	Entfernt das »Linefeed« aus der aktuellen Zeile
<CTRL-l>	Wandelt in einem vorher markierten Block alle Groß- in Kleinbuchstaben
<CTRL-m>	Zeigt den nächsten v. Compiler gefundenen Fehler
<CTRL-n>	Scrollt den Text nach oben
<CTRL-p>	Scrollt den Text nach unten
<CTRL-r>	Öffnet eine neue Datei
<CTRL-s>	Speichert den aktuellen Textpuffer
<CTRL-u>	Wandelt in einem vorher markierten Block alle Klein- in Großbuchstaben
<CTRL-v>	Liest eine Datei, ohne den Textpuffer zu löschen
<CTRL-w>	Speichert den Textpuffer unter neuem Namen
<CTRL-x>	Vertauscht den Cursor mit einer gesetzten Marke
<CTRL-z>	Verkleinert das aktuelle Fenster
<!>	Führt einen CLI-Befehl aus

<=>	Schreibt Informationen über die aktuelle Datei in die Statuszeile
<( )>	Leitet eine Makrodefinition ein
<>>	Beendet eine Makrodefinition
<0>	Schließt ein Fenster
<1>	Schließt alle Fenster mit Ausnahme des aktuellen
<2>	Teilt das aktuelle Fenster
<b>	Wechselt in einen anderen Textpuffer
<c>	Schaltet die Groß-/Klein-Unterscheidung für Suchbefehle um
<e>	Führt ein vorher definiertes Makro aus
<i>	Schaltet den Einrück-Modus aus und ein
<k>	Löscht einen Textpuffer
<n>	Setzt den Cursor in das nächste Fenster
<p>	Setzt den Cursor in das vorhergehende Fenster
<w>	Schaltet die Erzeugung von Icons an und ab
<z>	Vergrößert das aktuelle Fenster
<DEL>	Löscht die aktuelle Zeile

### Escape-Befehle

Diese Gruppe von Befehlen erreichen Sie mit der ESC-Taste. Nachdem Sie diese gedrückt haben, wartet M2Emacs auf die Eingabe eines weiteren Zeichens. Die erlaubten Zeichen sind:

<!>	Setzt die aktuelle Zeile an den oberen Fensterrand
<.>	Setzt eine Marke
<>>	Setzt den Cursor an das Ende des Textpuffers
<<<	Setzt den Cursor an den Anfang des Textpuffers
<b>	Setzt den Cursor ein Wort nach links
<c>	Wandelt in einem Wort den ersten Buchstaben in Groß-, alle anderen in Kleinschreibung
<d>	Löscht das Wort rechts vom Cursor
<f>	Setzt den Cursor ein Wort nach rechts
<l>	Wandelt in einem Wort alle Groß- in Kleinbuchstaben
<m>	Springt an eine anzugebende Zeile
<u>	Wandelt in einem Wort alle Klein- in Großbuchstaben
<v>	Setzt den Cursor eine Bildschirmseite nach oben
<w>	Kopiert einen markierten Bereich in den Zwischenspeicher
<y>	Kopiert einen markierten Bereich in eine Datei
<%>	Ersetzt einen angegebenen Begriff durch einen anderen

### Cursortasten und Maus

Sie können den Cursor selbstverständlich auch mit den Cursortasten und mit der Maus steuern. Die Cursortasten haben die gewohnten Funktionen. Gleichzeitig mit SHIFT gedrückt bewirken sie, daß der Cursor an den jeweiligen Rand springt (beziehungsweise an den Anfang/das Ende des Textpuffers). Mit den ALT-Tasten springen Sie von Wort zu Wort beziehungsweise von Bildschirmseite zu Bildschirmseite. Sie können mit der Maus den Cursor auf dem Bildschirm positionieren oder per Klick in die Statuszeile (ganz unten) seitenweise scrollen. Mit der rechten Maustaste markieren Sie den Anfang eines Blocks. Als Block gilt der Bereich zwischen Marke (mit der rechten Maustaste gesetzt) und Cursor. Mit einem Klick in die Statuszeile (unter der gestrichelten Linie) scrollen Sie den Bildschirminhalt um eine Seite. Verwenden Sie die linke Taste um vorwärts zu scrollen, die rechte für das Rückwärtsscrollen.

Die meisten Befehle sind auf verschiedenen Wegen erreichbar (per Menü, Mausclick, Funktionstaste oder Tastaturkürzel. Es bleibt Ihnen überlassen, welche Eingabe Sie bevorzugen.

Tabelle 1. Alle Steuercodes von »M2Emacs« auf einen Blick

zu werden, starten Sie den Editor (geben Sie dazu »m2emacs« ein) und machen Sie sich mit Hilfe der Befehlsliste in Tabelle 1 mit seiner Bedienung vertraut.

Wer den MicroEmacs von der Extras-Diskette kennt, wird bei dem Modula-Editor viele Funktionen wiedererkennen.

Selbstverständlich können Sie auch einen beliebigen an-

deren Editor verwenden. M2Emacs hat den Vorteil, daß alle Modula-Fehlermeldungen im Editor an der Stelle angezeigt werden, an der sie aufgetreten sind. Aus diesem Grund werden wir uns auf den folgenden Seiten auf diesen Editor beziehen. Wer einen anderen verwenden will, sollte M2Emacs für die Fehleranzeige »griffbereit« haben.



Aber bevor wir Sie jetzt noch länger auf die Folter spannen, starten Sie Ihren Editor und geben Sie die folgenden Zeilen ein:

```
MODULE mydir;
FROM Dos IMPORT Execute;
FROM SYSTEM IMPORT ADR;
VAR ok: INTEGER;
```

```
BEGIN
  ok := Execute (ADR
    ("dir"),0,0);
END mydir.
```

Diese Zeilen speichern Sie mit dem Menüpunkt »save as« unter dem Namen »mydir.mod«. Die Endung ».mod« wird Ihnen häufig begegnen. Sie sagt dem Compiler, daß es sich um einen Quelltext handelt, der in ein sogenanntes Objekt-File übersetzt werden soll.

Bitte beachten Sie in diesem Programm die Groß- und Kleinschreibung. Modula läßt Ihnen hier keine Unsauberkeit durchgehen, sondern erkennt jeden Unterschied zwischen MODULE und Module oder mODULE. Diese drei Begriffe, die im CLI als völlig gleich behandelt werden, sind für Modula drei unterschiedliche Namen. Bei selbst definierten Namen (wie »ok«) legen Sie die Schreibweise selbst fest. Alle fest eingebauten Schlüsselwörter sind grundsätzlich groß geschrieben. Alle anderen Namen sind entweder in Kleinschreibung oder in gemischter Schreibweise gehalten.

Verlassen Sie jetzt den Editor mit »Quit« (wenn Sie ihn mit Run gestartet haben, können Sie auch den Screen nach unten ziehen). Starten Sie Ihren Compiler mit »m2c«. Er meldet sich mit den Zeilen:

```
Amiga Modula-2 Compiler,
...
Demo Version
in>
```

## Ein geklonter »Dir«-Befehl

Der Cursor steht hinter »in>« und wartet auf Ihre Eingabe. Geben Sie Ihr soeben gespeichertes Programm »mydir.mod« an. Sie sehen vier Zeilen angezeigt. Zeilen mit einem einleitenden Minuszeichen bezeichnen von M2Amiga gelesene Files, das Pluszeichen kennzeichnet Dateien, die vom Compiler angelegt werden. »in>« fragt Sie nach dem nächsten File, das Sie compilieren möchten. Geben Sie nur <RETURN> ein, um den Compiler zu verlassen.

Sehen Sie sich jetzt Ihr aktuelles Directory an. Es weist zwei neue Einträge auf: »mydir.ref« und »mydir.obj«. Die Referenz-Datei (mit der Endung ».ref«) ist für einen Debugger gedacht, die Objekt-Datei (».obj«) wird vom Linker zu ausführbarem Code verarbeitet. Dabei werden die mit »FROM... IMPORT...« bezeichneten Dateien eingebunden. Starten Sie den Linker mit »m2l«. Sie erhalten eine ähnliche Meldung wie vom Compiler. Auf »in>« antworten Sie mit »mydir.obj«. Der Linker zeigt Ihnen an, welche Dateien zum fertigen Programm zusammengebunden werden. Verlassen Sie ihn beim nächsten »in>« mit <RETURN> und starten Sie Ihr erstes Programm mit »mydir«. Es bringt die gleiche Ausgabe auf den Bildschirm wie das AmigaDOS Dir-Kommando.

Sie können sich Ihr Directory übersichtlicher gestalten, indem Sie drei zusätzliche Unterverzeichnisse anlegen:

```
makedir sym
makedir ref
makedir obj
```

Für »sym« werden wir uns später noch interessieren, in »ref« und »obj« legt der Compiler automatisch alle Dateien mit der entsprechenden Endung ab. Achten Sie aber darauf, jetzt dem Linker den korrekten Pfad anzugeben (in unserem Beispiel »obj/mydir.obj«). Dieses Vorgehen hat einen immensen Vorteil: Sie haben im aktuellen Directory nur noch den Quellcode und das ausführbare Programm, alles andere »verschwindet« in den Unterverzeichnissen.

Bevor wir jetzt weiterarbeiten, wollen wir gemeinsam einen Blick auf »mydir.mod« werfen. Die erste Zeile legt den Namen des späteren ausführbaren Programms fest. Dieser kann sich durchaus vom Filenamen des Quelltexts unterscheiden, daher muß er hier nochmals angegeben werden. Machen Sie die Probe aufs Exempel und ändern Sie den Namen in der ersten Zeile in »yourdir« – den gleichen Namen sollten Sie in der letzten Zeile hinter »END« verwenden. Compilieren Sie das Programm neu. Sie werden feststellen, daß M2Amiga sich kein bißchen um den Namen Ihres Quelltexts schert, sondern alle Files stur nach dem Text hinter »MODULE« benennt.

Dieses sogenannte reservierte Wort »MODULE« muß in jedem Quelltext am Anfang stehen. Nach MODULE folgt der

Name des späteren ausführbaren Programms. Dieser muß mit einem Semikolon abgeschlossen sein, da er (fast) beliebig lang sein kann. Derselbe Name muß noch einmal auftauchen – am Ende des Quelltexts, hinter »END«, jetzt allerdings mit einem Punkt abgeschlossen.

Einen Quelltext, der von den Zeilen

```
MODULE name;
...
END name.
```

eingerahmt wird, nennt man »Modul«. Ein Programm kann aus mehreren Modulen beste-

## Namen

... können Sie in Modula für Variablen, Konstanten, Typen, Prozeduren und Module vergeben. Ein Name muß in der Deklaration vereinbart sein und kann von da an verwendet werden. Ein einmal vergebener Name kann nicht neu definiert werden (eine Variable »Wert«, die später als Konstante definiert wird, führt zu einer Fehlermeldung).

Sie können Namen global oder lokal definieren. Innerhalb einer Prozedur vergebene Namen (lokal) sind außerhalb dieser Prozedur nicht »sichtbar«. Folglich können sie dort für andere Zwecke neu vereinbart werden.

Namen können auch importiert werden. Sie erhalten dann automatisch die in der Bibliothek festgelegte Bedeutung (als Variable, Prozedur oder ähnliches).

hen, die unabhängig voneinander compiliert und getestet werden können.

Zwischen diesen beiden Zeilen finden Sie den eigentlichen Programmtext. Dabei unterscheidet Modula peinlich genau zwischen dem sogenannten Deklarationsteil und dem Anweisungsteil. Der Deklarationsteil legt alle Daten fest, mit denen gearbeitet werden soll. Der Anweisungsteil führt die »Arbeit« aus.

In unserem Beispiel verwenden wir sowohl interne als auch externe Deklarationen. Der externe Anteil muß mit einer »IMPORT«-Anweisung importiert werden. Die Zeile

```
FROM Dos IMPORT Execute;
spricht fast für sich. Der Name Execute wird in »Dos« gesucht und – falls vorhanden – übernommen. Das gleiche gilt für »ADR« und »SYSTEM«. M2Amiga sucht in den Bibliotheken nach dem angegebenen Na-
```

men. Aber wie erfährt der Compiler und der Linker, wo er die Bibliotheken zu suchen hat? Sehen Sie sich im »S:«-Ordner das File »m2path« an. Es enthält die Suchpfade. Wenn Sie eine eigene Bibliothek geschrieben haben und diese verwenden wollen, müssen Sie einen neuen Suchpfad hinzufügen.

Sie können eine Bibliothek auch komplett und »unbesehen« einbinden. Sie können dann alle dort definierten Namen verwenden:

```
IMPORT Dos;
```

Allerdings wird Ihr Programm dadurch unnötig aufgebläht (alle definierten Namen werden eingebunden). Intern deklarierte Namen können Sie fast beliebig vergeben. Modula akzeptiert allerdings nicht jeden. Es gibt einige Zeichen, die nicht auftreten dürfen. Dazu gehören alle Sonderzeichen wie ä, ö, ü, ß oder die Zeichen mit einer festen Bedeutung (beispielsweise das Semikolon, der Punkt, die Klammern).

Sie können aus allen alphanumerischen Zeichen (Buchstaben und Zahlen) bis zu 255 Zeichen lange Namen bilden, die Sie für Variablen, Konstanten, Prozeduren oder Module verwenden. Dabei sollten Sie sich allerdings an die Konvention halten, daß Großschreibung den vordefinierten Namen von Modula vorbehalten bleibt. Alle selbstdefinierten sollten in Kleinschreibung oder gemischt sein.

Sehen wir uns die Deklarationszeilen von »mydir« etwas genauer an. Die Zeile

```
FROM Dos IMPORT Execute;
importiert die Funktion »Execute« aus der Bibliothek »Dos«. Die nächste Zeile
```

```
FROM SYSTEM IMPORT ADR;
importiert die Funktion »ADR« aus der Bibliothek »SYSTEM«. Diese werden Sie vergeblich in den mit »m2path« angegebenen Pfaden suchen. SYSTEM ist im Compiler eingebaut (daher auch die Großschreibung).
```

Die letzte Deklaration definiert eine Variable »ok« mit dem Typ »INTEGER«. In Modula müssen Sie – im Gegensatz beispielsweise zu Basic – jede verwendete Variable vorher definieren. Dies ist eine große Hilfe bei der Fehlersuche. Denn dadurch kann der Compiler falsche (oder unsaubere) Zuweisungen aufdecken, die im anderen Fall eine aufwendige Fehlersuche nach sich ziehen würden. Der Anweisungsteil von Mydir enthält nur eine Zeile



(BEGIN markiert den Anfang des Anweisungsteils):

```
ok := Execute (ADR
("dir"),0,0);
```

Execute ruft ein beliebiges CLI-Kommando auf. Die Variable »ok« wird gebraucht, um den Rückgabewert von Execute aufzunehmen.

Dieser hat allerdings in der aktuellen Betriebssystemversion keine Funktion. Execute benötigt drei Parameter, die durch Kommata getrennt werden. Der erste (»ADR("dir")«) ist die Speicheradresse eines Strings, die anderen beiden werden nicht gebraucht und deshalb auf Null gesetzt.

Die Adresse eines Strings (der String besteht in unserem Fall aus den drei Buchstaben d, i und r) erhalten Sie mit der Funktion ADR. Diese hat einen sogenannten Rückgabewert vom Typ »ADDRESS«. Funktionen, die einen Wert an das aufrufende Programm zurückgeben, werden wir noch einige kennenlernen. Der Aufruf funktioniert immer nach demselben Schema: Einer Variablen wird der Wert zugewiesen, den die Funktion zurückgibt. Dies geschieht mit dem Zuweisungsoperator (»:«). Links von diesem steht die Variable, die den neuen Wert aufnehmen soll, rechts die Funktion (oder Variable), die den Wert liefert.

## Ein eigenes »Command Line Interface«

Betrachten Sie nochmals das Programm Mydir. Es ist etwas lästig, wenn Sie – um ein neues CLI-Kommando aufzurufen – jedesmal das Programm ändern müssen. Besser wäre es doch, den Anwender unseres Programms ein beliebiges Kommando eingeben zu lassen, das dann ausgeführt wird.

Dazu brauchen wir eine Variable, die den vom Anwender eingegebenen String aufnimmt – nennen wir sie »Kommando«. Außerdem muß der String von der Tastatur gelesen werden. Aber sehen Sie sich am besten unser Beispiel an:

```
MODULE myCLI;
```

```
FROM Dos IMPORT Execute;
FROM SYSTEM IMPORT ADR;
FROM InOut IMPORT Read
String, WriteString,
WriteLn;
```

```
VAR ok: INTEGER;
Kommando: ARRAY
[0..79] OF CHAR;
```

```
BEGIN
LOOP
WriteString (" ");
ReadString
(Kommando);
IF (CAP(Kommando[0])
="Q") THEN EXIT END;
ok := Execute (ADR
(Kommando),0,0);
WriteLn
END
END myCLI.
```

Um den String von der Tastatur zu lesen, verwenden wir die Zeile

```
ReadString (Kommando);
```

Statt dieses Rad neu zu erfinden, nehmen wir eine Prozedur aus der Bibliothek »InOut« (siehe Import-Zeile). Dieser wird als Parameter der String übergeben, der mit Zeichen gefüllt werden soll (in unserem Fall »Kommando«). Vorher bauen wir uns ein eigenes Prompt:

```
WriteString (" ");
```

WriteString schreibt eine beliebige Zeichenfolge auf den Bildschirm. Sie können als Argument direkt die Zeichenfolge übergeben – wie in unserem Beispiel – oder eine Variable (beziehungsweise Konstante) vom Typ String haben wir in der Zeile

```
Kommando: ARRAY [0..79]
OF CHAR;
```

definiert. Ein String besteht aus einer Aneinanderreihung einzelner Zeichen. Ein Zeichen hat in Modula den Namen »CHAR«, eine Aneinanderreihung heißt »ARRAY«. Für ein Array muß dem Compiler die Größe angegeben werden, dies geschieht mit der Bereichsangabe in eckigen Klammern (»[0..79]«). Unser String kann also maximal 80 Zeichen aufnehmen.

Jeder Variablen, die Sie verwenden, ist ein Typ zugeordnet. Sie vergeben einen Namen und wählen dazu einen Typ aus den vorhandenen. Zur Trennung dieser beiden Teile der Deklaration wird der Doppelpunkt verwendet. Vorerst werden wir uns auf die Standarddatentypen von Modula beschränken, Sie werden später noch Möglichkeiten kennenlernen, sich eigene Typen zu definieren.

Sie kennen bereits die Typen »INTEGER« (für ganzzahlige Werte zwischen 32768 und -32767), »CHAR« (für ASCII-Zeichen) und »ARRAY« (für Felder aus einem dieser Typen). Daneben existieren »CARDINAL« für positive ganzzahlige Werte (von 0 bis 65535), »BOOLEAN« für Ja-/Nein-Entscheidungen, »REAL« für Gleitpunktzahlen (im Bereich von -10)38 bis 10)38), »ADDRESS« und »POINTER« für die Adressierung von Variablen (ganzzahlige Werte im adressierbaren Speicherbereich), »SET« für Mengen (mit einer maximalen Anzahl von 32 Elementen) und »RECORD« für verbundene Strukturen aus unterschiedlichen Typen. Wenn Sie mit den meisten genannten Typen nichts anfangen können, ist dies kein Grund für Komplexe, Sie werden im Verlauf dieses Workshops alle kennenlernen.

Neben Variablen gibt es die Möglichkeit, Konstanten zu deklarieren. Diese behalten während des gesamten Programmablaufs einen zugewiesenen festen Wert. Konstanten müssen nicht mit dem Typ vereinbart werden, dieser ist für den Compiler aus dem Wert ersichtlich. Konstanten machen Ihre Programme »wartungsfreundlicher«. Stellen Sie sich vor, Sie programmieren eine Datenbank und rechnen anfangs mit höchstens 3000 Einträgen. Alle Routinen, die in irgendeiner Form auf diese Datensätze zugreifen, halten sich an diese Beschränkung und durchsuchen Ihre Datenbank nur bis zum dreitausendsten Satz. Nachdem Sie die Hälfte Ihrer Adressen eingegeben haben, merken Sie aber, daß Ihr Freundeskreis weit größer ist als ursprünglich angenommen, und Sie möchten statt 3000 lieber 6000 Einträge zulassen. Wenn Sie jetzt das ganze Programm nach der Zahl 3000 durchforsten und jedesmal überprüfen, ob diese 3000 auf 6000 erhöht werden soll oder eine andere Bedeutung hat, dann viel Spaß! Viel einfacher ist es, wenn am Anfang Ihres Programms einmal die Zeile

```
CONST maxEintraege =
3000;
```

steht. Von jetzt an verwenden Sie für die maximale Anzahl Einträge nicht mehr die Zahl 3000, sondern den Namen »maxEintraege«. Sollten Sie auf die Idee kommen, diese Zahl zu ändern, brauchen Sie nicht mehr lange zu suchen. Sie ändern nur die Konstante. Für unser Beispielprogramm würde sich eine Konstante für das Promptzeichen anbieten. Fügen Sie bei den Deklarationen eine Zeile ein:

```
CONST Prompt = " ";
```

und ändern Sie die Zeile mit WriteString in

```
WriteString (Prompt);
```

Sicher haben Sie den Unterschied zwischen der Wertzuweisung (»:«) und dem Gleichheitszeichen bemerkt. Das Gleichheitszeichen wird hier verwendet, um deutlich zu machen, daß der Name als Synonym für den dahinter stehenden Wert verwendet werden

## Stapelweise Arbeit für Ihre Schleifen

kann. Dagegen kann eine Wertzuweisung nur innerhalb des Anweisungsteils stehen und wirkt ausschließlich im Zusammenhang mit Variablen. Die Wertzuweisung hat zur Folge, daß eine Variable einen neuen Wert erhält – Konstanten können natürlich keinen neuen Wert annehmen (»Konstante« heißt Gleichbleibende).

Die letzte neue Anweisung ist »WriteLn«. Diese sendet einen Zeilenvorschub an den Bildschirm. Wir erreichen damit eine Leerzeile nach jedem ausgeführten Kommando.

Vielleicht haben Sie bemerkt, daß drei weitere Zeilen ins Programm eingebaut wurden. Diese haben keine direkte Auswirkung auf die Anweisungen. Es handelt sich um eine sogenannte Schleife. Eine Schleife ist eine Folge von Anweisungen, die mehrfach ausgeführt werden soll. Damit Sie nicht 5000mal dieselben Befehle untereinander schreiben müssen, dürfen Sie statt dessen dem Programm mitteilen »Bitte mach das, wenn du fertig bist, fang von vorne an«. Dies geschieht mit den reservierten Worten »LOOP« und »END«. Diese liegen wie eine Klammer um den sogenannten Schleifenkörper.

Wenn Sie eine Schleife nur aus LOOP und END bauen, werden Sie nach dem Start des Programms schnell ein Problem erkennen: Es gibt keine Möglichkeit, diese Schleife zu verlassen. Ihr Programm verläuft sich in einer sogenannten Endlosschleife. Diese Konstruktion sollten Sie beim Programmieren besser vermeiden.

Um eine Schleife zu verlassen, benötigen Sie eine genau definierte Abbruchbedingung. In unserem Beispiel ist dies ein »q« als erster Buchstabe des eingegebenen Kommandos. Der Anwender erfüllt diese Bedingung mit »Q«, »q« oder »quit« – aber auch mit »Quick« oder »Quintessenz«. Zugegeben dieses Verfahren ist nicht gerade elegant, aber es geht in erster Linie um eine Demon-



stration dieser Technik. Dazu wird eine bedingte Anweisung verwendet:

```
IF Bedingung THEN EXIT  
END;
```

Oder übersetzt: Wenn die Bedingung erfüllt ist, dann verlasse die Schleife - Ende der Anweisung. Das Schlüsselwort »EXIT« ist verantwortlich für das Verlassen der LOOP-Schleife. Dies soll aber nur geschehen, wenn die vorher abgefragte Bedingung erfüllt ist. Ist dies nicht der Fall, wird diese Anweisung ignoriert.

Als Bedingung haben wir »(CAP(Kommando[0]) = "Q")« gewählt. Sie ist genau dann erfüllt, wenn das erste Zeichen des eingegebenen Strings ein q oder ein Q ist. Die Funktion »CAP« verlangt als Argument ein Zeichen. Sie liefert dasselbe Zeichen als Großbuchstabe zurück. Damit ist es möglich, den Anwender von der korrekten Groß- oder Kleinschreibung zu entlasten. »Kommando[0]« ist eine Konstruktion, die bei ARRAYS auf das in eckigen Klammern angegebene Element zugreift. Nachdem wir unseren String mit einer Länge von 0 bis 79 definiert haben, erreichen wir mit der 0 das erste Element. Bitte beachten Sie, daß das Gleichheitszeichen hier in einer neuen Bedeutung verwendet wird. Es handelt sich hier um einen sogenannten »Vergleichsoperator«, der feststellt, ob die beiden Argumente den gleichen Wert haben.

Nachdem das Programm jetzt aus mehreren Zeilen besteht, lohnt es sich, kurz auf das Semikolon am Ende jeder Zeile einzugehen. Einzelne Anweisungen werden in Modula mit einem Semikolon getrennt. Vor END muß keines stehen (siehe WriteLn). Der Compiler erkennt am Semikolon das Ende einer Anweisung, daher können mehrere Anweisungen in einer Zeile stehen. Sie sollten damit allerdings vorsichtig umgehen, denn Ihr Programm kann dadurch nur unleserlicher werden.

Speichern Sie jetzt Ihr Programm und compilieren Sie es. Linken Sie es diesmal aber nicht auf die gewohnte Weise. Wir möchten erreichen, daß unser Programm nicht nur aus dem CLI, sondern auch von der Workbench gestartet werden kann. Dazu geben Sie ein

```
m21 -i mycli.obj
```

Wenn der Linker mit der Option »-i« gestartet wird, erzeugt er nicht nur das ausführbare Programm, sondern auch eine »info«-Datei, die für das Icon

zuständig ist. Geben Sie im CLI ein

```
loadwb
```

und verkleinern Sie das CLI-Fenster so weit, daß Sie das Symbol Ihrer Diskette erkennen. Klicken Sie dieses zweimal an. Sie sehen ein neues Programm-Icon. Mit einem Doppelklick starten Sie Ihr neues Programm, Sie werden feststellen, daß automatisch das benötigte CLI-Fenster geöffnet wird.

Sie können sowohl beim Linken als auch beim Compilieren den Filenamen als Argument übergeben. Dies sollten Sie aber nur machen, wenn Sie nicht mehr als ein File bearbeiten wollen. Denn Compiler und Linker werden dann automatisch »nach getaner Arbeit« beendet und aus dem Speicher entfernt. Für das zweite File müßten sie dann wieder neu geladen werden.

Testen Sie Ihr Programm ausführlich; Sie werden feststellen, daß es noch einige Nachteile aufweist. So können Argumente (beispielsweise für »copy«) nicht so leicht übergeben werden. Sie müssen dafür den String in Anführungszeichen setzen, da sonst das Leerzeichen als Trennung interpretiert wird. Dies liegt daran, daß die Routine ReadString jeden String nur bis zum nächsten Leerzeichen oder <RETURN> liest. Wollen wir diese Unsauberkeit vermeiden, so müssen wir uns eine eigene Routine schreiben. Doch bevor wir uns an diese Aufgabe wagen, wollen wir für unser Programm ein eigenes Fenster öffnen.

## Das erste eigene Fenster

Dazu fügen Sie zwischen BEGIN und LOOP eine zusätzliche Zeile ein:

```
File := OpenFenster ();  
und zwischen den beiden END  
CloseFenster (File);
```

Außerdem muß hinter dem ersten END jetzt ein Semikolon stehen, da dieses nicht mehr unmittelbar von einem weiteren END steht, sondern von einer Anweisung getrennt werden muß. Testen Sie mit einem Compiler-Lauf, was geschieht, wenn Sie dies vergessen sollten. Diese Art von Fehlern ist beim Programmieren in Modula sehr häufig, daher sollten Sie derartige Fehlermeldungen auch ohne nachzuschlagen erkennen. Entfernen Sie ein beliebiges Semikolon. Speichern

Sie das Programm. Compilieren Sie es und starten Sie den Editor. Welche Fehlermeldung erhalten Sie bei Druck auf <F2>?

Die Variable »File« muß definiert werden, dies geschieht hinter den beiden Variablendefinitionen (aber vor BEGIN) mit

```
VAR File: FileHandlePtr;
```

Der Datentyp »FileHandlePtr« ist kein Standardtyp von Modula, er ist in der Bibliothek »Dos« definiert. Fügen Sie der Importliste diesen Typ hinzu. Die Zeile »FROM Dos...« sollte dann folgendermaßen aussehen:

```
FROM Dos IMPORT  
FileHandlePtr, Execute;
```

Was jetzt noch fehlt, sind die beiden Funktionen OpenFenster und CloseFenster. Für beide schreiben wir uns ein eigenes Unterprogramm. Dafür benötigen wir die Dos-Routinen »Close« und »Open« sowie die Konstante »newFile«. Erweitern Sie Ihre Dos-Importliste also um die Namen Open, Close und newFile. Darauf fügen Sie zwischen den Variablendeklarationen und BEGIN folgende Zeilen ein:

```
PROCEDURE OpenFenster  
(): FileHandlePtr;  
VAR Handle: FileHandlePtr;  
BEGIN  
  Handle := Open (ADR  
  ("CON:0/0/640/256/  
  MyCli"), newFile);  
  RETURN Handle  
END OpenFenster;
```

```
PROCEDURE CloseFenster  
(f: FileHandlePtr);  
BEGIN  
  Close (f)  
END CloseFenster;
```

Diese Zeilen lassen sich sogar noch vereinfachen, wenn man den Rückgabewert der Open-Funktion nicht an eine Variable übergibt, sondern direkt an RETURN weiterleitet. Sie sparen dadurch die Variablendefinition und können das Programm somit auf vier Zeilen reduzieren. Dennoch bleiben wir wegen der Übersichtlichkeit vorerst bei obiger Version.

Die erste Zeile besteht aus der Prozedurdeklaration. Diese enthält das Schlüsselwort »PROCEDURE«, deren Namen, Parameter und Typ. Die Parameterliste kann durchaus wie in unserem Beispiel - leer sein. Sie werden später noch Prozeduren kennenlernen, denen einer oder mehrere Parameter übergeben werden. In

der zweiten Zeile wird eine lokale Variable mit Namen »Handle« und vom Typ »FileHandlePtr« deklariert. Die dritte und die letzte Zeile haben die gleiche

## Unbegrenzte Flexibilität mit Prozeduren

Funktion wie bei einem Modul. Der einzige Unterschied ist das Semikolon am Ende. Es kennzeichnet das Ende der Prozedur im Gegensatz zum Modulende. Der Anweisungsteil beinhaltet neben der Zeile, die für das Öffnen des Fensters zuständig ist, eine RETURN-Anweisung. Diese übergibt den Rückgabewert an das aufrufende Programm. Das Argument für RETURN ist eine Variable vom selben Typ wie die Prozedur. Allerdings kann auch eine Funktion mit einem Rückgabewert vom selben Typ als Argument benutzt werden. Beispielsweise:

```
RETURN Open (ADR ("CON:  
0/0/640/256/MyCli"),  
newFile);
```

Compilieren und linken Sie Ihr neues Programm. Nach dem Start werden Sie feststellen, daß sich tatsächlich ein Fenster mit Namen »mycli« öffnet, das den ganzen Bildschirm einnimmt. Mit der Ein-/Ausgabe haben wir aber ein kleines Problem: Sie läuft immer noch über das CLI-Fenster. Mit den Routinen aus InOut kommen wir nicht besonders weit. Diese beziehen sich auf das CLI-Fenster. Man könnte zwar diese Kanäle umlenken, aber nachdem wir auch mit der Parameterübergabe unzufrieden waren, ist es jetzt an der Zeit, daß wir unsere eigenen Routinen programmieren. Keine Angst - es ist einfacher als Sie denken.

Wir haben bislang drei Routinen aus InOut verwendet: ReadString, WriteString und WriteLn. Da wir InOut nicht mehr verwenden wollen, können Sie die Zeile

```
FROM InOut IMPORT ReadString, WriteString,  
WriteLn;
```

ersatzlos streichen. Fügen Sie statt dessen hinter der Importliste von Dos noch zwei Routinen an: »Read« und »Write«. Sie ahnen vermutlich schon, wo die Reise hingeht. Wir wollen auch diesmal nicht alles alleine machen, sondern wir sichern uns die Unterstützung des Betriebssystems.

Fügen Sie zwischen den Variablendeklarationen und der



BEGIN-Anweisung folgende Zeilen ein:

```
PROCEDURE WriteString
(fh: FileHandlePtr; str:
ARRAY OF CHAR);
BEGIN
    ok := Write (fh,
ADR(str), SIZE(str))
END WriteString;
```

```
PROCEDURE ReadString
(fh: FileHandlePtr; VAR
str:ARRAY OF CHAR);
VAR c: CHAR;
i: INTEGER;
```

```
BEGIN
IF HIGH (str) >=1
THEN
    i := 0;
    REPEAT
        ok := Read (fh,
ADR(c), SIZE(c));
        str[i] := c;
        INC(i)
    UNTIL (i >= HIGH
(str)) OR
(c = 12C)
        str[i-1] := 0C;
    END;
END ReadString;
```

```
PROCEDURE WriteLn (fh:
FileHandlePtr);
VAR LF: CHAR;
BEGIN
    LF := 12C;
    ok := Write (fh,
ADR(LF), SIZE(LF));
END WriteLn;
```

Außerdem ändern Sie die Zeile mit dem Execute-Befehl in

```
ok := Execute (ADR
(Kommando),0,File);
```

Dies bewirkt, daß die von Execute hervorgerufene Ausgabe nicht ins CLI-Fenster erfolgt (dies erreichen Sie mit der Angabe von »0«), sondern in das Fenster, das unserem »FileHandlePtr« zugeordnet ist.

Da wir sie nun ausgiebig verwenden, ist es an der Zeit, ein paar Worte über Prozeduren zu verlieren. Prozeduren sind Ansammlungen von Anweisungen, die sozusagen »mehrfach verwendbar« sind. Sie stellen sich eine Reihe von Befehlen zusammen, die eine besondere Aufgabe erfüllen. Dieser Ansammlung geben Sie einen Namen. Von jetzt an brauchen Sie nicht jedesmal die Befehle untereinander zu schreiben, wenn sie ausgeführt werden sollen, sondern es genügt, den Namen der Prozedur anzugeben.

Diese einfachste Form einer Prozedur trifft man relativ selten an. Eine Prozedur kann daneben zu einer Vielzahl von Spe-

zialaufgaben herangezogen werden. Aber sehen Sie sich zunächst das einfache Format der Prozedur an:

```
PROCEDURE ProcName;
BEGIN
    Anweisungen
END ProcName;
```

In diesem Beispiel machen Sie nichts weiter als alle Anweisungen unter einem Namen (»ProcName«) zusammenfassen.

Eine Prozedur kann aber weit mehr. Sie kann Werte aus dem Hauptprogramm übernehmen, dort verändern, Werte zurückgeben und einen eigenen Deklarationsteil beinhalten. Eine Prozedur, die einen Wert aus dem Hauptprogramm übernimmt, finden Sie in »CloseFenster«. Der Variablen »f«, die im Prozedurkopf definiert ist, wird beim Aufruf ein Wert übergeben (in unserem Beispiel »File«). Dieser übergebene Wert muß vom selben Typ sein wie der im Prozedurkopf angegebene. In dieser Form übergebene Variablen kopieren ihren Wert in die Prozedurvariable und bleiben selbst unverändert. Soll

## Der Schlüssel zu den Daten: Parameter

mit jeder Änderung des Wertes in der Prozedur auch der Wert im Hauptprogramm verändert werden, so muß die Variable »gebunden« werden. Dies geschieht - wie in der Prozedur »ReadString« - mit dem Schlüsselwort »VAR«. Die allgemeine Form der Variablenübergabe lautet also:

```
([VAR] Var1 [,Var2[.]]:
Typ1[; Var3: Typ2]);
```

Diese Parameterliste ist beliebig verlängerbar. Sie können für jede Variable entscheiden, ob nur der Wert übergeben wird, oder ob sie gebunden sein soll. Ebenso muß für jede Variable der Typ angegeben werden. Variablen vom gleichen Typ können zusammengefaßt werden. Dazu trennen Sie die Variablennamen durch Kommata, bevor nach einem Doppelpunkt der Name des Typs folgt.

Prozeduren können auch Werte zurückgeben. Dazu muß der Name der Prozedur mit einem eigenen Typ versehen werden. Dieser wird nach der Parameterliste hinter einem Doppelpunkt angegeben. Wird dieser Prozedur kein Parameter übergeben, so muß eine leere Liste (»0«) in den Prozedurkopf aufgenommen werden. Als Rück-

gabewert dürfen Sie keine verbundenen Typen (Arrays und Records) verwenden. Der Wert wird nicht automatisch zurückgegeben, sondern wird mit der »RETURN«-Anweisung an das aufrufende Programm übergeben. Ein Beispiel hierfür finden Sie in der Prozedur »OpenFenster«.

Eine Prozedur kann einen eigenen Deklarationsteil haben. Dieser steht zwischen dem Prozedurkopf und BEGIN. Alle hier definierten Namen gelten automatisch als lokal. Dies bedeutet, daß sie außerhalb dieser Prozedur unbekannt sind oder sogar eine andere Bedeutung haben können.

Die allgemeine Form einer Prozedur lautet:

```
PROCEDURE ProcName
[(Parameterliste):
RückgabeTyp];
Deklarationsteil
BEGIN
    Anweisungsteil;
    [RETURN Rückgabewert]
END ProcName;
```

Die einfachste unserer drei neuen Prozeduren ist »WriteString«. Sie verlangt die Übergabe von zwei Parametern - einem Zeiger auf ein FileHandle und einem String. Sie kennen Strings bisher nur mit einer festen Länge. Es wäre aber sehr unangenehm, wenn einer Prozedur immer nur Strings mit exakt derselben Länge übergeben werden könnten. Sie bräuchten dann für jede neue Stringlänge eine neue Prozedur (beispielsweise »WriteString20«, »WriteString21«...). Um diesen Umstand zu vermeiden, gibt es sogenannte »open arrays«. Diese werden nur mit der Angabe des Basistyps, aber ohne Größe festgelegt. Dadurch kann jede beliebige Größe verarbeitet werden.

Im Deklarationsteil wird eine Ganzzahlvariable definiert, die den Rückgabewert der »Write«-Funktion aufnimmt. Der Anweisungsteil besteht aus einer einzigen Zeile. Die Dos-Funktion »Write« verlangt die Übergabe von drei Parametern:

- fh ist ein Zeiger auf ein FileHandle. Dieses kann sich auf ein Fenster oder auf ein beliebiges anderes File beziehen.
- ADR(str) ist die Adresse des Strings, der in das Fenster geschrieben werden soll.
- SIZE(str) An dieser Stelle muß die Größe des Strings in Byte stehen. Nachdem diese bei der Übergabe nicht bekannt ist, ermitteln wir sie mit der SIZE-Funktion. Diese liefert von einer beliebigen Variablen

den belegten Speicherplatz in Byte.

Den Rückgabewert von Write übernehmen wir in die Variable »ok«, verwerten ihn aber nicht weiter. Er liefert die Anzahl der tatsächlich geschriebenen Zeichen. Damit können Sie kontrollieren, ob Ihr String vollständig geschrieben wurde oder ob ein Fehler aufgetreten ist. Um das Programm möglichst einfach zu halten, wurde auf diese Abfrage verzichtet.

Die Prozedur »WriteLn« verwendet dieselbe Systemroutine, um ein einzelnes Zeichen in das Fenster zu schreiben. Dazu muß außer dem FileHandle kein weiterer Parameter übergeben werden. Allerdings wird eine weitere Variable definiert (»LF«). Diese nimmt ein Steuerzeichen auf, nämlich das Zeichen mit dem ASCII-Code 12. Sie können jedes Zeichen neben seiner Bildschirmdarstellungsform (beispielsweise »"c"«) auch mit dem entsprechenden ASCII-Code erreichen. Den Code übergeben Sie dabei im oktalen Zahlensystem (mit der Basis 8). Das abschließende »C« dient zur Kennzeichnung dieser Darstellung. Steuerzeichen können Sie nur mit dem ASCII-Code erreichen, da sie auf dem Bildschirm nicht dargestellt werden. Jetzt werden Sie sich natürlich fragen, was der Sinn der Sache ist, wenn Steuerzeichen nicht dargestellt werden, diese in ein Fenster zu schreiben. Nehmen Sie WriteLn als Beispiel. Hier wird das Steuerzeichen mit der Nummer 12 in das angegebene Fenster geschrieben. »12C« ist die Kennung für »LineFeed«. Dies bedeutet, der Cursor wird eine Zeile tiefer und an den Anfang der Zeile gesetzt. Sie brauchen sich keine Umrechnungsformel für das oktale Zahlensystem zu merken, in Tabelle 2 finden Sie eine Aufstellung der wichtigsten Steuerzeichen mit einer kurzen Erläuterung, wofür diese verwendet werden.

»ReadString« verlangt zwei Parameter: das FileHandle und den String. Auch den Deklarationsteil kennen Sie bereits. Wirklich neu ist der Anweisungsteil. Dieser ist umfangreicher und enthält auch einiges Neue: Die Funktion »HIGH« kann nur bei »open-array«-Parametern angewendet werden. Sie liefert den höchsten Index des übergebenen Arrays. Dies hat nichts mit der Größe in Byte zu tun (diese wird mit »SIZE« bestimmt), sondern nur mit der Anzahl der Elemente.

Die IF-Abfrage ist nötig, da der Prozedur auch ein String



der Länge Null übergeben werden kann. In diesem Fall würden einige Anweisungen auf nicht erlaubte Elemente zugreifen. Um dies zu vermeiden, wird der Anweisungsteil nur ausgeführt, wenn der höchste Index mindestens 1 ist.

Im Anweisungsteil ist eine neue Form einer Schleife enthalten. Die »REPEAT«-Schleife ist ein sogenannter »Bottom-Testing-Loop«. Dabei wird bei jedem Erreichen des Schleifenendes (»Bottom«) geprüft, ob die Abbruchbedingung erfüllt ist oder nicht. Ist die Bedingung hinter dem Schlüsselwort »UNTIL« logisch wahr, dann wird die Schleife beendet. Im anderen Fall fährt das Programm am Anfang der Schleife (hinter REPEAT) fort. Die Verknüpfung mit »OR« bewirkt, daß der logische Ausdruck wahr ist, sobald eine der beiden Teilbedingungen erfüllt ist. Die erste:

```
(i >= HIGH(str))
```

testet, ob der Schleifenzähler die obere Grenze des Arrays erreicht hat. Die zweite:

```
(c = 120)
```

testet auf die RETURN-Taste. Der String soll abgeschlossen sein, wenn der Anwender beim Eingeben diese Taste gedrückt hat.

Vor der Schleife muß die Zählvariable (»i«) initialisiert werden. Sollten Sie dies einmal vergessen, erhalten Sie keinen Fehler beim Compilieren, sondern

## Effektivität durch Eigenbau

der während der Laufzeit. Testen Sie dies, indem Sie die Zeile entfernen und das Programm neu übersetzen. Sie erhalten einen Requester, der Sie auf eine Bereichsüberschreitung hinweist. »Bereichsüberschreitung« ist dabei etwas ungenau, denn der Wert von i ist nicht zu groß oder zu klein, sondern undefiniert.

In der Schleife werden mit der Read-Funktion einzelne Zeichen eingelesen und in den String kopiert (»str[i] := c«), bis die Abbruchbedingung erfüllt ist. Bei jedem Schleifendurchlauf wird die Zählvariable um eins erhöht (»INC(i)«).

Vielleicht ist Ihnen aufgefallen, daß unsere neuen Prozeduren jeweils einen Übergabeparameter mehr benötigen, als die Prozeduren aus InOut. Dies ist nötig, um das Fenster anzugeben, mit dem gearbeitet werden soll. Diesen Parameter müssen Sie natürlich beim Aufruf angeben. Fügen Sie bei al-

len drei Aufrufen als ersten Parameter »File« ein, sie sollten also folgendermaßen aussehen:

```
WriteString (File,
Prompt);
ReadString (File,
Kommando);
...
WriteLn (File)
```

Wenn Sie jetzt Ihr Programm übersetzen und starten, werden Sie feststellen, daß sich alle Ein- und Ausgaben in Ihrem neuen Fenster abspielen.

nochmals in der neuesten Version, in der nichts mehr von InOut importiert wird. Notieren Sie auch hier beide Werte.

Die Größe der beiden Objekt-Dateien dürfte sich nur minimal unterscheiden. Einen erstaunlichen Effekt werden Sie jedoch beim ausführbaren Programm feststellen. Mit den InOut-Funktionen ist es mehr als doppelt so groß wie mit unseren selbstgebauten!

Die Größe eines Files hat entscheidenden Einfluß auf die Ladezeit. Nicht zuletzt aus diesem

Nachteile. Wie bringen wir diese unter einen Hut?

Die Lösung ist eine selbstgeschriebene Bibliothek. Diese können wir so anlegen, daß nicht allzuviel unnötiger Ballast mitgeschleppt wird, und sie bleibt dennoch für alle unsere Programme erreichbar.

Eine eigene Bibliothek – das klingt kompliziert. Aber es ist wie so oft beim Programmieren: Die Lösung ist einfacher als Sie denken. Nachdem Sie noch etwas Theorie verdaut haben, geht es sofort wieder in die Praxis.

Eine Bibliothek besteht in Modula (fast) immer aus zwei Teilen: der Schnittstelle, Symbol-File genannt, und der Implementation. Zwischen diesen beiden Teilen einer Bibliothek herrscht eine strenge Aufgabenteilung. Die Schnittstelle legt alle für das aufrufende Programm »sichtbaren« Objekte fest. Das können Datentypen, Variablen, Konstanten oder Prozeduren sein. Der Implementationsteil dagegen sorgt dafür, daß diese »angeforderten« Objekte sinnvoll installiert werden. Dabei kann es vorkommen, daß Objekte bereits durch die Festlegung der Schnittstelle hinreichend definiert sind. In diesem Fall bleibt für die Implementation nichts mehr zu tun. In Extremfällen kann dies so weit gehen, daß eine Bibliothek bereits durch die Schnittstelle definiert ist. Dies ist allerdings nur dann möglich, wenn ausschließlich Daten definiert werden. Bei Prozeduren muß eine Implementation vorhanden sein.

Ein großer Vorteil gegenüber anderen Sprachen ist die Möglichkeit, Schnittstellen im voraus festzulegen und die Implementation unabhängig von diesen aufzubauen. Bei einem großen Projekt könnte man so in der ersten Phase definieren, was ein Programm leisten soll. Man überlegt sich, wie das Hauptprogramm aussieht und was in andere Module ausgelagert wird.

## Komfort mit Bibliotheken

Diese Information wird in der Schnittstelle festgehalten. Von diesem Moment an ist die Installation der benötigten Routinen völlig unabhängig. Beispielsweise kann die Programmierung dieser Routinen auf mehrere Programmierer verteilt werden.

Das Ganze klingt für Sie vermutlich im Moment noch wie Zukunftsmusik. Aber bedenken Sie es an folgendem

Steuerzeichen			
sind nicht druckbare Sonderzeichen. Sie werden beispielsweise verwendet, um die Peripheriegeräte zu steuern oder zwischen verschiedenen Modi hin- und herzuschalten. Die für diesen Workshop benötigten sind:			
Dez	Oct	Kürzel	Bedeutung
0	00C	nul	bildet den Abschluß von Strings
10	12C	lf	schaltet eine Zeile weiter
12	14C	ff	sendet einen Seitenvorschub (löscht den Bildschirm)
13	15C	cr	Wagenrücklauf, setzt den Cursor an den Anfang der Zeile
27	33C	esc	<ESC> wurde gedrückt
28	34C	eof	End-of-File-Markierung

**Tabelle 2. Mit diesen Steuerzeichen können Sie die Bildschirmdarstellung beeinflussen**

Bevor wir uns einen Schritt weiter wagen, verändern Sie Ihr Programm etwas. Experimentieren Sie mit dem bisher Gelernten. Sie können dabei nicht viel falsch machen, schlimmstenfalls erhalten Sie eine Fehlermeldung. Versuchen Sie beispielsweise die Prozedur »Open-Fenster« so zu verändern, daß beim Aufruf dieser Funktion die Fenstergröße angegeben werden kann.

Möglicherweise haben Sie während des Übersetzungsvorgangs die Ausgabe des Compilers und des Linkers aufmerksam beobachtet und sich über die Zahlen hinter den mit »+« markierten Files gewundert? Mit einem Pluszeichen werden – wie bereits kurz erwähnt – alle Files markiert, die nicht gelesen, sondern geschrieben werden. Die Zahl hinter dem Filenamen ist die Größe der geschriebenen Datei in Byte.

Machen Sie sich die Mühe und übersetzen Sie MyCLI nochmals in der Version mit den von InOut importierten Funktionen. Notieren Sie sich die Zahl hinter der Objekt-Datei. Diese entsteht beim Compilerlauf und hat die Endung ».obj«. Notieren Sie sich auch die Zahl, die Ihnen der Linker für das ausführbare Programm angibt. Übersetzen Sie jetzt MyCLI

Grund gilt ein kurzes Programm als effektiver gegenüber einem längeren mit gleichen Funktionen. Nachdem das Programm erst nach dem Linkerlauf an Länge zunimmt, müssen wir annehmen, daß der Grund dafür in den Bibliotheken zu suchen ist. Denn der Linker bindet das Hauptprogramm mit den importierten Bibliotheken zusammen. Tatsächlich ist die Bibliothek InOut hier der Hauptverantwortliche für die Speicherverschwendung.

Aber daraus zu folgern, die Entwickler dieser Bibliothek seien unfähig, kurze und effektive Programme zu schreiben, wäre ein zu verwegener Schluß. Eine Bibliothek enthält immer auch Dinge, die man für eine spezielle Anwendung nicht braucht. Diese belasten den Speicher, da sie dennoch importiert werden. Also werden wir in Zukunft am besten ganz ohne Bibliotheken programmieren? Vergessen Sie nicht, daß auch mit der Unterstützung des Betriebssystems nicht alles so einfach zu programmieren ist wie eine eigene Eingabe-Routine. Außerdem brauchen wir Bibliotheken, um Routinen von allen Programmen aus nutzbar zu machen. So haben also beide Varianten ihre Vor- und



Beispiel. Wir könnten in der ersten Phase der Programmentwicklung festlegen, was unser Hauptprogramm machen soll.

Etwa in dieser Form:

```
MODULE MyCLI;

FROM SYSTEM IMPORT ADR;
FROM Dos IMPORT Execute;
FROM DosTool IMPORT
Fenster, OpenFenster,
CloseFenster, Write
String, ReadString,
WriteLn;
FROM StringTool IMPORT
String, Compare;

CONST Prompt = " ";
ExitLoop = "Be-
ende";

VAR ok: INTEGER;
ex: BOOLEAN;
Kommando: String;

BEGIN
Fenster := Open
Fenster ();
LOOP
WriteString
(Fenster, Prompt);
ReadString (Fenster,
Kommando);
ex := Compare
(Kommando,
ExitLoop);
IF ex THEN EXIT END;
(* Abbruch ? *)
ok:=Execute (ADR
(Kommando), 0,
Fenster);
WriteLn (Fenster)
END; (* LOOP *)
CloseFenster (Fenster)
END MyCLI.
```

Bevor Sie dieses Programm speichern, sichern Sie das alte File »MyCLI.mod« unter einem anderen Namen, es wird noch gebraucht. Wenn Sie dieses Programm lesen, werden Sie

feststellen, daß es weniger den Eindruck eines fertigen Programms als den einer stichwortartigen Beschreibung macht. Aber genau das ist unser späteres Hauptprogramm. Sie brauchen kein einziges Zeichen daran ändern, um es zu übersetzen und zu starten!

Wenn Sie entgegen, das sei unglaublich, dann gebe ich Ihnen sogar recht, halte aber dennoch an meiner Behauptung fest. Wenn Sie sich außerdem wundern, warum nicht jeder in Modula programmiert, wundere ich mich mit Ihnen.

Wirklich neu ist in diesem Programm nur der Kommentar hinter der IF-Zeile und vor »CloseFenster«. Kommentare haben keinen Einfluß auf das Programm. Es wird dadurch aber lesbarer. Dies ist bei Modula sicher nur bei längeren Programmen unbedingt nötig - beispielsweise wenn zu viele END-Anweisungen auftauchen, als daß diese noch auf Anhieb richtig zugeordnet werden können.

Vielleicht haben Sie erkannt, daß in unserem Programm zwei neue Bibliotheken auftauchen. »DosTool« und »StringTool« sind keine Standardbibliotheken, folglich müssen wir sie selbst schreiben. Um den Compiler etwas näher kennenzulernen, sollten Sie vorher dieses Programm speichern (nicht vergessen, die alte Version in Sicherheit zu bringen!) und compilieren, ohne daß diese Bibliotheken existieren.

Sie haben es sicher schon vermutet: Wir erhalten eine Fehlermeldung. Sehen Sie sich diese mit M2Emacs an. Sie werden darauf hingewiesen, daß der Compiler kein Symbolfile mit Namen »DosTool« öffnen konnte. Das gleiche gilt für das File »StringTool«. Folglich kann er auch die von dort erwarteten Namen nicht erkennen. Auf den ersten Blick überrascht, daß

nach der Importliste die Meldung erscheint »No (more) errors found«. Man sollte doch erwarten, daß die oben nicht bekannten Namen auch im Anweisungsteil unbekannt bleiben. Der Grund dieser Meldung liegt jedoch darin, daß der Compiler abbricht, wenn er in der Importliste bereits feststellen muß, Sie haben so schlampig gearbeitet, daß weiteres Übersetzen sinnlos ist. Mit Recht werden Sie sich empören, ich hätte Sie aufs Glatteis geführt. Aber Sie sollen einige wichtige Fehlermeldungen ken-

ReadString übergibt die gleichen Parameter. Wir müssen aber darauf achten, daß das aufrufende Programm den eingelesenen String übergeben bekommt. Der einfachste Weg ist das »Binden« der Variablen mit dem Schlüsselwort VAR. Dadurch wird der in der Prozedur veränderte Wert auch im Hauptprogramm verändert. Compare soll zwei Strings vergleichen - wenn ein Unterschied festgestellt wird, soll der Wert »FALSE« zurückgegeben werden, im andern Fall »TRUE«. WriteLn und CloseFenster be-

```
DEFINITION MODULE StringTool;

TYPE String = ARRAY [0..79] OF CHAR;

VAR ExitLoop, ChangeDir: String;

PROCEDURE Compare (str1, str2: ARRAY OF CHAR): BOOLEAN;
PROCEDURE Length (str: ARRAY OF CHAR): INTEGER;
PROCEDURE Insert (str1: ARRAY OF CHAR; bei: INTEGER; VAR in: ARRAY OF CHAR);
PROCEDURE Copy (von: ARRAY OF CHAR; VAR nach: ARRAY OF CHAR; Start, Length: INTEGER);

END StringTool.
```

**Listing 2. »StringTool.def« legt die Definitionen für die Stringmanipulationen fest**

nenlernen, die Ihnen in Ihrer weiteren Programmierkarriere sicher nicht erspart bleiben.

Doch was müssen wir tun, damit der Compiler unser Programm weiter bearbeitet? Zwei komplette Bibliotheken schreiben? Falsch. Es reicht voll und ganz, die Schnittstellen festzulegen. Alles andere »vertagen« wir auf die Zeit zwischen Compiler- und Linkvorgang.

Lassen Sie uns zunächst das Hauptprogramm analysieren, um festzustellen, was wir in den Schnittstellen festlegen müssen. Neben der veränderten Importliste gibt es eine neue Konstante vom Typ String - »ExitLoop«. Diese wird als Stringkonstante definiert. Sie soll den Namen festlegen, den der Benutzer eingibt, um das Programm zu beenden. Eine Variable »ex« ist vom Typ BOOLEAN und soll das Ergebnis einer Funktion aufnehmen. Im Anweisungsteil stehen die entscheidenden Informationen für unsere Schnittstelle: Wir benötigen eine Funktion »OpenFenster«. Sie erhält keinen Übergabeparameter, gibt aber einen Wert »Fenster« zurück, den es ebenfalls in DosTool zu definieren gilt. WriteString hat als ersten Parameter das Fenster, in das geschrieben wird, und als zweiten die Zeichenkette.

nötigen als Parameter die Adresse eines Fensters, in dem ein Linefeed geschrieben beziehungsweise das geschlossen werden soll.

Das Aussehen einer Schnittstelle ist auf den ersten Blick relativ unkompliziert. Dennoch sollten Sie gerade hier sehr viel Sorgfalt aufwenden, das erspart später eine Menge unnötiger Arbeit. Für »DosTool« legen wir eine Schnittstelle fest, die Sie in Listing 1 finden.

Speichern Sie dieses Definitionsmodul unter dem Namen »DosTool.def«. Die Endung ».def« weist den Compiler darauf hin, daß dieses File nicht in ein Objekt, sondern in ein Symbolfile übersetzt werden soll. Die Funktionen kennen Sie bereits mit Ausnahme von »CD«. Begründen Sie sich vorerst damit, daß wir diese für eine spätere Version von MyCLI benötigen.

Das Definitionsmodul »StringTool« enthält ebenfalls mehr Funktionen als im Moment gebraucht werden (Listing 2).

Nachdem Sie »StringTool.def« gespeichert und kompiliert haben, stehen die benötigten Schnittstellen fest. Sicher sind Ihnen die zusätzlichen Prozeduren in »StringTool« aufgefallen. Diese werden im Moment nicht benötigt. Wenn wir sie dennoch hier festlegen, dann

```
DEFINITION MODULE DosTool;

FROM Dos IMPORT FileHandlePtr;

VAR Fenster: FileHandlePtr;

PROCEDURE OpenFenster(): FileHandlePtr;
PROCEDURE CloseFenster (f: FileHandlePtr);
PROCEDURE WriteString (fh: FileHandlePtr; str: ARRAY OF CHAR);
PROCEDURE ReadString (fh: FileHandlePtr; VAR str: ARRAY OF CHAR);
PROCEDURE WriteLn (fh: FileHandlePtr);
PROCEDURE CD (name: ARRAY OF CHAR): BOOLEAN;

END DosTool.
```

**Listing 1. Mit dieser Schnittstelle greifen Sie auf Ihre erste Bibliothek zu**



nur, weil wir damit das erneute Compilieren des gesamten Projekts vermeiden. Das wird erst später wirklich interessant. Begnügen Sie sich vorerst damit, daß wir mit den zusätzlichen Zeilen »in die Zukunft investiert« haben.

Wenn Sie jetzt MyCLI compilieren, werden Sie feststellen, daß der Compiler keine Bedenken mehr gegen Ihr Programm hat. Es ist syntaktisch völlig in Ordnung. Alle Namen wurden korrekt festgelegt oder importiert. Sollte es wirklich so einfach sein, sich eine eigene Bibliothek zu schreiben? Ich kann Sie beruhigen. Spätestens der Linker würde den Schwindel aufdecken. Aber machen Sie sich klar, was bisher geschehen ist:

Wir haben angefangen mit dem »Design« des Programms und haben uns noch kaum Gedanken darüber gemacht, was der Computer dabei genau machen soll. Es wurden lediglich einige »Vorstellungen« festgehalten (und nach Modula übersetzt), was in etwa passieren soll. Diese lauteten frei übersetzt:

1. Öffne ein Fenster.
2. Zeige dem Benutzer mit dem Prompt, daß etwas von ihm verlangt wird.
3. Lies seinen Wunsch ein.
4. Prüfe, ob er aufhören will. Wenn ja, weiter bei 8.
5. Führe seinen Wunsch aus.
6. Füge eine Leerzeile ein.
7. Mach weiter bei 2.
8. Schließe das Fenster und beende das Programm.

Im nächsten Schritt wird festgelegt, welche Aufgaben wir an einen »Spezialisten« vergeben wollen und wie dieser heißt. Zu diesem Zweck wird im Definitionsmodul genau festgelegt, welche Aufgaben dieser Spezialist erfüllen soll. An diesem

## Bibliotheken ermöglichen Arbeitsteilung

Punkt könnten Sie sich zurücklehnen, entspannen und alles weitere einem erfahrenen Programmierer anvertrauen. Denn mit der Festlegung der Definitionsmodule weiß dieser genau, was das verlangte Unterprogramm leisten muß. Sie kennen keinen erfahrenen Programmierer? Gut so, denn dabei hätten Sie ja nichts gelernt.

Also gehen wir nochmals an die Arbeit und schreiben unsere beiden Bibliotheken. Bei einer davon können wir auf Bekanntes zurückgreifen. Laden Sie die alte Version von »MyCLI.

```
IMPLEMENTATION MODULE DosTool; (* neu *)

FROM Dos IMPORT Open, Close, FileHandlePtr,
  newFile, Read, Write; (* neu *)
FROM SYSTEM IMPORT ADR;

PROCEDURE OpenFenster(): FileHandlePtr;
BEGIN
  RETURN Open (ADR("CON:0/0/640/256/MyCli"), newFile)
END OpenFenster;

PROCEDURE CloseFenster (f: FileHandlePtr);
BEGIN
  Close (f)
END CloseFenster;

PROCEDURE WriteString (fh: FileHandlePtr; str: ARRAY OF CHAR);
VAR ok: INTEGER;
BEGIN
  ok := Write (fh, ADR(str), SIZE(str))
END WriteString;

PROCEDURE ReadString (fh: FileHandlePtr; VAR str:ARRAY OF CHAR);
VAR c: CHAR;
  i, ok: INTEGER;
BEGIN
  IF HIGH (str) >= 1 THEN
    i := 0;
    REPEAT
      ok := Read (fh, ADR(c), SIZE(c));
      str[i] := c;
      INC(i)
    UNTIL (i >= HIGH(str)) OR (c = 12C);
    str[i-1] := 0C
  END;
END ReadString;

PROCEDURE WriteLn (fh: FileHandlePtr);
VAR LF: CHAR;
  ok: INTEGER;
BEGIN
  LF := 12C;
  ok := Write (fh, ADR (LF), SIZE(LF));
END WriteLn;

PROCEDURE CD (name: ARRAY OF CHAR): BOOLEAN;
END CD;

END DosTool. (* neu *)
```

### Listing 3. Mit »DosTool.mod« implementieren Sie die erste eigene Bibliothek

mod«. Löschen Sie alle Zeilen, die außerhalb der fünf Prozeduren liegen. Danach fügen Sie einen neuen Kopf und einen neuen Schluß an, so daß das Modul ebenso aussieht wie Listing 3 (die neuen Zeilen sind durch einen Kommentar gekennzeichnet):

Kümmern Sie sich vorerst nicht um die Prozedur CD. Sie wird später ausgebaut. Das Programm speichern Sie unter dem Namen »DosTool.mod« und compilieren es. Fertig ist die erste Bibliothek. Das war einfacher als Sie dachten, oder? Das einzig Neue ist die erste Zeile. Diese erklärt dem Compiler, daß es sich um ein Bibliotheksmodul handelt, zu dem bereits eine Schnittstelle in Form eines Symbolfiles existieren muß. Aufgrund dieses Symbolfiles wird überprüft, ob der Programmierer sich an die

gestellte Aufgabe gehalten hat oder nicht.

Bevor Sie weiterlesen, sollten Sie diese Form der Überprüfung intensiv testen. Streichen Sie einen beliebigen Parameter und compilieren Sie neu. Welche Fehlermeldung erhalten Sie? Was passiert, wenn Sie für den Typ des Übergabeparameters von WriteLn INTEGER statt FileHandlePtr einsetzen? Was geschieht, wenn Sie eine Prozedur ganz streichen? Welchen Fehler erhalten Sie für das Entfernen der Klammern und des Rückgabetyps bei OpenFenster?

Testen Sie auch, was geschieht, wenn Sie eine zusätzliche Prozedur schreiben (Sie könnten beispielsweise »WriteLn« kopieren und mit einem neuen Namen versehen). Der Compiler beklagt sich nicht? Kein Wunder – gegen Fleißar-

beit hat er nichts einzuwenden. Sie können jederzeit mehr produzieren als verlangt, aber nie weniger. Dies schafft die Möglichkeit, sogenannte »private« Prozeduren zu deklarieren, die nur innerhalb des Implementationsmoduls verwendet werden. Das Hauptprogramm hat darauf keinen Zugriff.

## Implementieren Stück für Stück

Aber bevor Sie MyCLI linken, benötigen Sie eine zweite Bibliothek. Von dieser verwenden wir vorerst zwei Namen: »String« und »Compare«. Alle anderen sind für dieses Programm uninteressant, daher werden wir sie vorerst etwas stiefmütterlich behandeln. String wurde im Definitionsmodul bereits hinreichend deklariert, folglich müssen wir uns nur noch die Routine für Compare schreiben. Diese soll zwei Strings auf Gleichheit testen und einen entsprechenden Wert an das Hauptprogramm zurückgeben. Daneben müssen wir jedoch beachten, daß die anderen Prozeduren zumindest soweit enthalten sind, daß der Compiler zufriedengestellt ist. Die Lösung finden Sie in Listing 4.

Alle Prozeduren, die im Moment nicht benötigt werden, setzen wir auf die »Warteliste«. Sie werden nur mit dem nötigsten Zubehör ausgestattet. Sollte jemand auf die Idee kommen, sie aufzurufen, geschieht einfach nichts. Sie bestehen aus einem leeren Deklarations- und einem leeren Anweisungsteil. Die einzige Prozedur, die mit Anweisungen versehen wird, ist Compare. Sie übernimmt etwa die Aufgabe der Zeile »IF CAP(Kommando[0]) = "Q") THEN...«. Der Unterschied ist, daß Compare einen ganzen String auf Gleichheit »abtastet«. Am Anfang wird die Variable »gleich« auf wahr gesetzt. Das heißt, wir gehen davon aus, daß beide Strings gleich sind, und versuchen dies in der Schleife zu widerlegen. »gleich« wird genau dann falsch, wenn die beiden Zeichenketten an einer beliebigen Stelle ein unterschiedliches Zeichen aufweisen, bevor das Ende des kürzeren von beiden erreicht wird.

Compare unterscheidet zwischen Groß- und Kleinschreibung. Das hat zur Folge, daß »Beende« etwas anderes ist als »beende« und letzteres nicht zum Beenden des Programms



führt. Wollen Sie nicht zwischen Groß- und Kleinbuchstaben unterscheiden, so müssen Sie die CAP-Funktion für beide Strings einfügen. Haben Sie gesehen, wo? Versuchen Sie es!

Sicher haben Sie sich über die RETURN-Anweisung an zwei verschiedenen Stellen gewundert. Wozu soll der Wert zweimal an das aufrufende Programm zurückgegeben werden? Keine Angst, das wird er nicht. RETURN bewirkt gleichzeitig einen Abbruch der Prozedur. Sobald der erste Unterschied gefunden ist, muß nicht nach weiteren gesucht werden.

Wenn Sie sich unser Programm jetzt ansehen, werden Sie feststellen, daß es einen fast identischen Nachbau des altbekannten CLI darstellt. Sicherlich stimmen Sie mir zu, daß das CLI nicht gerade ein erstrebenswertes Vorbild für Programmierer darstellt. Die Entwickler des Amiga haben damit alles andere als die ideale Benutzerschnittstelle geschaffen. Aber lassen Sie sich von den bisherigen Ergebnissen nicht entmutigen! Lassen wir Commodore ruhig weiterhin tapfer am CLI festhalten, wir streben nach höherem.

Wir wollen aus unserem altbackenen CLI eine Entwicklungsumgebung für den Modula-Compiler formen. Sie soll die ständig benötigten Befehle (wie Editor-, Compiler- und Linker-Aufruf sowie einige CLI-Kommandos) per Tastendruck aufrufbar machen. Daneben wäre es doch ganz nett, wenn Sie Ihre Arbeitsdatei in einem Menü einstellen könnten, anstatt sie immer wieder einzutippen. Die ständige Anzeige des aktuellen Directory kann ebenfalls nicht schaden.

Natürlich kann man das aktuelle Directory irgendwo ins Fenster schreiben. Aber eine elegante Lösung wäre doch die Titelzeile. In ihr stehen normalerweise so enorm wichtige Informationen wie »MyCLI« oder ähnliches. Mit der Anzeige des aktuellen Pfads ist die Titelzeile weit besser genutzt.

Das Problem ist, wie wir diese Zeile manipulieren können. Die einfachste Lösung wäre wohl, das Fenster zu schließen und mit neuem Namen zu öffnen. Aber abgesehen von der Zeit, die wir dabei verschwenden, ist es auch nicht nötig.

Allerdings taucht an dieser Stelle durchaus ein kleines Problem auf: Bisher haben wir stets nur mit einem von AmigaDOS verwalteten Fenster gearbeitet. Das soll auch in Zukunft

so bleiben. Die Änderung der Fenstertitel ist jedoch eine Aufgabe für einen anderen Betriebssystemteil: für Intuition. Intuition hat aber etwas andere Konventionen als AmigaDOS. So werden zur Spezifikation von Fenstern sogenannte »WindowPtr« verlangt, während AmigaDOS – wie Sie wissen – mit einem FileHandle arbeitet. Nun könnte man auf die Idee kommen, wenn es sich doch um dieselbe Sache (unser Fenster) handelt, einfach die eine Adresse in die andere Variable zu übernehmen. Sie brauchen

Dieser kleine Exkurs sollte Ihnen nur vorführen, daß auch Ihr Modula-Compiler nicht alle Fehler finden kann. Wenn wir das genannte Problem »sauber« lösen wollen, brauchen wir einen etwas umfangreicheren Ansatz.

Daß Ihr mit »Open« geöffnetes Fenster von AmigaDOS verwaltet wird, wissen Sie bereits. Auch daß viele Operationen, die im Zusammenhang mit Fenstern stehen, von Intuition erledigt werden, ist nichts Unbekanntes. Aber wenige wissen, daß Intuition auch AmigaDOS-

uns zunutze, um an den Zeiger für unser Fenster zu kommen. Denn nach dem Öffnen ist das Fenster automatisch aktiv.

Sie müssen nur wissen, wo Sie diesen Zeiger auslesen können. Es existiert eine Struktur »IntuitionBase«. In dieser sind alle Daten enthalten, die den Zustand der grafischen Benutzeroberfläche beschreiben. Dazu gehört auch die Information, welches Fenster gerade aktiv ist. IntuitionBase wird für unser Programm erst zugänglich, wenn wir ausdrücklich sagen, daß wir damit arbeiten wollen. Das erledigt die Funktion »OpenIntuition«. Diese öffnet die Intuition-Library und liefert einen Zeiger auf IntuitionBase.

Aber bevor wir uns in die Feinheiten der Intuition-Programmierung stürzen, wollen wir unser Hauptprogramm etwas verfeinern. Dies geschieht wohlgemerkt, ohne die bisherigen Schnittstellen zu verändern. Bibliotheken, die Sie einmal programmiert haben, können Sie problemlos weiterverwenden, wenn Sie sich an dieses Konzept halten. Das gleiche gilt, falls Ihr Freund, der Programmierer, auf die Idee kommt, die verwendeten Bibliotheken zu verbessern (womöglich nachdem das Programm längst fertig ist). Deshalb wird Ihr Programm nicht verändert und auch nicht neu kompiliert, Sie linken einfach die neuen Bibliotheken hinzu (von denen Sie nicht einmal den Source-Code kennen müssen) und fertig ist die Verbesserung.

Aber zurück zu unserem Projekt. Wenn Sie das alte Programm weiterverwenden wollen (sicher haben Sie noch eine Menge Ideen für Experimente), sollten Sie dem neuen einen anderen Namen geben. Laden Sie MyCLI.mod in den Editor und speichern Sie es unter dem Namen »MyShell.mod«. Anschließend verbessern Sie das Programm, bis es ebenso aussieht wie Listing 5.

Zugegeben – das war viel auf einmal. Aber lassen Sie uns dieses Programm gemeinsam analysieren, es sieht schwieriger aus als es ist (das ist eine Situation, der man beim Programmieren immer wieder begegnet).

Die Importliste dürfte Ihnen inzwischen keine Schwierigkeiten mehr bereiten. Sie wissen, daß wir für Bibliotheken, die Sie noch nicht kennen (in diesem Fall nur »IntuiTool«), erst noch eine Schnittstelle entwickeln müssen. Die anderen Bibliotheken sind entweder mitgeliefert

```
IMPLEMENTATION MODULE StringTool;

PROCEDURE Compare (str1, str2: ARRAY OF CHAR): BOOLEAN;
VAR i: INTEGER;
BEGIN
  i := 0;
  REPEAT
    IF (CAP(str1[i]) # CAP(str2[i])) THEN
      RETURN FALSE
    END;
    INC(i)
  UNTIL ((str1[i]=0C) OR (str2[i]=0C));
  RETURN TRUE
END Compare;

PROCEDURE Length (str: ARRAY OF CHAR): INTEGER;
END Length;

PROCEDURE Insert (str1: ARRAY OF CHAR; bei: INTEGER; VAR in: ARRAY OF CHAR);
END Insert;

PROCEDURE Copy (von: ARRAY OF CHAR; VAR nach: ARRAY OF CHAR; Start, Length: INTEGER);
END Copy;

END StringTool.
```

**Listing 4. In »StringTool.mod« finden Sie das Gerüst einer Bibliothek. Sie müssen nicht alle definierten Prozeduren ausformulieren, um Ihr Programm lauffähig zu machen.**

dazu nur eine Typumwandlung vorzunehmen. Dazu fügen Sie hinter »OpenFenster« eine neue Zeile ein:

```
iFenster := CAST
(WindowPtr, Fenster);
```

und definieren Sie im Deklarationsteil eine entsprechende Variable. Der Compiler hat gegen diese gewagte Manipulation nichts einzuwenden. Das heißt aber nicht, daß das Programm damit auch läuft. Bringen Sie alle nicht gespeicherten Daten in Sicherheit und starten Sie das Programm. Ihr Amiga bedankt sich mit einem Guru. Wenn Sie die Nummer genauer untersuchen, finden Sie eine »0000 0008«. Dies bedeutet Privilegverletzung. Anscheinend soll diese Art der unsauberen Programmierung das Privileg anderer Leute bleiben.

Fenster verwaltet. Auch bei DOS-Fenstern werden Arbeiten wie Verwaltung der Fenstertitel, Abfrage der Gadgets und Ähnliches von Intuition übernommen. Die einzige Schwierigkeit,

## Von DOS zu Intuition und zurück

wenn Sie selbst auf Intuition Einfluß nehmen wollen, ist der fehlende WindowPtr. Dabei handelt es sich um einen Zeiger auf die Struktur eines geöffneten Fensters. Es spielt keine Rolle, wer das Fenster angelegt hat. Intuition verwaltet in jedem Fall eine Liste aller Fenster. Es gibt einen Zeiger, der stets auf das aktive Fenster gerichtet ist. Diesen Umstand machen wir



oder bereits auf den vorangehenden Seiten besprochen.

Die Konstante ist Ihnen ebenfalls bekannt, es handelt sich um das Prompt von »MyCLI«. Die Variablen sind etwas umfangreicher geworden, wirklich neue sind aber nicht dabei – es tauchen nur einige Typen mehrfach auf. So haben wir inzwischen drei Strings und drei BOOLEAN-Variable, die jeweils

verschiedene Aufgaben erfüllen.

Im Anweisungsteil dagegen hat sich einiges getan. Nach dem Öffnen des Fensters wird ein String initialisiert. Dieser wird später in den Fenstertitel kopiert. Bitte verwechseln Sie diese Zuweisung nicht mit dem Verändern des Fenstertitels. Die Zeile hinter OpenFenster hat keinerlei Auswirkung auf

den tatsächlichen Fenstertitel. Mit der nächsten Zeile holen wir uns einen zweiten Zeiger auf unser Fenster. Dieser hat den Vorteil, daß er für alle Intuition-Funktionen verwendbar ist. Das eigentliche Auslesen überlassen wir einer Prozedur, die aus IntuiTool importiert wird. Wie diese arbeitet, können wir später entscheiden, vorerst genügt es zu wissen, daß wir irgendwie an diesen Zeiger kommen und daß wir ihn von der noch zu definierenden Funktion »ActiveWindow« geliefert bekommen. Das gleiche gilt für »InstallMenu«. Diese Funktion soll – wenn sie eingebaut ist – ein Menü an der Titelleiste des Screens befestigen.

## Die ganze Leistung von IF

Die folgende Schleife kennen Sie bereits, sie hat sich allerdings seit dem letzten Mal etwas verändert. Nach wie vor wird zuerst das Prompt ausgegeben und dann ein Kommando eingelesen. Dies geschieht weiterhin mit den bewährten Prozeduren. Allerdings geben wir uns jetzt nicht mehr damit zufrieden, dieses Kommando kritiklos ausführen zu lassen. Sie kennen bereits die Abfrage, ob es sich um den Wunsch des Benutzers handelt, Ihr Programm zu verlassen. Wenn wir stets den aktuell eingestellten Pfad anzeigen lassen wollen, müssen wir nicht nur die Arbeit von Endcli, sondern auch die von CD (»Change Directory«) selbst in die Hand nehmen. Alle anderen Wünsche übergeben wir weiterhin an Execute. Aber wie stellen wir fest, ob es sich um eines der beiden Kommandos handelt?

Die Grundlage hierfür kennen Sie bereits von MyCLI: Wir vergleichen mit Hilfe von Compare das eingegebene Kommando mit einer Vorgabe. Zunächst vergleichen wir das eingegebene Kommando mit dem importierten String »ChangeDir«. ChangeDir besteht aus drei Zeichen: »c«, »d« und <SPACE>, oder in lesbarer Schreibweise »cd«. Das angehängte Leerzeichen sorgt dafür, daß nur auf Gleichheit erkannt wird, wenn ein Argument übergeben wurde. Wird nur »cd« eingegeben, ist die Bedingung nicht erfüllt.

Sie sehen im folgenden Anweisungsblock zum ersten Mal eine relativ komplizierte Konstruktion aus IF-Verzweigungen. Dies ist eine willkommene Gelegenheit, diesen wichtigen

Bereich näher unter die Lupe zu nehmen:

Die einfachste Form lautet

```
IF Bedingung THEN
Anweisung END;
```

Dies führt dazu, daß »Anweisung« genau dann ausgeführt wird, wenn »Bedingung« erfüllt ist. Anders ausgedrückt: IF verhindert, daß die Anweisung ausgeführt wird, wenn die Bedingung nicht erfüllt ist. Dabei muß Bedingung eine Variable, Konstante oder ein Ausdruck vom Typ BOOLEAN sein. Eine Konstante ist wenig sinnvoll, denn dadurch wäre das Ergebnis vorher festgelegt und Sie könnten die Bedingung auch streichen. Eine Variable ist beispielsweise »Done« oder »gefunden«. Ein Ausdruck kann das Ergebnis einer Funktion sein, die einen BOOLEAN-Wert liefert oder auch das Ergebnis einer logischen Operation (beispielsweise »i = 0«).

Diese Konstruktion ist leicht nachzuvollziehen. Etwas komplizierter wird es, wenn wir die erweiterten Möglichkeiten von IF nutzen. Die einfachste Erweiterung ist eine »Vermehrung« der Anweisungen. Diese müssen dann durch ein Semikolon getrennt sein. Dieser sogenannte »Anweisungsblock« kann seinerseits wieder mehrere IF-Anweisungen enthalten. Jede dieser IF-Anweisungen gilt als beendet, wenn das zugehörige END erreicht wird. Dabei sind die von Ihnen gemachten Einrückungen im Text völlig unwesentlich. Der Compiler zählt mit, wie oft er ein IF »gesehen« hat und streicht jeweils eines davon bei jedem END. Übrigens besteht hier kein Unterschied zwischen IF und anderen Formen der Verzweigung (beispielsweise »FOR« oder »WHILE«). Jede wird durch END abgeschlossen. Wenn der Compiler drei FOR und vier IF gezählt hat, braucht er siebenmal die Anweisung END, bevor er alle Anweisungsblöcke als beendet betrachtet. Sie können dies leicht testen, indem Sie in MyCLI.mod ein END entfernen. Sie werden feststellen, daß der Compiler (der Ihren Source-Code ja von vorne nach hinten durchgeht) das END des Modulblocks als END für den letzten Anweisungsblock betrachtet, und dann erst feststellen kann, daß etwas fehlt.

Aber auch die Bedingungen können aus mehreren einzelnen bestehen. Diese müssen durch logische Operatoren getrennt sein. In Tabelle 3 finden Sie eine Liste aller zulässigen logischen Operatoren. Tabelle

```
MODULE MyShell;

FROM SYSTEM IMPORT ADR;
FROM Dos IMPORT Execute;
FROM DosTool IMPORT Fenster, OpenFenster, CloseFenster,
  ReadString, WriteString, WriteLn, CD;
FROM StringTool IMPORT String, ExitLoop, ChangeDir, Compare,
  Copy, Length, Insert;
FROM IntuiTool IMPORT iFenster, ActiveWindow, UpdateTitle, VerarbeiteMenu,
  InstallMenu, RemoveMenu, MenuStrip, nachricht, Class, pick;
FROM Exec IMPORT GetMsg, ReplyMsg;

CONST Prompt = " ";

VAR ok, i: INTEGER;
    Done, mitDP, gefunden: BOOLEAN;
    wTitel, Kommando, newDir: String;

BEGIN
  Fenster := OpenFenster ();
  wTitel := "DFO:";
  iFenster := ActiveWindow ();
  InstallMenu (iFenster, MenuStrip);
  LOOP
    WriteString (Fenster, Prompt);
    ReadString (Fenster, Kommando);
    IF (Compare (Kommando, ChangeDir)) THEN
      Copy (Kommando, newDir, 3, Length (Kommando));
      Done := CD (newDir);
      IF Done THEN
        mitDP := FALSE;
        FOR i := 0 TO Length (newDir) DO
          IF (newDir[i] = ":") THEN mitDP := TRUE END
        END;
        IF mitDP THEN wTitel[0] := OC
        ELSIF newDir[0] = "/" THEN
          i := Length (wTitel); gefunden := FALSE;
          REPEAT
            DEC (i);
            IF wTitel[i] = "/" THEN
              wTitel[i] := OC; gefunden := TRUE
            ELSIF wTitel[i] = ":" THEN
              wTitel[i+1] := OC; gefunden := TRUE
            END;
          UNTIL (gefunden OR (i = 0));
          Copy (newDir, newDir, 1, Length(newDir))
        ELSIF (Length (wTitel) > 4) THEN
          wTitel[Length(wTitel)+1] := OC;
          wTitel[Length(wTitel)] := "/"
        END;
        Insert (newDir, Length (wTitel), wTitel);
        UpdateTitle (iFenster, wTitel)
      END
    ELSIF (Compare (Kommando, ExitLoop)) THEN EXIT
    ELSE
      ok := Execute (ADR(Kommando),0,Fenster);
    END;
    WriteLn (Fenster)
  END;
  RemoveMenu (iFenster);
  CloseFenster (Fenster)
END MyShell.
```

**Listing 5. »MyShell.mod« in der fast endgültigen Version – das aktuelle Verzeichnis wird im Fenstertitel angezeigt**



Operator	Ergebnis
AND	logisches Und zweier Operanden
OR	logisches Oder zweier Operanden
NOT	logische Verneinung des nachfolgenden Operanden (alternativ: »#«)
=	liefert genau dann TRUE, wenn die beiden Operanden gleich sind
>	liefert genau dann TRUE, wenn der erste Operand größer ist als der zweite
<	liefert genau dann TRUE, wenn der erste Operand kleiner ist als der zweite
> =	liefert TRUE, wenn der erste Operand größer oder gleich dem zweiten ist
< =	liefert TRUE, wenn der erste Operand kleiner oder gleich dem zweiten ist

**Tabelle 3. Diese Operatoren wenden Sie auf Boolesche Variablen an**

4 gibt einige Beispiele zur Verdeutlichung.

Aber wie Sie aus dem Listing erkennen können, sind wir damit noch nicht am Ende. Sie können für Ihre Bedingung auch Alternativen zulassen:

```
IF Bedingung THEN
  Anweisung1
ELSE
  Anweisung2
END
```

Bitte beachten Sie, daß die Schreibweise keinen Einfluß auf den Compiler hat. Alle Befehle könnten ebenso gut in einer Zeile stehen. In oben dargestellter Form erhöhen Sie jedoch die Lesbarkeit für Sie selbst (oder für die Leser unserer Zeitschrift, falls Ihr Programm gedruckt wird).

Eine IF-Anweisung in dieser Form hat zur Folge, daß »Anweisung1« auch weiterhin nur ausgeführt wird, wenn die Bedingung wahr ist. »Anweisung2« dagegen wird ausgeführt, wenn die Bedingung falsch ist. Die typische Anwendung für diese

## »Multiple Choice« mit ELSE

Form sind zwei einander ausschließende Aktionen. »ELSE« hat hier die gleiche Bedeutung wie »andernfalls« in der Umgangssprache (oder eben »else« im englischen).

Zudem besteht noch die Möglichkeit, daß mehrere Bedingungen geprüft werden sollen, aber nur eine als erfüllt gelten kann. Damit ist nicht ausgeschlossen, daß weitere Bedingungen ebenfalls erfüllt sind. Lediglich die Prüfung soll abgeschlossen sein, wenn eine erfüllte Bedingung gefunden wird. Ist keine der angegebenen wahr, wird – falls er vorhanden ist – der ELSE-Zweig ausgeführt, fehlt dieser, geschieht nichts. Die Lösung in Modula

heißt »ELSIF«. Beachten Sie vor allem die von den meisten anderen Programmiersprachen abweichende Schreibweise. Die allgemeine Form der IF-Anweisung lautet also

```
IF Bedingung1 THEN
  Anweisung1
ELSIF Bedingung2
  Anweisung2
ELSIF ...
...
ELSE Anweisung3
END;
```

Dabei können beliebig viele (oder keine) ELSIF-Anweisungen enthalten sein und eine oder keine ELSE-Anweisung ist erlaubt. Testen Sie ruhig, was passiert, wenn Sie mehrere ELSE-Anweisungen nacheinander setzen. Die wichtigste Anweisung allerdings ist END. Wenn Sie diese vergessen, kommt es zu schwer nachvollziehbaren Fehlern, weil der Compiler einfach die nächste Anweisung übernimmt. Auch hier sollten Sie ruhig testen, wo der Fehler auftritt, wenn Sie die END-Anweisung bei einem IF streichen.

Die Schreibweise einer IF-Anweisung bleibt Ihnen überlassen. Wir halten uns in diesem Workshop aber daran, daß alle IF-Anweisungen eine einheitliche Schreibweise erhalten, wenn ihre Länge eine Zeile übersteigt. In diesem Fall finden Sie alle zu einem IF gehörigen Schlüsselwörter (IF, ELSIF, ELSE und END) in der gleichen Spalte untereinander. Der Bedingungsteil steht direkt hinter dem entsprechenden Schlüsselwort, der Anweisungsteil ist um mindestens zwei Spalten eingerückt.

Mit diesen Informationen ist es nicht mehr besonders schwierig, den neu hinzugekommenen Anweisungsblock nachzuvollziehen. Bei Blöcken von dieser Länge ist es wenig sinnvoll, sie von oben nach unten zu durchsuchen. Statt des-

sen sollten Sie sich in diesem Fall erst die zusammengehörenden Anweisungsblöcke markieren. Bitte führen Sie die folgenden Schritte genau so durch wie beschrieben, Sie werden feststellen, daß dieses Vorgehen das Verständnis ungemein fördert.

Markieren Sie mit einem roten Stift das erste IF. Es steht unter »ReadString«. Suchen Sie dann alle Schlüsselwörter, die zu diesem IF gehören und markieren Sie sie in der gleichen Farbe. Fertig? Sie müßten jetzt vier Schlüsselwörter markiert haben: Außer dem ersten IF ein ELSIF in der Zeile »ELSIF (Compare (Kommando, Exit-Loop)) THEN EXIT«, das ELSE in der nächsten Zeile und das END nochmals zwei Zeilen tiefer.

Fällt Ihnen auf, wie allein dadurch die Programmstruktur sehr viel deutlicher hervortritt? Der ganze Block zwischen IF und ELSIF wird nur ausgeführt, wenn die Bedingung hinter IF erfüllt ist. Solange wir davon ausgehen, daß dies nicht der Fall ist, haben wir die gleiche Situation wie in MyCLI. Das Neue an unserem Programm steht ausschließlich in diesem Block.

Gehen Sie noch einen Schritt weiter und markieren Sie im Listing 5 mit einem blauen Stift den nächsten Block. Dieser beginnt bei dem IF unter »Done« und hat nur noch eine END-Anweisung in der Zeile vor dem rot markierten ELSIF. Bitte machen Sie sich klar, daß der blaue Block vor der nächsten roten Markierung beendet sein muß. Es können keine Überschneidungen existieren.

Markieren Sie die »FOR«-Schleife (wofür diese verwendet wird, erfahren Sie weiter unten) mit gelb. Sie ist nur drei Zeilen breit. Direkt unter dem END dieser Schleife finden Sie das

nächste IF. Markieren Sie es mit einem grünen Stift. Dazu gehören zwei ELSIF-Anweisungen und ein END. Haben Sie diese gefunden? Dann markieren Sie mit orange die REPEAT-Schleife. Sie wird nicht mit END, sondern mit UNTIL abgeschlossen. In dieser ist die letzte IF-Anweisung enthalten. Markieren Sie diese mit violett und betrachten Sie jetzt die Programmstruktur. Sie dürfte um einiges klarer hervortreten als wenn Sie versuchen, das Programm von oben nach unten zu lesen.

## Markieren schafft Übersicht

Sie sollten dieses Verfahren immer dann anwenden, wenn Ihnen der Ablauf eines Programms nicht auf den ersten Blick klar ist. Es macht vieles sofort anschaulich, was im andern Fall in einer Ansammlung von Buchstaben untergeht.

Die rote IF-»Klammer« prüft zunächst, ob das eingegebene Kommando ein »cd« mit einem weiteren Argument ist. Wenn nicht, wird geprüft, ob es sich um ein Kommando handelt, das die Schleife beendet. Ist auch diese Bedingung nicht erfüllt, so kann es nur noch ein Kommando sein, das der Execute-Funktion übergeben wird. Der Rest des Programms ist bekannt außer »RemoveMenu«. Diese Funktion entfernt den von »InstallMenu« aufgebauten Menü-Strip.

Ist die erste Bedingung jedoch erfüllt, so wird das mit cd übergebene Argument in eine neue Variable (»newDir«) kopiert. Diese Variable übergeben wir der Funktion »CD«. CD sucht nach dem angegebenen Directory. Ist es vorhanden, so wird es zum aktuellen gemacht und TRUE zurückgegeben.

### Beispiele für Boolesche Logik

In folgender Tabelle finden Sie einige Beispiele, die Ihnen die Arbeitsweise von Booleschen Operatoren verdeutlichen. In den linken Spalten finden Sie den Inhalt der beiden Variablen a und b, die nächste Spalte enthält die Gleichung und ganz rechts finden Sie das Ergebnis.

a	b	Operation	Ergebnis
TRUE	TRUE	a AND b	TRUE
TRUE	FALSE	a AND b	FALSE
FALSE	TRUE	a AND b	FALSE
FALSE	FALSE	a AND b	FALSE
TRUE	TRUE	a OR b	TRUE
TRUE	FALSE	a OR b	TRUE
FALSE	FALSE	a OR b	FALSE
TRUE	TRUE	a AND (NOT b)	FALSE
TRUE	TRUE	(NOT a) OR b	TRUE

**Tabelle 4. Die Arbeitsweise der Booleschen Operatoren verdeutlichen Sie sich am besten mit diesen Beispielen**



```

DEFINITION MODULE IntuiTool;

FROM Intuition IMPORT WindowPtr, ScreenPtr, IDCMPFlagSet,
WindowFlagSet, WindowFlags, IDCMPFlags, IntuiMessagePtr, MenuPtr;
FROM SYSTEM IMPORT ADDRESS;

TYPE Class = IDCMPFlagSet;

CONST idcmpFlgs = IDCMPFlagSet {closeWindow};
winFlgs = WindowFlagSet {windowSizing, windowDepth, windowClose,
windowDrag, activate};
pick = IDCMPFlagSet{menuPick};

VAR iFenster: WindowPtr;
iScreen: ScreenPtr;
nachricht: IntuiMessagePtr;
MenuStrip: MenuPtr;

PROCEDURE ActiveWindow (): WindowPtr;
PROCEDURE OpenMinFenster (l,t,w,h: INTEGER): WindowPtr;
PROCEDURE OpenIFenster (l,t,w,h: INTEGER; name: ADDRESS; id:
IDCMPFlagSet; wi: WindowFlagSet): WindowPtr;
PROCEDURE CloseIFenster (iF: WindowPtr);
PROCEDURE VerarbeiteMenu (code: CARDINAL);
PROCEDURE InstallMenu (iF: WindowPtr; m: MenuPtr);
PROCEDURE RemoveMenu (iF: WindowPtr);
PROCEDURE VerarbeiteMenu (code: CARDINAL; fh: FileHandlePtr):
BOOLEAN;
PROCEDURE UpdateTitle (iF: WindowPtr; text: ARRAY OF CHAR);

END IntuiTool.def

```

**Listing 6. Das Definitionsmodul für eine vereinfachte Schnittstelle zu Intuition**

Kann dagegen newDir nicht gefunden werden, so gibt CD FALSE zurück.

Dieser Rückgabewert wird vom folgenden IF abgefragt. Nur wenn das Directory gewechselt wurde, muß der Titel des Fensters geändert werden. In diesem Fall sind allerdings eine Reihe weiterer Bedingungen zu prüfen:

Zunächst könnte es sein, daß der übergebene Name einen Doppelpunkt enthält. In diesem Fall ist das Directory auf einem anderen Device und der Name muß komplett gelöscht werden. Dann könnte der Anwender den Wunsch geäußert haben, per Schrägstrich ins nächsthöhere Directory befördert zu werden.

**Komplexe Manipulationen am Titel**

In diesem Fall muß der Pfad in der Titelleiste bis zum nächsten Schrägstrich oder bis zum Doppelpunkt gelöscht werden. Die letzte Möglichkeit, die es zu bedenken gilt, ist ein Unterverzeichnis, das nicht direkt unter dem Stammverzeichnis angesiedelt ist. In diesem Fall müssen wir einen Schrägstrich an der richtigen Stelle einfügen. In allen anderen Fällen sind keine weiteren Vorbereitungen nötig. Sind diese Bedingungen geprüft und die entsprechenden Aktionen ausgeführt, kann der

Name des neuen Directory an den alten angehängt werden (bitte beachten Sie, daß wir dazu die Insert-Funktion verwenden und als Position die letzte des zu erweiternden Strings übergeben). Danach wird eine Prozedur aufgerufen, die den neuen Titel einsetzt (»UpdateTitle«).

Sicher sind mit diesen Zeilen nicht alle Fragen ausgeräumt. Verdeutlichen Sie sich mehrmals die Bedingung, unter der ein bestimmter Zweig angesprungen wird. Machen Sie sich immer wieder klar, welche Bedeutung die einzelnen Anweisungen haben. Spielen Sie in Gedanken immer wieder verschiedene »Bedingungskonstellationen« durch. Nur so wird Ihnen wirklich klar, was in diesem Programm geschieht.

Wenn Sie »MyShell.mod« compilieren, werden Sie feststellen, daß auch diesmal zwei Bibliotheken fehlen. Daher wollen wir deren Schnittstelle festlegen, damit Sie Ihr Programm vom Compiler überprüfen lassen können. Das Definitionsmodul zu IntuiTool finden Sie in Listing 6.

Dieses Definitionsmodul enthält wenig Neues. Neben den unmittelbar benötigten Funktionen werden einige »auf Vorrat« definiert. Es handelt sich dabei größtenteils um nützliche Standardroutinen, die Sie in anderen Programmen verwenden können. Beispielsweise soll

»OpenFenster« eine Datenstruktur anlegen, die einer Betriebssystemroutine übergeben wird. Diese öffnet damit ein Intuition-Fenster. Sicher wundern Sie sich über die zweite Funktion mit ähnlichem Namen (»OpenMinFenster«). Diese hat eine verringerte Anzahl Parameter und öffnet ein Minimal-Fenster, das nur aus einem Rahmen besteht – weder Titel noch Gadgets oder Titelleiste sind enthalten. Diese Art von Fenstern können für die Anzeige von Informationen verwendet werden. Sie brauchen dafür keine komplette »NewWindow«-Struktur zu initialisieren.

Wirklich neu dürften Ihnen die beiden Konstanten-Definitionen sein – oder zumindest deren merkwürdige Schreibweise. Es handelt sich dabei um sogenannte FlagSets. Unter einem Set versteht man in etwa den mathematischen Mengenbegriff. Es handelt sich um eine Zusammenfassung einer Gruppe von gleichartigen Begriffen, von denen jeder genau ein Bit belegt (es ist nur die Information enthalten, ob das Element in der Menge enthalten ist – bei Flags spricht man auch von »gesetzt«). In Modula sind Sets auf die Größe eines Speicherwortes beschränkt. Für den Amiga heißt das 16 Bit. Ein Set kann also 16 Elemente enthalten. Daneben existieren Longsets, die bis zu 32 Elemente verwalten können. Flags sind beim Amiga-Betriebssystem als einzelne Bits definiert, die einen Zustand beschreiben. Von der Funktion her besteht also kein Unterschied zum Set-Konzept in Modula.

Für Sets existieren die Operatoren, die auch auf mathematische Mengen angewendet werden (siehe Tabelle 5). In unserem Beispiel werden die WindowFlags verwendet, um Eigenschaften eines Fensters zu beschreiben. Ist beispielsweise das Flag »windowClose« enthalten, so installiert Intuition ein Close-Gadget in dem Fenster. Entsprechend wird bei »windowSizing« mit dem Größen-Gadget verfahren, bei »windowDrag« kann das Fenster mit der Titelleiste verschoben werden und bei »windowDepth« kann es in den Hintergrund und nach vorne geklickt werden. »activate« aktiviert das Fenster sofort nach dem Öffnen (was bei Intuition-Fenstern nicht automatisch geschieht).

Intuition verwaltet drei Gadgets in eigener Regie. Um Size-, Drag- und Depth-Gadget brauchen Sie sich nicht zu kümmern, nachdem sie installiert sind. Dagegen wird für alles andere von Intuition lediglich eine Nachricht an Ihr Programm geschickt, daß der Anwender einen Wunsch geäußert hat. Ihre Sache ist es, diesen Wunsch auszuwerten. So könnte es sein, daß der Anwender ein Menü betätigt, eine Taste gedrückt, das Close-Gadget ausgewählt, eine Diskette entfernt oder eingelegt oder die Maus verschoben hat. In jedem Fall schickt Intuition eine Nachricht und Sie können darauf reagieren. Es wäre allerdings etwas mühsam, ständig alle Nachrichten auszuwerten. Um dies zu vermeiden, gibt es einen Filter. Sie legen selbst fest, welche Nachrichten Sie in Ihrem Fen-

Operator	Ergebnis
+	bildet die Vereinigungsmenge. Alle Elemente, die in einer der beiden Mengen enthalten sind, tauchen auch im Ergebnis auf.
-	bildet die Differenz zweier Mengen. Das Ergebnis enthält alle Elemente des ersten Operanden, die nicht im zweiten enthalten sind.
*	bildet die Schnittmenge zweier Mengen. Das Ergebnis enthält alle Elemente, die in beiden Mengen enthalten sind.
/	bildet die symmetrische Differenz. Das Ergebnis ist das gleiche wie bei der Vereinigung, außer daß alle Elemente entfernt werden, die in beiden Mengen enthalten sind.
Daneben haben folgende Booleschen Operatoren eine besondere Bedeutung:	
< =	ist enthalten (ist Untermenge).
> =	enthält (ist Obermenge).
IN	ist Element von (erster Operand muß ein Element sein).

**Tabelle 5. Sets können mit ähnlichen Operatoren bearbeitet werden, wie mathematische Mengen. Beachten Sie die veränderte Bedeutung der Booleschen Operatoren.**



```
DEFINITION MODULE Menues;

FROM Intuition IMPORT WindowPtr, MenuPtr;

VAR MenuStrip: MenuPtr;

END Menues.
```

## Listing 7. »Menues« ist zuständig für den Aufbau des »MenuStrip«

ster erhalten wollen – alle anderen werden nicht angenommen. Beispielsweise brauchen Sie für ein Fenster, das nur für die Ausgabe vorgesehen ist, keine Nachrichten zulassen. In diesem Fall setzen Sie den Filter auf Null. Zuständig für diesen Filter sind die »IDCMP-Flags«. Die Konstante, die wir in unserem Definitionsmodul festgelegt haben, läßt genau eine Form von Nachrichten zu Ihrem Fenster gelangen – die Nachricht, daß das Close-Gadget angewählt wurde.

## Minimal-Implementation von »Menues«

In IntuiTool wird ein weiteres Modul importiert: »Menues«. Dieses exportiert nur einen einzigen Namen, der von IntuiTool weiterverwendet wird: Die Variable »MenuStrip«. Entsprechend kurz ist das Definitionsmodul, das Sie in Listing 7 finden.

Compilieren Sie die beiden Definitionsmodule und danach »MyShell.mod«, linken Sie aber noch nicht. Bei dieser Länge der Programme dürfte es etwas schwieriger werden, alle Zeilen fehlerfrei abzutippen. Lassen Sie sich davon nicht entmutigen. Sie wissen inzwischen, wie Sie mit Fehlern umgehen. Modula-Fehlermeldungen weisen sehr genau auf die tatsächliche Ursache des Fehlers hin. Während in anderen Programmiersprachen der Fehler oft erst weit nach dem eigentlichen Auftreten vom Compiler erkannt wird, merkt der Modula-Compiler in fast jeder Situation sofort, wo der Fehler entstand. Dies ist vor allem durch die starke Typenkontrolle möglich und durch die sehr starre Sprachstruktur.

Wenn Sie alle drei Listings (5, 6, 7) fehlerfrei eingegeben haben, machen wir uns an die Implementation der Routinen. Fangen wir mit dem einfachsten an, mit dem Modul »Menues.mod«. Nachdem wir uns zunächst auf die Fenstertitel konzentrieren wollen, können wir Menues noch vernachlässi-

gen. Um das Programm zu testen, brauchen wir aber eine Minimal-Implementation. Diese umfaßt lediglich zwei Zeilen. Damit ist das Modul bekannt, weiter braucht nichts erledigt zu werden. Das Modul ist entsprechend kurz:

```
IMPLEMENTATION MODULE
Menues;

END Menues.
```

Damit haben wir das Modul ausreichend für den Linker vorbereitet und können uns auf das Wichtige konzentrieren. Später werden wir auf diese Implementation zurückkommen und sie erweitern. Zunächst benötigen wir eine Implementation von »IntuiTool«. Sehen Sie sich den Vorschlag aus Listing 8 an.

Wir beschränken uns darauf, die Prozeduren zu besprechen, die wir für die Shell benötigen. »ActiveWindow« liest den Zeiger auf das aktive Fenster aus IntuitionBase und gibt ihn an das aufrufende Programm zurück. »UpdateTitle« gibt einem angegebenen Intuition-Fenster einen neuen Titel. Den Zeiger für UpdateTitle erhalten wir über ActiveWindow. Das einzige Problem ist jetzt, daß ActiveWindow auf einen Bereich zugreift, von dem wir nicht wissen, ob er zugänglich ist. IntuitionBase dürfen wir erst ansprechen, wenn sie ausdrücklich geöffnet wurde. Es gibt zwei einfache Möglichkeiten, dies zu erreichen: Wir könnten eine zusätzliche Zeile in ActiveWindow einfügen. Dann würde diese bei jedem Aufruf abgearbeitet, obwohl es nur beim erstenmal nötig ist. Die zweite Möglichkeit ist, diese Zeile an den Anfang des Hauptprogramms zu setzen. Dann müßten wir aber jedem Benutzer unserer Bibliothek sagen, daß er »OpenIntuition« aufrufen muß, bevor er mit ActiveWindow (oder auch einer ähnlich aufgebauten Prozedur »ActiveScreen«) arbeiten kann. Beide Möglichkeiten sind also unbefriedigend.

Zum Glück läßt Modula hier aber eine äußerst elegante Lösung zu: Ein Implementationsmodul kann einen eigenen An-

```
IMPLEMENTATION MODULE IntuiTool;
```

```
FROM SYSTEM IMPORT ADR, ADDRESS;
FROM Intuition IMPORT WindowPtr, ScreenPtr, SetWindowTitles,
IntuitionBase, OpenIntuition, IntuiMessagePtr, NewWindow,
ScreenFlags, ScreenFlagSet, OpenWindow, CloseWindow, WindowFlagSet,
IDCMPFlagSet, IDCMPFlags, ModifyIDCMP, SetMenuStrip, ClearMenuStrip,
MenuPtr;
FROM Dos IMPORT FileHandlePtr, Execute;
IMPORT Menues;
```

```
VAR IntuitionBasePtr: POINTER TO IntuitionBase;
```

```
PROCEDURE ActiveWindow (): WindowPtr;
BEGIN
RETURN IntuitionBasePtr.activeWindow
END ActiveWindow;
```

```
PROCEDURE OpenMinFenster (l,t,w,h: INTEGER): WindowPtr;
BEGIN
RETURN OpenIFenster (.,t,w,h,0,IDCMPFlagSet[],WindowFlagSet[])
END OpenMinFenster;
```

```
PROCEDURE OpenIFenster (l,t,w,h: INTEGER; name: ADDRESS;
id: IDCMPFlagSet; wi: WindowFlagSet): WindowPtr;
```

```
VAR neu: NewWindow;
```

```
BEGIN
```

```
WITH neu DO
```

```
leftEdge := l; topEdge := t; width := w; height := h;
detailPen := 0; blockPen := 1;
idcmpFlags := idcmpFlgs; flags := winFlgs;
firstGadget := NIL; checkMark := NIL; title := name;
screen := NIL; bitMap := NIL;
minWidth:=0; minHeight:=0; maxWidth:=0; maxHeight:=0;
type := ScreenFlagSet{wbenchScreen}
```

```
END;
```

```
RETURN OpenWindow (neu)
```

```
END OpenIFenster;
```

```
PROCEDURE CloseIFenster (iF: WindowPtr);
```

```
BEGIN
```

```
IF iF # NIL THEN CloseWindow (iF) END
```

```
END CloseIFenster;
```

```
PROCEDURE InstallMenu (iF: WindowPtr; m: MenuPtr);
```

```
END InstallMenu;
```

```
PROCEDURE RemoveMenu (iF: WindowPtr);
```

```
END RemoveMenu;
```

```
PROCEDURE VerarbeiteMenu (code: CARDINAL; fh: FileHandlePtr);
```

```
END VerarbeiteMenu;
```

```
PROCEDURE UpdateTitle (iF: WindowPtr; text: ARRAY OF CHAR);
```

```
BEGIN
```

```
SetWindowTitles (iF, ADR(text), -1)
```

```
END UpdateTitle;
```

```
BEGIN
```

```
IntuitionBasePtr := OpenIntuition ();
```

```
END IntuiTool.mod
```

## Listing 8. »IntuiTool« vereinfacht die Arbeit mit wichtigen Intuition-Funktionen

weisungsteil haben. Dieser wird immer dann abgearbeitet, wenn mindestens ein Name aus diesem Modul in ein anderes importiert wird. Egal, wer jetzt etwas aus IntuiTool importiert, er bekommt unsere Zeilen »mitgeliefert« und diese werden bearbeitet, bevor das Hauptprogramm startet. Werden mehrere Module importiert, die einen Anweisungsteil enthalten, dann werden diese in der Reihenfolge ihres Aufrufs bearbeitet. Wir benötigen hier nur eine Zeile:

```
IntuitionBasePtr :=
OpenIntuition ();
```

»IntuitionBasePtr« wird in ActiveWindow gebraucht. Es handelt sich um einen Zeiger, der auf einen sogenannten »RECORD« zeigt. Ein Record besteht aus verschiedenartigen Elementen, die über ihren Namen angesprochen werden können. Dazu muß zunächst der Zeiger (der nichts anderes als eine Adresse darstellt) »dereferenziert« werden. Dereferenzieren bedeutet, daß Sie mit



der Variablen nicht mehr die Adresse, sondern direkt die Daten ansprechen. Dies geschieht in der Form

```
IntuitionBasePtr^
```

Danach machen Sie deutlich, daß Sie mit einem einzelnen Element des Records arbeiten wollen. Das geschieht durch Anhängen eines Punkts. Außerdem müssen Sie anmelden, um welches Element es sich handelt:

```
IntuitionBasePtr^
activeWindow
```

Testen Sie die Fehlermeldungen Ihres Compilers, indem Sie

eines der Elemente weglassen. Versuchen Sie auch statt activeWindow einen anderen Namen anzugeben. Welche Fehlermeldung erhalten Sie?

Theoretisch könnten Sie jetzt diese Bibliotheken compilieren, dann Ihr Programm linken und starten. Aber inzwischen werden einige Prozeduren verwendet, die in der letzten Version noch nicht oder nur unvollständig implementiert waren.

Lassen Sie uns also noch StringTool und DosTool vervollständigen. Das Besondere an diesem Vorgehen ist, daß die Verbesserung der Bibliotheken

- wenn man nicht gerade die Schnittstelle verändert - keinen

## Erweitern der Bibliotheken

Einfluß auf bestehende Programme hat. Sie brauchen MyCli nicht zu verändern, damit es mit der verbesserten Version zusammenarbeitet. Sie brauchen es noch nicht einmal neu compilieren. Sie müssen lediglich die neuen Bibliotheken dazulinken - fertig.

Beachten Sie zunächst die Veränderungen in »StringTool« (Listing 9).

Die Compare-Prozedur bleibt unverändert. Neu sind hier lediglich die drei anderen Prozeduren. »Length« stellt die Länge eines Strings fest. Dies geschieht, indem in einer Schleife der gesamte String durchlaufen wird. Ist das Ende des Strings erreicht, so enthält der Schleifenzähler (»i«) den Wert der letzten Position und somit die Länge in Zeichen. Die »WHILE«-Schleife hat große Ähnlichkeit mit REPEAT. Sie durchläuft den Schleifenkörper so lange, wie die Bedingung erfüllt ist. Der Unterschied zu REPEAT besteht darin, daß die Bedingung am Anfang der Schleife getestet wird. Achten Sie auf die Initialisierung von i vor dem ersten Durchlauf. Dies ist nötig, weil i nicht automatisch in der Bedingung initialisiert wird (wie dies beispielsweise eine FOR-Schleife kann).

Die »Copy«-Prozedur kopiert den angegebenen Teil eines Strings in einen neuen. Dabei wird er im Zielstring automatisch bei der Position Null abgelegt und mit einem Nullbyte (»0C«) abgeschlossen. Die verwendete FOR-Schleife ist nur für eine sehr spezielle Art von Anwendungen geeignet: Nur wenn die Anzahl der Schleifendurchläufe vorher exakt bekannt (oder zumindest berechenbar) ist, hat es Sinn, FOR zu verwenden. Denn eine FOR-Schleife können Sie nicht vor Erreichen des Endwerts (siehe unten) abbrechen. Die typische Anwendung von FOR ist die Initialisierung von ARRAYS. Diese haben eine bekannte Anzahl Elemente und können daher mit FOR optimal behandelt werden. Dazu wird eine Zählvariable definiert, die von einem Startwert bis zu (»TO«) einem Endwert gezählt wird. Für jeden Wert wird einmal der Schleifenkörper ausgeführt (der Bereich zwischen »DO« und »END«). Die Zählvariable ist dabei nur innerhalb der Schleife definiert. Außerhalb sollte sie niemals

verwendet werden. Eine Besonderheit taucht bei der Adressierung des Strings »nach« auf: Der Index wird nicht direkt angegeben, sondern berechnet.

»Insert« sieht auf den ersten Blick etwas komplizierter aus. Die ersten beiden Zeilen testen, ob der Zielstring groß genug ist, um zusätzlich den neuen String aufzunehmen. In unserem Beispiel dürfte dies kein Problem sein, denn er wurde im Hauptprogramm mit 80 Zeichen definiert, und welcher Fenstertitel hat schon mehr? Aber eine Bibliothek soll möglichst allen denkbaren Anwendungen gewachsen sein (Sie werden zu Recht entgegenen, daß diese Forderung illusorisch ist, aber beachten Sie das »möglichst«) und es könnte ja sein, daß der Benutzer Ihrer Bibliothek nicht so genau Bescheid weiß wie Sie und einen falschen Wert übergibt. In diesem Fall bricht die Prozedur ihre Arbeit kommentarlos ab. Natürlich wäre es besser, eine Fehlermeldung auszugeben, aber an dieser Stelle haben wir darauf verzichtet, um das Programm nicht noch komplizierter zu machen.

Die nächste IF-Anweisung testet, wo der neue String eingefügt werden soll und leitet, falls nötig, entsprechende Maßnahmen ein. So wird dafür gesorgt, daß kein Loch entsteht, indem für den Startwert (»bei«) kein größerer Wert zugelassen wird als die Länge des Zielstrings. Ist »bei« dagegen kleiner als die Länge des Zielstrings, so müssen diese Zeichen erst »in Sicherheit« gebracht werden, damit sie von dem neuen String nicht überschrieben werden.

## Erweiterte FOR-Schleife

Dies geschieht rückwärts, damit die kopierten Zeichen sich nicht selbst überschreiben. Zu diesem Zweck wird eine Erweiterung der FOR-Anweisung eingeführt: »BY -1« legt für den Zähler eine neue Schrittweite fest. Standardmäßig ist 1 eingestellt. Um rückwärts zu zählen, geben wir eine negative Zahl an. Man könnte auch beliebige andere Konstanten verwenden. Nach diesen Vorbereitungen wird der String in den Zielstring kopiert und das Ganze mit einem Nullbyte abgeschlossen. Ein Rückgabewert ist nicht nötig, da der Zielstring an die Variable im Hauptprogramm »gebunden« ist (mit der »VAR«-Anweisung im Prozedur-Kopf).

Auch in dieser Bibliothek nutzen wir die Möglichkeit, im Im-

```
IMPLEMENTATION MODULE StringTool;

PROCEDURE Compare (str1, str2: ARRAY OF CHAR): BOOLEAN;
VAR i: INTEGER;
BEGIN
  i := 0;
  REPEAT
    IF (CAP(str1[i]) # CAP(str2[i])) THEN
      RETURN FALSE
    END;
    INC(i)
  UNTIL ((str1[i]=0C) OR (str2[i]=0C));
  RETURN TRUE
END Compare;

PROCEDURE Length (str: ARRAY OF CHAR): INTEGER;
VAR i: INTEGER;
BEGIN
  i := 0;
  WHILE (str[i] # 0C) AND (i <= HIGH (str)) DO
    INC(i)
  END;
  RETURN i
END Length;

PROCEDURE Insert (str1: ARRAY OF CHAR; bei: INTEGER; VAR in: ARRAY
OF CHAR);
VAR i, L1, L2: INTEGER;
BEGIN
  L1 := Length (str1); L2 := Length (in);
  IF (L2 + L1) > HIGH (in) THEN RETURN END;
  IF (L2 < bei) THEN
    bei := L2
  ELSIF (L2 > bei) THEN
    FOR i := L2 TO bei BY -1 DO
      in[i+L1] := in[i]
    END
  END;
  FOR i := bei TO bei+L1-1 DO
    in[i] := str1[i-bei]
  END;
  in [L1 + L2 + 1] := 0C
END Insert;

PROCEDURE Copy (von: ARRAY OF CHAR; VAR nach: ARRAY OF CHAR;
Start, Length: INTEGER);
VAR i: INTEGER;
BEGIN
  FOR i := Start TO Length DO
    nach[i-Start] := von[i]
  END
END Copy;

BEGIN
  ExitLoop := "endli";
  ExitLoop[6] := 0C;
  ChangeDir := "ed ";
  ChangeDir[3] := 0C
END StringTool.
```

Listing 9. »StringTool« wird den neuen Anforderungen angepaßt



plementationsmodul Anweisungen einzufügen, zur Initialisierung von Variablen: »ExitLoop« und »ChangeDir« erhalten die später benötigten Werte (beachten Sie, daß wir für ExitLoop jetzt »endcli« verwenden, um das eventuell noch laufende CLI nicht zu stören.

Mit diesen Änderungen liefern alle im Hauptprogramm

verwendeten Prozeduren ein sinnvolles Ergebnis (die einzige Ausnahme bilden die Menü-Prozeduren, die wir erst später benötigen). Dennoch wollen wir die DosTool-Funktionen noch etwas verbessern. Sehen Sie sich dazu Listing 10 an.

Die drei Prozeduren zur Ein- und Ausgabe (»ReadString«, »WriteString« und »WriteLn«)

bleiben unverändert. Die erste Verbesserung ist das Einfügen eines eigenen Anweisungsteils. Diese Technik ist Ihnen bereits bekannt. In diesem besorgen wir uns mit Hilfe der »Lock«-Funktion einen Zeiger (einen sogenannten »FileLockPtr«) auf das aktuelle Directory. Dieser muß als globale Variable definiert sein, weil alle Variablen, die innerhalb einer Prozedur deklariert werden, nur so lange bestehen bleiben, wie die Prozedur aktiv ist. Allerdings brauchen wir unser Hauptprogramm nicht mit dieser Variablen belasten. Sie ist in DosTool definiert und nur dort sichtbar. Alle DosTool-Prozeduren können sie nutzen, außerhalb dieser Bibliothek ist sie unbekannt, da sie nicht exportiert wird (sie taucht nicht im Definitionsmodul auf).

In diesem Zeiger auf ein FileHandle wird das Directory gespeichert, das vor dem Start unseres Programms eingestellt war (die Systemroutine »Lock« liefert einen Zeiger auf das aktuelle Directory, wenn ihr eine Null als Argument übergeben wird). Beim Beenden des Programms (genauer: beim Schließen des Fensters) machen wir mit Hilfe dieses Zeigers und der Prozedur »CurrentDir« das vorher eingestellte Directory wieder zum aktuellen. Dadurch ist es möglich, daß andere Programme von unseren Directory-Wechseln unbeeinflusst bleiben.

Drei neue Zeilen können Sie auch in OpenFenster erkennen. Mit der ersten besorgen wir uns einen Zeiger auf ein Directory, das bei jedem Amiga unabhängig von der eingelegten Diskette vorhanden ist (»DF0:«). Mit der zweiten Zeile machen wir dieses Directory zum aktuellen. Die dritte gibt den Zeiger wieder frei. Sie sollten sich angewöhnen, vom Betriebssystem angeforderte Speicherbereiche möglichst schnell wieder freizugeben, sobald sie nicht mehr benötigt werden. Der Lock auf das Directory, das vor dem Start unseres Programms aktiv war, muß natürlich beibehalten werden, solange das Programm läuft. Aber den Zeiger auf »DF0:« in OpenFenster können Sie freigeben, sobald DF0: per CurrentDir zum aktiven Directory gemacht wurde. Das gleiche gilt für auch für die Prozedur CD. Allerdings wird hier erst getestet, ob der erhaltene Zeiger gültig ist (es könnte ja sein, daß der Anwender Ihres Programms versucht, in ein Directory zu wechseln, das nicht existiert). Vollkommen neu ist die Pro-

zedur »CD«. Sie macht das in »name« übergebene Directory zum aktuellen. War dies möglich, wird TRUE als Ergebnis zurückgeliefert, wenn nicht, ist das Ergebnis FALSE. Das Vorhandensein des gewünschten Directory erkennen wir am Rückgabewert der Lock-Funktion. Diese gibt eine Null zurück, wenn kein Directory mit diesem Namen vorhanden ist. Im andern Falle ist das Ergebnis ein FileLockPtr, der auf dieses Directory zeigt.

Compilieren Sie diese beiden Bibliotheken und linken Sie »MyShell.obj« erneut. Starten Sie das entstandene Programm und testen Sie alle Funktionen.

## Die letzten Feinheiten

Achten Sie beim CD-Befehl darauf, daß dieser in der vorgestellten Version noch keine zwei aufeinanderfolgenden Schrägstriche verträgt (um beispielsweise zwei Directories auf einmal nach oben zu wechseln). Ebenso sollten Sie vermeiden, mit der Eingabe von »cd :« direkt ins Stammverzeichnis zu wechseln. Dieser Befehl funktioniert zwar (ebenso wie die beiden Schrägstriche keinen Fehler produzieren), aber die Directory-Anzeige reagiert nicht korrekt. Testen Sie diese beiden Unzulänglichkeiten ausführlich und versuchen Sie eine Lösung zu finden. Dazu ein Tip: Der String »newdir« muß auf diese beiden Spezialfälle untersucht werden. In Abhängigkeit vom Ergebnis korrigieren Sie den bestehenden Fenstertitel (»WTitel«). Haben Sie die Stelle gefunden, an der die Abfrage eingefügt werden muß? Haben Sie eine funktionsfähige Lösung gefunden? Dann haben Sie den schwierigsten Schritt im sogenannten »Debugging« (Fehlersuche) bereits hinter sich. Denn die einfachste Stufe ist die Suche nach syntaktischen Fehlern. Diese werden vom Compiler angezeigt. Fast ebenso einfach ist die Suche nach Fehlern, die einen Absturz verursachen. Auch hier erhalten Sie eine Fehlermeldung, die genaue Rückschlüsse auf den aufgetretenen Fehler erlauben, wenn dies auch immer noch wenig bekannt ist.

Die linke Hälfte der Guru-Nummer ist hierfür zuständig. Im Amiga-Sonderheft 4 (Thema C und Assembler) finden Sie eine Liste der bekannten Fehlernummern mit einer kurzen Erklärung der Ursache. Der schwierigste Schritt ist die Suche nach Fehlern im Programm-

```
IMPLEMENTATION MODULE DosTool;

FROM Dos IMPORT Open, Close, FileHandlePtr, newFile, Read, Write,
    CurrentDir, Lock, FileLockPtr, Unlock, sharedLock;
FROM SYSTEM IMPORT ADR;

VAR L: FileLockPtr;

PROCEDURE OpenFenster(): FileHandlePtr;
VAR l: FileLockPtr;
BEGIN
    l := Lock (ADR("df0:"), sharedLock);
    l := CurrentDir (l);
    Unlock (l);
    RETURN Open (ADR("CON:0/0/640/256/DF0:"), newFile)
END OpenFenster;

PROCEDURE CloseFenster (f: FileHandlePtr);
BEGIN
    l := CurrentDir (L);
    Close (f)
END CloseFenster;

PROCEDURE WriteString (fh: FileHandlePtr; str: ARRAY OF CHAR);
VAR ok: INTEGER;
BEGIN
    ok := Write (fh, ADR(str), SIZE(str))
END WriteString;

PROCEDURE ReadString (fh: FileHandlePtr; VAR str: ARRAY OF CHAR);
VAR c: CHAR;
    i, ok: INTEGER;
BEGIN
    IF HIGH (str) >= 1 THEN
        i := 0;
        REPEAT
            ok := Read (fh, ADR(c), SIZE(c));
            str[i] := c;
            INC(i)
        UNTIL (i >= HIGH(str)) OR (c = 12C);
        str[i-1] := 0C
    END;
END ReadString;

PROCEDURE WriteLn (fh: FileHandlePtr);
VAR LF: CHAR;
    ok: INTEGER;
BEGIN
    LF := 12C;
    ok := Write (fh, ADR (LF), SIZE(LF));
END WriteLn;

PROCEDURE CD (name: ARRAY OF CHAR): BOOLEAN;
VAR l: FileLockPtr;
BEGIN
    l := Lock (ADR(name), sharedLock);
    IF l # 0 THEN
        l := CurrentDir (l);
        Unlock (l);
        RETURN TRUE
    END;
    RETURN FALSE
END CD;

BEGIN
    L := Lock (0, sharedLock)
END DosTool.
```

**Listing 10. In dieser Version verarbeitet »DosTool« jeden Directory-Wechsel korrekt**



Design. Ein Denkfehler beim Programmwurf ist sehr schwierig zu entdecken, weil man bei der Fehlersuche meist immer noch in denselben Bahnen denkt wie während des Entwerfens. Hilfreich sind hier Maßnahmen wie das Herausnehmen einzelner Programmblöcke, um den Fehler einzugrenzen. Ebenso kann es helfen, einzelne Variablen während des Programmablaufs anzeigen zu lassen. Eine weitere Maßnahme ist das Auslagern von Programmteilen in Prozeduren. Dadurch wird die Struktur leichter erkennbar.

Bevor Sie jetzt weiterlesen, sollten Sie das bisher Gelernte nochmals durchgehen und auf sich wirken lassen. Machen Sie eine kurze Pause, verdeutlichen Sie sich einzelne Punkte in der Reihenfolge wie sie Ihnen in den Sinn kommen (versuchen Sie dabei nicht, angestrengt an einen Bereich zu denken, sondern lassen Sie Ihre Gedanken einfach kreisen). Danach überprüfen Sie, ob Ihnen folgende Begriffe geläufig sind:

- Deklarationsteil und Anweisungsteil
- Anweisungsblock
- Rückgabewert
- Prozedur
- Schleife und Verzweigung
- Betriebssystemroutine
- Variable und Konstante
- Aufruf von Prozeduren
- Wertzuweisung und Vergleich
- gebundene Variablen
- Initialisierung von Variablen
- Variablentypen
- Parameterübergabe
- Arrays und Records
- Boolesche Variablen
- Integers und Strings

Wenn Sie damit keine Probleme mehr haben, können Sie sich auf einige Aufgaben stürzen. Diese sollen das Verständnis des bisher behandelten Stoffs vertiefen. Bearbeiten Sie sie daher aufmerksam, denn alle folgenden Programmbeispiele werden etwas anspruchsvoller sein als die bisherigen. Lösen Sie folgende Aufgaben, bevor Sie weiterlesen:

- Schreiben Sie ein Programm, das dem Benutzer erlaubt, einen String einzugeben. Dieser soll zunächst auf ein vorher definiertes Trennzeichen untersucht werden (beispielsweise das Leerzeichen) und an diesen Stellen in einzelne Strings zerlegt werden. Verwenden Sie StringTool für diese Aufgabe.
- Schreiben Sie ein Programm, das einen beliebigen String umformt. Das Ergebnis soll ein alphabetisch geordnetes Abbild der ursprünglichen Zei-

chenkette sein. Schreiben Sie von diesem Programm zwei Versionen. Einmal sollen alle Kleinbuchstaben vor den Großbuchstaben auftauchen. Die zweite Version soll keinen Unterschied zwischen Groß- und Kleinschreibung machen.

- Öffnen Sie fünf verschiedene große Fenster. Schreiben Sie beliebige Texte in die einzelnen Fenster und ändern Sie die Fenstertitel jedesmal, wenn ein String in ein Fenster geschrieben wurde. Hinweis: Sie brauchen fünf Variablen vom Typ FileHandlePtr.

## Ihre Lösungen sind gefragt

- Ändern Sie das Programm aus der letzten Aufgabe, so daß jedesmal das Fenster in den Vordergrund bewegt wird, in das gerade geschrieben wird. Sie benötigen dazu zwei Intuition-Prozeduren (»WindowToFront« und »WindowToBack«). Beide verlangen nur einen Parameter: einen Zeiger auf Ihr Fenster vom Typ »WindowPtr«. Beide haben keinen Rückgabewert.

- Schreiben Sie ein Programm, das ein Feld von hundert Integer-Zahlen mit Eingaben des Benutzers füllt (in der Bibliothek »InOut« ist eine Prozedur »ReadInt« enthalten - als Parameter übergeben Sie eine Integervariable, die von der Prozedur gefüllt wird). Dieser Array soll - nachdem er aufsteigend sortiert wurde - im CLI-Fenster ausgegeben werden (mit »WriteInt« aus InOut. WriteInt hat zwei Parameter: die Integerzahl für die Ausgabe und eine Integervariable für die Breite der Ausgabe in Zeichen).

- Schreiben Sie oben genanntes Programm so um, daß es die Eingabe abbricht, wenn der Benutzer eine Null eingibt. Außerdem soll für alle eingegebenen Werte ein ganzzahliger Mittelwert und ein Median berechnet werden. Dafür benötigen Sie den Operator für die ganzzahlige Division (»z := x DIV y«). Der Mittelwert ist der Durchschnittswert eines Zahlenfelds, der Median ist der Wert, der in einem sortierten Zahlenfeld die mittlere Position einnimmt.

Sicher werden Sie diese Aufgaben nicht innerhalb weniger Minuten erledigt haben (ich würde vor Neid erblassen). Einige davon sind vielleicht auch nicht ohne längeres Ausprobieren zu lösen. Aber lassen Sie sich davon nicht entmutigen! Es ist vorteilhaft, diese Art von Schwierigkeiten bei kleinen überschaubaren Problemen

kennenzulernen. Später werden Sie die so gewonnene Routine sehr nützlich finden.

Einige Tips sollen Ihnen aber nicht vorenthalten bleiben: Fangen Sie bei jeder Aufgabe mit dem kleinsten Teil an. Wenn Sie mit mehreren Fenstern arbeiten, öffnen Sie erst eines, bevor die anderen nachgeschoben werden und testen Sie das Programm in dieser Zwischenphase. Wo immer dies möglich ist, sollten Sie eine Prozedur zur Erledigung von klar begrenzten Aufgaben einsetzen. Überlegen Sie sich vor jeder Aufgabe: Was wird vom Benutzer an Eingaben erwartet? Was soll ausgegeben werden? Welche Vorbereitungen müssen für die Fenster getroffen werden?

Haben Sie alle Aufgaben erfolgreich gelöst? Wenn nicht, sollten Sie das betreffende Thema weiter vorne nochmals nachlesen.

Bevor wir tiefer ins Betriebssystem einsteigen, muß ich Sie mit etwas Theorie belästigen. Sie haben bisher die einfachen Datentypen von Modula kennengelernt. Dazu gehören Integer-, Char- und Boolean-Variable. Daneben kennen Sie Arrays und den Zugriff auf einzelne Elemente eines Arrays (mit Hilfe des Index in eckigen Klammern). Oberflächlich kennengelernt haben Sie Records und den Zugriff auf deren Elemente (mit dem Punkt zur Trennung). Bisher haben wir allerdings konsequent die ausführliche Behandlung von Zeigern und Adressen vermieden. Wir haben Sie lediglich verwendet und festgestellt, daß es funktioniert. Gerade die letzten beiden Punkte (Zeiger und Records) sind aber für das Betriebssystem sehr wichtige Daten. Sobald Sie über den »Tellerrand« von AmigaDOS hinausschauen, werden Sie Ihren Weg mit Zeigern und Strukturen (was gleichbedeutend mit Record ist) gepflastert sehen. Daher werden wir bevor, wir uns an die damit überreichlich gesegneten Menüs wagen, erst diese beiden Datentypen ausführlich behandeln.

Lassen Sie uns mit den Zeigern beginnen. Dabei handelt es sich um eine spezielle Form von Adressen. Der Zeiger ist eine Variable, die die Adresse einer anderen Variable enthält. Sie werden sich sicher fragen, wozu der Umstand, wenn wir doch die zugehörige Variable kennen. Aber erstens kennen wir sie nicht immer (Betriebssystemroutinen liefern oft nur Adressen zurück), außerdem vermeidet man damit den Auf-

wand, jedesmal die ganze Datenstruktur »mitzuschleppen«, wenn ein Parameter übergeben wird. Zudem sind gewisse Techniken (die Sie noch kennenlernen werden) nur mit Zeigern möglich.

Eine spezielle Form von Zeigern stehen im Zusammenhang mit Strings. Das Amiga-Betriebssystem verlangt die Übergabe von Stringparametern stets als Adresse des ersten Zeichens dieses Strings. Das Ende wird am Zeichen »0C« erkannt, das jedem String automatisch angehängt wird.

In den Libraries (deren Definition Sie teilweise im Directory »defs« der Fish-Disk 113 nachlesen können) erkennen Sie einen Zeiger an der Endung »Ptr« in seinem Namen. Wenn Sie einen Zeiger selbst definieren, brauchen Sie erst einen sogenannten Basistyp. Anschließend definieren Sie eine Variable als Zeiger auf diesen Typ:

```
FROM Tool IMPORT Window;
VAR FensterPtr: POINTER
TO Window;
```

Von jetzt an kann dieser Zeiger keine beliebigen Adressen mehr aufnehmen, sondern nur noch die Adresse einer Variable des zugehörigen Basistyps (»Window«). Zeiger sind also an den Basistyp gebunden. Dieses Konzept ist keine Kleinkariertheit Ihres Modula-Compilers, sondern verhindert sehr zuverlässig den Mißbrauch von Zeigern. Denken Sie nur an das eindrucksvolle Beispiel vom Anfang dieses Workshops. Wir hatten einen FileHandlePtr mit Gewalt (genauer: mit der CAST-Funktion) in einen WindowPtr umgewandelt. Das Ergebnis war ein eleganter Absprung ins System-Nirwana.

## Zeiger - der Schlüssel zum System

Wenn Sie häufigen Gebrauch von Zeigern machen, ist es ratsam, gleichartige Zeiger unter einem Namen zusammenzufassen. Dies erreichen Sie mit der Definition eines eigenen Datentyps. Die Syntax ist einfach:

```
FROM Tool IMPORT
Fenster;
TYPE FensterPtr =
POINTER TO Fenster;
VAR Fenster1, Fenster2,
Fenster3: FensterPtr;
```

Sie hätten fast das gleiche erreicht, wenn Sie jede Variable als »POINTER TO Fenster« definiert hätten. Allerdings hätte



dies den Nachteil gehabt, daß die drei Fenster-Zeiger nicht kompatibel gewesen wären. Das bedeutet, daß der Compiler den Versuch, einen Zeiger zu kopieren, mit einer Fehlermeldung quittiert hätte. Dies liegt daran, daß ein so definierter Zeiger als »anonym« gilt, solange ihm kein eigener Typ-Name gegeben wurde. Anonyme Typen werden aber ebenso verwaltet wie inkompatible: mit strikter Trennung.

riable fPtr in die Variable »f« kopieren:

```
f := fPtr
```

Falls Fenster ein Record ist, haben Sie damit auch Zugriff auf die einzelnen Felder.

Als Record bezeichnet man einen Verbund verschiedener Datentypen. In allen höheren Programmiersprachen sind ähnliche Datenstrukturen implementiert. Beispielsweise haben Sie in C den Typ »struct« zur

ren, bevor der Record selbst definiert ist. Fügen Sie dazu die Zeile

```
TYPE RecPtr = POINTER TO RecBeispiel;
```

vor der Record-Deklaration ein. Wenn Sie diese Zeilen in ein beliebiges Programm einfügen, hat der Compiler nichts dagegen, obwohl er sonst ziemlich kleinkariert über jeden (noch) unbekannt Namen meckert. Das ist keineswegs eine von den Entwicklern übersehene Hintertür, sondern unbedingt notwendig. Denn um die - besonders auf Betriebssystemebene oft gebrauchten - verketteten Listen zu implementieren, muß irgendwann die Frage beantwortet werden, was zuerst da war (die Henne oder das Ei?). Denn verkettete Listen enthalten einen Zeiger auf eine gleichartige Struktur (das bedeutet für die Typ-Deklaration: auf sich selbst!). Durch die »Vorwärts-Deklaration« von Zeigern wird es überhaupt erst möglich, einen Verbund der folgenden Art herzustellen:

```
TYPE ListPtr = POINTER TO Liste;
Liste = RECORD
  vor: ListPtr;
  nach: ListPtr;
  Kopf: ListPtr;
  Fuss: ListPtr;
  Element: CHAR;
END;
```

Stellen Sie sich nur den Aufwand vor, der Ihnen bevorstehen würde, wenn dem Compiler ein Name erst bekannt wäre, nachdem er deklariert ist. Welchen Typ würden Sie zuerst definieren, die Liste oder den Zeiger darauf?

Records (oder Strukturen wie dieser Datentyp häufig in Anlehnung an die C-Konventionen genannt wird) sind im Amiga-Betriebssystem ein zentraler Begriff. Die meisten Daten werden in Form von Records verwaltet. Um einen Überblick zu gewinnen, starten Sie Ihren Editor und laden Sie das File »Intuition.def« aus dem Defs-Ordner der Fish 113. Sehen Sie sich den Code ausgiebig an. Sie werden einige interessante Deklarationen sehen, die Sie auch später wiederfinden. Besonders ausführlich sollten Sie sich die Strukturen und Sets im Zusammenhang mit den Menüs ansehen (diese finden Sie gleich am Anfang des Files). Daneben sollten Sie die Deklaration der NewWindow- und der Window-Struktur (kurz vor den Prozeduren) aufmerksam studieren. Wenn Sie sich für Gadgets interessieren, finden Sie

die entsprechenden Strukturen zwischen Menu- und NewWindow-Struktur.

Diese Definitionenmodule sind eine interessante Informationsquelle, sobald Sie die Möglichkeiten der mitgelieferten Bibliotheken voll nutzen wollen. Nicht umsonst werden bei der kommerziellen Version von M2Amiga die Definitionenmodule im Handbuch abgedruckt. Sie sollten viel Zeit auf das Studieren dieser Informationen verwenden. Es lohnt sich in jedem Fall, die investierte Zeit sparen Sie mehrfach bei der Entwicklung Ihrer Programme wieder ein.

## Definitions- module - ein unerschöpfliches Nachschlage- werk

Lassen Sie diese Informationen zu Zeigern und Records einen Moment lang auf Sie wirken. Stellen Sie sich die Zusammenarbeit dieser wichtigen Typen vor und überprüfen Sie, ob Sie alles verstanden haben, bevor Sie weiterlesen. Ohne das Verständnis dieser Datentypen können Sie den Amiga nicht sinnvoll programmieren.

Bevor wir uns an die Aufgabe machen, die gewünschten Menüs zu installieren, machen Sie sich erst klar, wie Menüs arbeiten. Eine Menüstruktur besteht aus den in Tabelle 6 dargestellten Elementen. Die Menüstruktur verwaltet die sogenannten Menü-Header. Das sind die Menüpunkte, die in der Titelleiste auftauchen, wenn Sie die rechte Maustaste betätigen. Wenn Sie diese Struktur genauer betrachten, werden Sie feststellen, daß ein Zeiger (»nextMenu«) auf eine weitere Menüstruktur verweist. Jede Menüstruktur verwaltet genau einen Menü-Header. Wenn Sie mehrere in Ihre Menüleiste aufnehmen wollen, müssen Sie die entsprechende Anzahl Menüstrukturen verketteten, indem Sie im Feld nextMenu jeweils den Zeiger der nächsten Menüstruktur eintragen. Das Ende dieser Liste wird vom Betriebssystem am vordefinierten Wert »NIL« erkannt. Sie sehen diese Abhängigkeiten in Bild 1 dargestellt.

Die nächsten vier Felder der Menüstruktur geben die Position des Menü-Headers an. Dabei haben in der aktuellen Betriebssystemversion nur zwei eine Bedeutung: »leftEdge« und »width«. »topEdge« und »height« sollten Sie immer mit

Feld	Typ	Bedeutung
nextMenu:	MenuPtr;	Zeiger zum nächsten Menü-Header.
leftEdge:	INTEGER;	Abstand der linken Kante des Menü-Headers von der linken Kante des Screen in Bildpunkten.
topEdge:	INTEGER;	hat zur Zeit noch keine Bedeutung.
width:	INTEGER;	Breite des Menü-Headers in Bildpunkten.
height:	INTEGER;	hat zur Zeit noch keine Bedeutung.
flags:	BITSET;	kann »menuEnabled« setzen (BITSET{0}) oder kein Flag (BITSET{}).
menuName:	ADDRESS;	Adresse des Strings, der im Header angezeigt werden soll.
firstItem:	MenuItemPtr;	Zeiger zum ersten Menü-Punkt.
jazzX:	INTEGER;	wird von Intuition intern verwendet.
jazzY:	INTEGER;	wird von Intuition intern verwendet.
beatX:	INTEGER;	wird von Intuition intern verwendet.
beatY:	INTEGER;	wird von Intuition intern verwendet.

**Tabelle 6. Eine Menü-Struktur enthält zwölf Felder, von denen aber nur sechs für den Programmierer wichtig sind**

Zeiger können mit einer Wertzuweisung initialisiert oder kopiert werden. Es gibt jedoch keine Arithmetik für Zeiger. Die Berechnung einer Speicherstelle etwa nach dem Motto

```
FensterPtr3 := FensterPtr1 + FensterPtr2;
```

ist unzulässig. Diese Unsauverkeit hätte auf dem Amiga - wenn es möglich wäre - katastrophale Folgen.

Der einzige zulässige Operator für Zeiger ist (neben der Wertzuweisung) die Dereferenzierung mit dem »\*«-Operator. Sie könnten beispielsweise zwei Variablen in der folgenden Form definieren:

```
FROM Tool IMPORT
Fenster;
VAR f: Fenster;
    fPtr: POINTER TO
    Fenster;
```

Wenn Sie jetzt die Variable »fPtr« dereferenzieren, haben Sie Zugriff auf den zugrunde liegenden Datentyp. Sie könnten also die dereferenzierte Va-

Verfügung. Auch das Listenkonzept von LISP weist einige Ähnlichkeiten zu den Modul-Records auf (es handelt sich dabei aber um einen umfassenderen Ansatz).

Bei Records ist es aus Gründen der Übersichtlichkeit sinnvoll, sich für jede Anwendung einen eigenen Typ zu »bauen«. Dies geschieht mit der soeben kurz beschriebenen »TYPE«-Deklaration. Sie finden dazu viele Beispiele in den Definitionenmodulen der Betriebssystemschnittstellen, die auf der Fish-Disk 113 im »Defs«-Ordner liegen. Dennoch hier ein eigenes Beispiel:

```
TYPE RecBeispiel =
RECORD
  a: INTEGER;
  b: CHAR;
  c: ARRAY
    [0..398] OF
    CHAR;
END;
```

Sie können einen Zeiger auf diesen Record bereits definie-



Null initialisieren, weil sie zur Zeit einen nicht veränderbaren Wert haben. Sollte in einer späteren Version eine Bedeutung eingeplant sein, so wissen wir jetzt noch nichts Genaues über den Sinn dieser Werte. Null dagegen ist fast immer ein Wert, der gegen diesbezügliche Änderungen relativ »resistent« ist.

Die Variable »flags« ist vom Typ »BITSET«. Dabei handelt es sich um einen Set, der aus den Elementen 0 bis 15 bestehen kann (oder in Modula-Schreibweise: »{0..15}«). Sie können hier sinnvollerweise nur den Wert {0} angeben, da die anderen Flags nicht abgefragt beziehungsweise für interne Zwecke verwendet werden (der Wert {8}). Die Null hat die Bedeutung »menuEnabled«. Dadurch ist das Menü für den Benutzer an-

## Menü-»Ghosting« mit Flags

wählbar. Ist kein Flag gesetzt, so erscheint das Menü in Geisterschrift und kann vom Benutzer nicht gewählt werden.

»firstItem« ist ein Zeiger auf den ersten Unterpunkt. Hierhin kopieren Sie einen »MenuItemPtr«, nachdem die entsprechende Struktur initialisiert ist. Die restlichen vier Variablen haben für den Programmierer keine Bedeutung, sie werden vom Betriebssystem verwaltet.

Ähnlich wie Menüs sind Menü-Items in einer verketteten Liste verbunden. Das erste Menü-Item hängt dabei - wie bereits erwähnt - im Zeiger firstItem der Menü-Struktur. Alle weiteren sind über das Feld »nextItem« der MenuItem-Struktur miteinander verbunden. Sind zu den Menü-Items sogenannte »SubItems« definiert, dann sind diese in einer Liste organisiert, die mit ihrem ersten Zeiger im Feld »subItem« der MenuItem-Struktur hängt (siehe Bild 2).

SubItems werden mit der gleichen Struktur initialisiert wie MenuItems - der »MenuItem«-Struktur (Tabelle 7). Diese enthält die von der Menüstruktur bekannten Felder. Allerdings haben hier die Felder topEdge und height eine Bedeutung: Items können unterschiedlich hoch sein, daher wird die Höhe in Bildpunkten angegeben. Auch bei topEdge sollten Sie den tatsächlichen Abstand von der Unterkante der Menüleiste angeben. Die »flags« sehen bei den Items schon etwas interessanter aus als bei Menüs. Sehen Sie sich die Aufzählung der »MenuItem-Flags« im File »Intuition.def« an.

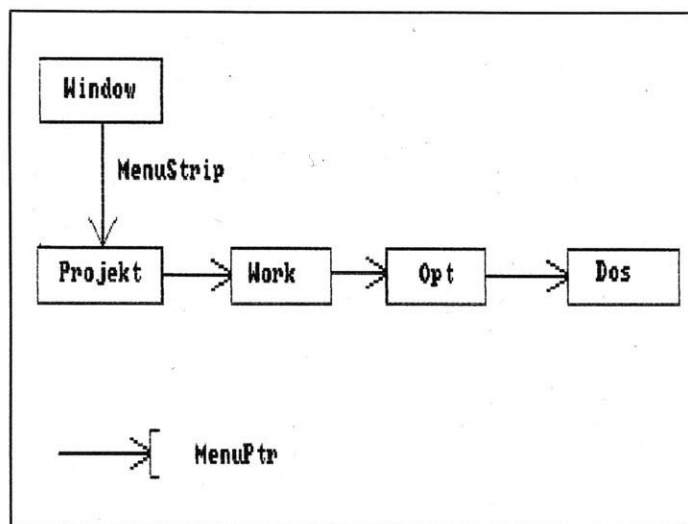


Bild 1. Menü-Header werden mit Zeigern verkettet

Die Flags mit dem Kürzel »mi« und einer anschließenden Zahl sind für den Benutzer uninteressant. Wir wollen uns daher auf die anderen konzentrieren. Für unser späteres Programm am wichtigsten sind »commSeq«, »itemEnabled«, »highComp«, »highBox« und »itemText«. Deren Bedeutung im einzelnen:

**commSeq** zeigt Intuition an, daß dieses Item über einen Shortcut (AMIGA-Taste in Verbindung mit einem Zeichen) angewählt werden kann (damit ist aber noch kein Zeichen festgelegt, über welches das Menü erreicht wird).

**itemEnabled** macht das Item anwählbar. Sie sollten dieses Flag immer setzen, es sei denn Sie möchten, daß dieser Menüpunkt anfangs in Geisterschrift erscheint und nicht angewählt werden kann.

**highComp** legt den Modus des »Highlighting« fest (in diesem Fall wird das Item komplett dargestellt).

**highBox** zeichnet einen Rahmen um den angewählten Menüpunkt.

**itemText** zeigt Intuition an, daß das Item mit Text gefüllt werden soll. Theoretisch könnte statt dessen auch Grafik enthalten sein. In diesem Fall setzen Sie kein Flag.

Die anderen Flags werden von uns nicht verwendet, daher können wir sie hier übergehen. Wenn Sie sich für deren Bedeutung interessieren, finden Sie im C-Kurs aus dem vierten AMIGA-Sonderheft einige interessante Hinweise.

Das nächste Feld im MenuItem-Record heißt »mutualExclude«. Es ist nur bei sogenannten »Attribut«-Menüs interessant, die einen Zustand beschreiben - anstatt wie in unseren Beispielen eine Aktion zu

veranlassen. Diese Art von Menü-Items finden Sie beispielsweise in Textverarbeitungsprogrammen. Dort legen sie fest, welche Schrift eingestellt ist. Nachdem immer nur eine Schrift eingestellt sein kann, werden die anderen automatisch von Intuition abgeschaltet, sobald eine Schriftart angewählt wurde. Dafür muß in den mutualExclude-Flags festgelegt sein, welche Items abgeschaltet werden sollen.

In »itemFill« ist eine Adresse eingetragen, die entweder auf eine sogenannte »IntuiText«-Struktur verweist oder auf eine

»Image«-Struktur. Dies ist abhängig vom Flag »itemText«. Ist itemText gesetzt, so erwartet Intuition eine IntuiText-Struktur, im andern Fall handelt es sich um eine Image-Struktur.

In »selectFill« tragen Sie eine alternative Struktur ein, die angezeigt werden soll, wenn der Menüpunkt angewählt (»selected«) ist. Sicher kennen Sie die Menüs, die einen Menüpunkt »Ende« enthalten und in dem Moment, wo dieser angewählt wird, auf »Wirklich ?« umschalten.

Im Feld »command« ist das Zeichen eingetragen, das in Verbindung mit der AMIGA-Taste die gleiche Wirkung hat wie das Anwählen des Menüs. In »SubItem« ist der erste Zeiger der SubItem-Liste eingetragen. Sind keine SubItems definiert, so steht hier »NIL«. Das Feld »nextSelect« ist für den internen Gebrauch. Hier legt Intuition die Information ab, welches Menü als nächstes gewählt wurde, falls der Benutzer mehrere nacheinander mit der linken Maustaste aktiviert hat, ohne die rechte Taste loszulassen (sogenanntes »Dragselcting«).

Für jedes Menü-Item (oder SubItem) muß eine IntuiText-Struktur initialisiert werden, auf die das Feld »itemFill« zeigt (oder eine Image-Struktur, falls Sie grafische Menüs verwenden).

Feld	Typ	Bedeutung
nextItem: leftEdge:	MenuItemPtr; INTEGER;	Zeiger auf das nächste Item. Abstand der linken Kante von der linken Kante des Menü-Headers in Bildpunkten.
topEdge:	INTEGER;	Abstand der Oberkante von der Unterkante des Menü-Headers in Bildpunkten.
width:	INTEGER;	Breite des Menüpunkts in Bildpunkten.
height:	INTEGER;	Höhe des Menüpunkts in Bildpunkten.
flags:	MenuItem-FlagSet;	legt die Eigenschaften des Menüpunktes fest (s. Text).
mutual Exclude:	LONGSET;	legt bei Attribut-Menüs die Punkte fest, die bei Anwahl dieses Punktes ausgeschlossen werden sollen.
itemFill:	ADDRESS;	Adresse der Struktur, die in die Fläche des Menüpunktes eingetragen werden soll (IntuiText- oder ImagePtr).
selectFill:	ADDRESS;	Adresse einer alternativen Anzeige.
command:	CHAR;	Zeichen, das in Verbindung mit der rechten Amiga-Taste als Shortcut verwendet werden soll.
subItem: nextSelect:	MenuItemPtr; CARDINAL;	Zeiger auf das erste SubItem; Nummer des nächsten gewählten Menüpunkts bei »Dragselcting«.

Tabelle 7. Die »MenuItem«-Struktur enthält alle Informationen für Items und SubItems



den wollen). Diese IntuiText-Struktur sehen Sie in Tabelle 8. Sie enthält in »frontPen« beziehungsweise »backPen« je eine Farbummer, die für die Vorder- und Hintergrundfarbe verwendet werden soll.

Das Feld »drawMode« legt die Art der Textdarstellung fest. Wir werden vorerst nur »jam1« verwenden. Dies entspricht der Standard-Textdarstellung.

tend mit unübersichtlich ist, so sollten Sie sich dieses Programm doch einmal recht gründlich ansehen. Doch vorher überlegen wir uns, welche Funktionen unserer Shell wir per Menü (beziehungsweise Shortcut) aufrufbar machen wollen.

Selbstverständlich können Sie Ihr Menü später nach Ihren Wünschen ausbauen, aber vor-

und »Link Mainfile«. Für das Mainfile soll jeweils die korrekte Endung automatisch angehängt werden.

Das letzte Menü-Item ist »Testlauf«. Es startet das in Mainfile eingestellte Programm.

Es bleibt die Frage, woher wir die Informationen beziehen, was gerade als »Mainfile« eingestellt ist. Zu diesem Zweck lassen wir den Benutzer das

Aber lesen Sie zunächst das Listing 11.

Sie finden eine ungewöhnlich umfangreiche Import-Liste für die Intuition-Library. Die einzelnen Prozeduren und Daten werden weiter unten erklärt. Daneben ist erstmals eine Import-Anweisung für die Graphics-Library enthalten. Diese verwaltet einige komfortable Routinen für das Zeichnen in Intuition-Fenster und zur Textausgabe. Sehen Sie sich die Datei »Graphics.def« genau an, Sie werden viele interessante Anregungen finden, die an dieser Stelle leider nicht behandelt werden können.

Die drei Prozeduren wurden bereits kurz erwähnt. »Struct-Menu« benötigt vier Parameter. »m« ist ein Record als gebundene Variable, der von StructMenu initialisiert werden soll. »l« und »w« bezeichnen Position und Breite des Header-Felds. »text« ist die Adresse des Namens, den Sie in diesem Menü-Header anzeigen möchten. Als Rückgabewert wird ein »MenuPtr« festgelegt. Sie werden sich fragen, wozu neben der Struktur, die ja an die entsprechende Variable im Aufruf gebunden ist, auch noch deren Adresse zurückgegeben werden muß. Dies vereinfacht die Verkettung der einzelnen Header. Sehen Sie sich dazu die ersten vier Zeilen im Anweisungsteil des Moduls an.

Im Anweisungsteil der Prozedur sehen Sie eine neue Technik des Zugriffs auf Records. Bisher hatten wir die einzelnen Felder mit dem Namen der Struktur, einem Punkt und dem Feldnamen angesprochen. Bei großen Records ist dies sehr umständlich, da für jedes Feld der gleiche »Vorspann« verwendet wird (Recordname - eventuell dereferenziert - und Punkt). Dies läßt sich mit der »WITH«-Anweisung vereinfachen. WITH legt fest, daß alle Felder des Records bis zum zugehörigen END in die Variablenliste aufgenommen werden. Das bedeutet, daß jedes Feld mit seinem Namen angesprochen werden kann, ohne den Recordnamen davorzustellen.

Die Menü-Struktur wird mit den übergebenen Variablen initialisiert. Daneben werden einzelne Felder mit festen Werten belegt. Dazu gehören »topEdge« und »height«, da beide - wie bereits erwähnt - im Moment keine Bedeutung haben. »flags« wird mit »menuEnabled« belegt und »firstItem« wird auf »NIL« gesetzt. Daraus zu schließen, daß keine Items in

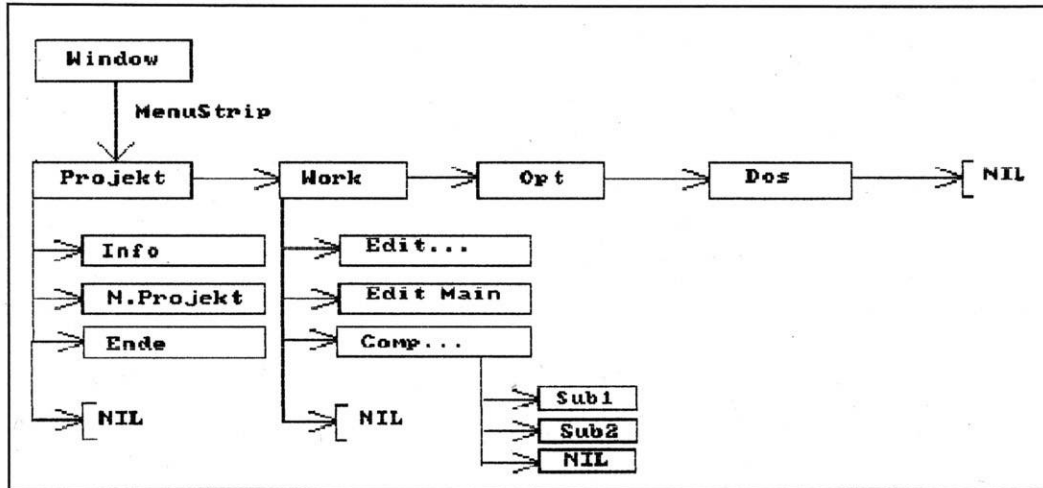


Bild 2. Der komplette »MenuStrip« sieht komplizierter aus als er ist

»leftEdge« und »topEdge« legen den Ausgangspunkt des Schriftzugs fest. Dies ist relativ zur späteren Position des Texts zu sehen.

In »TextFonts« können Sie einen Zeiger auf eine andere Schriftart eintragen. Wenn Sie mit der Standardschrift zufrieden sind, tragen Sie »NIL« ein. »nextText« können Sie ebenfalls mit NIL initialisieren, es sei denn, Sie wollen mehrere IntuiText-Strukturen verketteten.

»iText« endlich ist die Adresse des Strings, der angezeigt werden soll. Sie werden sich natürlich fragen, wozu der ganze Aufwand. Aber diese Strukturen brauchen Sie nur zu initialisieren (das bedeutet mit sinnvollen Werten versorgen). Alles weitere übernimmt Intuition. Aber dafür muß dieser Betriebssystemteil eben auch mit den nötigen Informationen versorgt werden.

Sie sehen vermutlich jetzt schon, daß relativ viel Aufwand getrieben werden muß, um ein Menü aufzubauen. Sie müssen für jeden Menü-Header eine Struktur initialisieren, für Menü-Items sogar je zwei. Außerdem müssen die einzelnen Elemente in sogenannten »verketteten Listen« verbunden werden.

Sicher befürchten Sie schon, daß dazu ein relativ langes Programm entstehen wird? Dann befürchten Sie richtig. Wenn »lang« auch nicht gleichbedeu-

erst sollten Sie sich an folgenden Vorschlag halten:

Wir installieren vier Menü-Header. Den ersten nennen wir »Projekt«. Unter diesem hängen drei Menü-Items:

»Info« gibt allgemeine Informationen zum Programm aus. Hier können Sie Copyright-Vermerke, Ihre Adresse oder auch eine Online-Hilfefunktion unterbringen.

»Neues Projekt« legt für den Projekt-Namen ein Unterverzeichnis an. Zusätzlich werden die Unterverzeichnisse (»obj«, »ref« und »sym«) angelegt.

»Ende« setzt das Ende-Flag des Hauptprogramms (über den Rückgabewert von »VerarbeiteMenu«) und kehrt sofort aus der Prozedur zurück.

Den zweiten Menü-Header nennen wir »Work«. Er erledigt die wichtigste Arbeit. Hier sind die Editor-, Compiler- und Linkeraufrufe zusammengefaßt:

»Edit...« startet den Editor ohne Angabe eines Files.

»Edit Mainfile« startet den Editor mit dem Source-Code des eingestellten Mainfiles.

»Compile...« startet den Compiler ohne Angabe eines Files. Dieses muß wie gewohnt nach dem Start eingegeben werden.

»Comp Mainfile« startet den Compiler mit dem Source-Code Ihres Mainfiles.

Auf die gleiche Weise verfahren wir mit den Items »Link...«

Mainfile selbst definieren und zeigen es in einem eigenen Fenster an. Außerdem sollten die Compiler- und Linkeroptionen fest eingestellt bleiben, damit Sie wegen einer neuen Option nicht jedesmal »per Hand« starten müssen. Sehen Sie sich dazu die Menü-Items unter »Opt« näher an. Die Namen sprechen für sich.

## Unbegrenzt erweiterbar

Der letzte Menü-Header heißt »Dos« und stellt Shortcuts für häufig gebrauchte Dos-Befehle bereit. Vorerst sind hier »dir«, »list« und »info« implementiert.

Das - relativ kurz gehaltene - Definitionsmodul kennen Sie bereits. In der Implementation nutzen wir den Anweisungsblock, um die benötigten Strukturen zu initialisieren. Dadurch sind sie sofort beim Programmstart verfügbar. Der Aufruf der Prozeduren verlangt einige Parameter. Betrachten Sie diese zunächst in den drei Prozeduren (»StructMenu«, »StructItem« und »StructText«). Die Aufrufe wiederholen sich ständig: Zunächst werden die vier Header initialisiert, danach für jedes Item eine IntuiText-Struktur bereitgestellt, die in die darauf initialisierte Menüitem-Struktur eingebunden wird.



dieser Menü-Struktur vorhanden sind, wäre allerdings etwas voreilig. Die Item-Liste kann auch später »eingeklinkt« werden.

Die Prozeduren »StructItem« und »StructText« funktionieren auf ähnliche Weise. Sehen Sie sich die Prozeduren etwas näher an, Sie werden feststellen, daß sie sich von »StructMenu« nur minimal unterscheiden.

An dieser Stelle sollten Sie einen beliebigen und schwer auffindbaren Fehler ausprobieren. Geben Sie das Listing bis zum Ende der letzten Prozedur ein, lassen Sie aber den Anweisungsblock weg. Compilieren Sie es, um sich von der Funktionsfähigkeit zu überzeugen. Wenn Sie alles korrekt eingegeben haben, ändern Sie die Importliste. Sie werden an keiner Stelle des Programms den Namen »MenuItemFlags« entdecken. Streichen Sie ihn also aus der Importliste, denn offensichtlich ist er unnötig. Compilieren Sie das Programm jetzt erneut. Welche Fehlermeldung erhalten Sie und vor allem wo?

Die MenuItemFlags sind eine Aufzählung der möglichen Werte von MenuItemFlagSet. Fehlt diese Aufzählung, so kann der Set nicht mit den vordefinierten Namen initialisiert werden. Dies ist ein häufiger Flüchtigkeitsfehler. Der Name der zugehörigen Flags wird leicht übersehen, da er nirgends auftaucht. Dennoch wird er gebraucht, um die enthaltenen Namen zugänglich zu machen. Fügen Sie ihn also wieder in die Importliste ein.

## Eine Bibliothek als Arbeitstier

Sehen Sie sich jetzt den Anweisungsteil an. Er ist für ein Bibliotheksmodul ungewohnt umfangreich. Er besteht allerdings größtenteils aus sich ständig wiederholenden Wertzuweisungen. Wenn diese auch auf den ersten Blick kompliziert aussehen, decken sie sich doch mit den Prozedurdeklarationen, sind also im Prinzip bereits bekannt. Interessant sind lediglich die übergebenen Werte.

Die erste Zeile weist dem im Definitionsmodul deklarierten Namen »MenuStrip« die Adresse aus einem Aufruf von »StructMenu« zu. Der Menü-Header beginnt an der linken Seite der Menü-Leiste (0 für leftEdge) und ist 80 Punkte breit (80 für width). Er soll den Text »Project« anzeigen. Für die weiteren Header wird der Zeiger jeweils direkt in die vorherige Menü-Struktur geschrieben.

Statt

```
menu[0].nextMenu
```

könnten Sie auch schreiben:

```
MenuStrip^.nextMenu
```

Feld	Typ	Bedeutung
frontPen:	UByte;	Farbnummer für die Zeichenfarbe;
backPen:	UByte;	Farbnummer für die Hintergrundfarbe;
drawMode:	DrawMode;	Zeichenmodus (siehe Text).
leftEdge:	INTEGER;	Abstand des ersten Zeichens von der linken Kante des Textbereichs.
topEdge:	INTEGER;	Abstand der Textoberkante von der Oberkante des Textbereichs.
iTextFont:	TextAttrPtr;	Zeiger auf einen alternativen Font (wird der Standard gewünscht, dann kann Null angegeben werden).
iText:	ADDRESS;	Adresse des Strings.
nextText:	IntuiTextPtr;	Zeiger auf den nächsten Text, falls mehrere verkettet sind.

**Tabelle 8. Mit der »IntuiText«-Struktur kann Intuition komfortabel Texte verarbeiten**

Ebenso könnten Sie sich bis zum letzten Menü durchhangeln und für

```
menu[2].nextMenu
```

schreiben

```
MenuStrip^.nextMenu  
.nextMenu^.nextMenu
```

Aber zum Glück haben wir ein Array aus Menü-Strukturen definiert, mit dem wir diese Zeiger etwas leichter ansprechen können. Das Array ist übrigens nötig, um die Datenstrukturen während des ganzen Programmablaufs zugänglich zu halten.

Nach der Kommentarzeile sehen die Zuweisungen etwas anders aus. Zunächst wird ein Zeiger auf eine IntuiText-Struktur initialisiert. Dieser wird in der nächsten Zeile an die MenuItem-Struktur übergeben. Daher kann er hinterher gefahrlos überschrieben werden. Die Prozeduraufrufe erklären sich fast von selbst, interessant ist allerdings die zweite Zeile des Aufrufs von StructItem. Der Wert für leftEdge ist relativ zur linken Kante des zugehörigen Headers zu sehen, deshalb können ab dem zweiten Menü auch negative Werte übergeben werden. Die Konstante »commWidth« wurde aus Intuition importiert. Sie enthält die Anzahl Punkte, die benötigt werden, um einen Shortcut darzustellen. Sie sollten grundsätzlich diese Konstante verwenden, anstatt selbst den Wert 27 einzusetzen, da solche Werte durchaus in späteren Betriebssystemversionen geän-

dert werden können. Der letzte Parameter ist das Zeichen, welches in Verbindung mit der rechten Amiga-Taste ein angeähltes Menü »simuliert«.

es nicht mehr gebraucht wird, wenn das Menü entfernt wird.

Sie brauchen weder »menus.def« oder »IntuiTool.def« noch »MyShell.mod« neu zu compilieren, linken Sie einfach neu (»m2l MyShell.obj«) und starten Sie das Programm. Sie sehen auf den ersten Blick keine Veränderung, aber wenn Sie die rechte Maustaste drücken, erscheint eine Menüleiste. Sehen Sie sich Ihre Menüs an. Es könnte sein, daß einzelne nicht sichtbar sind. In diesem Fall sollten Sie die Angabe der Position überprüfen. Vermutlich überschneiden sich die Werte mit anderen Menüitems.

Sie haben jetzt ein komplettes Menü installiert. Versuchen Sie eines der Menüs anzuwählen. Was passiert? Richtig, nichts. Es handelt sich dabei keineswegs um einen besonders perfiden Trick von Intuition. Wir haben lediglich bis jetzt noch nichts vorgesehen, um die Menüs auszuwerten.

Bevor wir diese letzten Zeilen implementieren, experimentieren Sie etwas mit dem Einrichten von Menüs. Versuchen Sie auch einem Menü-Item Sub-Items anzuhängen. Sie gehen dabei genauso vor wie bei der Initialisierung der Items (Sie können sogar dieselben Prozeduren verwenden). Experimentieren Sie vor allem mit verschiedenen Werten für die Position. Geben Sie andere Zeichen für die Shortcuts an. Testen Sie auch die Wirkung unterschiedlicher Flags.

Beim Auswerten der Menüs stoßen wir auf ein Problem. Menüs können nicht auf die gleiche Weise ausgewertet werden wie eingegebener Text. Sie werden von Intuition verwaltet und über sogenannte »IntuiMessages« abgefragt. Der eingegebene Text taucht in dieser Nachricht, die von Intuition an unser Fenster gesendet wird, nicht auf. Dieser wird weiterhin von der Prozedur »ReadString« eingelesen. An dieser Stelle stoßen wir auf unser Problem: Wir können beide Arten von Eingaben nur parallel auswerten, wenn sie über dieselbe »Leitung« übertragen werden. Wollen wir dagegen alle Eigenschaften unseres Fensters beibehalten (sowohl die AmigaDOS- als auch die Intuition-Eigenschaften), dann müssen wir Kompromisse eingehen. Entweder wir werten die Menüs sauber aus, dann haben wir Probleme mit ReadString, weil die Shell dann so lange auf eine Nachricht von Intuition wartet, bis ein Menü betätigt wurde. Umgekehrt können wir die Ein-

Wenn Sie alle Zeilen eingegeben haben, speichern Sie das Programm unter dem Namen »menus.mod« und compilieren Sie es.

Um die Menüs in Ihr Programm einzubauen, erweitern Sie nur noch IntuiTool um die beiden Prozeduren InstallMenu und RemoveMenu, die bisher nur aus einer Leerprozedur bestehen. Fügen Sie dazu die beiden Prozeduren aus Listing 12 in »IntuiTool.mod« ein und fügen Sie dem Anweisungsteil zwei Zeilen hinzu:

```
MenuStrip := Menues.  
MenuStrip;  
minWin := OpenMinFenster  
(200,0,390,12);
```

Die erste Zeile überträgt das in Menues definierte Menü in die Variable »MenuStrip«. Die zweite Zeile öffnet ein Fenster zur Ausgabe für den Menüpunkt »MainFile«. Compilieren Sie jetzt auch diese Bibliothek neu. Die Prozedur »ModifyDCMP« verändert den »Intuition Direct Communications Message Port«. Dieser bildet den Filter für die Nachrichten, die Sie für Ihr Fenster zulassen wollen. Mit »menuPick« geben wir an, daß nur Nachrichten angenommen werden, wenn diese sich auf ein Menü beziehen. Die Prozeduren »SetMenuStrip« und »ClearMenuStrip« verbinden einen Menü-Zeiger mit einem Fenster. »CloseFenster« schließt das Fenster, das wir vorher zum Anzeigen geöffnet haben. Da dies nur für einen Menüpunkt eingerichtet wird, gehen wir davon aus, daß



gaben über AmigaDOS vorrangig abfragen, dann reagieren jedoch die Menüs nicht wie gewohnt.

Natürlich läßt sich dieses Problem auch »sauber« lösen, aber vorerst wollen wir – im Interesse eines überschaubaren Programms – auf die Abfrage der sogenannten »RawKeys« verzichten. Der Kompromiß, den wir eingehen, ist erträglich:

## Saubere Lösung mit »RawKeys«

Ein Menü wird erst dann bearbeitet, wenn ReadString abgeschlossen ist. Im Klartext heißt

das, jede Menüabfrage muß durch <RETURN> bestätigt werden. Dies läßt sich aber verkraften, da wir für jedes Menü Shortcuts definiert haben, so daß Sie die Tastatur nicht verlassen müssen.

Bauen Sie also die Prozedur »VerarbeiteMenu« soweit aus, daß sie aussieht wie die in Listing 13 abgedruckte. Danach fügen Sie folgende Zeilen in »MyShell.mod« ein:

```
nachricht := GetMsg
(iFenster^.userPort);
WHILE nachricht #
NIL DO
class := nachricht
.class;
```

```
code := nachricht
.code;
ReplyMsg (nachricht);
IF (class=pick) AND
(code # menuNull) THEN
Ende := Verarbeite
Menu (code, Fenster)
END;
nachricht := GetMsg
(iFenster^.userPort)
END;
```

Diese Zeilen sollen innerhalb der Repeat-Schleife zwischen WriteString und ReadString stehen. Die Funktion »GetMsg« liest die Nachrichten, die an unser Fenster geschickt werden. Liegt keine Nachricht an (»nach-

richt = NIL«), so werden die Zeilen, die als Reaktion darauf vorgesehen sind, einfach übersprungen. Ist dagegen eine Nachricht vorhanden, so werden die interessantesten Informationen zunächst in zwei Variablen zwischengespeichert (»class« und »code«), damit die Nachricht beantwortet werden kann. »ReplyMsg« gibt den von der Nachricht belegten Speicher frei. Damit verhindern Sie, daß Sie zweimal dieselbe Nachricht erhalten. Danach wird zunächst überprüft, ob class vom Typ Menü ist (zwar ist im Moment nichts anderes möglich, aber vielleicht wollen Sie ja noch andere Nachrichten

IMPLEMENTATION MODULE Menues;

```
FROM SYSTEM IMPORT ADR, ADDRESS, BITSET;
FROM Intuition IMPORT WindowPtr, MenuPtr, MenuItemPtr, Menu,
IntuiMessagePtr, IDCMPFlags, IDCMPFlagSet, checkWidth,
commWidth, IntuiTextPtr, ClearMenuStrip, SetMenuStrip,
MenuItemFlags, MenuItemFlagSet, IntuiText, MenuItem;
FROM Graphics IMPORT jam1;
```

```
PROCEDURE StructMenu (VAR m: Menu; l,w: INTEGER;
text: ADDRESS): MenuPtr;
BEGIN
WITH m DO
nextMenu := NIL;
leftEdge := l; topEdge := 0; width := w; height := 0;
flags := BITSET[0]; menuName := text; firstItem := NIL
END;
RETURN ADR(m)
END StructMenu;
```

```
PROCEDURE StructItem (VAR i: MenuItem; l,t,w,h: INTEGER;
text: ADDRESS; c: CHAR): MenuItemPtr;
BEGIN
WITH i DO
nextItem := NIL;
leftEdge := l; topEdge := t; width := w; height := h;
flags := MenuItemFlagSet{itemEnabled, itemText, highComp,
commSeq};
itemFill := text; selectFill := 0;
command := c
END;
RETURN ADR(i)
END StructItem;
```

```
PROCEDURE StructIText (VAR it: IntuiText; text: ADDRESS;
l,t: INTEGER): IntuiTextPtr;
BEGIN
WITH it DO
frontPen := 2; backPen := 0;
drawMode := jam1; iTextFont := NIL;
leftEdge := l+2; topEdge := t;
iText := text; nextText := NIL
END;
RETURN ADR(it)
END StructIText;
```

```
VAR menu: ARRAY [0..3] OF Menu;
ProjektItems: ARRAY [0..3] OF MenuItem;
ProjektText: ARRAY [0..3] OF IntuiText;
WorkItems: ARRAY [0..6] OF MenuItem;
WorkText: ARRAY [0..6] OF IntuiText;
OptItems: ARRAY [0..2] OF MenuItem;
OptText: ARRAY [0..2] OF IntuiText;
DosItems: ARRAY [0..2] OF MenuItem;
DosText: ARRAY [0..2] OF IntuiText;
t: IntuiTextPtr;
```

```
BEGIN
MenuStrip := StructMenu (menu[0],0,80,ADR("Project"));
menu[0].nextMenu := StructMenu (menu[1],80,50,ADR("Work"));
```

```
menu[1].nextMenu := StructMenu (menu[2],130,40,ADR("Opt"));
menu[2].nextMenu := StructMenu (menu[3],170,40,ADR("Dos"));
(* Items initialisieren *)
t := StructIText (ProjektText[0],ADR("Info"),0,0);
menu[0].firstItem := StructItem (ProjektItems[0],
0,0,120+commWidth,9,t,"i");
t := StructIText (ProjektText[1],ADR("sYsinFo"),0,0);
ProjektItems[0].nextItem := StructItem (ProjektItems[1],
0,10,120+commWidth,9,t,"y");
t := StructIText (ProjektText[2],ADR("Neues Projekt"),0,0);
ProjektItems[1].nextItem := StructItem (ProjektItems[2],
0,20,120+commWidth,9,t,"n");
t := StructIText (ProjektText[3],ADR("Ende"),0,0);
ProjektItems[2].nextItem := StructItem (ProjektItems[3],
0,30,120+commWidth,9,t,"q");
t := StructIText (WorkText[0],ADR("Edit ..."),0,0);
menu[1].firstItem := StructItem (WorkItems[0],
-10,0,130+commWidth,9,t,"e");
t := StructIText (WorkText[1],ADR("Edit Mainfile"),0,0);
WorkItems[0].nextItem := StructItem (WorkItems[1],
-10,10,130+commWidth,9,t,"1");
t := StructIText (WorkText[2],ADR("Compiliere ..."),0,0);
WorkItems[1].nextItem := StructItem (WorkItems[2],
-10,20,130+commWidth,9,t,"c");
t := StructIText (WorkText[3],ADR("Comp Mainfile"),0,0);
WorkItems[2].nextItem := StructItem (WorkItems[3],
-10,30,130+commWidth,9,t,"2");
t := StructIText (WorkText[4],ADR("Link ..."),0,0);
WorkItems[3].nextItem := StructItem (WorkItems[4],
-10,40,130+commWidth,9,t,"l");
t := StructIText (WorkText[5],ADR("Link Mainfile"),0,0);
WorkItems[4].nextItem := StructItem (WorkItems[5],
-10,50,130+commWidth,9,t,"3");
t := StructIText (WorkText[6],ADR("Testlauf"),0,0);
WorkItems[5].nextItem := StructItem (WorkItems[6],
-10,60,130+commWidth,9,t,"t");
t := StructIText (OptText[0],ADR("Change Mainfile"),0,0);
menu[2].firstItem := StructItem (OptItems[0],
-15,0,135+commWidth,9,t,"m");
t := StructIText (OptText[1],ADR("Comp-Optionen"),0,0);
OptItems[0].nextItem := StructItem (OptItems[1],
-15,10,135+commWidth,9,t,"k");
t := StructIText (OptText[2],ADR("Link-Optionen"),0,0);
OptItems[1].nextItem := StructItem (OptItems[2],
-15,10,135+commWidth,9,t,"o");
t := StructIText (DosText[0],ADR("Dir"),0,0);
menu[3].firstItem := StructItem (DosItems[0],
0,0,90+commWidth,9,t,"d");
t := StructIText (DosText[1],ADR("List"),0,0);
DosItems[0].nextItem := StructItem (DosItems[1],
0,10,90+commWidth,9,t,"s");
t := StructIText (DosText[2],ADR("Info"),0,0);
DosItems[1].nextItem := StructItem (DosItems[2],
0,20,90+commWidth,9,t,"v");
END Menues.
```

Listing 11. »Menues.mod« hat einen ungewöhnlich langen Anweisungsteil. Dieser verkettet den »MenuStrip«.



zulassen) und ob code ungleich »menuNull« ist. menuNull erhalten wir, wenn die Menüleiste aktiviert wurde, ohne ein Menü zu wählen. Sind beide Bedingungen erfüllt (»AND«), dann wird die Variable code zur weiteren Verarbeitung an die Prozedur VerarbeiteMenu übergeben, die einen Wert zurückgibt, der TRUE enthält, wenn im Projekt-Menü »Ende« angewählt wurde. Die letzte Anweisung in der While-Schleife liest nochmals denselben Port aus, damit die Schleife, falls mehrere Menüs gewählt waren, von vorne beginnt.

Die Prozedur »VerarbeiteMenu« enthält einiges Neue. Fügen Sie zunächst die beiden Importzeilen und die Variablen-deklarationen aus Listing 13 an die Import- beziehungsweise Variablenliste von IntuiTool an. Außerdem sollten Sie die Importliste bei »Intuition« um den Namen »WindowToFront« erweitern. Bauen Sie darauf die beiden Prozeduren kurz vor dem Beginn des Anweisungsteils ein.

Die Prozedur »InitStrings« wird von VerarbeiteMenu aufgerufen, wenn der Benutzer ein neues MainFile einstellt. Wir gehen davon aus, daß für dieses File auch neue Compiler- und Linkeroptionen gelten. Daher werden auch diese Strings neu initialisiert.

Die Prozedur VerarbeiteMenu deklariert zunächst einige lokale Variablen. Im Anweisungsteil wird dann die Variable code zerlegt, um an die Nummer des gewählten Menüs und des Items zu kommen. Diese werden mit einer »CASE«-Anweisung ausgewertet. CASE läßt im Gegensatz zu IF das Auswerten von Mehrfach-Verzweigungen zu. Deshalb eignet sich CASE besser als IF für die Auswertung unserer Nachricht. Nehmen wir die äußere Klammer als Beispiel. CASE fragt hier die Variable »menuNum« nach ihrem Inhalt ab. Diese ist vom Typ CARDINAL, sie kann also positive, ganzzahlige Werte zwischen 0 und 65535 aufnehmen. Wir unterscheiden vier Fälle: menuNum kann Werte zwischen 0 und 3 annehmen, daher fragen wir nur diese vier Fälle ab.

Ist der Variableninhalt 0 (dies bedeutet, daß das linke Menü angewählt wurde), so wird mit einer weiteren CASE-Anweisung geprüft, welchen Inhalt die Variable itemNum hat. Auf gleiche Weise wird verfahren, wenn eines der anderen Menüs angewählt wurde. Hinter dem jeweiligen Item-Zweig werden

die Anweisungen eingefügt, die als Reaktion auf das jeweilige Menü eingeplant sind.

Die Anweisungen innerhalb der CASE-Verzweigungen dürften Ihnen größtenteils vertraut sein. Die einzigen unbekannteren werden von dem Bibliotheksmodul »MenuTool« importiert. Interessant sind daneben höchstens die Stringverkettungen:

Wird der Menüpunkt »MainFile« gewählt, so werden zu-

den in der gängigen Modul-Literatur häufig eine andere Darstellungsweise finden (mit dem senkrechten Strich am Ende jedes Falles). Wir ziehen diese Form jedoch vor, da wir sie für übersichtlicher halten. Selbstverständlich können Sie Ihre senkrechten Striche aber auch an das Ende der Zeilen setzen.

Wenn Sie diese Prozedur eingefügt sowie neu compiliert

```
PROCEDURE InstallMenu (iF: WindowPtr; m: MenuPtr);
BEGIN
  IF SetMenuStrip (iF, m) THEN
    ModifyIDCMP (iF, IDCMPFlagSet{menuPick})
  END
END InstallMenu;

PROCEDURE RemoveMenu (iF: WindowPtr);
BEGIN
  CloseIFenster (minWin);
  ModifyIDCMP (iF, IDCMPFlagSet{});
  ClearMenuStrip (iF)
END RemoveMenu;
```

#### Listing 12. Mit diesen Prozeduren erweitern Sie »IntuiTool.mod«, um die Menüs zu installieren

nächst alle Strings mit Hilfe der Funktion InitStrings initialisiert. Diese setzt »MainText«, »CompOpts« und »LinkOpts« auf LeerStrings. »EdText«, »CompText« und »LinkText« werden mit dem Namen des zugehörigen Programms (beispielsweise »m2-emacs«) und der üblichen Endung initialisiert. Diese wird

## CASE - Das etwas andere IF

einfach an den Namen angehängt, damit der Name des Arbeitsfiles (»MainText«) nur noch per »Insert«-Funktion einkopiert werden muß. Dies geschieht in den folgenden Zeilen, nachdem mit der Funktion »GetString« der Text eingelesen wurde.

Danach wird mit »DisplayString« der Name in einem Fenster angezeigt und dieses Fenster (das beim Programmstart geöffnet wurde) in den Vordergrund geholt (mit »WindowToFront«). Hier ist die CASE-Anweisung beendet und das Programm verzweigt zur END-Zeile.

Bitte beachten Sie die senkrechten Striche zwischen den einzelnen Fallunterscheidungen der CASE-Anweisungen. Diese werden benutzt, um die einzelnen Fälle voneinander zu trennen. Das Semikolon würde hier nicht ausreichen, da mehrere Anweisungen bei jedem Einzelfall enthalten sein können. Die END-Anweisung ist ebenfalls bereits belegt. Daher mußte ein neues Zeichen zur Trennung erhalten. Sie wer-

den gelinkt haben, funktionieren die meisten Menüs problemlos. Sie können die Dos-Funktionen aus dem Menü oder per Shortcut wählen, den Editor und den Compiler starten oder per Shortcut die Shell beenden. Um allerdings die entscheidenden Verbesserungen zu nutzen (die Einstellung des Mainfiles im Menü und das automatische Einfügen dieses Namens), fehlen noch die Funktionen, die aus MenuTool importiert werden.

Das Definitionsmodul hatten wir bereits festgelegt. Sehen Sie sich jetzt das Implementationsmodul (Listing 14) an. Es legt eine Konstante fest (»InfoText«), die als Platzhalter fungiert für Texte, die Sie anzeigen möchten, wenn der Benutzer das Info-Menü wählt. Tragen Sie hier Ihren Namen und Ihre Anschrift ein oder einen Copyright-Vermerk. Oder wie wäre es mit einer Auskunft über den aktuellen Systemzustand? Die Dos- und die Exec-Library geben hier einiges an Informationen her.

Die Prozedur »IText« arbeitet wie »StructIText« aus IntuiTool. Sie initialisiert eine IntuiText-Struktur, die in einem Intuition-Fenster angezeigt werden soll (hier mit der Prozedur »PrintIText«). Mit der Prozedur »Informiere« können Sie in einem beliebigen Intuition-Fenster Texte anzeigen. Sie können dafür das Fenster der Shell verwenden oder ein beliebiges anderes, dessen Zeiger Sie im Aufruf übergeben. In dieser Version wird einfach nur der Text aus

der Konstante InfoText angezeigt. Sie können die Anzeige aber auch von Bedingungen abhängig machen und entsprechend verschiedene Texte verwenden.

Die Prozedur »MakeProject« ist noch nicht implementiert, sie soll in einer der Aufgaben weiter unten entwickelt werden. An »GetString« wird eine Stringvariable übergeben, die an den Aufruf gebunden ist. Die Prozedur öffnet ein eigenes Dos-Fenster, das für die Eingabe mit ReadString verwendet wird. Ebenso gut hätten Sie natürlich auch einen Requester verwenden können, was allerdings etwas mehr Arbeit bei der Auswertung bedeuten würde (allerdings auch mehr Komfort für den Benutzer).

»DisplayString« wird verwendet, um das MainFile anzuzeigen. Dazu werden zunächst dem übergebenen String die Zeichen »Main: « vorangestellt (mit Insert). Danach wird - wie bei Informiere - der String in einen IntuiText umgewandelt und im übergebenen Fenster angezeigt. Wenn Sie hier ein anderes Fenster verwenden wollen, steht dem ebenfalls nichts im Wege.

Compilieren Sie diese letzte Bibliothek. Linken Sie das Hauptprogramm nochmals und starten Sie Ihre Shell. Drücken Sie gleichzeitig die rechte Amigataste und »i«. Was passiert? Nichts. Drücken Sie jetzt <RETURN> und beachten Sie das Fenster, das sich in der Mitte des Bildschirms öffnet. Es zeigt unseren InfoText an und schließt sich nach einigen Sekunden von selbst wieder.

Drücken Sie die rechte Amigataste und »m«. Sie sehen ein Fenster im linken oberen Viertel des Bildschirms. Es fordert Sie mit der Titelzeile auf, einen Text einzugeben. Geben Sie »MyShell« (und <RETURN>) an. Das Fenster schließt sich und Sie sehen rechts oben ein neues Fenster mit dem Inhalt »Main: MyShell«. Stellen Sie jetzt für den Compiler und für den Linker je eine Option ein. Wählen Sie dazu das Menü »Comp-Optionen« (entweder mit der rechten Maustaste oder mit <rechte Amiga-k>). Sie sehen das gleiche Fenster wie beim Einstellen des Mainfiles. Tragen Sie als Text »i« ein (hinter dem i muß eine Leerstelle sein). Das gleiche machen Sie mit den Linkeroptionen. Von jetzt an produzieren der Compiler und auch der Linker ein Icon für jede erzeugte Datei. Die Optionen werden mit Insert in den Text eingefügt, der an die



Execute-Funktion übergeben wird, wenn Sie den Menüpunkt »Comp Mainfile« oder »Link Mainfile« wählen. Wählen Sie eine der beiden Funktionen an, der Compiler beziehungsweise Linker wird automatisch mit Ihren Optionen und Ihrem Mainfile gestartet und bricht danach ab. Die Shortcuts zu den einzelnen Menüpunkten finden Sie hinter den Texten eingetragen, wenn Sie die Menüs anwählen.

Natürlich ist diese Shell noch lange nicht das Optimale an Benutzerfreundlichkeit. Sie könnten noch viele Funktionen einbauen, angefangen von Funktionstastenbelegung über alternative Namen für Kommandos oder Editiermöglichkeiten mit den Cursortasten bis hin zu Resident-Funktionen oder einer Help-Funktion. Aber damit wären Sie schnell an der Grenze der Leistungsfähigkeit des PD-Compilers. Dieser kann nur bis zu einer Grenze von 2 KByte Objektcode compilieren. Außerdem können Sie nicht mehr als fünf Bibliotheken in ein Modul einbinden.

Die einfachste Möglichkeit zur Optimierung der Shell führt über die RawKey-Messages. Sie können ein Fenster als »RAW:« anstelle von »CON:« öffnen und erhalten dann automatisch Informationen über alle Arten von Eingaben einschließlich Funktionstasten, Cursortasten und Ziffernblock. Allerdings müssen Sie diese dann selbst auswerten und anzei-

gen. Der Komfort des Con-Fensters geht Ihnen dabei verloren.

## Besser geht's mit WaitPort

Eine andere Möglichkeit führt über Intuition-Fenster. Auch hier können Sie RawKey-Messages auswerten, indem Sie das entsprechende IDCMP-Flag setzen (»rawKey«). Bei Intuition haben Sie zusätzlich den Vorteil, daß Ihnen alle Möglichkeiten von Gadgets, Requestern und Menüs offenstehen. Außerdem können Sie dann alle Messages »sauber« mit der »WaitPort«-Funktion abfragen. Diese versetzt Ihr Programm in den Wartezustand, solange der Benutzer das Fenster nicht anspricht. Erst wenn eine der erwarteten Nachrichten eintrifft, wird das Programm wieder »geweckt« und kann darauf reagieren. In der Zwischenzeit wird keine Rechenzeit an den »schlafenden« Task verschwendet. »WaitPort« finden Sie in der Exec-Library. Die Prozedur benötigt nur einen Parameter, den Message-Port, in dem die Nachricht eintreffen soll. Dieser wird von Intuition angelegt, wenn mindestens ein IDCMP-Flag gesetzt ist. Wir haben ihn bereits für »GetMsg« verwendet. Er liegt in der Window-Struktur und heißt »userPort«.

Bevor Sie sich jetzt in eigene Experimente stürzen, sollten Sie noch folgende Aufgaben

bearbeiten, um das Gelernte zu vertiefen:

- Implementieren Sie die Prozedur MakeDir. Diese soll beim Anwählen des Menüpunkts »Neues Projekt« ein Fenster öffnen, in dem der Name des Projekts eingegeben wird. Verwenden Sie dazu die Prozedur »GetString«. Legen Sie unter diesem Namen ein Directory im aktuellen Verzeichnis an, sowie drei Unterdirectories für die obj-, sym- und ref-Dateien (in der Dos-Bibliothek gibt es eine Prozedur »CreateDir«). Testen Sie vorher, ob ein gleicher Name schon existiert. Wenn ja, geben Sie eine Fehlermeldung aus, wenn nein werden die Verzeichnisse angelegt.

- Schreiben Sie eine Prozedur, die den Source-Code des Mainfile untersucht. Sie soll feststellen, ob die Importliste Files enthält, die im eingestellten Suchpfad für die Symbol-Dateien nicht aufzufinden sind (in »m2path« aus dem »S:«-Ordner). Wenn dies der Fall ist, fehlen Definitionsmodule. Gestalten Sie in diesem Fall den Compiler-Aufruf so um, daß diese - falls vorhanden - automatisch vor dem Mainfile compiliert werden (Sie können dem Compiler mehrere Namen durch Leerstellen getrennt übergeben). Fehlt auch der Source-Code dieser Definitionsmodule, dann soll in den Editor verzweigt werden (mit dem Namen des fehlenden Definitionsmoduls als neues File).

- Bauen Sie die gleiche Abfrage vor den Linkerlauf, so daß bei fehlenden Objekt-Dateien automatisch entweder der Compiler gestartet wird oder - wenn noch kein Source-Code vorhanden ist - der Editor. Das gleiche können Sie beim Testlauf machen. Allerdings müssen Sie jetzt immer mehr bedenken: Ist das Hauptprogramm schon compiliert und gelinkt? Enthält es noch nicht compilierte Bibliotheken? Sind in diesen - falls im Sourcecode vorhanden - uncompierte Bibliotheken enthalten? Dabei reicht es nicht, zu testen, ob die Bibliotheken compiliert wurden. Sie müssen auch testen, ob die Schnittstellen nach der letzten Veränderung der Implementation oder des Hauptprogramms verändert worden sind.
- Schreiben Sie ein Programm, das ein Intuition-Fenster öffnet und mit Menüs versieht. Diese sollen während des Programmlaufs abgefragt werden. Warten Sie mit der Exec-Funktion »WaitPort« auf eine Nachricht von Intuition, bevor Sie die Menüs auswerten. Das Fenster soll alle Systemgadgets erhalten. Bevor das Fenster geschlossen wird, soll eine Meldung an den Benutzer ausgegeben werden (dies kann geschehen, um ihn zu informieren, daß er seine Arbeit noch nicht gespeichert hat).

Wenn Sie alle Aufgaben gelöst haben, fällt es Ihnen sicher nicht schwer, sich neue zu stel-

```

FROM MenuTool IMPORT Informiere, MakeProject, GetString,
    DisplayString;
FROM StringTool IMPORT Insert, String, Copy, Length;

VAR minWin: WindowPtr;
    MainText, EdText, CompText, LinkText: String;
    CompOpts, LinkOpts: String;

PROCEDURE InitStrings;
BEGIN
    EdText := "m2emacs .mod"; EdText[12]:=0C;
    MainText[0] := 0C;
    CompText := "m2c .mod"; CompText[8] := 0C;
    LinkText := "m2l obj/.obj"; LinkText[12] := 0C;
    CompOpts[0] := 0C; LinkOpts[0] := 0C
END InitStrings;

PROCEDURE VerarbeiteMenu (code: CARDINAL; fh: FileHandlePtr):
BOOLEAN;
VAR menuNum, itemNum: CARDINAL;
    f: WindowPtr;
    ok: INTEGER;
BEGIN
    menuNum := code MOD 32; itemNum := (code DIV 32) MOD 32;
    CASE menuNum OF
        | 0: CASE itemNum OF (* Project *)
            | 0: f := OpenMinFenster (30, 90, 580, 50);
                Informiere (f);
                CloseWindow (f);
            | 1: MakeProject;
            | 2: RETURN TRUE
        END; (* 0: CASE itemNum *)
        | 1: CASE itemNum OF (* Work *)
            | 0: ok := Execute (ADR("m2emacs"),0,fh);
            | 1: ok := Execute (ADR(EdText),0,fh);
            | 2: ok := Execute (ADR("m2c"),fh,fh);
            | 3: ok := Execute (ADR(CompText),0,fh);
            | 4: ok := Execute (ADR("m2l"),fh,fh);
            | 5: ok := Execute (ADR(LinkText),0,fh);
            | 6: ok := Execute (ADR(MainText),0,fh)
        END; (* 1: CASE ... *)
        | 2: CASE itemNum OF (* Opt *)
            | 0: InitStrings; GetString (MainText);
                Insert (MainText, 8, EdText);
                Insert (MainText, 4, CompText);
                Insert (MainText, 8, LinkText);
                DisplayString (minWin, MainText);
                WindowToFront (minWin);
            | 1: GetString (CompOpts);
                Insert (CompOpts, 4, CompText);
            | 2: GetString (LinkOpts);
                Insert (LinkOpts, 4, LinkText);
        END; (* 2: CASE ... *)
        | 3: CASE itemNum OF (* Dos *)
            | 0: ok := Execute (ADR("dir"),0,fh);
            | 1: ok := Execute (ADR("list"),0,fh);
            | 2: ok := Execute (ADR("info"),0,fh)
        END (* 3: CASE ... *)
    END; (* CASE menuNum *)
    RETURN FALSE
END VerarbeiteMenu;

```

**Listing 13. Menüs werden am einfachsten mit der »CASE«-Anweisung ausgewertet. Die Prozedur »VerarbeiteMenu« fügen Sie in »IntuiTool.mod« ein.**



len. Wir freuen uns schon auf die nächsten Programmeinsendungen in Modula.

Natürlich war es auf diesen wenigen Seiten nicht möglich, das Thema abschließend zu behandeln. Sicher sind bei Ihnen noch viele Fragen offen. Ebenso kann nicht allzu tief auf die Programmierung des Amiga eingegangen werden. Beide Themenbereiche sollten nur angerissen werden, um Sie für diese Sprache zu interessieren.

Wenn Sie mehr über Modula wissen wollen, gibt es dazu einige interessante Bücher. Eine Auswahl:

- Edward J. Joyce, A Seafarer's Guide and Shipyard Manual vom Addison-Wesley-Verlag, ist eine lockere und verständliche Einführung. Es ist das beste Buch für den Einsteiger.

- Dal Cin/Lutz/Risse, Programmieren in Modula-2, erschienen im Teubner Verlag, erklärt anhand vieler Beispiele, wie man in Modula effektiv programmiert. Mit etwas Grundwissen ist es eine Fundgrube für jeden Programmierer.

- Niklaus Wirth, Programmie-

rung in Modula-2, erschienen im Springer-Verlag, gilt als das Standardwerk zu Modula. Es eignet sich als Nachschlagewerk. Sie finden alle Informationen zum Standard-Sprachumfang von Modula.

- Niklaus Wirth, Algorithmen und Datenstrukturen, erschienen im Springer-Verlag, ist eine Sammlung von Beispielprogrammen. Wenn Sie auf der Suche nach einem Algorithmus zur Implementation und Behandlung von verketteten Listen oder anderen Datenstrukturen sind, sollten Sie als erstes dieses Buch aufschlagen. Es erspart Ihnen das Wühlen in Bergen Literatur, die das gleiche leisten, dafür aber wesentlich mehr Seiten brauchen.

## Amiga-Extra im Herbst

Daneben wird voraussichtlich im Herbst ein Extra-Heft im Markt & Technik Verlag erscheinen, das sich speziell mit der Programmierung des Amiga in Modula beschäftigt. Auch im letzten Sonderheft finden Sie

ausführliche Informationen zur Programmierung des Amiga. Mit oberflächlicher Kenntnis von C und einiger Erfahrung in Modula sollten Sie in der Lage sein, alle Beispiele zu übertragen. Natürlich werden Sie sehr bald an die Grenzen Ihres PD-Compilers stoßen. Bereits die in diesem Workshop gezeigten Beispiele sind nicht weit entfernt. Um dies zu demonstrieren, fügen Sie im Bibliotheksmodul »IntuiTool.mod« einige Zeilen Sourcecode hinzu. Wenn Sie das Programm neu compilieren, erhalten Sie eine bisher unbekanntes Fehlermeldung: »Program too long«. Der PD-Compiler kann nur bis zu einer Länge von etwas über 2 KByte Objektcode compilieren. Ebenso ist es unmöglich, mehr als fünf Bibliotheksmodule zu importieren.

Wenn Sie Geschmack an Modula gefunden haben und Programme entwickeln wollen, die über diese Beschränkungen hinaus gehen, sollten Sie sich die professionelle Version dieses Compilers kaufen. Diese enthält außerdem zusätzliche

Bibliotheken, die die Arbeit erleichtern, ein sehr gutes Handbuch und auf Wunsch eine Shell, Source-Level-Debugger und einige Utilities. Sie erhalten die professionelle Version von M2Amiga für etwa 350 Mark bei der Schweizer Firma A+L AG, Im Spaeten 23, CH-8906 Bonstetten/ZH, Schweiz. Daneben dürfte er auch bei den meisten Software-Händlern erhältlich sein. Einige Interessante Anregungen finden Sie auf den PD-Disketten der »Amok«-Reihe (siehe Seite 24).

Aber auch wenn Sie weiterhin in Basic, C oder Assembler programmieren, hat sich die durch Modula gewonnene Erfahrung sicher gelohnt. Modula ist eine Sprache, die in Lesbarkeit und Strukturierung noch lange ihresgleichen sucht. Ein Modula-Programm in eine andere Sprache zu übertragen, dürfte nie Probleme bereiten.

Eines sollten Sie aus diesem Workshop gelernt haben: Auch wenn Sie beim Programmieren zu Spaghetti à la Basic neigen, machen Sie die Rechnung nie ohne den Wirth! so

```

DEFINITION MODULE MenuTool;

FROM Intuition IMPORT WindowPtr;
FROM StringTool IMPORT String;

PROCEDURE Informiere (w: WindowPtr; sys: BOOLEAN);
PROCEDURE MakeProject;
PROCEDURE GetString (VAR s:String);
PROCEDURE DisplayString (w: WindowPtr; s: String);

END MenuTool.

IMPLEMENTATION MODULE MenuTool;

FROM SYSTEM IMPORT ADR, ADDRESS;
FROM Intuition IMPORT WindowPtr, IntuiTextPtr, IntuiText,
PrintIText;
FROM Dos IMPORT FileHandlePtr, Open, Close, Delay, newFile;
FROM StringTool IMPORT String, Insert, Copy, Length;
FROM DosTool IMPORT WriteString, ReadString, WriteLn;
FROM Graphics IMPORT jam1, SetAPen;

CONST InfoText = "Hier können Sie einen beliebigen Text anzeigen";
      SysInfoText = "Noch nicht implementiert!!";

PROCEDURE IText (VAR it: IntuiText; text: ADDRESS;
1,t: INTEGER): IntuiTextPtr;
BEGIN
  WITH it DO
    frontPen := 2; backPen := 0;
    drawMode := jam1; iTextFont := NIL;
    leftEdge := 1+2; topEdge := t;
    iText := text; nextText := NIL
  END;
  RETURN ADR(it)
END IText;

PROCEDURE Informiere (w: WindowPtr; sys: BOOLEAN);
VAR t: IntuiTextPtr;
    it: IntuiText;
BEGIN
  IF sys THEN

```

```

      t := IText(it, ADR(SysInfoText), 0, 0)
    ELSE
      t := IText(it, ADR(InfoText), 0, 0)
    END;
    SetAPen (w.rPort, 1);
    PrintIText (w.rPort,t,10,20);
    Delay (180);
  END Informiere;

PROCEDURE MakeProject;
END MakeProject;

PROCEDURE GetString (VAR s:String);
VAR fh: FileHandlePtr;
BEGIN
  fh:=Open(ADR("CON:30/30/290/35/Geben Sie den Programmnamen ein:"),
newFile);
  WriteString (fh, " * ");
  ReadString (fh, s);
  Close (fh)
END GetString;

PROCEDURE DisplayString (w: WindowPtr; s: String);
VAR yPos: INTEGER;
    t: IntuiTextPtr; it: IntuiText;
BEGIN
  Main := "Main: "; Main[6] := 0C;
  Insert (Main,0,s);
  IF (s[0] = "M") THEN
    yPos := 0
  ELSE
    yPos := 10
  END;
  t:= IText (it, ADR(s),0,0);
  PrintIText (w.rPort,t,3,yPos)
END DisplayString;

END MenuTool.

```

Listing 14. Die Bibliothek »MenuTool« unterstützt Sie bei der Auswertung der Menüs



**Ein Chamäleon paßt sich seiner Umgebung perfekt an. Diese herausragende Eigenschaft zeichnet auch den »DME« aus. Wie Sie diesen PD-Editor perfekt auf Ihre Bedürfnisse abstimmen, erfahren Sie in diesem Workshop.**

# Das Chamäleon

Von Heiko Schlichting

**E**ditoren sind ein heikles Thema. Wer bereit ist, viel Geld zu investieren, der findet sicher einen kommerziellen Editor mit erstaunlichen Fähigkeiten. Aber sobald Spezialanforderungen gewünscht werden, fühlt man sich bald auch von den teuersten Produkten alleingelassen. Da ist es schon das höchste der Gefühle, wenn man die Funktionstasten mit Text belegen darf. Oder eine Schnittstelle zu Amiga-DOS geboten wird, so daß man nicht wegen jedem »list« oder »cc +l myprog« den Editor verlassen muß. Gerade Programmierer wünschen sich aber nicht irgendeinen tollen Editor, der die tausendste unwichtige Funktion eingebaut hat (und dafür oft über 100 KByte belegt), sondern eben »ihren« Editor. Der eine schwört auf MicroEmacs-kompatible Tastenbelegung, der andere auf Turbo-Pascal-ähnliche Compileraufrufe. Der dritte will die Fehlermeldungen seines Compilers im Editor angezeigt haben, und wieder andere legen Wert auf besonders ausgefeilte Suchen- und Ersetzen-Funktionen. All diesen Wünschen mit einem Programm gerecht zu werden, ist unmöglich. Es sei denn, man hat eine gute Idee – wie Matthew Dillon. Dieser bekannte PD-Autor (siehe Kasten) schrieb einen Editor mit allen genannten Fähigkeiten.

Wenn Sie jetzt vermuten, daß ein so leistungsfähiges Programm mindestens einige 100 KByte Speicher belegen und ebenso viele Mark kosten müßte, haben Sie falsch getippt. Der DME ist Public Domain und somit umsonst (ausgenommen eventuell anfallende Kopier- und Versandgebühren für die Diskette). Auch entpuppt sich der DME nicht als »Speicher-

vernichter«, er belegt in der Grundversion nur 40 KByte. Wenn Sie jetzt erschrocken irgendeinen Trick vermuten, liegen Sie richtig. Natürlich lassen sich in 40 KByte nicht alle genannten Funktionen implementieren. Der DME ist dabei auf die Mitarbeit des Benutzers angewiesen. Er kann immer genau das, was Sie ihm beibringen. Wenn Sie jetzt entrüstet einwenden, dann könnten Sie auch gleich Ihren eigenen Editor programmieren, warten Sie die folgenden Erklärungen ab. Den DME auf die eigenen Bedürfnisse einzustellen, ist unglaublich einfach.

Wir beziehen uns im folgenden Workshop auf die Version 1.35 des DME. Wenn Sie diese nirgends auf einer Fish- oder sonstigen PD-Diskette finden

können, verzweifeln Sie nicht. Sie wird spätestens in einigen Monaten auf einer Fish-Disk erhältlich sein. Wenn Sie allerdings nicht so lange warten wollen, empfehlen wir Ihnen, sich die Programmservice-Diskette zu diesem Heft zu besorgen. Sie enthält den DME in der brandneuen Version.

Sie können selbstverständlich für diesen Workshop auch eine ältere Version verwenden. Testen Sie in diesem Fall aber sorgfältig, welche Funktionen enthalten sind.

Wenn Sie den DME V1.35 von der Programmservice-Diskette verwenden, muß dieser erst entpackt werden. Um wenigstens den größten Teil der in diesem Heft vorgestellten Programme auf einer Diskette unterzubringen, mußten wir die einzelnen Files mit dem Programm »Zoo« archivieren. Dieses liegt der Diskette bei. Sie finden auf Seite 125 eine Anleitung.

Formatieren Sie eine leere Diskette und übertragen Sie al-

le Files darauf (wenn Sie nur ein Laufwerk haben, auf dem Umweg über die RAM-Disk). Kopieren Sie eine Original-Workbench. Im weiteren Verlauf dieses Kurses gehen wir davon aus, daß Sie mit dieser Kopie arbeiten.

Booten Sie von dieser Kopie und schaffen Sie zunächst etwas Platz: Löschen Sie die Demos-Schublade (anklicken und »Discard« aus dem Workbench-Menü wählen). Starten Sie das CLI und entfernen Sie die beiden mitgelieferten Editoren aus dem C-Ordner (»Ed« und »Edit«). Löschen Sie danach alle nicht benötigten Drucker- und Tastaturtreiber (in »devs/printers« beziehungsweise »devs/keymaps«). Sollten Sie mit diesen Arbeitsschritten nicht zurechtkommen, lesen Sie zunächst den Artikel »CLI – Starthilfe« auf Seite 76 in diesem Heft. Damit dürfte genügend Platz geschaffen sein, um alle benötigten Files auf diese Workbench zu übertragen. Kopieren Sie folgende Dateien:

- »DME« in den C-Ordner
  - »null-handler« in den I-Ordner
  - »dres.library« in den libs-Ordner
  - »dme.ref« in den s-Ordner
- Außerdem fügen Sie die enthaltene Mountlist an Ihre eigene an. Geben Sie im CLI ein:
- ```
mount null:
```

Damit sind alle Vorbereitungen abgeschlossen. Das Kopieren und »mounten« des »null-handlers« ist nicht unbedingt notwendig. Dieser beseitigt jedoch die Unzulänglichkeiten des (Pseudo-)Device »NIL:« von Amiga-DOS und erleichtert Ihnen daher die Arbeit mit dem »Execute«-Befehl des DME.

Starten Sie jetzt den DME (aus dem CLI durch Eingabe von »DME <RETURN>«). Auf dem Bildschirm öffnet sich ein Fenster mit dem Titel »DME

```

DME V1.35 ©Copyright 1989 by Matthew Dillon, All Rights Reserved
SAMPLE .EDRC file
copy to $:.EDRC after removing unwanted stuff

# By default DME does not put tabs in the output file.  'savetabs on'
# causes DME to optimize the output file with tabs.

savetabs on
#map c-tab 'left left left left'
#map cs-tab 'del del del del'
#map a-tab 'goto block while cb 'c-tab down''
#map as-tab 'goto block while cb 'cs-tab down''
#map LR-move 'tomouse tlate 32'

# Some people do not like the default cursor key mappings
# (see DME.DOC). Here is an example of remapping them.
#map 'c-up' 'scrollup'
#map 'c-dow' 'scrolldown'
#map 's-up' 'pageup'
#map 's-dow' 'pagedown'
#map 's-lef' 'wleft'
#map 's-riq' 'wright'
#map 'a-dow' 'bottom'
#map 'a-up' 'top'
    
```

Bild 1. »DME« öffnet ein Fenster, in das mit dem Aufruf schon ein File geladen werden kann



V1.35 Copyright 1989 Matthew Dillon, All Rights Reserved« (Bild 1). Um einen Eindruck von den Möglichkeiten zu erhalten, die der DME bietet, klicken Sie einmal mit der rechten Maustaste in das Fenster. Das bisherige Eingabefenster reduziert sich auf ein Icon. Wie erhalten Sie wieder Ihren Ausgangszustand? Klicken Sie nochmals mit der rechten Maustaste. Das Icon vergrößert sich sofort wieder auf die vorher eingestellten Werte.

Der DME ist so wie er sich auf Diskette befindet, bereits voll benutzbar (wenn auch nicht gerade komfortabel). Seine volle

Leistungsfähigkeit offenbart sich aber erst mit einer eigenen Konfiguration. Der DME verhält sich dabei anders als »normale« Editoren, seine Befehle ähneln eher einer einfachen Programmiersprache. Etwas Geduld und Freude am Experimentieren sollten Sie mitbringen, um mit dem Programm vertraut zu werden. Das heißt aber nicht, daß wir Sie von hier ab Ihrem Schicksal überlassen. Sehen Sie sich zuerst die Übersicht über alle Befehle an. Sie finden sie in Tabelle 1. Für den täglichen Gebrauch sind nur wenige Befehle wirklich nötig, man braucht also keineswegs

alle im Kopf zu behalten. Am besten kopieren Sie sich diese Tabelle und hängen sie über oder neben Ihren Monitor.

Wie bereits angedeutet, ermöglicht es der DME die gesamte Oberfläche nach eigenen Wünschen zu gestalten. Dies geht so weit, daß man selbst entscheiden kann, ob man Menüsteuerung wünscht oder ob man lieber mit Tastenkombinationen arbeitet – oder beides. Die Menüs sind keineswegs vorgeformt, Sie können Ihre eigenen Befehle über ein Menü aufrufbar machen und selbstverständlich auch eigene Formulierungen für den Menü-

text wählen. Aber auch für alle, die einen Editor nicht über Menüs bedienen wollen, ist gesorgt. Zusätzlich zu allen Tasten lassen sich auch die Mausknöpfe und die Mausbewegung mit Funktionen belegen. Voreingestellt ist zum Beispiel die Funktion »iconify« auf der rechten Maustaste, die das aktuelle Window zu einem kleinen Icon schrumpfen läßt und damit den Blick auf die dahinterliegenden Fenster freigibt. Für menügewöhnte Benutzer ist dies oft eine Überraschung, andererseits erleichtert der Iconify-Befehl die Arbeit mit mehreren Editorfenstern erheb-

| Befehl                                                                                             | Bedeutung                                                                                                                             |
|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| (text)                                                                                             | Angegebener Text wird als Argument eingefügt                                                                                          |
| Tastenbezeichner                                                                                   | Führt die Befehlsfolge, die auf Taste liegt, aus (wenn diese vorher mit »map« belegt war)                                             |
| Menü-Punkt                                                                                         | Führt einen Menüpunkt aus (beispielsweise »(Projekt Speichern)«)                                                                      |
| addpath Pfad                                                                                       | Setzt einen Suchpfad für Querverweislisten (für »ref« und »ctags«)                                                                    |
| arpinsfile                                                                                         | Arbeitet wie »insfile«, aber es wird der Filerequester der ARP.library aufgerufen                                                     |
| arpload                                                                                            | Überschreibt den geladenen Text durch ein neues Textfile und verwendet dazu den ARP-Filerequester (entspricht »newfile«)              |
| arpsave                                                                                            | Arbeitet wie »saveas«, aber mit Filerequester                                                                                         |
| <b>Alle »ARP«-Befehle können nur zusammen mit der »Arp.Library« verwendet werden (siehe Text).</b> |                                                                                                                                       |
| back                                                                                               | Entspricht »bs« (Backspace, Zeichen vor dem Cursor löschen)                                                                           |
| backtab                                                                                            | Rücktabulator, springt um eine Tabulatorstelle zurück                                                                                 |
| bcopy                                                                                              | Kopiert einen vorher definierten Block vor die aktuelle Zeile                                                                         |
| bdelete                                                                                            | Löscht einen vorher definierten Block                                                                                                 |
| block                                                                                              | Blockanfang beziehungsweise Blockende markieren                                                                                       |
| bmove                                                                                              | Verschiebt einen vorher definierten Block vor die aktuelle Zeile                                                                      |
| bottom                                                                                             | Springt zum Textende                                                                                                                  |
| bs                                                                                                 | Backspace (Zeichen vor dem Cursor löschen)                                                                                            |
| bsave Name                                                                                         | Speichert einen vorher definierten Block unter dem angegebenen Namen                                                                  |
| bsource                                                                                            | Interpretiert einen Block als Befehlsfolge und führt ihn aus                                                                          |
| chfilename Name                                                                                    | Aktuellen Text in »Name« umbenennen                                                                                                   |
| ctags                                                                                              | Wenn der Cursor auf einer Funktionsdefinition der Programmiersprache C steht, wird zur Funktionsdefinition gesprungen (siehe Bericht) |
| del                                                                                                | Löscht das Zeichen unter dem Cursor                                                                                                   |
| deline                                                                                             | Löscht die aktuelle Zeile                                                                                                             |
| down                                                                                               | Bewegt den Cursor nach unten                                                                                                          |
| downadd                                                                                            | Bewegt den Cursor nach unten. Wenn das Textende bereits erreicht ist, wird eine Zeile angefügt                                        |
| esc                                                                                                | Befehlsmodus ein / aus                                                                                                                |
| escimm Arg                                                                                         | Befehlsmodus ein und Anzeigen von »Arg«                                                                                               |
| execute Bef                                                                                        | Führt den CLI-Befehl »Bef« aus                                                                                                        |
| find String                                                                                        | Suchmuster auf »String« setzen und zum nächsten Auftreten des Suchmusters springen                                                    |
| findr s1 s2                                                                                        | Suchmuster auf »s1« und Ersetzungsmuster auf »s2« setzen das nächste Auftreten von »s1« suchen und durch »s2« ersetzen                |
| findstr String                                                                                     | Suchmuster auf »String« setzen                                                                                                        |
| first                                                                                              | Springt in die erste Spalte                                                                                                           |
| firstnb                                                                                            | Springt in die erste Spalte, die kein Leerzeichen ist                                                                                 |

|                       |                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| goto block            | Springt an den Anfang des aktiven Blocks                                                                                                                                       |
| goto start            | Springt zum markierten Blockanfang                                                                                                                                             |
| goto end              | Springt zum Ende des markierten Blocks                                                                                                                                         |
| goto [+/-] #          | Springt zur Zeile »#« oder um »#« Zeilen vorwärts oder rückwärts (abhängig vom Vorzeichen)                                                                                     |
| height #              | Alle zukünftigen Fenster sind »#« Pixels hoch                                                                                                                                  |
| iconify               | »Iconifiziert« das aktuelle Fenster                                                                                                                                            |
| if Bed Bef            | Führt den Befehl »Bef« aus, wenn Bedingung »Bed« erfüllt ist (siehe auch Tabelle 5)                                                                                            |
| ifelse Bed Bef1 Bef2  | Führt »Bef1« aus, wenn Bedingung »Bed« erfüllt ist, im andern Fall wird »Bef2« ausgeführt (siehe auch Tabelle 5)                                                               |
| ignorecase Art        | Groß- und Kleinbuchstaben werden bei allen Such- und Ersetz-Funktionen als gleichwertig behandelt. Dabei kann Art drei Werte annehmen: »on«, »off« oder »toggle«               |
| insertmode Art        | Setzt den Einfügemodus. Art kann drei Werte annehmen: »on«, »off« oder »toggle«                                                                                                |
| insfile Name          | Fügt das File »Name« in den aktuellen Text ein                                                                                                                                 |
| insline               | Fügt eine Zeile ein                                                                                                                                                            |
| ipc Programm          | Mit diesem Befehl steuern Sie andere Programme (arbeitet zur Zeit nur mit Programmen von Matthew Dillon zusammen).                                                             |
| Projekt Befehl        | »IPC« heißt Interactive Process Communication                                                                                                                                  |
| join                  | Verbindet die aktuelle Zeile mit der folgenden                                                                                                                                 |
| last                  | Springt zum letzten Nicht-Leerzeichen der aktuellen Zeile                                                                                                                      |
| left                  | Bewegt den Cursor um eine Position nach links                                                                                                                                  |
| leftedge #            | Setzt den Abstand vom linken Rand für alle neuen Fenster auf # Pixels                                                                                                          |
| map Tas Bef           | Belegt die Taste »Tas« mit der Befehlsfolge »Bef«                                                                                                                              |
| margin #              | Setzt den rechten Seitenrand für Zeilenumbruch und Neuformatierung auf den #.ten Buchstaben (gilt für »wordwrap« und »reformat«)                                               |
| menuadd Kopf Elem Bef | Der Menüpunkt »Elem« wird zum Menü »Kopf« hinzugefügt. Der Menüpunkt wird mit der Befehlsfolge »Bef« belegt. Das Menü »Kopf« wird erzeugt, wenn es nicht bereits vorhanden war |
| menudel Kopf Elem     | Löscht den Menüpunkt »Elem« aus Menü »Kopf«                                                                                                                                    |
| menudelhdr Kopf       | Löscht das Menü »Kopf«                                                                                                                                                         |
| menuclear             | Löscht alle Menüs                                                                                                                                                              |
| menuon                | Schaltet die Menüs ein                                                                                                                                                         |
| menuoff               | Schaltet die Menüs ab                                                                                                                                                          |
| newfile Name          | Ersetzt den aktuellen Text durch das Textfile »Name«                                                                                                                           |
| newwindow             | Eröffnet ein neues DME-Fenster                                                                                                                                                 |
| next                  | Findet das nächste Auftreten des Suchmusters                                                                                                                                   |
| nextx                 | Findet das nächste Auftreten des Suchmusters und ersetzt es durch das Ersetzungsmuster                                                                                         |
| null                  | Stellt eine leere Anweisung dar                                                                                                                                                |
| pagedown              | Blättert eine Seite weiter (siehe pageset)                                                                                                                                     |
| pageset #             | Beim Seitenblättern wird um »#« Prozent weitergeblättert (# = 0 bis 100)                                                                                                       |



lich. Eine sehr gute Möglichkeit, sich an die Bedienung des DME zu gewöhnen, ist eine alternative Bedienung über Menüs und Shortcuts. Später können Sie die Speicherplatzschluckenden Menüs wieder entfernen.

Geben Sie jetzt einen kurzen Text ein. Verwenden Sie dafür mehrere Zeilen. Sobald Sie mit der Eingabe beginnen, wechselt die Anzeige der Titelzeile. Statt des Copyright-Vermerks erscheint jetzt die Anzeige des Filenamens (im Moment noch »unnamed«) sowie die Anzeige der Zeilen und Spaltennummer. Schreiben Sie mit der ersten

Zeile über den rechten Rand hinaus. Sie werden feststellen, daß DME jeweils eine halbe Seite weiterscrollt, wenn Sie an den rechten Rand stoßen. In der StandardEinstellung wird die Seite also nicht umbrochen. Sie können in einer Zeile bis zu 255 Zeichen eingeben.

## 255 Zeichen pro Zeile

Versuchen Sie jetzt das Fenster per Klick auf das Close-Gadget zu schließen. Sie erhalten die Meldung »\*\*\* File has been modified \*«. Wenn Sie mit einem weiteren Klick bestäti-

gen, daß Sie akzeptieren, den eingegebenen Text zu verlieren, verlassen Sie DME. Klicken Sie statt dessen noch einmal innerhalb des Fensters. Sie können ganz normal weiterarbeiten. Geben Sie jetzt so viele Zeilen ein, daß etwas mehr als eine Seite beschrieben ist. Klicken Sie in den oberen Bereich des Fensters und bewegen Sie die Maus nach oben. Sie sehen, daß der Text unter dem Cursor wegschrollt.

Drücken Sie einmal auf <ESC>. Am unteren Rand des Bildschirms öffnet sich eine neue Eingabezeile. Geben Sie dort ein: `saveas test`

DME speichert Ihren Text unter dem Namen »test« im aktuellen Directory. Mit den bisher beschriebenen Funktionen kann der DME nicht mehr als jeder andere Editor auch (ausgenommen vielleicht das Iconifizieren). Zur Besonderheit wird er erst durch die sehr flexiblen Erweiterungsmöglichkeiten.

Wie Sie sehen, besitzt DME zwei verschiedene Modi: den Textmodus und den Kommandomodus. Sie gelangen mit <ESC> vom Text- in den Kommandomodus. Befehle geben Sie in der Zeile am unteren Bildschirmrand ein. Der wichtigste Befehl für eigene Erweiterun-

|                 |                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pageup          | Eine Seite zurückblättern (siehe pageset)                                                                                                                                     |
| ping #          | Setzt eine Textmarkierung (# = 0-9)                                                                                                                                           |
| pong #          | Springt zur angegebenen Textmarkierung (# = 0-9)                                                                                                                              |
| prev            | Springt zum vorhergehenden Auftreten des Suchmusters                                                                                                                          |
| prevr           | Springt zum vorhergehenden Auftreten des Suchmusters und ersetzt es durch Ersetzungsmuster                                                                                    |
| pushmark        | Legt aktuelle Blockmarkierung auf einen Stapel und löscht die Blockmarkierung                                                                                                 |
| popmark         | Setzt die letzte Blockmarkierung des Stapelspeichers                                                                                                                          |
| purgemark       | Löscht den Stapelspeicher für Blockmarkierungen                                                                                                                               |
| quit            | Beendet DME                                                                                                                                                                   |
| recall          | Aktiviert den Befehlsmodus und zeigt den letzten Befehl nochmals an. Dieser Befehl muß mittels einer Taste aufgerufen werden und kann nicht im Befehlsmodus eingegeben werden |
| ref             | Sucht nach einer Referenz für den Text unter dem Cursor (siehe Bericht)                                                                                                       |
| reformat        | Formatiert den aktuellen Abschnitt neu unter Beachtung des Seitenrands (siehe margin)                                                                                         |
| remeol          | Löscht den Text von der Cursorposition bis zum Zeilenende                                                                                                                     |
| rempath Pfad    | Löscht den Pfad für die Suche nach Querverweislisten                                                                                                                          |
| repeat Bed Bef  | Wiederholt den Befehl »Bef«, solange die Bedingung »Bed« erfüllt ist (siehe Tabelle 5)                                                                                        |
| repstr String   | Setzt das Ersetzungsmuster auf »String«                                                                                                                                       |
| resettoggle #   | Löscht den Umschalter (toggle) für das Feldelement »#« (# = 0 bis 255)                                                                                                        |
| resize S Z      | Verändert die Größe des aktuellen Fensters auf S Spalten und Z Zeilen                                                                                                         |
| return          | entspricht »(first downadd)«                                                                                                                                                  |
| right           | Bewegt den Cursor um eine Position nach rechts                                                                                                                                |
| rx Mac          | Ruft ARexx-Macro »Mac« auf                                                                                                                                                    |
| rx1 Mac Arg     | Ruft ARexx-Macro »Mac« mit Argument »Arg« auf                                                                                                                                 |
| rx2 Mac Arg Arg | Ruft ARexx-Macro »Mac« mit 2 Argumenten auf                                                                                                                                   |
| saveas Name     | Speichert den aktuellen Text unter dem angegebenen Namen. Der Name des aktuellen Textes (Titelzeile) wird nicht verändert                                                     |
| savemap Name    | Speichert die eigenen Tastenbelegungen unter dem angegebenen Namen                                                                                                            |
| saveold         | Speichert den aktuellen Text unter dem in der Titelzeile angegebenen Namen                                                                                                    |
| savesmap Name   | Speichert alle Tastenbelegungen (einschließlich der Systemtastenbelegungen) unter »Name«                                                                                      |
| savetabs on/off | Optimierung der Textspeicherform (Ersetzen der Leerzeichen durch Tabulatorbefehle) wird an- oder ausgeschaltet. Die Voreinstellung ist aus (off)                              |
| scanf KontrSt   | Liest den String in der aktuellen Position in die Variable \$scanf (siehe Bericht)                                                                                            |

|                        |                                                                                                                                                                                                       |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| screenbottom           | Springt in die letzte Zeile auf dem Bildschirm                                                                                                                                                        |
| screen top             | Springt in die erste Zeile auf dem Bildschirm                                                                                                                                                         |
| scrollup               | Aufwärtsscrollen ohne den Cursor zu bewegen                                                                                                                                                           |
| scrolldown             | Abwärtsscrollen ohne den Cursor zu bewegen                                                                                                                                                            |
| set Var String         | Erzeugt oder verändert die interne Variable »Var«                                                                                                                                                     |
| setenv Var Str         | Erzeugt oder verändert die Environment-Variable »Var« (Workbench 1.3 ENV:-Verzeichnis nötig)                                                                                                          |
| setfont Font #         | Setzt einen Font (Beispiel: »setfont topaz.font 9«)                                                                                                                                                   |
| settoggle #            | Setzt Umschalter (toggle) Feldelement # (# = 0 bis 255)                                                                                                                                               |
| source Name            | Das File »Name« wird als Befehlsfolge aufgefaßt und ausgeführt (" # " in der ersten Spalte bedeutet Kommentar)                                                                                        |
| split                  | Zeile wird an der Cursorposition aufgespalten                                                                                                                                                         |
| swapmark               | Blockmarkierung wird mit der letzten auf dem Stapelspeicher vertauscht                                                                                                                                |
| tab                    | Vorwärtstabulator                                                                                                                                                                                     |
| tabstop #              | Setzt Tabulatorstop auf jede »#«-te Position                                                                                                                                                          |
| tlate [+/-] #          | Verändert den ASCII-Code des Zeichens unter dem Cursor um + # oder - #, beziehungsweise setzt das Zeichen auf genau ASCII(#), wenn kein + oder - angegeben ist (beispielsweise »tlate +65«, tlate 13) |
| tmpheight #            | Setzt die Fensterhöhe nur für das nächste zu öffnende DME-Fenster auf »#« Pixel                                                                                                                       |
| tmpwidth #             | Setzt die Fensterbreite nur für das nächste zu öffnende DME-Fenster auf »#« Pixel setzen                                                                                                              |
| toggle #               | Schaltet den Umschalter (toggle) # (# = 0 bis 255) um                                                                                                                                                 |
| tomouse                | Bewegt den Cursor zur Mausposition                                                                                                                                                                    |
| top                    | Bewegt den Cursor zum Textanfang                                                                                                                                                                      |
| topedge #              | Setzt den Abstand vom oberen Rand für alle neuen Fenster auf # Pixel                                                                                                                                  |
| unblock                | Löscht alle Blockmarkierungen                                                                                                                                                                         |
| undo                   | Macht Änderungen in der aktuellen Zeile rückgängig (muß auf eine Taste gelegt werden, funktioniert nicht im Befehlsmodus)                                                                             |
| unmap Taste            | Löscht die Tastenbelegung von »Taste«                                                                                                                                                                 |
| up                     | Bewegt den Cursor um eine Position nach oben                                                                                                                                                          |
| while Bed Bef          | Wiederholt »Bef« solange Bedingung »Bed« erfüllt ist (siehe Tabelle 5)                                                                                                                                |
| width #                | Setzt die Breite für neue Fenster auf # Pixels                                                                                                                                                        |
| wleft                  | Springt zum Anfang des vorherigen Wortes. Steht der Cursor innerhalb eines Wortes, so wird zum Anfang dieses Wortes gesprungen                                                                        |
| wordwrap on/off/toggle | Zeilenumbrech ein/aus/umschalten (siehe auch margin)                                                                                                                                                  |
| wright                 | Springt zum Anfang des folgenden Wortes                                                                                                                                                               |

**Tabelle 1. Mit diesen Befehlen steuern Sie den DME im Direktmodus. Sie können aber auch Tastenkombinationen damit belegen oder sie beim Start automatisch ausführen lassen – der Komfort ist fast unbegrenzt.**



gen ist »map«. Mit diesem Befehl lassen sich Tasten mit Funktionen oder Strings belegen. Soll zum Beispiel mit <F6> an den Anfang des Textes gesprungen werden, so erreichen Sie das mit »map (f6) (top)«. Nach der Eingabe dieses Befehls ist die Taste F6 mit dem Kommando »top« belegt (Springen an den Textanfang). Befehlsfolgen oder Tasten wer-

## Bis zu 128 Funktionen pro Taste

den dabei mit runden Klammern umschlossen, Texte kennzeichnet man durch ein weiteres Klammerpaar. Jede Nicht-Sondertaste kann im Unterschied zu anderen Editoren mit Befehlen oder Text belegt werden. Als Sondertasten gelten die beiden Amiga-Tasten, <Ctrl>, <Shift> und die Alt-Tasten. Normalerweise ist beispielsweise die »Return«-Taste mit »Return« belegt und die Cursorrichtungstasten sind mit UP, DOWN, LEFT und RIGHT. Jedoch sind auch hier beliebige Änderungen möglich. Tasten werden nach ihrem Aufdruck benannt (die Taste für die Ziffer 6 heißt »6«). Es gibt jedoch folgende Ausnahmen:

- Backspace (Korrekturtaste <->) ist »BS«.
- Tasten des Zehnerblocks erhalten ein vorangestelltes »NK«, ausgenommen Enter, welche »ENTER« genannt wird.
- Die Cursorrichtungstasten werden »UP«, »DOWN«, »LEFT« und »RIGHT« genannt.
- Die drei Mausknöpfe heißen »LMB«, »MMB«, »RMB«. (Die Amiga-Maus hat keinen mittleren Mausknopf (MMB). Falls eine Maus mit drei Knöpfen angeschlossen ist, kann aber auch der mittlere genutzt werden).

Jede Taste kann mit einer beliebigen Kombination von <CTRL>, <ALT>, <SHIFT> <AMIGA> und den Maustasten belegt werden. Mit diesen sechs Sondertasten (sieben, wenn die Maus drei Knöpfe hat), kann man jede Nicht-Sondertaste bis zu 64mal (128mal) verschieden belegen. Bei über 70 Tasten sind das mehr als 4000 Kombinationen. Vorerst dürfte das genügen. Einige Beispiele finden Sie in Tabelle 2.

An den DME können beim Aufruf unendlich viele Parameter übergeben werden. Jedes Argument ohne vorangestelltes »-« wird als ein File interpretiert, welches in einem eigenen Fenster dargestellt wird. Alle Einstellungen des Benutzers für die Tastenbelegung, die Menüs

etc. liest der DME aus einem File namens ».EDRC«, welches im S-Ordner liegt. Zusätzlich kann noch ein weiteres ».EDRC«-File im aktuellen Verzeichnis liegen, wenn für besondere Anlässe spezielle Tastenbelegungen nötig sind. Die in diesem .EDRC-File enthaltenen DME-Befehle werden nacheinander ausgeführt, bevor der Editor startet (ähnlich einem sogenannten Batch-File). Alle im DME benutzbaren Befehle dürfen dort enthalten sein. Alle Aufrufparameter finden Sie in Tabelle 3. Gibt es keine .EDRC-Files, so wird die voreingestellte Tastenbelegung des DME benutzt, was auch für alle die Ta-

| Sondertaste      | Bezeichner |
|------------------|------------|
| Shift            | s          |
| Ctrl             | c          |
| alt              | a          |
| Amiga            | A          |
| Zehnerblock      | nk         |
| linke Maustaste  | L          |
| rechte Maustaste | R          |

| Beispiele zur Tastenbezeichnung: |                                                               |
|----------------------------------|---------------------------------------------------------------|
| Bezeichner                       | bezeichnete (Maus-)Taste                                      |
| a                                | Taste »a«                                                     |
| tab                              | Tabulatortaste                                                |
| a-a                              | alt»a«                                                        |
| A-a                              | Amiga»a«                                                      |
| sA-a                             | Shift+Amiga »a«                                               |
| L-a                              | »a« bei gedrückter linker Maustaste                           |
| ac-?                             | alt+ctrl »?«                                                  |
| s-f5                             | Shift+Funktionstaste Nr.5                                     |
| nk0                              | »0« auf dem Zehnerblock                                       |
| cs-nk0                           | Ctrl Shift »0« auf dem Zehnerblock                            |
| L-lmb                            | Drücken der linken Maustaste                                  |
| L-mmove                          | Mausbewegung bei gleichzeitigem Drücken der linken Maustaste  |
| LR-lmb                           | Drücken der rechten Maustaste bei gedrückter linker Maustaste |

Tabelle 2. Alles neu macht DME: Mit einigen einfachen Befehlen belegen Sie die Tastatur

sten gilt, die im ».EDRC« nicht umdefiniert werden. Dies ist die Situation, die Sie vorfinden, wenn Sie alle Schritte genau so ausgeführt haben, wie oben beschrieben. Die voreingestellte Tastenbelegung des DME 1.35 ist in der Tabelle 4 zusammengestellt. Testen Sie außerdem

## Start mit Ihrer Konfiguration

das mitgelieferte ».EDRC«-File. Kopieren Sie es mit

```
copy sample.edrc s:.edrc
```

in den S-Ordner. Starten Sie den DME neu und testen Sie die jetzt verfügbaren Funktionen. Beispielsweise sind jetzt Menüs installiert und einige Tasten mit speziellen Funktionen

| Mögliche Parameter beim Aufruf von DME                                                                                                 |                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Der Aufruf hat in allgemeiner Schreibweise folgendes Format:<br>DME [-fFilename] [-b] [-t#] [-l#] [-w#] [-h#] [Filename [Filename]...] |                                                                                                                                                      |
| Parameter                                                                                                                              | Erklärung                                                                                                                                            |
| -fFilename                                                                                                                             | »Filename« anstatt von .EDRC laden und ausführen                                                                                                     |
| -b                                                                                                                                     | Normaldimensioniertes Fenster öffnen, wenn mehrere Files angegeben wurden (normalerweise werden die Fenster verkleinert, um Speicherplatz zu sparen) |
| -t#                                                                                                                                    | Fenster in Zeile # eröffnen                                                                                                                          |
| -l#                                                                                                                                    | Fenster ab Spalte # eröffnen                                                                                                                         |
| -w#                                                                                                                                    | Fensterbreite # Pixels                                                                                                                               |
| -h#                                                                                                                                    | Fensterhöhe # Pixels                                                                                                                                 |

Tabelle 3. Sie können »DME« auch mit den Aufrufparametern bequem steuern

Argumenten. Das Trennzeichen zwischen den Argumenten und Befehlen ist das Leerzeichen. Um einen Text mit Leerzeichen als ein Argument zu definieren, setzen Sie den Text in Klammern oder versehen ihn mit einfachen Anführungszeichen, beispielsweise:

(Dies ist ein Text mit Leerzeichen)

'Dieses ebenfalls'

Text, der ausgegeben werden soll, muß nochmals zusätzlich geklammert werden! Erwartet ein Befehl nur ein Argu-

## Befehle ohne Ende

ment, so wird das Wort nach diesem Argument als nächster Befehl aufgefaßt. Das bedeutet, daß mehrere Befehle geklammert werden müssen, um vom DME als ein Befehlsblock verstanden zu werden. Zwei einfache Beispiele: der Befehl »map (f4) ((Hallo))« belegt die Funktionstaste F4 mit dem Text »Hallo«. Mit »map (f4) (down down right (Hallo))« wird F4 mit einer Befehlsfolge (darum die äußeren Klammern) belegt: Cursor zwei Zeilen nach unten und eine Spalte nach rechts, anschließend »Hallo« ausgeben. Aufgrund der Interpretationsart von Befehlen muß Text also unbedingt in zusätzliche Klammern gesetzt werden. Dies wird von DME-Neulingen oft vergessen und führt dann zur Fehlermeldung »unknown command«.

Der map-Befehl ist das »Arbeitstier« des DME. Mit ihm lassen sich - wie bereits kurz erwähnt - beliebige Tasten mit Befehlen oder Text belegen. Um die Arbeitsweise des DME zu verstehen, stellen Sie sich am besten alle Tasten mit ei-

belegt. Um das Vorgehen nachzuvollziehen, laden Sie das .edrc-File. Drücken Sie dazu <ESC> und geben Sie ein:

```
newfile s:.edrc
```

Sie finden im Editor-Fenster einige Kommentare (mit einem »#« eingeleitet) und Befehlszeilen. Sehen Sie sich die Kommandos an. Sie finden vor allem »map«- und »menuadd«-Befehle. Bereits aus der Struktur dieser Beispiele können Sie einiges lernen. Weiter unten wird genauer auf die Befehle eingegangen.

Der DME hat einen reichhaltigen Befehlssatz, der beispielsweise auch Befehlschachtelung und vieles mehr erlaubt. Ein Befehl besteht aus einem Schlüsselwort, gefolgt von einer festgelegten Zahl von



nem map-Befehl belegt vor. So ist die Taste »a« mit der Zeichenfolge »a« belegt. Drücken Sie einmal <ESC> und geben Sie in der Befehlszeile ein:

```
map a ((hallo))
```

Drücken Sie jetzt im Editor-Fenster die Taste a. Was sehen Sie auf dem Bildschirm? Um diese Spielerei wieder rückgängig zu machen (manchmal wird das a ja noch gebraucht), geben Sie (nach Drücken von <ESC>) ein:

```
unmap a
```

Die Taste »a« ist also gewöhnlich mit »map a ((a))« belegt. Ein einfacher Weg für das Ausführen von Befehlen oder Befehlsfolgen ist die Eingabe im Befehlsmodus. In diesen gelangen Sie durch Drücken der ESC-Taste. Die letzte Zeile des Fensters wird dann für die Befehlseingabe abgetrennt. Wird zusätzlich zu <ESC> noch <CTRL> gedrückt, so wird der zuletzt ausgeführte Befehl

angezeigt und kann editiert werden. Testen Sie dies, indem Sie jetzt <CTRL ESC> drücken. Sie sehen in der Befehlszeile den letzten Befehl (unmap a). Dieses Verfahren kennen Sie vielleicht von der History-Funktion einer Shell.

Experimentieren Sie mit den map-Befehlen. Belegen Sie sich Ihre Tasten nach freier Wahl. Beispielsweise könnten Sie Floskeltexte auf einzelne Tasten legen. Nehmen wir an, Sie programmieren in Modula und wollen Ausdrücke wie »IMPLEMENTATION MODULE« nicht jedesmal ausschreiben. Belegen Sie die Tastenkombination <CTRL i> damit (in DME-Kommandos wird diese mit »C-i« angesprochen) und Sie haben die Zeile auf Abruf parat. Als C-Programmierer können Sie häufig benötigte Strukturen mit einem Tastendruck initialisieren (vielleicht mit <AMIGA w> für die New-Window-Struktur?). Oder auch

das beliebte »Sehr geehrte Damen und Herren« für Briefe auf <ALT s> läßt sich damit realisieren. Stellen Sie sich Ihre individuelle Belegung zusammen und testen Sie diese. Wenn Sie damit zufrieden sind, schreiben Sie alle Belegungen in das »edrc«-File im S-Ordner. Sie können dieses File natürlich jederzeit in den DME laden und modifizieren. Dazu tragen Sie die gewünschten Befehle in diesem File ein und löschen alles, was Sie nicht brauchen. Beim nächsten Start von DME sind alle Kommandos wie gewünscht verfügbar. Es empfiehlt sich allerdings, die Wirkung aller Befehle vorher im Direktmodus (nach Druck auf <ESC>) zu testen. Natürlich können Sie auch jeden beliebigen Befehl (oder jede Befehlskombination) auf Tastendruck ausführen lassen. Beispielsweise ließe sich damit der Ablauf

File in das neue Fenster

Fenster öffnen  
laden  
altes Fenster  
iconifizieren

leicht realisieren. Haben Sie die entsprechenden Befehle gefunden?

Eine komplette Befehlsliste des DME 1.35 finden Sie in Tabelle 1. Viele der Befehle erklären sich anhand der Tabelle von selbst. Aber ein paar komplexe Befehle bedürfen doch noch weiterer Erklärungen:

## Komfort mit ARP-Befehlen

Die Befehle »arpload«, »arp-save« und »arpinsfile« funktionieren nur, wenn im LIBS-Ordner die »ARP.library« vorhanden ist. Dies ist eine Bibliothek mit vielen hilfreichen Funktionen, die ebenfalls als Public Domain-Programm erhältlich ist (auf der Fish 123 oder RPD 122, Version 1.1) und

| Die DME-Standardereinstellungen |                              |                                                          |
|---------------------------------|------------------------------|----------------------------------------------------------|
| Taste                           | Befehl                       | Bedeutung                                                |
| map (a-c)                       | (bcopy)                      | Kopiere Block                                            |
| map (a-d)                       | (bdelete)                    | Lösche Block                                             |
| map (a-down)                    | (scroll down)                | Nach unten scrollen                                      |
| map (a-l)                       | (while cu (tlate +32 right)) | Wort unter dem Cursor in Kleinbuchst. übersetzen         |
| map (a-m)                       | (bmove)                      | Block zur Cursorposition verschieben                     |
| map (a-r)                       | (nextr)                      | Letztes Suchen und Ersetzen wiederholen                  |
| map (a-s)                       | (bsource)                    | Block als Befehlsfolge interpretieren und ausführen      |
| map (a-u)                       | (while cl (tlate -32 right)) | Wort unter dem Cursor in Großbuchstaben übersetzen       |
| map (a-up)                      | (scroll up)                  | Hochscrollen                                             |
| map (bs)                        | (bs)                         | Löschtaste (Backspace)                                   |
| map (c-/)                       | (escimm (find ))             | Befehlsmodus und Suchen                                  |
| map (c-)                        | (ref)                        | Referenz suchen                                          |
| map (c-l)                       | (ref)                        | Funktionsdefinition suchen                               |
| map (c-1)                       | (goto block)                 | Springe zum Blockanfang                                  |
| map (c-b)                       | (block)                      | Markiere Blockanfang oder Blockende                      |
| map (c-c)                       | ()                           | Stoptaste (Break)                                        |
| map (c-del)                     | (remeol)                     | Lösche bis Zeilenende                                    |
| map (c-down)                    | (pagedown)                   | Seite weiterblättern                                     |
| map (c-esc)                     | (recall)                     | Befehlsmodus einschalten und den letzten Befehl anzeigen |
| map (c-f)                       | (reformat)                   | Neu formatieren                                          |
| map (c-g)                       | (escimm (goto ))             | Befehlsmodus und Springe                                 |
| map (c-i)                       | (insertmode on)              | Einfügemodus einschalten                                 |
| map (c-j)                       | (join)                       | Zeilen verbinden                                         |
| map (c-l)                       | (wleft)                      | Ein Wort nach links                                      |
| map (c-n)                       | (next)                       | Letztes Suchen wiederholen                               |
| map (c-o)                       | (insertmode off)             | Einfügemodus ausschalten                                 |
| map (c-p)                       | (prev)                       | Letztes Suchen in Gegenrichtung wiederholen              |
| map (c-q)                       | (quit)                       | DME verlassen                                            |
| map (c-r)                       | (wright)                     | Ein Wort nach rechts                                     |
| map (c-s)                       | (split first down)           | Zeile auftrennen                                         |
| map (c-u)                       | (unblock)                    | Blockmarkierungen löschen                                |
| map (c-up)                      | (pageup)                     | Seite zurückblättern                                     |
| map (c-w)                       | (wordwrap toggle)            | Zeilenumbruch ein/aus                                    |

|               |                                 |                                                                                                     |
|---------------|---------------------------------|-----------------------------------------------------------------------------------------------------|
| map (del)     | (del)                           | Zeichen löschen (Delete-Taste)                                                                      |
| map (down)    | (down)                          | Cursor nach unten                                                                                   |
| map (enter)   | (return)                        | An den Anfang der nächsten Zeile springen                                                           |
| map (esc)     | (esc)                           | Befehlsmodus einschalten                                                                            |
| map (f1)      | (escimm (insfile ))             | Befehlsmodus und Einfügen eines Files                                                               |
| map (f10)     | (saveold quit)                  | Text speichern und DME ↵ verlassen                                                                  |
| map (f2)      | (escimm (newfile ))             | Text mit neuem Textfile überschreiben (Vorsicht, der alte Text wird nicht automatisch gespeichert!) |
| map (f3)      | (escimm (newwindow newfile ))   | Neues Fenster mit neuem Textfile öffnen                                                             |
| map (f6)      | (saveold iconify)               | Text speichern und Fenster »iconifizieren«                                                          |
| map (f7)      | (escimm (bsave ))               | Befehlsmodus und Speichern                                                                          |
| map (f8)      | (saveold escimm (newfile ))     | Text speichern, Befehlsmodus und Laden                                                              |
| map (f9)      | (saveold)                       | Text speichern                                                                                      |
| map (L-lmb)   | (tomouse)                       | Cursor springt zur Maus (linke Maustaste)                                                           |
| map (L-mmo)   | (tomouse)                       | Cursor springt zur Maus (mittlerer Mausknopf, wenn vorhanden)                                       |
| map (left)    | (left)                          | Cursor nach links                                                                                   |
| map (R-rmb)   | (iconify)                       | Fenster »iconifizieren« (rechte Maustaste)                                                          |
| map (return)  | (return inline up firstnb down) | Neue Zeile erzeugen und Cursor einrücken                                                            |
| map (right)   | (right)                         | Cursor nach rechts                                                                                  |
| map (s-)      | (( ))                           | Leerzeichen                                                                                         |
| map (s-del)   | (deline)                        | Zeile löschen                                                                                       |
| map (s-down)  | (bottom)                        | Springe zum Textende                                                                                |
| map (s-left)  | (first)                         | Springe zum Zeilenanfang                                                                            |
| map (s-right) | (last)                          | Springe zum Zeilenende                                                                              |
| map (s-tab)   | (backtab)                       | Rücktabulator                                                                                       |
| map (s-up)    | (top)                           | Springe zum Textanfang                                                                              |
| map (sa-s)    | (unblock block block bsource)   | Zeile als Befehl interpretieren und ausführen                                                       |
| map (tab)     | (tab)                           | Tabulator                                                                                           |
| map (up)      | (up)                            | Cursor nach oben                                                                                    |

Tabelle 4. Diese Tastenbelegung ist voreingestellt, sie kann mit dem »map«-Befehl jederzeit verändert werden



von zahlreichen Programmen (beispielsweise der neuen CShell) benutzt wird. Der DME funktioniert auch ohne die ARPLibrary einwandfrei. Doch mit den ARP-Befehlen des DME ist es möglich, den Filere-

veas« und »insfile« aus dem Standard-Befehlssatz von DME.

Mit den Befehlen »menuadd«, »menudel«, »menudelhdr« und »menuclr« hängen Sie eigene Menüs an das DME-Fenster oder verändern vorhandene. Diese können dann mit »menuon« und »menuoff« an- und abgeschaltet werden. Ist eine größere Menge von menuadd-Befehlen in dem aktuellen .EDRC-File vorhanden, so wird der Start von DME gebremst. Sie können den Ablauf aber beschleunigen, indem Sie vor den Menü-Befehlen menuoff ge-

ben. Am Ende können die Menüs dann mit menuon eingeschaltet werden. DME zählt jeden Aufruf von menuoff mit, so daß ein doppelter Aufruf von menuoff zwei menuon-Befehle erfordert. Dieses gilt jedoch nicht für den umgekehrten Fall.

Zwei ähnliche Befehle machen die Arbeit mit dem DME besonders für Programmierer zum echten Vergnügen - »ctags« und »ref«. Mit diesen verarbeiten Sie auf sehr einfache Weise sogenannte Querverweise aus beliebigen Textfiles. Setzen Sie den Cursor auf einen beliebigen Suchbegriff und rufen Sie dann per Tastendruck den Befehl »ref« auf. Dieser durchsucht anhand der Suchtabelle im File »DME.ref« aus dem S-Ordner die dort aufgeführten Files. Anschließend öffnet DME ein neues Fenster, welches alle gesuchten Informationen zu dem Begriff enthält, auf den der Cursor zeigt. Können Sie sich etwas einfacheres vorstellen? Dieser Befehl kann Programmierern sehr viel Arbeit ersparen, wenn man zum Beispiel die Include-Files als Referenzquelle verwendet. Modula-Programmierer würden entsprechend ihre Bibliotheksmodule ange-

ben. Auf einen Tastendruck kann man zu dem Begriff »struct Screen« die komplette Screen-Struktur anzeigen lassen und so die gesuchten Strukturelemente finden. Der DME verwendet beim Referenz-Befehl temporäre Dateien, damit man durch einen Bedienungsfehler nicht aus Versehen die Include-Files zerstört.

Nicht weniger sinnvoll ist der Befehl »ctags«, der besonders C-Programmierern gefallen wird. Mit einem gleichnamigen Programm (wird bei Aztec C mitgeliefert, ist aber auch in Public Domain-Sammlungen zu finden) kann aus C-Programmen eine Querverweisliste namens »tags« erzeugt werden, in der alle Funktionsdefinitionen vermerkt sind. Nun kann man bei einem Funktionsaufruf in einem C-Programm mit dem ctags-Befehl an den Anfang der Funktionsdefinition springen, ohne sich Gedanken über die Lage der entsprechenden Funk-

## Menüs selbstgestrickt

quester der ARPLibrary zu verwenden. Dieser läßt eine komfortable Auswahl der zu ladenden Datei zu (siehe Bild 2). Ansonsten bewirken die Befehle das gleiche wie »newfile«, »sa-

## Kurz vorgestellt: Matthew Dillon

Matthew Dillon ist Autor vieler begeisternder Public Domain-Programme. Seine bekanntesten veröffentlichten Projekte sind der DME, DMOUSE und DNET. Der kalifornische Spitzenautor ist 22 Jahre jung. Er lebt in Berkeley, Kalifornien. An der University of California hat er im vergangenen Dezember promoviert. Der engagierte Programmierer arbeitet jetzt halbtags im Rechenzentrum der Universität an Datenbanken auf Unix-Großrechnern. Zusätzlich ist Mathew in einem kleinen Softwarehaus engagiert.

Seine Freizeit verbringt Matthew Dillon im Sommer vorwiegend am Tahoe-See mit Wasserski-Fahren. Auch im Winter schnallt er die Skier nicht ab. Er wechselt auf gefrorenes Wasser - cross-country-Ski ist in der kalten Jahreszeit sein Ausgleich für die Arbeit am Computer. Matthew programmiert die Hilfsprogramme auf dem Amiga und auf Unix-Anlagen, hauptsächlich, um sich selbst die Arbeit zu erleichtern. Da aber auch viele andere Programmierer diese Tools gut gebrauchen können, werden sie ausgereift und kostenlos zur Verfügung gestellt. Mathew's wichtigste Prinzipien beim Programmieren der Tools sind Flexibilität und Absturzicherheit. Auf verspielte Benutzeroberflächen und aufwendige Grafik legt er weniger Wert als auf Funktionalität. Seine Programme sind deshalb auch besonders für Programmierer interessant, können trotzdem aber sicher jedem Amiga-Anwender Nutzen bringen.

Alle Programme werden nicht kommerziell vertrieben, weil Matthew, wie er sagt, es gar nicht eilig hat, reich zu werden. Später möchte er aber auch kommerzielle Software erstellen, die sich von den »kleineren« Tools aber deutlich abheben soll.

Der Kalifornier programmiert am liebsten in C. In dieser Sprache erstellt er neben wenigen Assembler-routinen alle seine Programme. Modula hält Matthew für eine gute Programmiersprache, alle anderen Sprachen sagen ihm nicht zu. Matthew amüsiert sich über die vielen guten Programme, die auch heute noch in Fortran und Cobol geschrieben werden, obwohl sich schon wesentlich bessere Programmiersprachen etabliert haben.

Hier eine kurze Vorstellung der wichtigsten Projekte, die derzeit für den Amiga in der PD-Szene zu erhalten sind:

**DMOUSE** - dunkelt den Bildschirm nach einer einstellbaren Zeit ab, beschleunigt die Maus und schaltet diese auf Wunsch ab. Das Tool kann (NEW)CLI-Windows auf Tastendruck öffnen und vieles mehr.

**DME** - der hier vorgestellte Editor ist besonders für Programmierer geeignet. Durch seine große Flexibilität läßt sich der DME jedoch auch an andere Bedürfnisse anpassen.

**DMAKE** - ein verbesserter »Make«-Befehl (für C-Programmierer)

**DPIPE** - ein verbesserter »Pipe«-Handler

**DNET** - ein Netzwerkprogramm und eine Mailbox-Treibersoftware, die auch für Anbindungen an Großrechenanlagen (Unix) geeignet ist. Außerdem ist DNET ein verbesserter Ersatz für den AUX-Handler, weil es auch die Steuerung des Amiga über die serielle Schnittstelle erlaubt.

**DRES.LIBRARY** ist eine Bibliothek, die Interaktionen zwischen verschiedenen Tasks ermöglicht (»IPC« - Interactive Process Communication)

**TERRAIN** - dient zum Entwerfen von fraktalen 3D-Landschaftsoberflächen

**DTerm** - ein Terminalprogramm. Alle Programme von Matthew sind Public Domain und auch der Quelltext kann kostenlos bezogen werden, wenn er nicht bereits dem Programm beigefügt ist. Wir freuen uns auf die kommenden Projekte, die dieser Autor in den PD-Pool ein-speisen wird.

Heiko Schlichting/rs

## Ein Editor, der lesen kann?

tion zu machen. Liegt die Funktionsdefinition in einem anderen File, so öffnet der DME ein

| Befehl                                                                                                                                                                                                                                                                                                          | Folge                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| repeat Bed Arg                                                                                                                                                                                                                                                                                                  | Falls »Bed« eine Zahl ist: Wiederhole den Befehlsblock »Arg« »Bed«-mal. Falls Bed eine Bedingung ist: Wiederhole den Befehlsblock »Arg« solange die Bedingung erfüllt ist. |
| Folgende Bedingungen stehen zur Programmierung zur Verfügung:                                                                                                                                                                                                                                                   |                                                                                                                                                                            |
| Bedingung                                                                                                                                                                                                                                                                                                       | Bedeutung                                                                                                                                                                  |
| line                                                                                                                                                                                                                                                                                                            | Aktuelle Zeilennummer (Die Zeilennummerierung beginnt mit 1)                                                                                                               |
| lbot                                                                                                                                                                                                                                                                                                            | Anzahl Zeilen bis zum Textende einschließlich der aktuellen Zeile                                                                                                          |
| cleft<br>cright                                                                                                                                                                                                                                                                                                 | Spaltennummer (beginnt bei 0)                                                                                                                                              |
| tr                                                                                                                                                                                                                                                                                                              | Anzahl der Zeichen bis zum Zeilenende, einschließlich des Zeichens unter dem Cursor                                                                                        |
| tl                                                                                                                                                                                                                                                                                                              | Anzahl der Zeichen bis zum nächsten Tabulatorstop                                                                                                                          |
|                                                                                                                                                                                                                                                                                                                 | Anzahl der Zeichen bis zur Rücktabulatormarkierung.                                                                                                                        |
| Einige Befehle können eine repeat-Schleife abbrechen. Dies sind zum einen die Befehle »find«, »findr«, »next«, »next«, »prev« und »prevr«, wenn das gesuchte Stringmuster nicht gefunden wird, zum anderen Befehle, bei denen Operationen an Grenzen stoßen, wie zum Beispiel »up«, »left«, »right« und »down«. |                                                                                                                                                                            |
| Wenn Bed auf -1 gesetzt wird, wird repeat zu einer Endlosschleife (genauer: 0xFFFFFFF mal), wenn nicht einer der obengenannten Befehle die Schleife abbricht.                                                                                                                                                   |                                                                                                                                                                            |
| Eine Benutzerunterbrechung kann durch Drücken von <CTRL c> erfolgen.                                                                                                                                                                                                                                            |                                                                                                                                                                            |
| Syntax und Bedingungen für die Befehle »if«, »ifelse« und »while«:<br>if [!]Bed Akt<br>ifelse [!]Bed Akt1 Akt2<br>while [!]Bed Akt                                                                                                                                                                              |                                                                                                                                                                            |



neues Window und springt an die erste Zeile der gesuchten Funktion. Ist die Funktion bereits geladen, so wird das entsprechende Fenster nach vorn gebracht und in die gesuchte Zeile gesprungen. Einfacher kann man die Suche an Funktionen wohl nicht mehr gestalten. Eine solche Querverweisliste läßt sich mit neueren »ctags«-Programmen nicht nur aus Quelltexten der Programmiersprache C erzeugen, sondern auch aus Texten anderer Programmiersprachen. Ein neues Programm dieser Art ist zum Beispiel auf der Fish-Disk Nr. 197 veröffentlicht.

Mit dem »scanf«-Befehl ist es möglich, einen unter dem Cursor stehenden Text zu analysieren und in eine Variable einzulesen. Die Syntax des scanf-Befehls entspricht dabei weitestgehend der Programmiersprache C, mit der Einschränkung, daß nur ein Argument eingelesen werden kann. Mit »scanf %s« wird der String unter dem Cursor in die Variable \$scanf eingelesen, mit »scanf %4s« entsprechend nur die ersten vier Buchstaben. Sollen nur bestimmte Zeichen eingelesen werden, so können diese in eckigen Klammern angegeben

werden. »scanf %[0123456789]« liest also bis zum ersten Zeichen ein, welches keine Ziffer ist. Mit »scan %()b.« kann bis zum ersten Auftreten eines Punktes eingelesen werden. Alle eingelesenen Strings werden in der Variablen \$scanf gespeichert und können als Parameter in den Befehlen dienen. So kann man beispielsweise nach dem nächsten Auftreten des Wortes unter dem Cursor suchen.

## Bedingte Verzweigungen

Für eine variable Programmierung von eigenen Editorfunktionen sind sie unerlässlich: die bedingten Verzweigungen und Schleifen. Auch solche Befehle gibt es natürlich im DME. Man kann dabei zwischen »repeat«, »while«, »if« und »ifelse« wählen. Mit while und repeat läßt sich ein Anweisungsblock so oft wiederholen, wie es die als Parameter angegebene Bedingung vorgibt. Die zur Verfügung stehenden Bedingungen sind in Tabelle 5 dargestellt. Eine Schleife wird außerdem abgebrochen, wenn ein mit »find«, »prev« oder »next« gesuchter String nicht gefunden wird,

oder wenn die Befehle »up«, »right«, »left« und »down« an die Textgrenzen stoßen. Die Befehlsfolge »repeat -1 (findr (e) (a))« würde also jedes »e« im Text hinter der aktuellen Cursorposition durch ein »a« ersetzen. Schleifen können im DME mit <CTRL c> abgebrochen werden, was besonders bei versehentlich erzeugten Endlosschleifen unentbehrlich ist. Die Bildschirmdarstellung des DME kann durch die Programme FastFonts (»ff«) oder BlitzFonts beschleunigt werden. Dieses gilt allerdings nur für acht Pixel

gen der Zeilenbreite (mit dem »margin«-Befehl) und Einschalten des Wordwrap-Modus (mit »wordwrap on«) wird auch das Schreiben von Texten im DME zum Kinderspiel. Trotzdem ist der DME damit noch kein Textverarbeitungsprogramm, denn Blocksatz, automatische Trennung und Proportionalfonts werden nicht unterstützt. Wer aber keinen allzu großen Wert auf optisch ausgefeilte Texte legt, nur kurze Anleitungen oder einen ASCII-Text für seine Mailbox schreibt, ist mit dem DME gewiß gut bedient.

| Bedingung | Trifft zu,                                                                                                                                                                    |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ###       | wenn der Umschalter (toggle) ### gesetzt ist (### = 0 bis 255)                                                                                                                |
| t         | wenn Cursor in der obersten Zeile des Textes                                                                                                                                  |
| b         | wenn Cursor in der letzten Zeile des Textes                                                                                                                                   |
| l         | wenn Cursor in der 0. Spalte                                                                                                                                                  |
| r         | wenn Cursor hinter dem letzten Nicht-Leerzeichen einer Zeile                                                                                                                  |
| m         | wenn der Text nach dem Einladen verändert wurde (»modified« in der Titelzeile)                                                                                                |
| i         | wenn Einfügemodus eingeschaltet                                                                                                                                               |
| x[<=>]#   | wenn Spaltenposition (beginnt mit 1) verknüpft mit der Zahl # wahr ergibt. Beispielsweise x<20 : Spaltenposition kleiner als 20. x<=5 : Spaltenposition kleiner oder gleich 5 |
| y[<=>]#   | entspricht x[<=>]#, es wird jedoch mit der Zeilennummer (beginnt mit 1) verglichen                                                                                            |
| cl        | wenn Zeichen in Cursorposition ein Kleinbuchstabe ist                                                                                                                         |
| cu        | wenn Zeichen in Cursorposition ein Großbuchstabe ist                                                                                                                          |
| ca        | wenn Zeichen in Cursorposition ein Buchstabe oder eine Ziffer ist                                                                                                             |
| cn        | wenn Zeichen in Cursorposition eine Ziffer ist                                                                                                                                |
| c[<=>]#   | wenn Zeichen in Cursorposition dem ASCII-Code # entspricht (# ist eine Dezimalzahl. Für die möglichen Bedingungen siehe x[<=>]#)                                              |
| cb        | wenn Cursor innerhalb der Blockmarkierungen.                                                                                                                                  |

Tabelle 5: Syntax und Bedingungen für die Schleifenbefehle

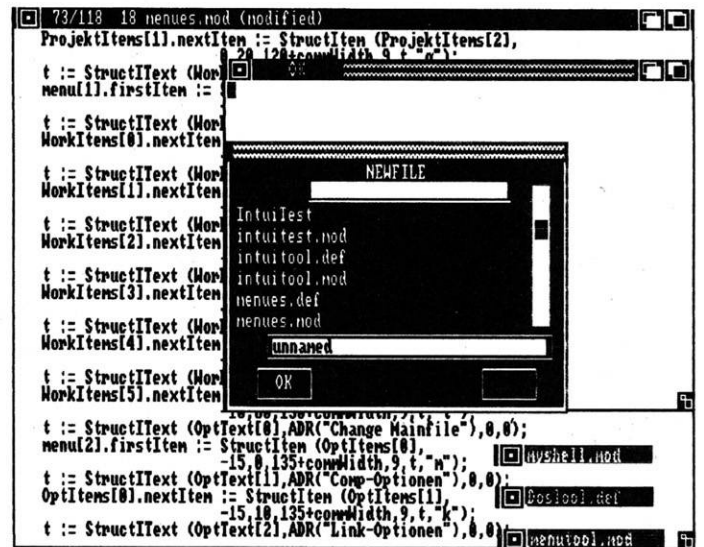


Bild 2. Mehrere Fenster sind kein Problem für »DME«. Rechts unten sehen Sie drei »iconifizierte« Fenster. Darüber wird soeben ein neues File mit dem Filerequester der ARP-Library geladen.

breite Fonts. Wer es vorzieht, andere Fonts als den voreingestellten Topaz-Font zu verwenden, kann mit dem Befehl SetFont auch auf andere nicht-proportionale Fonts umschalten. Die inzwischen bei Editoren schon zum Standard gewordene Schnittstelle zu »AREXX« ist beim DME ebenfalls implementiert, damit ist auch die Kommunikation zu anderen Programmen gewährleistet, ohne daß der Editor verlassen werden muß. ARexx ist eine programmierbare Shell, die verschiedenen Programmen eine problemlose Kommunika-

## Umbruch mit Wordwrap

tion ermöglicht. Sie finden eine Demo-Version auf Fish 188. Die professionelle Version von ARexx erhalten Sie bei Philgerma, Barerstr. 32, 8000 München 2.

Wordwrap ist eine weitere interessante Funktion, die auch das komfortable Schreiben von Texten erlaubt. Durch Festle-

Der DME ist der zur Zeit leistungsfähigste Editor auf dem Amiga. In Flexibilität und Geschwindigkeit übertrifft er selbst professionelle Programme um Längen. Daß diese unbegrenzte Flexibilität mit einer anfangs etwas spartanisch erscheinenden Oberfläche erkauft wird, ist unserer Meinung nach kein Nachteil. Es ist allerdings unerlässlich, etwas mit dem DME zu probieren. Studieren Sie außerdem die für Sie wichtigen Teile der Anleitung, damit Sie die Qualitäten des DME für Ihre Bedürfnisse optimal zu nutzen wissen.

Wer sich seine individuelle Tastenbelegung mit den gewünschten Funktionen zusammengestellt hat, wird dieses Programm ganz gewiß nicht mehr missen wollen. so

Der DME wird von Matthew Dillon ständig verbessert. Jeweils die neueste Version (einschließlich Quelltext und deutscher Anleitung) erhalten Sie gegen Zusendung einer Diskette und eines frankierten und an Sie selbst adressierten Umschlages kostenlos vom Autoren dieses Artikels, Heiko Schlichting. Bitte senden Sie dazu Diskette und Rückumschlag mit dem Stichwort »DME« an: Heiko Schlichting, Marinesteig 34a, 1000 Berlin 38.



# Der Blick

**Multitasking-Betriebssysteme werden oft als etwas sehr Kompliziertes angesehen. Tatsächlich ist es nicht ganz einfach, alles zu verstehen, was im Innern beispielsweise des Amiga abläuft.**

**D**as Amiga-Betriebssystem gilt als extrem schwierig zu verstehen. Es wimmelt nur so von Tasks und Prozessen, Libraries und Devices, Ports und Semaphores. Der erste Eindruck von »Xoper« bestätigt diese Erfahrung. Xoper, programmiert von dem Heidelberger Werner Gunther, listet alle »Systemaktivitäten« zusammen mit den verfügbaren Informationen. Aber mit diesem »Einblick« ist Xoper noch längst nicht am Ende. Sie

ser Beziehung unproblematisch sind. Alle Befehle, die katastrophale Folgen haben könnten, sind durch eine längere Eingabe-Sequenz gegen versehentliches Benutzen gesichert. Wir können an dieser Stelle kein ausführliches Wissen über die internen Eigenschaften des Amiga-Betriebssystems vermitteln. Bereits mit einem kurzen Überblick würden wir dieses Heft füllen. Wenn Sie mit dem Verständnis der besprochenen

grammservice-Diskette zu diesem Heft. Falls Sie letztere verwenden, entpacken Sie alle Files im Ordner »Xoper« mit Hilfe von Zoo. Eine Anleitung hierfür finden Sie auf Seite 95. Kopieren Sie die entpackten Files auf eine neue Diskette und starten Sie Xoper entweder vom CLI oder mit einem Doppelklick von der Workbench. Sie sehen ein Fenster, das alle laufenden Tasks und Prozesse anzeigt (Bild 1). Dieses Fenster kann bei Ihnen natürlich etwas anders aussehen, je nachdem, welche Programme bei Ihnen gerade laufen. Aber in jedem Fall ist es doch erstaunlich, was alles läuft, ohne daß Sie davon viel merken. Starten Sie einige neue Tasks, indem Sie einige Programme vom CLI mit »run« starten oder auf der Workbench anklicken. Sie werden feststel-

gnusEd«). Ziehen Sie den Screen nach unten, bis Sie die Taskanzeige von Xoper beobachten können. Sie sehen diese um den neuen Task bereichert (in der Spalte »TASK-NAME« finden Sie den Namen). Gehen Sie von dort zwei Spalten nach links. Die Spalte »CPUSE« zeigt an, in welchem Umfang die einzelnen Tasks den Prozessor belasten. Ziehen Sie jetzt den Editor-Screen ganz nach unten und warten Sie, bis die Liste erneuert wird. Sie werden feststellen, daß der Wert gesunken ist. Ziehen Sie den Screen wieder soweit nach oben, daß Sie etwas eingeben und dabei das Xoper-Fenster beobachten können. Geben Sie beliebige Zeichen ein und beobachten Sie dabei das Xoper-Fenster. Tippen Sie einmal langsam, einmal schnell. Was geschieht dabei? Wenn Sie CygnusEd oder SuperEd verwenden, können Sie den Editor beenden und in einen »Wartezustand« versetzen. Der Wert unter CPUSE

| ID     | TYPE    | STATE   | PRI | CPUSE | NUM | TASKNAME         |
|--------|---------|---------|-----|-------|-----|------------------|
| c16eb8 | Process | Running | 0   | 19.5% | --- | -Xoper-          |
| c1f3d8 | Process | Waiting | 5   | 20.0% | --- | CON              |
| c047a6 | Task    | Waiting | 5   | 4.6%  | --- | trackdisk.device |
| c047a6 | Task    | Waiting | 5   | 4.6%  | --- | trackdisk.device |
| c18748 | Process | Waiting | 1   | 4.2%  | 0   | Workbench        |
| c00a98 | Process | Waiting | 10  | 4.2%  | --- | File System      |
| c05dc0 | Process | Waiting | 10  | 2.9%  | --- | File System      |
| c0b0b0 | Process | Waiting | 10  | 2.9%  | --- | DH0              |
| c11120 | Process | Waiting | 10  | 2.9%  | --- | DH1              |
| c02672 | Task    | Waiting | 20  | 25.1% | --- | input.device     |
| c1d228 | Process | Waiting | 20  | 8.5%  | 4   | Background CLI   |
| c2c500 | Process | Waiting | 5   | 0.0%  | --- | CON              |
| c292c0 | Process | Waiting | 0   | 0.0%  | 2   | SHELL            |
| c10cb8 | Process | Waiting | 0   | 0.0%  | --- | harddisk.device  |

**Bild 1.** Nach dem Aufruf von »Xoper« erhalten Sie eine Liste aller installierten Tasks

können Tasks manipulieren, aus dem System entfernen und in der Priorität verändern. Sie können zusätzliche Informationen über Libraries oder Devices erhalten und nutzlos gewordene Fenster und Screens entfernen.

Aber um alle Funktionen von Xoper zu nutzen, ist einige Erfahrung notwendig. Bevor Sie Befehle verwenden, die Programme beeinträchtigen könnten oder gar entfernen, überlegen Sie sich genau, was dabei geschieht. Grundsätzlich können Sie davon ausgehen, daß alle Befehle, die aus nur einem Buchstaben bestehen, in die-

Gebiete Probleme haben, empfehlen wir Ihnen folgende Bücher:

- Amiga Intern (Band 1 und 2), Data Becker, ISBN 3-89011-104-1 und 3-89011-268-4, Preis je 69 Mark;

- AmigaDOS-Handbuch, Markt & Technik-Verlag, ISBN 3-89090-465-3, Preis 59 Mark.

Alle drei Bände ergänzen sich optimal für das Einarbeiten in die Besonderheiten des Amiga-Betriebssystems. Allerdings sollten Sie schon etwas Erfahrung in einer Programmiersprache gesammelt haben.

Sie finden Xoper auf der Fish-Disk 171 sowie auf der Pro-

```

XOPER V1.2 © 1988 Werner Gunther
USAGE: Xoper [T] [S] [L] [D] [R] [E] [M] [P] [I]
OR Xoper -b
DISPLAY:
[Tasks] [S]=taskstate [L]ibraries [D]eVICES [R]esources
r[E]lident [M]emory [P]orts [I]nterrupts [Q]uit
COMMANDS:
Time (secs)
Taskpri (priority) [processnum] (taskname)
Myprpri (priority)
Signal (mask(hex)) [processnum] (taskname)
Break [processnum] (taskname)
Freeze/Warn [processnum] (taskname)
Cancel [processnum] (taskname)
Snoop [processnum] (taskname)
Pri (priority) (nodename)
Info (name)
Clear (longword(hex))
Hunks [processnum] (procname)
CD (directory) [processnum] (procname)
Unlock (lock)
Openlib | Closelib (libraryname)
Closewindow | Closescreen (title)
Lockdrive | Freedrive (DF:)
Flush,Devices,Currentdir,Files,Locks,Lastalent,More
Taskports,Windows,Screens,Fonts,Windowfonts,Capture
ClrCool,ClrCold,ClrWarn,Usage,Hold,Exit
    
```

**Bild 2.** Dieses Help-Fenster ist jederzeit über den »H«-Befehl aufrufen

len, daß diese Programme sofort der Taskliste hinzugefügt werden.

Starten Sie jetzt einen Editor. Verwenden Sie möglichst ein Programm, das einen eigenen Screen verwendet (z.B. »MicroEmacs«, »SuperEd« oder »Cy-

sinkt dadurch auf Null. Die Gesamtbelastung des Prozessors finden Sie in der obersten Zeile hinter »CPU activity:«. Dieser Wert gibt sozusagen die »Auslastung« des Prozessors an. Die Werte unter CPUSE ergeben zusammengerechnet 100



der Task sein, der sie liest, also Xoper.

**PRI** zeigt die Priorität des Tasks an. Diese ist dafür verantwortlich, wieviel Rechenzeit er vom System zugeteilt bekommt.

**NUM** enthält nur dann einen Wert,

wenn das Programm mit »run« aus dem CLI gestartet wurde. Dann kann es nicht mit Sicherheit über den Namen in **TASKNAME** identifiziert werden.

Unter der Taskliste finden Sie ein Prompt mit einem Cursor. Hier können Sie alle Xoper-Kommandos eingeben. Angenommen, Sie fühlen sich durch den schnellen Wechsel im Bildschirmaufbau gestört. Dann verändern Sie diesen mit dem »time«-Kommando. Geben Sie dazu ein:

```
time 50
```

Sie werden feststellen, daß das Fenster nur noch in langsamem Wechsel erneuert wird. Gleichzeitig sinkt der Anteil von Xoper an der Prozessorausnutzung auf etwa 10 Prozent. Geben Sie statt dessen an:

```
time 0.1
```

dann wird der Bildschirm im schnellen Wechsel erneuert. Der CPUSE-Anteil von Xoper steigt damit auf über 30 Prozent. Wenn Sie dabei den Wert hinter »CPU activity« beobach-

tet haben, ist Ihnen sicher aufgefallen, daß dieser sich fast verdoppelt hat. Unter diesem Aspekt macht die Belastung des Systems durch Xoper weit mehr aus, als der Anstieg von 10 auf 30 Prozent vermuten läßt.

Dieser »Taskwächter« ist zwar ganz nett, um zu erfahren, was sich alles im System tummelt (vielleicht auch, was die Startup-Sequence dort implementiert), aber wenn das alles war – na ja.

Um sich einen Überblick der Fähigkeiten von Xoper zu schaffen, geben Sie »h« ein. Sie sehen die Hilfsseite von Xoper (Bild 2).

Testen Sie zunächst das »Q«-Kommando. Damit beenden Sie Xoper. Starten Sie jetzt erneut. Diesmal allerdings geben Sie im CLI ein:

```
Xoper -b
```

Die Option »-b« bewirkt, daß zunächst nichts geschieht. Xoper wartet lediglich darauf, aktiviert zu werden. Dies geschieht mit der Tastenkombination

Mit dem Kommando »T« erhalten Sie wieder die ursprüngliche Taskliste. Mit »S« erhalten Sie zusätzliche Informationen zu den einzelnen Tasks (Bild 3). Beispielsweise finden Sie in der Spalte »SIGALLOC« die Maske der allozierten Signale. In »SIGWAIT« dagegen finden Sie die Maske der Signale, auf die der Task gerade wartet. Diese muß beim laufenden Task (hier immer Xoper) natürlich Null sein. Unter »SIGRECVD« finden Sie das letzte erhaltene Signal. Die Spalte »PC« zeigt den Stand des Programmzählers (»Program-Counter«) an. Dieser steht bei allen Tasks im Wartezustand auf der Adresse der Exec-Funktion, die für das Wiedereinklinken (»Taskswitching«) verantwortlich ist (fc08b8). Beim laufenden Task dagegen zeigt der Wert immer auf die Stelle des Programms, die die Informationen über die Taskliste beschafft. Alle diese Informationen sind in hexadezimaler Darstellung angezeigt.

Sehen Sie sich mit »L« die Liste der Libraries (Bild 4) an. Diese umfaßt alle Bibliotheken, die sich entweder ständig im ROM befinden oder die gerade von einem Programm benutzt werden.

Die erste Spalte gibt die aktuelle Basisadresse dieser Bibliothek an. Wie Sie leicht erkennen können, ist die Basis-

# ins Innere

Prozent. Dies ist natürlich als 100 Prozent vom Wert in der ersten Zeile zu sehen.

Sehen Sie sich die anderen Spalten an:

**ID** ist die Anfangsadresse des Tasks. Mit dieser Adresse ist er in der Exec-Taskliste »angemeldet«.

**TYPE** kann zwei Werte annehmen: »Task« oder »Process«. Normalerweise wird ein Programm als Prozeß gestartet. Ein Task ist eine Untermenge von einem Prozeß. Er kann auch ausdrücklich von einem Prozeß initialisiert werden. Nähere Informationen hierzu finden Sie im Amiga Intern.

**STATE** erkennt die Werte »Running«, »Waiting«, »Ready« oder »Frozen«. Frozen ist ein Zustand, in dem Programme nur von Xoper gebracht werden können (siehe weiter unten). Als »Ready« gilt ein Task, wenn seine Programmausführung nur deswegen gestoppt ist, weil er keine Rechenzeit mehr erhält. Im Gegensatz dazu sind Tasks im Zustand »Waiting«, wenn Sie auf eine Nachricht warten, die sie wieder aktiviert. Es kann immer nur einen Task ausgelesen werden, kann dies nur

| ID     | STATE   | SIGALLOC | SIGWAIT  | SIGRECVD | PC     | TASKNAME         |
|--------|---------|----------|----------|----------|--------|------------------|
| c16eb0 | Running | e000ffff | -----    | 00000000 | c356f8 | -Xoper-          |
| c1f3d8 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | CON              |
| c02672 | Waiting | c000ffff | c0000000 | 00000000 | fc08b8 | input.device     |
| c1d220 | Waiting | c000ffff | c0000000 | 00000000 | fc08b8 | Background CLI   |
| c0142e | Waiting | 0000ffff | 00000300 | 00000000 | fc08b8 | trackdisk.device |
| c047a6 | Waiting | 0000ffff | 00000300 | 00000000 | fc08b8 | trackdisk.device |
| c11120 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | DH1              |
| c00a98 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | File System      |
| c05dc0 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | File System      |
| c0b0b0 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | DH0              |
| c18748 | Waiting | 0000ffff | 00000000 | 00000110 | fc08b8 | Workbench        |
| c2c500 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | CON              |
| c292c0 | Waiting | 0000ffff | 00000100 | 00000000 | fc08b8 | SHELL            |
| c10cb8 | Waiting | 0000ffff | 00000000 | 00000100 | fc08b8 | harddisk.device  |

Bild 3. Mit »TaskState« erhalten Sie weitere Informationen über die Signale der einzelnen Tasks

| NODE   | CNT | VER | REV | FLAGS    | LIBRARY NAME      |
|--------|-----|-----|-----|----------|-------------------|
| c00276 | 2   | 33  | 192 | 00000100 | exec.library      |
| c018f6 | 0   | 33  | 121 | 00000100 | expansion.library |
| c01dfe | 1   | 33  | 97  | 00000100 | graphics.library  |
| c038fe | 1   | 33  | 33  | 00000100 | layers.library    |
| c03cb4 | 1   | 33  | 190 | 00000100 | intuition.library |
| c04628 | 1   | 33  | 7   | 00000100 | mathfp.library    |
| c04af8 | 32  | 33  | 124 | 00000100 | dos.library       |
| c04f02 | 1   | 33  | 90  | 00000100 | ramlib.library    |
| c069ec | 2   | 33  | 127 | 00000100 | icon.library      |
| c09f00 | 0   | 34  | 4   | 00000100 | version.library   |
| c2cbe8 | 1   | 1   | 4   | 00000100 | dres.library      |
| c17604 | 0   | 31  | 1   | 00000100 | arp.library       |
| c123dc | 0   | 2   | 0   | 00000100 | environment       |

Bild 4. Libraries können von einer unterschiedlichen Anzahl Programmen benutzt werden

<rechte-Amiga linke-Amiga x>. Diese Form des Aufrufs eignet sich besonders, um Xoper bereits in der Startup-Sequence zu starten und sozusagen auf Abruf bereitzuhalten. Beenden Sie jetzt nochmals mit »Q«. Drücken Sie darauf wieder dieselbe Tastenkombination wie beim Start. Xoper ist immer noch aktiv.

adresse der »exec.library« nicht die »4«. Dies ist ein häufiges Mißverständnis. In der Speicherstelle 4 liegt lediglich die (auch hier angegebene) Adresse der Basis von Exec, aber nicht die Basis selbst. Die nächste Spalte enthält die Anzahl der Programme, die auf die jeweilige Bibliothek zugreifen. Dabei werden auch Zugriffe



von anderen Libraries mitgezählt. Nur so ist es zu verstehen, daß 32 Programme auf die »Dos.library« zugreifen. In der dritten Spalte finden Sie die Versionsnummer und in der vierten die Revisionsnummer der Bibliotheken. Die letzte Spalte vor dem Library-Namen enthält den Wert der »Flags«-Variablen aus der Library-

chen. Die meisten Namen sind Ihnen bereits aus den drei vorherigen Tabellen bekannt.

Alle Devices und Resources sowie die meisten Libraries sind resident, das heißt, sie werden nicht - wie die meisten Anwendungsprogramme - bei Bedarf in den Speicher geladen und danach wieder entfernt. Statt dessen sind sie dort fest

Sehen Sie sich die Prioritäten der einzelnen Einträge an. Sie finden ungewöhnlich hohe Werte bei allen System-Dateien. Bei Ihren eigenen Programmen sollten Sie die Priorität möglichst nicht über 20 setzen, um diese »Hackordnung« nicht durcheinander zu bringen. Der Eintrag mit der höchsten Priorität ist die exec.library. Diese muß ständig verfügbar sein, sie ist sozusagen »lebenswichtig« für das System. Das gleiche gilt für die Resources und Devices. Einträge mit dem Typ »unknown« (im »TYPE«-Feld) sind meist residente Anwendungsprogramme, die sogenannte »HotKeys« abfragen oder mit dem Resident-Befehl fest im Speicher verankert wurden.

Mit dem »M«-Befehl erhalten Sie Informationen über den Zustand des RAM-Speichers. Neben den freien Bytes werden die Unter- und Obergrenzen der Speicherbereiche angezeigt sowie die Priorität der verschiedenen Speicherarten (»CHIP« oder »FAST«).

Wesentlich interessanter ist allerdings die Liste der verwendeten Ports (Bild 6). Diese erhalten Sie mit dem »P«-Befehl. Ports sind die Message-Ports, über die einzelne Tasks kommunizieren können. Beispielsweise erhält ein Programm,

steht die Adresse des Ports in hexadezimaler Schreibweise.

Die letzte Liste erreichen Sie mit »I« (Bild 7). Es handelt sich dabei um die beim System angemeldeten Interrupts. Sie finden in der »NODE«-Spalte die zugehörige Adresse. Unter »DATA« stehen die Adressen der Datensegmente der Interrupt-Struktur, unter »CODE« die der Programm-teile, die beim Auslösen des Interrupts angesprungen werden sollen.

Alle diese Funktionen können Sie auch kombinieren, indem Sie mehrere gleichzeitig angeben. Die einzelnen Listen werden dann nacheinander angezeigt. Das gleiche Ergebnis erreichen Sie, wenn Sie Xoper mit diesen Befehlen als Parameter aufrufen, beispielsweise:

```
Xoper tsmi
```

In diesem Fall würde nach dem Start die Task-, TaskState-, Memory- und Interrupt-Liste jeweils nacheinander im Fenster erscheinen.

Mit allen bisherigen Funktionen konnten Sie nicht viel mehr anfangen, als sich im System umzusehen und Informationen gewinnen. Das ist einiges mehr, als Sie von anderen Programmen an Information erhalten. Aber die eigentliche Leistung von Xoper fängt hier erst an.

Sie können mit Xoper auch direkten Einfluß auf das System nehmen. Geben Sie beispielsweise ein:

```
T
Mypri 20
```

Mit »T« schalten Sie in die Taskanzeige. Beobachten Sie die Priorität von Xoper, nachdem Sie die zweite Zeile eingegeben haben. Sie steht anfangs auf 0 und ab dem nächsten Erneuern des Fensters ist sie 20. Mit »Mypri« verändern Sie also die Priorität von Xoper. Das gleiche können Sie mit jedem beliebigen Task machen, indem Sie statt mypri das Kommando »Taskpri« verwenden. Es benötigt als ersten Parameter die gewünschte Priorität. Als zweiten Parameter übergeben Sie den Namen des Tasks. Ist dieser »Background CLI«, so muß vor dem Namen noch die Prozeß-Nummer übergeben werden, damit der Prozeß eindeutig identifiziert werden kann.

Mit dem »taskpri«-Befehl haben Sie jedoch nur Zugriff auf Programm-Strukturen. Um die Priorität innerhalb von verketteten Listen zu ändern, benötigen Sie einen anderen Befehl. »pri« versorgen Sie mit zwei Parametern: der gewünschten Priorität und dem Namen des »Nodes«

| ADDR   | PRI | FLAGS    | VER | TYPE     | RESIDENT NAME     |
|--------|-----|----------|-----|----------|-------------------|
| fc00b6 | 120 | 00000000 | 33  | Library  | exec.library      |
| fc4afc | 110 | 10000001 | 33  | Library  | expansion.library |
| fe4800 | 100 | 00000001 | 33  | Resource | potgo.resource    |
| fe4fe4 | 100 | 00000001 | 33  | Resource | keymap.resource   |
| fc450c | 80  | 00000001 | 33  | Resource | cia.resource      |
| fc4794 | 70  | 00000001 | 33  | Resource | disk.resource     |
| fe4774 | 70  | 00000001 | 32  | Resource | nisc.resource     |
| fe49cc | 70  | 10000000 | 33  | Library  | panlib.library    |
| fc3378 | 65  | 00000001 | 33  | Library  | graphics.library  |
| fe502e | 60  | 00000001 | 33  | Device   | keyboard.device   |
| fe507a | 60  | 00000001 | 33  | Device   | gameport.device   |
| fe90ec | 50  | 00000001 | 33  | Device   | timer.device      |
| fc34cc | 40  | 10000001 | 33  | Device   | audio.device      |
| fe50c6 | 40  | 00000001 | 33  | Device   | input.device      |
| fe0d90 | 31  | 00000001 | 33  | Library  | layers.library    |
| fe510e | 20  | 00000001 | 33  | Device   | console.device    |
| fe90e4 | 20  | 00000001 | 33  | Device   | trackdisk.device  |
| fd3f5c | 10  | 10000001 | 33  | Library  | intuition.library |
| fc323a | 5   | 00000001 | 33  | Unknown  | alert.hook        |
| fe424c | 0   | 10000001 | 33  | Library  | nathftp.library   |
| feb400 | 0   | 00000000 | 33  | Task     | workbench.task    |
| ff425a | 0   | 00000000 | 33  | Library  | dos.library       |
| fe8004 | -60 | 00000001 | 33  | Unknown  | strap             |

Bild 5. Die höchste Priorität aller residenten Programme hat immer die Exec.library

Struktur. Deren Bedeutung ist nirgends dokumentiert. Da sich der Wert nicht ändert, nehmen wir an, daß in der aktuellen Betriebssystemversion keine Bedeutung vorgesehen ist.

Öffnen Sie eine beliebige Library, die in Ihrem Libs-Ordner liegt. Verwenden Sie dazu den Befehl

```
Openlib <Name>
```

Setzen Sie für »Name« einen beliebigen Namen ein, der im Libs-Ordner enthalten ist und noch nicht in der Liste von Xoper auftaucht. Nach dem nächsten Erneuern des Fensters ist dieser Name in der Liste enthalten. Dieselbe Library können Sie gefahrlos mit

```
Closelib <Name>
```

entfernen. Bei allen anderen Libraries sollten Sie mit diesem Befehl sehr vorsichtig sein. Sie könnten von anderen Programmen noch gebraucht werden.

Das Auslesen der »Resources« und »Devices« funktioniert analog zu den Libraries, daher muß an dieser Stelle nicht weiter darauf eingegangen werden.

Interessante Informationen finden Sie dagegen in der Liste der residenten Programme (Bild 5), die Sie mit »E« errei-

| NODE   | PORTNAME      | FLAGS   | SIGBT | NUM | SIGTASK         |
|--------|---------------|---------|-------|-----|-----------------|
| c08c20 | Timer Port    | Signal  | 31    | 0   | Background CLI  |
| c09340 | IDCMP         | Ignore  | 30    | 0   | Background CLI  |
| c1c890 | IDCMP         | Signal  | 30    | 0   | Background CLI  |
| 01526e | Workbench     | Signal  | 31    | 0   | Workbench       |
| 015290 | Workbench     | Signal  | 31    | 0   | Workbench       |
| c054e0 | IDCMP         | Ignore  | 30    | 1   | -----           |
| c12a56 | serial.device | Softint | 0     | 0   | serial.device   |
| c12ab6 | serial.device | Softint | 0     | 0   | serial.device   |
| c38814 |               | Signal  | 0     | 0   | -Xoper-         |
| c10d14 |               | Signal  | 0     | 0   | harddisk.device |
| c2960c |               | Signal  | 0     | 0   | SHELL           |
| c346c4 |               | Signal  | 0     | 0   | CON             |
| c187a4 |               | Signal  | 0     | 0   | Workbench       |
| c1d204 |               | Signal  | 0     | 0   | Background CLI  |
| c00af4 |               | Signal  | 0     | 0   | File System     |
| c05e1c |               | Signal  | 0     | 0   | File System     |
| c0b10c |               | Signal  | 0     | 0   | DH0             |
| c1117c |               | Signal  | 0     | 0   | DH1             |
| c21ca4 |               | Signal  | 0     | 0   | CON             |

Bild 6. Ein wichtiger Bereich jeder Multitasking-Maschine wie dem Amiga ist das Message-Port-System

»verankert«. Daneben gibt es die Möglichkeit, eigene Programme speicherresident zu schreiben oder mit Hilfe des »Resident«-Befehls der Workbench 1.3 in diese Liste aufzunehmen. Programme, die auf Tastendruck aktiv werden (wie »CygnusEd«, »TurboPrint«) nutzen diese Möglichkeit.

wenn der Anwender das CloseGadget betätigt, eine entsprechende Meldung von Intuition in den Message-Port »gelegt«. In der Liste finden Sie alle installierten Ports mit ihren Namen (wenn möglich) und den Signalen, auf die sie »reagieren« (in der Spalte »SIGBT«). In der ersten Spalte (»NODE«)



```
Xoper V1.2 @ 1988 Werner Gunther
```

| NODE   | DATA   | CODE   | PRI | TYPE       | INTERRUPT NAME  |
|--------|--------|--------|-----|------------|-----------------|
| c01b2c | c01ade | fc4a80 | 59  | Disk Block | -----           |
| c016aa | c01680 | fc461c | 0   | CIAA       | ciaa.resource   |
| c01e48 | c01dfe | fc6cdc | 0   | VBea       | -----           |
| c02224 | c020f4 | fe5cc8 | 0   | VBea       | gameport.device |
| c02320 | c022f6 | fe96da | 0   | VBea       | timer.device    |
| c01e74 | c01dfe | fc6d8e | 0   | Blitter    | -----           |
| c02478 | c02446 | fc35a4 | 0   | Audio Ch.0 | audio.device    |
| c024e4 | c024b2 | fc35a4 | 0   | Audio Ch.1 | audio.device    |
| c02550 | c0251e | fc35a4 | 0   | Audio Ch.2 | audio.device    |
| c025bc | c0258a | fc35a4 | 0   | Audio Ch.3 | audio.device    |
| c01b42 | c01ade | fc4a98 | 0   | Disk Sync  | -----           |
| c01742 | c01710 | fc4610 | 0   | CIAA       | ciab.resource   |

**Bild 7. Interrupts spielen eine bedeutende Rolle bei der Hardware-Programmierung**

in der Liste. Dabei kann es sich um ein Device, Ressource oder auch um einen Speicherbereich (CHIP- oder FAST-RAM) handeln.

Haben Sie schon einmal ein Programm geschrieben, einen Testlauf gestartet, und dann festgestellt, daß Sie eine Endlosschleife eingebaut haben? Wie nützlich wäre es doch, wenn man statt des hier üblichen Drei-Finger-Griffs nachträglich eine Abbruchbedingung einbauen könnte. Der Break-Befehl macht's möglich. Er erlaubt den Abbruch eines beliebigen Programms mit <CTRL c> oder einer Batch-Datei mit <CTRL d>. Dies gilt natürlich auch für Programme, für die Sie dies ursprünglich nicht vorgesehen hatten. Achten Sie aber darauf, daß Sie vorher das richtige Fenster aktivieren, denn auch Xoper läßt sich mit <CTRL c> abbrechen.

Wollten Sie schon immer wissen, wie einzelne Programmsegmente intern verkettet sind? Dann ist der »hunks«-Befehl genau richtig für Sie. Er wirkt im Zusammenhang mit einem beliebigen Prozeß oder Task und zeigt alle belegten Speicherbereiche an.

Eine ähnliche Funktion hat der »Snoop«-Befehl. Er »verfolgt« ein Programm während der Laufzeit und führt eine Liste über alle belegten und freigegebenen Speicherbereiche.

Dies kann mit <CTRL c> im Xoper-Fenster abgebrochen werden. Neben der Liste des belegten Speichers erhalten Sie die Information, mit welchen Attributen (»MEMF\_CHIP«, »MEMF\_PUBLIC« etc.) dieser belegt wurde.

Ein hilfreicher Befehl ist »Locks«. Er wird ohne Parameter eingegeben und zeigt alle gesetzten FileLocks mit Adresse und AccessMode. Eine typische Ausgabe sehen Sie in Bild 8. Damit kann häufig erklärt werden, warum Sie immer wieder aus Ihnen unverständlichen Gründen die Meldung erhalten, »Please insert volume... in any drive«. Meist ist noch ein Lock auf eine Datei oder ein Directory dieser Diskette gesetzt.

Zusätzliche Informationen über den System-Zustand erhalten Sie mit den Kommandos »devices«, »capture«, »Screens«, »windows« und »info«. Devices bringt eine Liste aller physikalischen Geräte auf den Bildschirm. Sektororientierte Geräte (Massenspeicher) werden mit der Anzahl der Schreib-/Leseköpfe, Spuren, Sektoren und Puffer angezeigt. Daneben ist im »STATE«-Feld vermerkt, ob das Device im Moment aktiv ist. Mit capture erhalten Sie eine Liste der Reset-Vektoren. Diese bilden eine Ansatzstelle für resetfeste Programme (beispielsweise auch Viren). Mit Screens erhalten Sie eine Liste aller ge-

| LOCK     | ACCESS | SIZE  | PATH/FILENAME            |
|----------|--------|-------|--------------------------|
| 3014a5   | Read   | ----- | AmigaLibDisk171:Xoper/   |
| 30148b   | Read   | 3899  | Amok#2:DMError/DMErr.doc |
| 3014c9   | Read   | ----- | 20MB_dh0:System/         |
| 302565   | Read   | ----- | 20MB_dh0:System/         |
| 30042b   | Read   | ----- | 20MB_dh0:                |
| 3004b1   | Read   | ----- | 20MB_dh0:libs/           |
| 30206b   | Read   | ----- | 20MB_dh0:devs/           |
| 30045f   | Read   | ----- | 20MB_dh0:fonts/          |
| 300473   | Read   | ----- | 20MB_dh0:C/              |
| 300487   | Read   | ----- | 20MB_dh0:L/              |
| 302065   | Read   | ----- | 20MB_dh0:S/              |
| (RETURN) |        |       |                          |

**Bild 8. Mit »Locks« erhalten Sie eine Liste aller für laufende Programme eingestellten Locks auf Dateien oder Directories**

öffneten Screens, Windows macht das gleiche für Fenster. Zusätzlich wird bei Fenstern angegeben, auf welchem Screen sie geöffnet wurden.

Sie kontrollieren damit, ob Fenster vorhanden sind, die Sie keinem Programm zuordnen können. In diesem Fall können Sie die Fenster nachträglich mit »Closewindow« (siehe unten) schließen. Info liest die vom Autoren »hinterlassenen« Kommentare in der Library- oder Device-Struktur.

Hatten Sie schon gelegentlich den Verdacht, daß manche Programme nicht allen belegten Speicher freigeben? »Flush« versucht, solche Speicherbereiche aufzuspüren und freizugeben. Gleichzeitig werden unbenutzte Bibliotheken aus dem Speicher entfernt und Devices geschlossen.

Manchmal kann es verwirrend sein, wenn verschiedene Programme unterschiedliche Directories zum aktuellen gemacht haben. Noch größer wird das Durcheinander, wenn einzelne Programme verlassen werden, ohne den Lock zurückzugeben. Mit »CD« können Sie für jedes beliebige Programm den Pfad neu setzen. Sie sollten damit aber **äußerst** vorsichtig umgehen. Dieser Befehl kann zu unkontrollierbaren Reaktionen führen.

Das gleiche gilt für alle folgenden Befehle. Wenden Sie diese nur an, wenn Sie genau wissen, was Sie bewirken. Sie

Der »cancel«-Befehl beendet nur einen Task. Er läßt eventuell geöffnete Fenster und Screens unberührt. Diese entfernen Sie mit folgenden Befehlen:

**CloseWindow** und **CloseScreen** schließen nicht mehr benötigte Fenster und Screens. Sie übergeben den Namen als Argument, den Sie eventuell vorher mit »windows« ausgelesen haben.

Wenn Sie die letzten drei Befehle verwenden, sollten Sie sich stets an die korrekte Reihenfolge halten: Entfernen Sie zuerst den Task, danach eventuell nicht benötigte Windows und erst dann die zugehörigen Screens (für Xoper ist es gleichgültig, was Sie schließen, es führt vollkommen kritiklos Ihre Wünsche aus). Wenn Sie diese Reihenfolge nicht einhalten, kann es sein, daß der Task noch einmal auf sein vermeintlich noch vorhandenes Fenster zugreift, das Sie ihm soeben entzogen haben. Die Folge ist der blinkende rote Kasten, den Sie vermutlich vermeiden wollten.

Der letzte hier besprochene Befehl ist »signal«. Mit diesem können Sie ein beliebiges Programm »anfunken«, wenn es einen Message-Port installiert hat. Sie senden einfach das Signal, auf das es gerade wartet (das Sie vorher aus der Taskstate-Liste gelesen haben). Einige Programme warten lediglich auf eine Aktivierung durch ein Signal - vielleicht kennen Sie das Beispiel für die Abfrage

## Programmservice

Zu diesem Heft ist eine Programmservice-Diskette erhältlich (siehe Seite 33). Auf dieser finden Sie eine Auswahl der in diesem Heft vorgestellten PD-Software, darunter auch »Xoper«.

Bereits für diese Auswahl war es nötig, alle Files mit Hilfe von »Zoo« zu packen. Zu diesem Programm finden Sie eine Anleitung auf Seite 95 sowie auf der Diskette. Sie erhalten diese Diskette mit dem diesem Heft beigelegten Bestellschein.

sollten möglichst alle Dateien gespeichert haben, bevor Sie eines dieser Kommandos starten. Alle diese Befehle sind dafür gedacht, »Datenmüll« aus dem Speicher zu räumen, wenden Sie sie **niemals** auf »gesunde« Bereiche an. Sie können damit sehr leicht Abstürze provozieren.

**cancel** wirft den Task, dessen Namen Sie übergeben, aus der Taskliste. Dies kann sehr hilfreich sein, wenn ein Task mit dem beliebigen »Task held. Finish all Disk activity«-Requester sein Leben ausgehaucht hat. Wenden Sie den Befehl möglichst nicht auf andere Tasks an, diese könnten das restliche System mit in den Abgrund reißen.

des Message-Ports aus dem ROM-Kernel Reference Manual, das nur das SigBit des Message-Ports ausliest, statt mit »GetMsg« die komplette Nachricht zu holen. Alle Programme, die auf ähnliche Weise warten, können Sie mit diesem Befehl »wecken«. Wenn allerdings darauf der Message-Port ausgelesen wird, deckt das Programm Ihren »Schwindel« auf. Denn in der aktuellen Version ist es nicht möglich, eine komplette Message zu übergeben.

Mit seinen leistungsfähigen Funktionen erlaubt Xoper einen tiefen Einblick in den Amiga. Gehen Sie aber vorsichtig mit allen Befehlen um, die Programmcode manipulieren.



# Public Domain

## - was steckt dahinter?

Von Dietmar Inäbnit

**P**ublic Domain – bei diesen zwei Worten gehen die Meinungen der Amiga-Anwender weit auseinander. Was für die eine Gruppe »unnützes Zeug« ist, das »so-wieso nicht funktioniert«, ist für die andere eine unerschöpfliche Fundgrube an Spielen, Utilities und vielem mehr. Bevor wir die konträren Gesichtspunkte untersuchen, müssen wir zuerst einmal wissen, was PD-Software überhaupt ist, woher sie kommt und wer diese Programme schreibt.

Public Domain-Software ist nicht gleich Public Domain-Software. Es gibt ein paar kleine Unterschiede, die der Anwender beachten sollte. Eigentlich gibt es drei verschiedene Arten von PD-Software:

### Die Wurzeln der PD

**Die Shareware.** Das Konzept der Shareware beruht darauf, daß ein Entwickler ein Programm erstellt und es möglichst vielen Anwendern zur Verfügung stellt. Entspricht dieses Programm den Erwartungen des Anwenders, so überweist dieser an den Entwickler einen geringen Beitrag und erwirbt somit die Benutzungslizenz für das Programm. Sehr häufig kommt es vor, daß man nach der Bezahlung des Programmes eine ausführliche Dokumentation erhält.

**Die Freeware.** Das Konzept der Freeware ist eigentlich das gleiche wie bei der Shareware. Die Programme können ebenfalls frei kopiert und weitergegeben werden. Das Urheberrecht bleibt jedoch immer beim Autor.

**Die original Public Domain.** Mit dieser Software dürfen Sie alles anstellen, was Ihr Herz begehrt. Sie dürfen das Programm auseinandernehmen, einzelne Teile für eigene Programme verwenden, oder was Ihnen sonst noch einfällt. All diese Möglichkeiten stehen of-

**Was versteckt sich hinter diesen zwei Worten, die für viele das Glück auf Erden bedeuten? Gibt es den typischen PD-Anwender? Wie sollte man beim Austesten einer PD-Diskette vorgehen? Fragen über Fragen... Antwort gefällig?**

fen, da diese Software nicht mit einem Urheberrecht (Copyright) geschützt ist.

Eines ist ganz wichtig: Wenn Sie eine PD-Diskette im Handel erwerben, entrichten Sie keine Gebühr für die Programme, sondern bezahlen den Material- und Zeitaufwand, den der Händler für das Kopieren hatte.

Die Anfänge der PD- und Shareware-Software sind in Amerika zu finden. Lange Zeit waren Computer fast ausschließlich an Universitäten verbreitet. Die Software war nur sehr spärlich gesät und noch sündhaft teuer. Die Studenten warfen deshalb ihre selbst geschriebenen Programme in einen gemeinsamen Pool. Der Leitgedanke war, daß jedes Programm jedem zugänglich gemacht wurde. Die Entwicklung lief immer weiter. Die eigenen Programme landeten regelmäßig im Software-Pool und man nahm sich aus dem Pool, was man gerade an Software brauchte. Bald war es soweit, daß man zuerst beim PD-Verwalter nachfragte, ob ein gewisses Programm schon existierte. Die Autoren dieser Programme erhielten Kritiken und Korrekturvorschläge von ihren Kollegen. Viele gute Programmierer wurden auch durch die rege Tätigkeit im PD-Bereich von Software-Häusern entdeckt. Schnell wurden Vorteile klar, die sich kein Programmierer entgehen läßt: Das Austesten der Software durch viele verschiedene Anwender ist einer davon. Der steigende Bekanntheitsgrad guter Software-Entwickler ist ein weiterer. Es ist

also kein Wunder, daß der Begriff »Public Domain« heute in der ganzen Welt bekannt ist.

Der Berg an PD-Software wurde immer größer. Studenten waren längst nicht mehr die einzigen PD-Lieferanten. Bald darauf kamen die ersten cleveren Geschäftsleute auf die Idee, PD-Software professionell zu vertreiben.

Oft stellt man sich die Frage: Was sind denn das für Menschen, die PD-Programmierer? Warum geben die denn ihre Programme in den PD-Pool?

Die Antworten auf diese zwei Fragen sind einerseits ganz einfach, andererseits sehr schwer zu beantworten. Man kann doch nicht einfach sagen, daß ein PD-Programmierer sehr bleich und schlecht rasiert ist, oder viereckige Augen besitzt. Nein, das wäre zu einfach. PD-Programmierer erkennt man nicht an ihrem Aussehen. Alle PD-Programmierer haben eigentlich nur eines gemeinsam: Sie arbeiten gerne mit ihrem Computer und lassen Andere an ihrem Erfolg und ihrer Erfahrung teilhaben. Zwei kleine Beispiele geben Antwort auf die zweite Frage und verdeutlichen Ihnen die Beweggründe der PD-Autoren.

Ein Programmierer hat eine superschnelle Sortieroutine geschrieben. Er glaubt, daß diese Routine vielen Programmierern weiterhelfen wird. Aus diesem Grund möchte er, daß möglichst viele seine Routine erhalten. Was macht er? Er schickt seine Routine als Shareware oder Public Domain an einen PD-Vertrieb.

Ein anderer Programmierer schreibt einen neuen Texteditor. Da er noch nicht ganz sicher ist, ob alle Fehler beseitigt sind, gibt er eine Test-Version (Beta-Version) in den PD-Pool. Er erhofft sich damit zwei Dinge: Er erwartet ein Echo auf sein Programm. Dieses Echo kann eine Hilfe zur Beseitigung eines Fehlers (Bugs), eine Kritik oder der freundliche Brief eines begeisterten Anwenders sein. Außerdem steigert der Autor dadurch seinen Bekanntheitsgrad und wird vielleicht von einer Softwarefirma als aussichtsreiches Talent entdeckt.

Das Ganze ist in der Wirklichkeit nicht ganz so rosig, wie es scheint. Der Programmautor erhält oftmals nicht das gewünschte Echo. Und viele Menschen »vergessen« auch das Programm zu bezahlen, wenn es als Shareware vertrieben wird. Schreiben Sie doch einmal einen netten Brief an einen PD-Autor, dessen Programm Sie häufig benutzen, oder bezahlen Sie die angegebene Benutzergebühr für ein Shareware-Programm. Dabei können sehr interessante Kontakte entstehen. Sie geben einem anderen Menschen damit auch das Gefühl der Anerkennung geleisteter Arbeit.

Ein Nebenprodukt der Public Domain ist die Veröffentlichung eines Listings in einem Computermagazin (zum Beispiel dem AMIGA-Magazin). Das Programm wird einer großen Öffentlichkeit vorgestellt. Es ist schon häufig vorgekommen,

### Blühender Nebenprodukt

das dies der erste Schritt in eine große Karriere war. Der Autor wird von einem Software-Haus entdeckt und kann danach für diese Firma Programme schreiben. Dies ist ein Aufruf an alle Programmierer, die noch irgendwo ein Superlisting parat haben. Lassen Sie Ihr Listing nicht in einer Schublade verstauben, sondern senden Sie



es uns. Lassen Sie sich keinesfalls von Fehlschlägen entmutigen.

Jeder kennt dieses eigen tümliche Gefühl, wenn man etwas gekauft hat, es ausprobieren will, und es funktioniert nicht. In diesen Situationen finden sich im menschlichen Wortschatz oft nicht mehr die Worte, um die entstehende Stimmung auszudrücken (Worte, die doch gefunden werden, können wir an dieser Stelle nicht drucken). Genau in dieser Situation befinden sich viele Amiga-Besitzer, die zum ersten Mal eine PD-Diskette gekauft haben. Sie legen die Diskette ins Laufwerk und... nichts passiert. Ohne jede Grundkenntnisse des Amiga-Systems funk-

## Auf die Plätze, Fertig, Los!

tioniert das Ganze wohl doch nicht. Sie sollten sich im CLI etwas auskennen, über allgemeine Kenntnisse der Struktur von Dateiverzeichnissen verfügen. Das soll nicht bedeuten, daß Anfänger keine Chance haben, mit PD-Software zurechtzukommen. Es genügt schon, wenn Sie die Grundbefehle wie CD, DIR, ED etc. kennen. Wenn Sie die folgenden Tips im Umgang mit einer PD-Diskette beherrsigen, wird sich Ihr Adrenalin-Spiegel sicher nicht verändern.

Schalten Sie den Amiga ein und legen die PD-Diskette ins Laufwerk. Es passiert nichts? Des Rätsels Lösung ist einfach:

Die Diskette ist nicht bootfähig (siehe "Install"-Befehl im Handbuch). Es gibt noch eine zweite Möglichkeit: der Rechner bootet und Sie sind plötzlich im CLI. Auf dieser Diskette wurde zwar der Install-Befehl ausgeführt, aber es existiert keine Startup-Sequence. Spätestens jetzt ist es an der Zeit, Ihnen etwas über den Aufbau einer PD-Diskette zu erzählen.

Der Hersteller versucht, auf einer PD-Disk soviel Programme wie möglich zu kopieren. Das heißt, alles Überflüssige muß weg. Überflüssig sind die CLI-Befehle, die Workbench und die Icons der Programme.

Das Beste ist, Sie starten den Amiga mit einer Workbench-Diskette und legen die PD-Disk ins df1: (Sollten Sie nur ein Laufwerk besitzen, so wird es etwas schwieriger. Alle notwendigen Dateien, sowie die CLI-Befehle müssen in diesem Fall ins RAM kopiert und mit dem ASSIGN-Befehl umgeleitet werden). Klicken Sie jetzt das Icon der PD-Disk an. Grundsätzlich können nun zwei Dinge geschehen:

1. Ein Fenster wird geöffnet und es erscheinen Programmsymbole (Icons) und Schubladen
2. Es erscheint ein Fenster ohne Icons.

Wenn Icons im Fenster vorhanden sind, können Sie jetzt versuchen, die einzelnen Programme zu starten und auszutesten. Sie haben keine Icons? Glauben Sie nicht, daß die Diskette leer ist. Die Icons sind nur das Opfer einer Sparmaßnahme. Diese sollen nur den Um-

gang mit dem Amiga erleichtern. Jedes Icon verbraucht Platz auf der Diskette, ist aber für den korrekten Programmablauf nicht erforderlich (Icons erkennt man im CLI an der Endung ».info«).

Als nächsten Schritt begeben wir uns ins CLI des Amiga und schauen uns einmal das Inhaltsverzeichnis der PD-Diskette an. Das Root-Verzeichnis einer PD-Diskette besteht meistens aus einer langen Liste von Unterverzeichnissen (z.B. SOUND-DEMO (DIR)). Jedes Verzeichnis steht meistens für ein Programm auf der Diskette. Mit dem Befehl »cd« wechseln Sie in das gewünschte Unterverzeichnis. Beispiel:

```
cd sound-demo <RETURN>
```

Wenn Sie nun DIR eingeben, werden Sie meistens den gleichen Programmnamen mehrmals sehen, aber mit verschiedenen Endungen (.doc, .txt, .obj, usw.). Der Eintrag ohne eine Endung ist fast immer das lauffähige Programm.

Die Anleitung zu diesem Programm wird oft mit der Endung ».doc« oder ».txt« gekennzeichnet. Diese Anleitungen sind meistens ASCII-Files. Sie können sich diese mit dem type-Befehl ausdrucken, oder nur am Bildschirm anzeigen lassen. Natürlich können Sie die ASCII-Files auch in einer Textverarbeitung (z.B. BeckerText) betrachten. Es ist sehr wichtig, diese Anleitungen genau durchzulesen, um unliebsame Überraschungen zu vermeiden. Alle anderen Endungen

bei den Programmnamen deuten darauf hin, daß dies der Quell-Code (Source) des Programmes ist. Je nach vom Entwickler verwendeter Programmiersprache kann das ein C-Source, Modula2-Source, oder ein Assembler-Source sein. Diese Files sind nur für Programmierer interessant, eine detaillierte Anleitung für die Handhabung von PD-Disketten würde den Rahmen dieses Artikels sprengen. Aber, Sie werden schnell feststellen, daß es nicht so schwierig ist, wie es sich anhört. Weitere Hinweise zur Handhabung der PD-Disketten finden Sie auf Seite 125 in diesem Heft.

Als PD-Anwender sollten Sie im Prinzip nur drei Voraussetzungen erfüllen. Sie kennen das CLI, den Aufbau der Dateiverzeichnisse und haben Spaß am Testen neuer Programme.

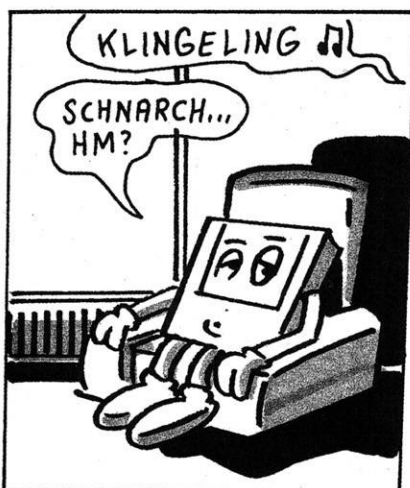
Ganz am Anfang dieses Artikels steht, daß die Meinungen der Amiga-Anwender bei PD-Software weit auseinandergehen. Jetzt dürfte Ihnen dies klar sein. Wenn jemand ohne die

## Spannender als ein Krimi

oben genannten Voraussetzungen eine PD-Diskette kauft, wird er später vielleicht nichts mehr von diesem interessanten Software-Pool wissen wollen. Jeder PD-Freak kann Ihnen versichern, daß der Inhalt einer PD-Diskette spannender als ein Krimi ist. Wann gehören Sie zum PD-Clan? rs

# \*AMICUS

DIE ABENTEUER EINER FLOPPY-DISC





# Frisieren Sie die Workbench

Es ist Zeit für einen Frühjahrsputz auf der Workbench-Diskette! Löschen Sie den ganzen »Datenschrott«, den Sie sowieso nicht gebrauchen können. Werten Sie Ihre Workbench statt dessen

mit leistungsfähigen PD-Programmen auf.  
Alle nötigen Arbeitsschritte für den Frühjahrsputz lernen Sie im folgenden Artikel kennen.

Von Peter Sack

Die Original-Workbench ist so angelegt, daß die auf ihr enthaltenen Programme und Dateien möglichst alle Bereiche abdecken. Dadurch wird aber soviel unnötiger Ballast mitgeschleppt, daß kein Platz für sinnvolle Erweiterungen mehr vorhanden ist. Dateien wie die im Fonts-Ordner enthaltenen werden Sie aber nur brauchen, wenn Sie mit einer Textverarbeitung schreiben. Der sinnvollste Weg wäre also, sich für diese Textverarbeitung eine eigene Workbench anzulegen und die Fonts auf der Standard-Workbench zu löschen.

In diesem Artikel erfahren Sie, welche Dateien Sie auf Ihrer Workbench bedenkenlos löschen können und mit welchen PD-Programmen Sie den gewonnenen Speicherplatz sinnvoller verwenden. Zunächst sollten Sie sich eine Arbeitskopie der Workbench-Diskette machen, die Sie täglich benutzen.

Beginnen wir mit dem Löschen der überflüssigen Programme: Entfernen Sie den Schreibschutz von Ihrer Kopie, booten Sie mit der Kopie Ihrer Workbench-Diskette (wir gehen im Verlauf des Artikels davon aus, daß Sie die Diskette modifizieren, von der Sie gebootet haben). Sobald die Workbench erscheint, klicken Sie zweimal auf das Diskettensymbol der Workbench-Diskette.

Sie sehen jetzt in einem Fenster den Inhalt dieser Diskette. Als erstes löschen Sie die »Demos«-Schublade, da sie nur Programme enthält, die für den täglichen Gebrauch völlig nutzlos sind. Klicken Sie dazu einmal auf das Icon der Schublade

und wählen Sie »Discard« aus dem »Workbench«-Menü. Anschließend öffnen Sie die »Utilities«-Schublade. Haben Sie jemals den »Calculator« oder das »Notepad« verwendet? Wahrscheinlich nicht. Klicken Sie auf den Calculator und löschen Sie ihn mit »Discard«. Führen Sie das gleiche mit dem Notepad durch, lassen Sie jedoch die »Utilities«-Schublade bestehen. Öffnen Sie nun das »System«-Directory durch einen Doppelklick. Zwei Programme, die Sie sicherlich selten brauchen, sind »Say« und »IconEd«, also löschen Sie diese mit Discard. Lassen Sie das »System«-Fenster geöffnet und klicken Sie im Haupt-Directory einmal auf das »System«-Icon. Wählen Sie »Clean Up« aus dem »Special«-Menü auf der Workbench, wor-

aufhin die Programm-Icons neu geordnet und formiert werden. Halten Sie die Shift-Taste gedrückt und klicken alle Icons in der System-Schublade an. Anschließend wählen Sie »Snapshot« (ebenfalls aus dem »Special«-Menü der Workbench). Schließen Sie darauf das System-Fenster.

Durch diese Löschaktionen sind bereits 160 KByte auf der Workbench-Diskette frei, doch wir können durchaus noch mehr einsparen:

## Platzsparen auf Disketten

Stellen Sie in Preferences fest, welchen Druckertreiber Sie benötigen (meist nur der für Ihren Drucker eingestellte) und

schreiben Sie sich den Namen auf. Danach öffnen Sie das CLI (im Ordner »System« auf der Workbench-Diskette das Programmsymbol anklicken). Auf der Workbench-Diskette finden sich einige Druckertreiber, wobei jeder zwischen 1 und 6 KByte belegt. Somit können Sie sich auch hier freien Speicher »schinden«, wenn Sie alle nicht benötigten entfernen. Geben Sie also ein:

```
cd devs:Printers
copy <Name> RAM:
delete #?
copy RAM:<Name> devs:
Printers
```

Zuerst wird in das Drucker-Verzeichnis gewechselt, und Ihr Druckertreiber wird auf die RAM-Disk gerettet. Dabei setzen Sie für <Name> den vor-

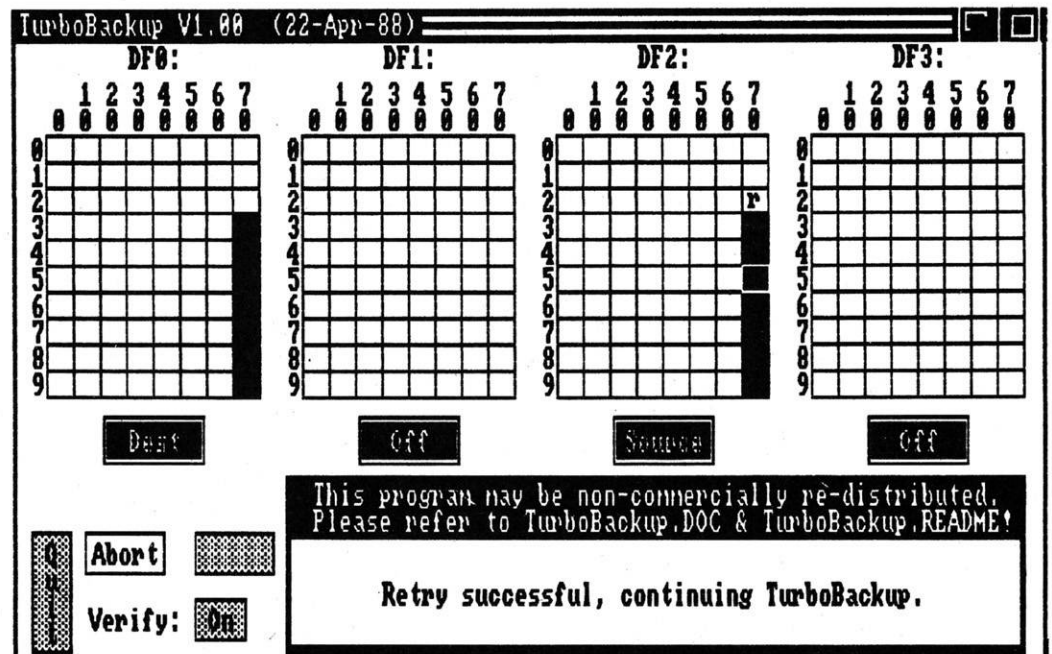


Bild 1. »TurboBackup« ist ein komfortables Diskettenkopierprogramm, das auftretende Lesefehler anzeigt und – wenn möglich – beseitigt



her notierten ein. In der nächsten Zeile werden alle Einträge im Verzeichnis der Druckertreiber gelöscht. Schließlich wird der gewünschte Druckertreiber von der RAM-Disk wieder in das »Printers«-Verzeichnis kopiert.

Daneben befinden sich mindestens 14 Tastaturreiber für die verschiedenen nationalen Zeichensätze im Verzeichnis »keymaps«. Davon brauchen Sie normalerweise nur einen: den deutschen. Wechseln Sie im CLI zunächst das Verzeichnis mit

```
cd /keymaps
```

und wenden Sie das gleiche Verfahren an wie bei den Druckertreibern: Kopieren Sie das File »d« ins RAM. Danach löschen Sie alle Einträge im aktuellen Verzeichnis (»devs:keymaps«) und kopieren den deutschen Tastaturreiber wieder in das richtige Directory. Sie sparen durch diese beiden »Säuberungsaktionen« etwa 100 KByte.

Insgesamt sind jetzt zwischen 250 und 280 KByte frei auf Ihrer Workbench-Diskette. Es gibt noch einige Programme, die man löschen kann, die aber immerhin manchmal gebraucht werden. Vielleicht lagern Sie diese auf eine »Tools«-Diskette aus? Damit sind sie nicht unerreichbar, aber Sie sparen dennoch Platz für eventuell häufiger Benötigtes.

Die ersten Streich-Kandidaten sind die verschiedenen Schriftarten (Fonts), das »Preferences«- und das »Clock«-Programm. Die Fonts werden nur in sehr seltenen Fällen gebraucht, beispielsweise für das Notepad. Nachdem wir diesem aber bereits die Nützlichkeit abgesprochen haben, können wir auch die Fonts löschen. Der Amiga hat zwei Zeichensätze im ROM eingebaut. Nur wenn Sie die sogenannten »Disk-fonts« verwenden wollen, müssen Sie auf das Fonts-Directory zurückgreifen. Ansonsten hat aber folgende Zeile (im CLI eingegeben) keinerlei negative Auswirkung:

```
delete sys:fonts all
```

Es könnte allerdings sein, daß in der Startup-Sequence Ihrer Workbench ein »Assign«-Befehl auf das »Fonts«-Directory angewendet wurde. In diesem Fall erhalten Sie die Fehlermeldung »Object in use«. Dies ist allerdings noch lange kein Grund zum Verzweifeln. Sie können das logische Device »Fonts:« ebensogut auf ein beliebiges anderes setzen. Am besten verwenden wir hierzu das Device »NIL:«. Folgende

Zeile geben Sie hierfür im CLI ein:

```
Assign Fonts: NIL:
```

Aus der Startup-Sequence löschen Sie die Assign-Zeile. Darauf ist Ihr Fonts-Directory endgültig zum Löschen freigegeben. Geben Sie also oben

auf Ihrer Workbench-Diskette etwa 460 KByte frei geworden. Diese stehen zur »Veredelung« mit PD-Programmen zur Verfügung.

Im folgenden werden nun einzelne Programme vorgestellt, die auf keiner Workbench fehlen dürfen. Sie finden in die-

geärgert? Die Editiermöglichkeiten sind mehr als spartanisch, eine einmal falsch eingegebene Zeile muß komplett wiederholt werden. Doch auch hierfür gibt es eine Public Domain-Lösung, die sogar die »AmigaShell« von der Workbench 1.3 um Längen schlägt: Die neueste Version von »ConMan« finden Sie auf Fred Fish 133 und RPD 112. Mit diesem Programm können Sie mit den Cursorstasten die aktuelle Eingabezeile editieren, die letzten Eingaben erneut oder nach Korrektur ausführen lassen oder das CLI-Window auf Minimal- oder Maximalgröße bringen. Ein Programm, auf das kein CLI-Benutzer verzichten sollte. (Speicherbedarf: ca. 40 KByte.)

## Was die Uhr geschlagen hat

- Das Programm »Clock« von der Fish-Disk 36 kann mehr als nur die Anzeige der aktuellen Uhrzeit. Es informiert außerdem über den momentan verfügbaren CHIP- beziehungsweise FAST-Speicher. Dazu kommt ein weiterer Vorteil: Der Speicherbedarf für »Clock« beträgt nur 16 KByte.

- Wie oft ist es Ihnen schon passiert, daß das CLI-Fenster trotz »EndCLI« nicht geschlossen wurde, weil Sie gerade ein Programm aus dem CLI mit »Run« gestartet haben, das noch nicht beendet war? Mit »Runback« von der Fish-Disk 73 hat dieses ein Ende. Es erfüllt den gleichen Zweck wie »Run«, erlaubt aber, daß man das CLI-Window schließen kann, ohne die aufgerufenen Tasks beenden zu müssen. Zu Runback finden Sie eine ausführliche Beschreibung auf Seite 95 in diesem Sonderheft. (Speicherbedarf: 6 KByte).

- CLI-Feinde, die lieber eine Batch-Datei auf der Workbench anklicken würden, als das CLI zu laden und dort komplizierte Befehlsfolgen einzugeben, haben mit »Xicon« von der Fish-Disk 102 das richtige Tool gefunden. Xicon bietet für diesen Zweck viele Optionen. Das Programm öffnet zur Abarbeitung der Befehle ein eigenes Fenster, dessen Größe man selbst festlegen kann (10 KByte).

- »TurboBackup« (Bild 1) ist eine schnelle und zuverlässige Alternative zum »Diskcopy«-Befehl der Workbench. Sie finden es auf der Fish-Disk 139. Es öffnet ein eigenes Fenster und kopiert eine Quelldiskette auf bis zu drei Zieldisketten gleichzeitig.

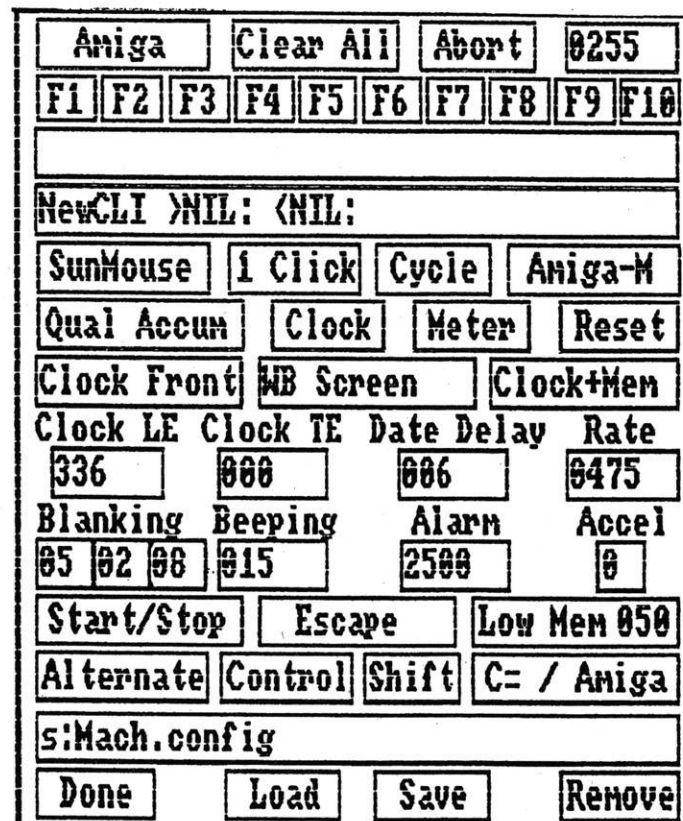


Bild 2. Bei »MachII« finden Sie eine Vielzahl nützlicher Funktionen (siehe auch Seite 131)

genannte Zeile (»delete ...«) nochmals ein, und das Directory ist endgültig verschwunden.

Die einmal eingestellten Preferences werden in einer eigenen Datei gespeichert (»System-Configuration«), man braucht das Programm Preferences also nur, wenn man etwas an den Einstellungen ändern will. Preferences braucht sowohl in der Version 1.2 als auch 1.3 etwa 57 KByte. Stellen Sie also Ihre bevorzugten Einstellungen ein und löschen Sie darauf Preferences.

## Veredelung der Workbench

Für die Clock gibt es einen grafisch nicht ganz so tollen, dafür aber weit leistungsfähigeren Ersatz auf der Fish-Disk 36. Löschen Sie also auch diese 20 KByte.

Wenn Sie alle vorgeschlagenen Programme und Directories gelöscht haben, dann sind

ser Aufstellung neben einer kurzen Programmbeschreibung jeweils die Angabe, wieviel Speicherplatz die lauffähigen Programme (ohne die auf den PD-Disketten ebenfalls enthaltene Dokumentation) belegen.

Zu einigen dieser Programme finden Sie einen gesonderten Artikel. Im Einzelfall ist die Seitennummer angegeben.

- Ein völlig unverzichtbares Public Domain-Programm ist die ASDG-RAM-Disk, deren Installation in diesem Heft auf Seite 58 ausführlich beschrieben ist. Mit ihr erhalten Sie ein schnelles und dennoch sicheres Speichermedium. Ihr großer Vorteil gegenüber der mitgelieferten Workbench RAM-Disk ist jedoch, daß sie ihre Dateien über einen Reset und fast jeden Absturz rettet, so daß sie beim nächsten Installieren wieder zur Verfügung stehen. (Speicherbedarf: 20 KByte.)

- Wer hat sich noch nicht über die Unfreundlichkeiten des CLI



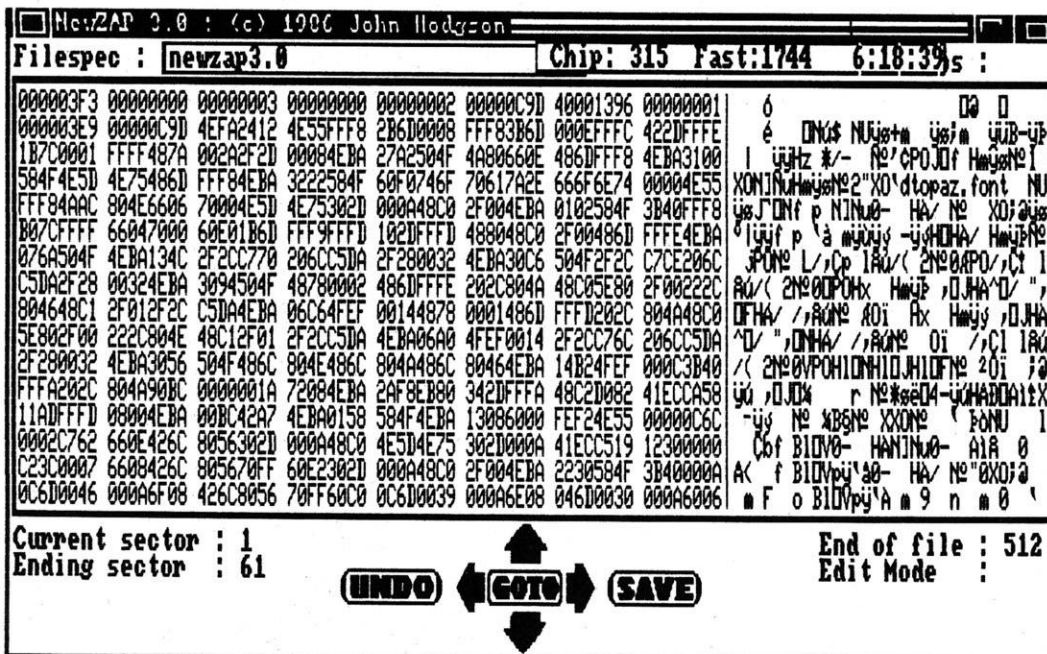


Bild 3. Mit »NewZAP« entreißen Sie Ihren Disketten die letzten Geheimnisse

Nebenbei kontrolliert Turboback-up selbständig, welche Laufwerke angeschlossen sind. Daneben werden die Kopien auf Schreib-/Lesefehler überprüft. Erstaunlich ist bei dieser Mehrarbeit die benötigte Zeit: Diese ist nicht länger als beim guten alten »Diskcopy«. Zu Turboback-up finden Sie eine ausführliche Anleitung auf Seite 129. (Speicherbedarf: 27 KByte.)

- Mit Editoren ist der AmigaDOS-Anwender nicht gerade verwöhnt. Weder »Ed« noch »Edit« können überzeugen. Auch der »MicroEmacs« von der Extras-Diskette ist nicht gerade der neueste Stand der

Technik. Es ist auch nicht leicht, die sehr unterschiedlichen Bedürfnisse der verschiedenen Anwender zu befriedigen. Dieses Problem hat Matthew Dillon erkannt und deshalb einen programmierbaren Editor (den »DME«) geschaffen. Er wurde bereits in zahlreichen Versionen weiterentwickelt. Aufgrund seiner hohen Flexibilität - man kann sich sehr einfach Pull-Down-Menüs und neue eigene Befehle definieren - und seinen unglaublichen Möglichkeiten ist er wirklich für jeden ein empfehlenswerter Editor. Man definiert die Funktionen genauso weit, wie man für nötig hält - die

Fähigkeiten des Editors wachsen mit zunehmender Übung. In diesem Heft finden Sie auf Seite 61 einen ausführlichen Workshop zum Thema DME. Daneben enthält die PD-Diskette zum Sonderheft 5 die brandaktuelle Version dieses genialen Programms. (Speicherbedarf: 48 KByte.)

- Wer schon einmal eine Datei versehentlich gelöscht hat - der CLI-Befehl »delete #?« eignet sich hierfür hervorragend -, wird »undelete« zu schätzen wissen. Dieses Programm nutzt die Eigenschaft von AmigaDOS, gelöschte Dateien nicht physikalisch zu löschen -

dies geschieht erst beim nächsten Schreibzugriff. Bis dahin ist noch nichts verloren. Sie können die vermißten Dateien mit dem Programm »undelete« von der RPD 103 zurückholen. (Speicherbedarf: 17 KByte.)

- Eine Zusammenfassung von vielen nützlichen Kleinigkeiten ist »MachII« (Bild 2). Sie finden eine ausführliche Beschreibung dieses Programms auf Seite 131. MachII bietet eine enorme Vielfalt an Funktionen, die man bereits nach kurzer Eingewöhnungszeit nicht mehr vermissen möchte. MachII finden Sie auf der Fish-Disk 163. (Speicherbedarf: 40 KByte.)

- Oft will man sich mit gewissen Eigenschaften eines Programms nicht abfinden. So öffnet sich beispielsweise das CLI-Fenster oft genug gerade an der Stelle, wo auch alle anderen Fenster liegen.

In diesem Fall könnte man doch den Programmcode nach dem entsprechenden Befehl absuchen, diesen ändern, das Programm wieder speichern, und schon hat man eine neue Version. Die Standardworkbench sieht keine Möglichkeit dieser Art vor. Die Lösung heißt »NewZAP« (Bild 3) von der Fish-Disk 58. (Speicherbedarf: 25 KByte.)

Sicherlich ist dies nur eine kleine Auswahl der Utilities, die wir für die wichtigsten halten. Testen Sie die neuen Möglichkeiten, sicher werden Sie noch viele Anregungen finden, wenn Sie Ihre PD-Disketten nach hilfreichen Programmen durchforsten.

# CLI-Starthilfe

Von Peter Sack

**E**insteiger und Workbench-verwöhnte Mausfans klagen immer wieder über die verwirrenden Eigenarten des CLI. Dies beginnt schon damit, daß man ein Close-Gadget vergeblich sucht. Um das Fenster zu schließen, muß man erst das Handbuch zum Amiga gründlich gelesen und den Befehl »endcli« gefunden haben. Das Editieren einer Eingabe ist auf die Verwendung der Taste Backspace beschränkt. Ginge das nicht auch

**Das CLI ist für viele Amiga-Besitzer ein Buch mit sieben Siegeln. Dennoch ist es oft unumgänglich, sich damit zu beschäftigen. Im folgenden finden Sie nützliche Tips zum richtigen Umgang Ihrem »Command Line Interface«.**

einfacher? Vielleicht schon, aber nach der Lektüre dieses Artikels sind Sie in der Lage, alle Klippen zu umschiffen.

Wenn Sie diesen Artikel aufmerksam lesen, werden Sie Ihre Public Domain-Disketten

problemlos vom CLI aus bearbeiten. Wir gehen davon aus, daß Sie mit der Workbench vertraut sind und beispielsweise wissen, was mit dem Begriff »Unterverzeichnis« gemeint ist, oder wie Sie das CLI starten.

Wenn nicht, empfehlen wir Ihnen, sich vorher im Handbuch die Kapitel 3 (»Inbetriebnahme«), 4 (»Der Arbeitstisch«) und 5 (»Nützliche Hilfsmittel«) anzusehen.

Legen Sie nach dem Einschalten des Amiga Ihre Workbench-Diskette in das interne Laufwerk. Bitte arbeiten Sie immer mit einer Kopie der Original-Workbench, verschließen Sie das Original am besten in einer Schublade. Warten Sie, bis die Workbench geladen wurde. Öffnen Sie deren Fenster mit einem Doppelklick auf das Icon der Workbench-Di-



ette. Wenn Sie Workbench 1.2 benutzen, öffnen Sie durch einen weiteren Doppelklick die »System«-Schublade und starten das Programm »CLI«. Arbeiten Sie mit Workbench 1.3, so starten Sie das CLI durch einen Doppelklick auf das »Amiga-Shell«-Icon. Nach kurzer Zeit erscheint nun in der Mitte des Bildschirms das CLI-Fenster.

## Geheimnissen der Diskette auf der Spur

Ziehen Sie es an den oberen Bildschirmrand und bringen Sie es auf die volle Größe. Dann geben Sie folgendes ein:

```
dir
und drücken <RETURN>. Der Amiga gibt das Inhaltsverzeichnis (»Directory«) der Workbench-Diskette aus. Als erstes werden alle Unterverzeichnisse (»Subdirectories«) angezeigt. Sie erkennen sie daran, daß sie alle eine »(dir)«-Kennung hinter dem Namen tragen. Danach folgen sämtliche Dateien des Hauptverzeichnisses in alphabetischer Reihenfolge. Vielleicht fällt Ihnen auf, daß einige Dateien ähnliche Namen haben, beispielsweise »Clock« und »Clock.info«. Bei Clock.info handelt es sich um das Symbol (»Icon«), mit dem Sie das Programm »Clock« von der Workbench starten können. Eine Datei wird also nur dann auf der Workbench angezeigt, wenn zu ihr eine weitere Datei mit demselben Namen und der Namens-erweiterung ».info« existiert. In dieser sind die Daten für das Icon und einige weitere – im Moment weniger wichtige – Angaben enthalten.
```

Es zeigt sich, daß eine Diskette mehr Dateien und Directories enthält, als auf der Workbench angezeigt werden. Dies hat für den unerfahreneren Benutzer den Vorteil, daß er nicht mit Unwesentlichem oder Unverständlichem konfrontiert wird. Beispielsweise wird auf der Original-Workbench das CLI nicht angezeigt (sondern erst nachdem Sie in Preferences auf »CLI ON« geklickt haben). Ebenso wird keine Datei angezeigt, in der beispielsweise die italienische Tastaturbelegung gespeichert ist, weil diese Information für die Workbench völlig uninteressant ist.

Sehen Sie sich das Inhaltsverzeichnis einmal genau an: Es gibt eine Datei »System.info«. Sie enthält das Icon für das Unterverzeichnis »System«. Für das Unterverzeichnis »c« gibt

es jedoch keine derartige Datei (etwa »c.info«), folglich wird dieses Verzeichnis nicht auf der Workbench angezeigt. Das ist sinnvoll, da in ihr nur CLI-Befehle gespeichert werden, die man von der Workbench aus ohnehin nicht anwenden kann. Sie benötigen zur Ein- und Ausgabe ein CLI-Fenster.

Die verfügbaren CLI-Befehle können Sie auflisten. Geben Sie ein:

```
dir c
und <RETURN>. Sie sehen ein anderes Inhaltsverzeichnis als beim letzten »dir«-Befehl –
```

nen dies anschaulich testen, wenn Sie folgende Zeile im CLI eingeben:

```
rename dir inhalt
Versuchen Sie jetzt nochmals mit
dir
den Inhalt eines Directorys auszugeben. Welche Fehlermeldung erhalten Sie? Versuchen Sie das gleiche mit der Zeile
inhalt
```

Sie wundern sich, daß Sie dieselbe Ausgabe erhalten wie vorher mit »dir«? Dies hat einen einfachen Grund: Die Befehle

## Sie besitzen nur ein Laufwerk?

Wenn Sie nur ein Diskettenlaufwerk besitzen, booten Sie von Ihrer Workbench. Danach möchten Sie eine beliebige PD-Diskette ansehen, diese hat aber keine Icons, die von der Workbench sichtbar sind. Also öffnen Sie ein CLI-Fenster. Dabei taucht zum erstenmal ein Requester am linken oberen Bildschirm auf, der Sie freundlich auffordert »Please insert Volume Workbench in any drive«. Da Sie nur ein Laufwerk besitzen, gibt es nur einen Weg: PD-Diskette raus – Workbench rein. Tatsächlich verschwindet der Requester nach kurzem Zögern. Aber schon fangen die Probleme an: Wenn Sie jetzt »dir« eingeben, erhalten Sie das Directory der Workbench. Wechseln Sie vorher die Diskette, so fängt der Requester an, lästig zu werden.

An dieser Stelle wird er fast schon penetrant. Wechseln Sie jetzt die Diskette, wird der Requester freundlich befiehlt, so erhalten Sie wieder das Directory der Workbench. Jetzt heißt es Nerven behalten und nicht gleich die Diskette an die Wand werfen. Denn es gibt noch einen Weg.

Mit drei Zeilen ist das Problem gelöst:

```
makedir RAM:c
copy c: RAM:c
assign c: RAM:c
```

Von jetzt an kann das CLI auf die Unterstützung der Workbench-Diskette verzichten. Denn alle benötigten Befehle sind auch im C-Directory der RAM-Disk zu finden. Beachten Sie die dritte Zeile. Ohne diese hätten Sie nichts erreicht, denn erst mit dieser weiß das CLI, daß der C-Ordner ins RAM »verlegt« wurde.

Von jetzt an werden alle CLI-Befehle in »RAM:c« gesucht. Sie können jetzt die Diskette wechseln und eine beliebige andere untersuchen, ohne ständig darauf hingewiesen zu werden, daß die Workbench fehlt. Natürlich können Sie diese Befehle auch bei jedem Start Ihrer Workbench ausführen lassen. Dazu fügen Sie die drei Zeilen an einer beliebigen Stelle (empfehlenswert ist möglichst früh) in die Datei »Startup-sequence« aus dem s-Directory ein. Falls Sie nur 512 KByte zur Verfügung haben, sollten Sie einige Befehle aus dem c-Directory löschen, da dieses sonst zu voll wird. Achten Sie darauf, daß keines gelöscht wird, das in der Startup-Sequence benötigt wird. Überlegen Sie sich bei allen anderen, ob Sie sie jemals verwendet haben. Wenn nicht, haben Sie einen Streichkandidaten gefunden.

den Inhalt des Unterverzeichnisses »c«. Und dort finden Sie neben einer Reihe Dateien auch die Datei »dir«. Jedesmal wenn Sie einen Befehl eingeben, überprüft das CLI, ob es sich bei diesem um ein Programm aus dem Directory »c« der Boot-Diskette handelt. Dies ist auch der Grund, warum beim Aufruf eines CLI-Befehls häufig die Meldung »Please insert volume Workbench« erscheint. Sie können das Inhaltsverzeichnis einer Diskette nur dann mit dem dir-Befehl abrufen, wenn sich die Datei »dir« im c-Directory befindet. Sie kön-

nen dies anschaulich testen, wenn Sie folgende Zeile im CLI eingeben:

```
path <DirName> add
ein neues Directory in die Liste aufnehmen. Dabei ist »DirName« der Name des gewünschten Directorys.
```

Im c-Directory einer unveränderten Workbench-Diskette –

wie sie Ihrem Amiga beiliegt, befinden sich alle in diesem Artikel erwähnten CLI-Befehle. Beim Aufruf brauchen Sie nicht auf Groß- oder Kleinschreibung zu achten, Sie können das Inhaltsverzeichnis einer Diskette mit »Dir«, »DiR«, »dir« oder »dIR« abrufen – Das CLI sucht in jedem Fall nach dem richtigen Programm.

Ein weiterer wichtiger CLI-Befehl ist »Cd«, was ausgeschrieben »Change directory« heißt. Dies bedeutet keineswegs, daß Sie damit irgendwas an Ihren Unterverzeichnissen ändern, sondern Sie wechseln das aktuelle Verzeichnis. Warum ist dieser Befehl so wichtig? Jedesmal wenn Sie einen CLI-Befehl eingeben, bezieht er sich auf das Directory, in dem Sie sich im Moment befinden. Geben Sie ein:

```
dir
Sie sehen das Inhaltsverzeichnis des aktuellen Directories (vermutlich das Stammverzeichnis Ihrer Boot-Diskette). Darauf geben Sie ein:
```

```
cd devs
dir
```

Es erscheint ein neues Verzeichnis (»devs«). In diesem sind weitere Directories enthalten. Wenn Sie wollen, können

## Von Bäumen, Zweigen und versteckten Daten

Sie jetzt das Spiel noch weiter treiben, indem Sie beispielsweise mit

```
cd keymaps
dir
```

das nächste Directory auslesen. Spätestens dann werden Sie aber feststellen, daß es kein Directory mehr gibt, in das Sie wechseln könnten. Was den sogenannten Directory-Baum betrifft, sind Sie sozusagen am Ende. Schalten Sie jetzt nicht erschreckt Ihren Computer aus – Sie haben sich noch lange nicht »verlaufen«. Geben Sie folgende Zeile ein:

```
dir :
Sie sehen wieder das Stammverzeichnis. Es ist also auch dann noch erreichbar, wenn Sie längst in ein tief verschachteltes Unterverzeichnis abgetaucht sind. Mit
```

```
cd :
oder
cd df0:
sind Sie wieder im Stammver-
```



zeichnis gelangt. Genau ein Verzeichnis nach oben gelangen Sie mit

```
cd /
```

Dieser Befehl bezieht sich immer auf das aktuelle Verzeichnis. Er sucht das übergeordnete Verzeichnis und macht dieses zum aktuellen.

Allerdings kann der Amiga unterschiedliche Speichermedien verarbeiten: Dies beginnt beim Diskettenlaufwerk, geht über die RAM-Disk bis hin zur Festplatte oder in ferner Zukunft dem optischen Laufwerk. Jedem Gerät wird von Amiga-DOS ein Name gegeben, unter denen Sie es ansprechen können. Das interne Diskettenlaufwerk hat beispielsweise die Kennung »df0:«. Weitere Diskettenlaufwerke werden mit »df1:«

## Wie viele echte Laufwerke besitzen Sie?

bis »df3:« durchnumeriert, die Workbench-RAM-Disk heißt »RAM:«, eine Festplatte »dh0:« oder »jh0:«. Unter diesen Bezeichnungen können Sie die verschiedenen Geräte im CLI ansprechen. Eine komplette Liste dieser Namen finden Sie in Tabelle 1. So gelangen Sie mit

```
cd dh0:
```

in das Stammverzeichnis Ihrer Festplatte (falls Sie eine besitzen). Kopieren Sie mit Hilfe der Workbench beispielsweise das Uhr-Programm (»clock«) oder das CLI auf die RAM-Disk und geben Sie im CLI ein:

```
dir RAM:
```

so erscheint das Inhaltsverzeichnis der RAM-Disk mit dem vorher kopierten Programm. Geben Sie ein:

```
cd RAM:
```

so wechselt der Amiga auf die RAM-Disk und alle weiteren Befehle beziehen sich auf das Hauptverzeichnis der RAM-Disk. Ein weiterer wichtiger Geräte name ist »SYS:«. Dieser bezeichnet die Diskette, mit der der Amiga nach einem Reset gebootet wurde – in unserem Fall also die Workbench-Diskette. Der Vorteil bei der Benutzung dieses symbolischen Namens ist, daß man sich bei mehreren Laufwerken nicht mehr zu merken braucht, in welchem Laufwerk sich die Boot-Diskette und somit das momentan benutzte »c«-Directory befindet. Sie schreiben jetzt einfach:

```
cd SYS:
```

und der Amiga wechselt in die-

sem Fall in das Stammverzeichnis der Workbench-Diskette. Sollte diese nicht in DF0: liegen, ist das kein Problem. Wenn sie in einem anderen Laufwerk liegt, wird dies sofort erkannt. Wenn nicht, werden Sie aufgefordert, die Workbench einzulegen. Wie kann man beispielsweise das System-Verzeichnis auf die RAM-Disk kopieren? Dies geht vergleichsweise einfach:

```
copy SYS:Utilities/
Notepad#? RAM:
```

Die Schreibweise »Notepad #?« heißt, daß der Amiga jede Datei, deren Name mit »Notepad« anfängt, kopieren soll, dabei ist es egal, ob kein, ein oder mehrere Zeichen folgen. Noch etwas zum Suchpfad: Sie haben nach »Utilities« einen Schrägstrich (»/«) eingegeben und dann einen Programmnamen. Dieser Strich trennt ein-

sicht, in der alle besprochenen CLI-Befehle mit Syntax und kurzer Beschreibung zusammengestellt sind. Schwerwiegende Probleme tauchen meist erst dann auf, wenn Sie nur ein Laufwerk besitzen, und damit eine andere als die Workbench-Diskette untersuchen wollen. In diesem Fall finden Sie einige Hinweise in nebenstehendem Kasten (»Please insert...«).

Wenn Sie vom CLI aus ein Programm starten, das die Ein- und Ausgabe des CLI nicht blockiert (Sie möchten vielleicht weitere CLI-Befehle eingeben), verwenden Sie dafür den »Run«-Befehl. Geben Sie dazu ein

```
run Programmname
```

»Programmname« ist der Name des Programms, wie Sie ihn im Directory eingetragen finden. Sie können dabei auch einen Suchpfad angeben, der als letztes Element diesen Dateinamen enthält. Beispielsweise:

```
run df1:tools/
turbobackup
```

Neben diesen Grundlagen gibt es im Zusammenhang mit Public Domain noch einige Besonderheiten, die Sie beachten sollten.

Unbekannte Public Domain-Disketten überprüfen Sie am besten zuerst von der Workbench aus. Suchen Sie nach Dokumentationen oder Hinweisen in Form einer »ReadMe«- oder ».doc«-Datei im Hauptdirectory der Diskette. Sind solche vorhanden, lesen Sie diese zuerst.

Wenn Sie weitere Schublade finden, sehen Sie sich auch diese an. Die Programmbeschreibungen finden Sie meist zusammen mit den Programmen in einer gemeinsamen Schublade.

Ist dies der Fall, lesen Sie alle Dokumentationen zu den Sie interessierenden Programmen. Diese sind meist mit einer der folgenden Extensionen versehen:

- ».hlp« für Help-Files,
- ».txt« für Text-Dateien,
- ».man« oder ».doc« für Dokumentationen.

Lesen Sie in diesem Fall auch den Abschnitt »Ausgeben und Drucken«.

Ganz selten scheint auf einer Diskette »gar nichts« zu sein, wenn man sie auf der Workbench anklickt. In diesem Fall öffnen Sie das CLI und lassen sich dort das Directory des Stammverzeichnisses ausgeben. Geben Sie »ReadMe«- und ».doc«-Files mit dem »Type«-Befehl aus, beispielsweise mit

## Die gebräuchlichsten Gerätenamen

Amiga-DOS braucht – um verschiedene Geräte zu unterscheiden – für jedes Gerät einen anderen Namen. Einige sind Ihnen vertraut, andere haben Sie vielleicht noch nie gehört. Grundsätzlich können Sie davon ausgehen, daß es sich immer dann um einen Geräte- (»Device«-) Namen handelt, wenn am Ende ein Doppelpunkt steht.

### Name Art des Geräts

|       |                                                                                 |
|-------|---------------------------------------------------------------------------------|
| DFx:  | Diskettenlaufwerk (»x« steht für die Nummer des Laufwerks), beispielsweise DF0: |
| DHx:  | Festplatte (»x« steht für die Nummer des Laufwerks)                             |
| JHx:  | Festplatte, der PC-Karte. Diese kann vom Amiga mitbenutzt werden                |
| RAM:  | RAM-Disk                                                                        |
| VDO:  | Resetfeste RAM-Disk von ASDG (PD-Programm)                                      |
| PRT:  | Drucker                                                                         |
| PAR:  | Parallele Schnittstelle                                                         |
| SER:  | Serielle Schnittstelle                                                          |
| RAD:  | Resetfeste RAM-Disk der Workbench 1.3                                           |
| S:    | Directory, aus dem Batch-Dateien gelesen werden (logisches Device)              |
| DEVS: | Directory für Devices (logisches Device)                                        |
| LIBS: | Directory für Libraries (logisches Device)                                      |
| L:    | Directory für Handler (logisches Device)                                        |
| NIL:  | Pseudo-Device, das ins »Nichts« zeigt                                           |

Die Gruppe der logischen Devices ist kein echtes Gerät, sondern eine reine Festlegung für das Betriebssystem. Ebenso können Sie sich mit dem Assign-Befehl ein beliebiges Device definieren, das Sie dann unter diesem Namen ansprechen. Auf der Workbench 1.3 finden Sie einige Beispiele für Devices, die mit Hilfe einer »Mountlist« angemeldet werden müssen.

**Tabelle 1. Die wichtigsten Gerätenamen von Amiga-DOS. Die Übersicht enthält auch sogenannte logische Devices.**

```
copy SYS:System RAM:
```

Die Angabe, wo sich die gewünschten Programme befinden – in diesem Fall »SYS:System« – nennt man Suchpfad, da Amiga-DOS diesen Weg verwendet, um an die gewünschten Dateien zu kommen. Wenn Sie wissen, daß die Workbench-Diskette im Laufwerk »df0:« liegt, können Sie ebenso gut:

```
copy df0:System RAM:
```

als Suchpfad angeben. Genauso können Sie einzelne Dateien kopieren. Wollen Sie aus dem Verzeichnis »Utilities« das Programm »Notepad« kopieren, so schreiben Sie:

```
copy SYS:Utilities/
Notepad RAM:
```

```
copy SYS:Utilities/
Notepad.info RAM:
```

oder

zelle Unterverzeichnisse oder ein Verzeichnis von einer darin liegenden Datei. Bei »Utilities« handelte es sich um ein Verzeichnis, Sie hätten nach dem Schrägstrich noch einige weitere Directories – jeweils durch einen Querstrich getrennt – eingeben können. Als letzten Befehl müssen Sie

```
EndCli
```

kennen. Mit Endcli beenden Sie die Arbeit mit dem CLI und

## Der sanfte Ausstieg: Schluß mit EndCli

schließen das aktuelle CLI-Fenster. Diese Beispiele sollten Ihnen die Benutzung des Amiga-CLI deutlich gemacht haben. In Tabelle 2 finden Sie eine Über-



type VirusX.doc

Sie können die Ausgabe mit der Space-Taste anhalten; um fortzufahren, drücken Sie <-> (Backspace).

## Type-Befehl: Lesen Sie zuerst die Anleitung

Anhand der Extensionen der Dateien (meist die letzten drei

Buchstaben hinter dem Punkt) kann man einiges über diese herausfinden. In Tabelle 3 finden Sie die häufigsten Extensionen aufgeführt und erklärt.

Meist ist jedes Programm in einer eigenen Schublade untergebracht, so daß Sie dort auch zugehörige Bedienungsanleitungen, Hilfsdateien und Sourcecodes finden.

Falls Sie von der Workbench aus keine Programme in einer

Schublade finden, so probieren Sie es vom CLI aus noch einmal. Vermutlich existieren nur keine »info«-Dateien zu den Programmen. Lesen Sie in diesem Fall das Root-Directory mit »dir« oder »list« und wechseln Sie dann in die Directories, die Sie besonders interessieren (mit dem Befehl »cd«).

Natürlich ist es oft angenehmer, die Doc-Files nicht nur am Bildschirm zu betrachten, son-

parallele Schnittstelle übertragen. Durch die Angabe von »prt:« sorgen Sie dafür, daß er vorher durch das Printer-Device für die Druckausgabe aufbereitet wird. Geben Sie dazu folgenden Befehl im CLI ein:

```
type >prt: Filename
oder
```

```
copy Filename prt:
```

Für »Filename« setzen Sie den Namen der auszudruckenden

## CLI-Befehle und ihre Wirkung

Diese Befehle halten wir für die wichtigsten bei der Arbeit mit dem CLI. Natürlich ist dies Geschmackssache (es soll Leute geben, die mit »dir« und »endcli« auskommen). Aber diese Befehle sollten Sie auf keinen Fall aus Ihrem C-Ordner löschen.

|                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cd &lt;Suchpfad&gt;</b>                     | Auf diese Eingabe hin wechselt der Amiga in das in »Suchpfad« angegebene Directory und macht es zum aktuellen Directory, so daß sich alle folgenden Befehle auf dieses Directory beziehen, sofern Ihnen kein eigener Suchpfad übergeben wird. Beispiele: »cd df0:c« - Wechsle in das »C«-Directory der Diskette in »df0:«. »cd Texte« - Wechsle in das Unterverzeichnis »Texte« des aktuellen Directories.                                                                                                                                                                               |
| <b>Copy &lt;Datei1 &gt;<br/>&lt;Datei2&gt;</b> | Kopiere »Datei1« nach »Datei2«, so daß Datei1 noch einmal unter dem Namen Datei2 auf der Diskette abgelegt wird. Datei1 kann ebenso wie Datei2 einen Suchpfad enthalten. Beispiele: »Copy RAM:Help.txt df0:Devs/Printers« - Kopiere die Datei »Help.txt« von der RAM-Disk auf das Diskettenlaufwerk »df0:« in das Verzeichnis »Devs/Printers« beziehungsweise in das in dem Verzeichnis »devs« enthaltene Unterverzeichnis »Printers«. »Copy Brief.txt Brief.backup« - Dupliziere die Datei »Brief.txt« im momentanen Directory und speichere sie erneut unter dem Namen »Brief.backup«. |
| <b>Dir (Suchpfad)</b>                          | Dieser Befehl gibt das Inhaltsverzeichnis eines Directories aus. Wird kein »Suchpfad« angegeben, so wird das Inhaltsverzeichnis des aktuellen Directories ausgegeben, im anderen Falle gibt Amiga-DOS das Inhaltsverzeichnis des im Suchpfad spezifizierten Directories aus. Beispiele: »Dir df0:« - Das Directory der Diskette in »df0:« wird ausgegeben »Dir Texte« - Das Inhaltsverzeichnis des Directories »Texte« im momentanen Directory wird angezeigt.                                                                                                                           |
| <b>Endcli</b>                                  | Beendet die Arbeit mit dem CLI und schließt das CLI-Window.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Run<br/>&lt;Programm&gt;</b>                | Startet »Programm«. Bei »Programm« kann es sich um einen einfachen Programmnamen oder einen Suchpfad mit Programmname handeln. Beispiele: »Run Notepad« - Startet das Programm Notepad aus dem aktuellen Directory. »Run SYS:System/NoFastMem« - startet das Programm »NoFastMem« aus dem »System«-Directory der Bootdiskette.                                                                                                                                                                                                                                                           |
| <b>Type &lt;Datei&gt;</b>                      | Gibt die Text-»Datei« auf den Bildschirm aus. Bei »Datei« kann es sich auch um einen Suchpfad handeln, der als letztes Element eine Datei enthält. Beispiele: »Type Hilfe.txt« - Gib die Datei »Hilfe.txt« aus dem momentanen Directory auf den Bildschirm aus. »Type df0:Devs/Mountlist« - Gib die Datei »Mountlist« auf der Diskette in df0: im Verzeichnis »devs« aus.                                                                                                                                                                                                                |
| <b>&lt; Ausdruck &gt;<br/>(Ausdruck)</b>       | bedeutet, daß der Parameter »Ausdruck« angegeben werden muß. heißt, daß der Parameter »Ausdruck« optional ist und somit angegeben werden kann.                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Tabelle 2. Mit diesen CLI-Befehlen sollten Sie sehr gut vertraut sein

| Extension              | Bedeutung                                                                 |
|------------------------|---------------------------------------------------------------------------|
| .ass, .a, .asm         | Assemblersourcecode                                                       |
| .bas                   | Basic-Programm                                                            |
| .c                     | C-Sourcecode                                                              |
| .h                     | Teil eines C-Programms (Header-Datei)                                     |
| .mod                   | Modula-Sourcecode                                                         |
| .def                   | Modula-Definitionsmodul (Schnittstelle)                                   |
| .font                  | Font-Definition                                                           |
| .hlp, .fnf, .doc, .man | Bedienungsanleitung                                                       |
| .info                  | enthält das Icon zu einem Programm                                        |
| .lib, .library         | Software-Bibliothek, die von verschiedenen Programmen genutzt werden kann |
| .txt                   | Textdatei                                                                 |
| .device                | Erweiterung zur Benutzung von Hardware                                    |

Tabelle 3. Aus diesen Extensionen können Sie Rückschlüsse auf den Inhalt eines Files ziehen

dern ausgedruckt zur Verfügung zu haben. Wenn Sie einen Drucker besitzen, drucken Sie zuerst das File aus, das Sie über den Inhalt der Diskette informiert. Sie brauchen dazu weder eine Textverarbeitung noch ein anderes Hilfsprogramm. Geben Sie einfach folgende Zeile im CLI ein:

```
copy Contents prt:
```

Dabei kann für »Contents« natürlich auch ein anderer Name stehen (der auf Ihrer Diskette enthalten ist). Falls Sie dabei die Fehlermeldung »Cannot open printer« erhalten, sollten Sie überprüfen, ob in Ihrem Devs-Ordner das »printer.device« enthalten ist und ob der richtige Druckertreiber in Preferences eingestellt war. Oder ist vielleicht einfach nur der Drucker nicht eingeschaltet?

Drucken Sie jetzt alle für Sie interessanten Files aus. Wenn Sie nicht genau wissen, ob ein bestimmtes File für Sie interessant ist, bringen Sie seinen Inhalt mit

```
type <name>
```

auf den Bildschirm. Sie können die Ausgabe mit der Leertaste anhalten und mit <-> (oben rechts auf der Tastatur) fortsetzen.

Das Drucken funktioniert immer auf dieselbe Art: Sie kopieren ein File »auf den Drucker«. Dies geschieht nicht wie beim Kopieren auf der Workbench, sondern der Text wird auf die

den Datei ein. Mit der Umleitung »>prt:« können Sie auch Directories ausgeben:

```
dir >prt:
```

Alle Unterdirectories einer Diskette drucken Sie mit der zusätzlichen Angabe von »all«:

```
dir >prt: all
```

Wenn Sie es geschafft haben, alle Dokumentationen auszudrucken, legen Sie diesen Artikel zur Seite. Sie haben jetzt genügend Informationen ausgedruckt, die Sie am besten sofort durcharbeiten sollten.

## Der Blick ins Handbuch hilft bei Schwierigkeiten

Falls später irgendwelche Probleme auftauchen, erinnern Sie sich vielleicht wieder an Ihren »Freund und Helfer«.

Die Arbeit mit dem CLI sollte für Sie nun kein Buch mit sieben Siegeln mehr sein. Mit jeder Benutzung dieser Oberfläche gewinnen Sie Erfahrungen. Heute noch kompliziert scheinende Vorgänge wurden später selbstverständlich sein. PD-Programme, die nur über das CLI sinnvoll anzuwenden sind, bereiten Ihnen inzwischen sicher keine Probleme mehr. Und sollten Schwierigkeiten auftreten, verzweifeln Sie nicht. Schnell werden Sie die richtige Antwort finden - vielleicht im Handbuch? so



**E**ine Übersicht über zwei der größten und wichtigsten PD-Serien, nämlich »Fred Fish« und »RPD«, finden Sie auf den folgenden Seiten. Die Auflistung erfolgt dabei nach Disketten geordnet. Neben einer kurzen Funktionsbeschreibung finden Sie zu jedem Programm/File ein Zuordnungs-Kürzel. Dieser erleichtert das systematische Suchen nach bestimmten Arten von Programmen. Die Kürzel bedeuten:

|   |                     |
|---|---------------------|
| A | Anwendung           |
| G | Grafik              |
| M | Musik, Midi         |
| P | Programmiersprachen |
| T | Tools               |
| S | Spiele              |
| U | Utilities           |

# Große Public Domain- Übersicht

**Ihre Suche nach Programmen in der Flut von PD-Disketten erleichtert unsere Liste. Dabei werden die Programme nicht nur bestimmten Rubriken zugeordnet, sondern auch kurz beschrieben.**

von Martin Jobst

Mit Hilfe unserer Liste können Sie ohne großen Aufwand bestimmte Programme aus den Bereichen herauspicken, die Sie besonders interessieren. Die Kurzbeschreibung der einzelnen Programme erleichtert dies ungemein.

Noch ein Tip: Auf der Programmservice-Diskette zu diesem Sonderheft finden Sie ein File mit dem Namen »PD-Liste.ASC«. Dieses ASCII-File enthält die vollständige Liste. Mit Hilfe eines Editors oder einer Textverarbeitung können Sie die Liste systematisch durchsuchen. Besitzen Sie »Documentum«, können Sie sich mit der Funktion »Querverweislste« eine Liste mit Zeilen, in denen sich Ihr Suchbegriff befindet, erstellen lassen. rs

| Diskette: | Programm:                 | Bemerkung:                                | Art:                                      |   |
|-----------|---------------------------|-------------------------------------------|-------------------------------------------|---|
| Fish 1    | Amigademo                 | Grafik-Benchmark-Test                     | U                                         |   |
|           | Amigaterm                 | Terminal-Emulation                        | U                                         |   |
|           | Balls                     | Bewegungssimulation                       | G                                         |   |
|           | Colorful                  | HAM-Demo                                  | G                                         |   |
|           | Dhystone                  | Benchmark-Test                            | U                                         |   |
|           | Dotty                     | Source-Code zum Workbench-Demo »Dots«     | G                                         |   |
|           | Freedraw                  | einfaches Malprogramm                     | G                                         |   |
|           | Gad                       | Gadget-Demo                               | G                                         |   |
|           | Gfxmem                    | zeigt die Speicherbelegung                | U                                         |   |
|           | Halfbrite                 | demonstriert den 64-Farben-Modus          | G                                         |   |
|           | Hello                     | Window-Demonstration                      | G                                         |   |
|           | Latfp                     | zeigt Zugriff auf FFP-Library             | T                                         |   |
|           | Palette                   | Beispielprogramm für Farbpaletten         | G                                         |   |
|           | Trackdisk                 | demonstriert Trackdisk-Treiber            | T                                         |   |
|           | Requesters                | Beispiel für die Erzeugung von Requestern | T                                         |   |
|           | Speech                    | Sprach-Demo                               | M                                         |   |
|           | Speechtoy                 | Sprach-Demo                               | M                                         |   |
| Fish 2    | Alib                      | Programmierhilfe (»libs«)                 | T                                         |   |
|           | cc                        | Aufruf-Hilfe für Lattice C                | U                                         |   |
|           | Dbug                      | Debugging-Hilfe für C-Programme           | U                                         |   |
|           | Make                      | Emulation des Unix-Kommandos »make«       | T                                         |   |
|           | Make2                     | weitere »make«-Version                    | T                                         |   |
|           | Microemacs                | Editor                                    | U                                         |   |
|           | Portar                    | File-Archivierungsprogr.                  | U                                         |   |
| Xrf       | »Cross-reference«-Utility | U                                         |                                           |   |
| Fish 3    | Gothic                    | druckt Banner aus                         | G                                         |   |
|           | Roff                      | Text-Formatierung                         | T                                         |   |
|           | Ff                        | Text-Formatierung                         | T                                         |   |
|           | CForth                    | Forth-Implementation                      | P                                         |   |
|           | XLisp                     | Lisp-Implementation, Version 1.4          | P                                         |   |
| Fish 4    | Banner                    | druckt Banner aus                         | G                                         |   |
|           | Bgrep                     | Suchalgorithmus                           | T                                         |   |
|           | Bison                     | Anlehnung an das Unix-Kommando »yacc«     | T                                         |   |
|           | Bm                        | Suchalgorithmus                           | T                                         |   |
|           | Brep                      | Suchalgorithmus                           | T                                         |   |
|           | Kermit                    | DFÜ-Tool, alte Version                    | A                                         |   |
|           | MyCLI                     | CLI-Abwandlung                            | U                                         |   |
|           | Mandel                    | Mandelbrot-Programm                       | G                                         |   |
|           | Fish 5                    | Bons                                      | demonstriert Zugriff auf »console.device« | T |

| Diskette: | Programm:                              | Bemerkung:                           | Art: |
|-----------|----------------------------------------|--------------------------------------|------|
|           | Freemap                                | Anzeige des freien Speichers         | T    |
|           | Input.dev                              | Tastaturabfrage (vor Intuition)      | T    |
|           | Joystick                               | Joystick-Abfrage                     | T    |
|           | Keyboard                               | Demo für Tastatur-Abfrage            | T    |
|           | Layers                                 | Anwendung der »layers.library«       | T    |
|           | Mandelbrot                             | Mandelbrot-Programm (IFF-Verarbeit.) | G    |
|           | Mouse                                  | Maus-Abfrage am Game-Port            | T    |
|           | One.window                             | Windows öffnen mit »console.device«  | T    |
|           | Parallel                               | Zugriff auf den Parallel-Port        | T    |
|           | Printer                                | zeigt Zugriff auf den Drucker        | T    |
|           | Print.support                          | Druckerrountinen                     | T    |
|           | Proctest                               | Erzeugen eines Prozesses             |      |
|           | Region                                 | Grafikroutine                        |      |
|           | Samplefont                             | Zeichensatz                          |      |
|           | Serial                                 | Zugriff auf den seriellen Port       | T    |
|           | SinglePlayfield                        | erzeugt ein »Single-Playfield«       | T    |
|           | Speechtoy                              | Sprach-Demo                          | M    |
|           | Speech.demo                            | vereinfachte Version von »Speechtoy« | M    |
|           | Text.Demo                              | zeigt verfügbare Fonts an            | T    |
| Fish 6    | Timer                                  | Zugriff auf »timer.device«           | T    |
|           | Trackdisk                              | Verwendung des »Trackdisk«-Treibers  | T    |
|           | Compress                               | Packer                               | U    |
|           | Dadc                                   | Digital/Analog-Uhr                   | U    |
|           | Microemacs                             | erweiterte Version dieses Editors    | U    |
|           | Mult                                   | File-Tool                            | T    |
|           | Scales                                 | Demonstration der Audio-Device       | M    |
|           | SetParallel                            | Änderung der Parallel-Port-Parameter | T    |
|           | SetSerial                              | Änderung der Seriell-Port-Parameter  | T    |
|           | Sortc                                  | Sortieroutine                        | T    |
| Stripc    | entfernt Kommentare von C Source-Files | T                                    |      |
| Fish 7    | Hack                                   | Version des berühmten Unix-Spiels    | S    |
| Fish 8    | Hack.source                            | Source-Code zu »Hack«                | S    |
| Fish 9    | Moire                                  | malt Moire-Muster                    | G    |
|           | MVP-Forth                              | Forth-Implementation                 | P    |



| Diskette: | Programm:                                                                                                                             | Bemerkung:                                                                                                                                                                                                                                                                                                         | Art:                                                             |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|           | Proff<br>Setlace<br>Skewb<br>Sparks                                                                                                   | (Vers. 1.00.03A)<br>Text-Formatieroutine<br>Interlace-Modus ein/aus<br>Grafik-Demo<br>Grafik-Demo                                                                                                                                                                                                                  | T<br>T<br>G<br>G                                                 |
| Fish 10   | Conquest<br>Dehex<br>Filezap<br>Fixobj<br>Iff<br><br>Id<br><br>Ls<br><br>Sq.Usq<br><br>Trek73<br>Yachtc                               | Weltraum-Adventure wandelt Hex- in Binär-Code<br>Patch-Programm für Files säubert XModem-Dateien<br>Lese-/Schreibroutine für IFF-Files<br>erweitertes CLI-Kommando »list«<br>Emulation des Unix-Kommandos »ls«<br>Packer (langsamer als der von Fish 6)<br>Weltraumspiel<br>Würfelspiel                            | S<br>T<br>T<br>T<br>T<br><br>T<br><br>T<br><br>U<br><br>S<br>S   |
| Fish 11   | Dpslide<br>Pictures                                                                                                                   | Diashow-Programm<br>verschiedene, teilweise digitalisierte Bilder                                                                                                                                                                                                                                                  | G<br>G                                                           |
| Fish 12   | Amiga3d<br><br>ArgoTerm<br><br>Arrow3d<br><br>Ld4<br><br>IconExec/<br><br>SetWindow<br><br>Images<br><br>SetAlternate<br><br>StarTerm | dreidimensionales Grafik-Demo<br>Terminal-Emulation Version 0.20<br>dreidimensionales Grafik-Demo<br>erweitertes CLI-Kommando »list«<br>diese beiden Programme erlauben den Start von Programmen per Icon<br>verschiedene, teilweise digitalisierte Bilder<br>Utility zum »mergen« von Icons<br>Terminal-Emulation | G<br><br>A<br><br>G<br><br>T<br><br>T<br><br>G<br><br>U<br><br>A |
| Fish 13   | Basic-Programme                                                                                                                       | enthält jede Menge Basic-Programme aus verschiedenen Bereichen                                                                                                                                                                                                                                                     | T                                                                |
| Fish 14   | Amiga3d<br>Beep<br>Dex<br>Dimensions<br>Filezap<br><br>Gfxmem<br><br>von Fish 1<br>Gi<br><br>PdTerm<br>Shell<br>Termcap               | Grafik-Demo generiert Beep-Sound<br>holt Text aus Source-Files<br>Grafik-Demo Update des Patch-Progr. von Fish 10<br>Update der Speicheranzeige<br><br>konvertiert DPaint-Brushes in C-Source<br>Terminal-Emulation einfache Shell<br>Implementation einer Termcap-Library                                         | G<br>M<br>U<br>G<br>T<br><br>U<br><br>T<br><br>A<br>U<br>T       |
| Fish 15   | Blobs<br>Clock<br>Dazzle<br>Fish<br>Monopoly<br>OkidataDump<br><br>Polydraw<br>Polyfractals                                           | Grafik-Demo<br>Digital-Uhr<br>Grafik-Demo<br>Grafik-Demo<br>Unterhaltungsspiel<br>Hardcopy-Routine für Oki ML92<br>Malprogramm<br>Fraktal-Generator                                                                                                                                                                | G<br>U<br>G<br>G<br>S<br>U<br><br>G<br>G                         |
| Fish 16   | Developer's-IFF                                                                                                                       | Kopie der »Amiga Developers IFF-disk« von Commodore-Amiga                                                                                                                                                                                                                                                          | T                                                                |
| Fish 17   | HAM-Demo                                                                                                                              | Diskette enthält HAM-Bilder                                                                                                                                                                                                                                                                                        | G                                                                |
| Fish 18   | AmigaDisplay<br>Ash<br>Browser                                                                                                        | Terminal-Emulation<br>Shell<br>»Workbench für Programmierer«                                                                                                                                                                                                                                                       | A<br>U<br>U                                                      |

| Diskette: | Programm:                                                                                                                                                | Bemerkung:                                                                                                                                                                                                                                                                                                                        | Art:                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
|           | MC68010<br><br>Multidim<br>PigLatin<br>Scrimper<br>Clisp1.6                                                                                              | Anleitung zur<br>Installation eines 68010<br>Grafik-Demo<br>Sprachausgabe in Latein<br>Hardcopy-Routine<br>einfache Lisp-Implementation                                                                                                                                                                                           | T<br><br>G<br>M<br>U<br>P                                           |
| Fish 19   | BlackJack<br>JayMinerSlides<br><br>Keymap__Test<br><br>LockMon                                                                                           | Kartenspiel<br>Funktionsdiagramme der Amiga-Chips<br>dient zum Testen der Keymap-Funktionen<br>zeigt geschützte Dateien an                                                                                                                                                                                                        | S<br>G<br><br>T<br><br>T                                            |
| Fish 20   | AmigaToAtari<br><br>DiskSalv<br>Hash<br><br>Hd<br>MandelBrots<br>MultiTasking<br>Pack<br>PortHandler<br>Random<br><br>SetMouse2<br>SpeechTerm<br><br>Ted | konvertiert Objekt-Code in ST-Format<br>File-Retter<br>berechnet Hash-Werte von Amiga-DOS<br>Hex-Dump-Routine<br>Mandelbrot-Grafiken<br>Multitasking-Demo<br>Packer für C-Source<br>Demo eines Port-Handlers<br>Zufallszahlen-Generator in Assembler<br>wechselt den Mouse-Port<br>Terminal-Emulation mit Sprachausgabe<br>Editor | U<br><br>U<br>T<br><br>T<br>G<br>T<br>U<br>T<br><br>T<br>A<br><br>U |
| Fish 21   | Mandelbrot-Disk                                                                                                                                          | enthält ein komplettes Paket zur Erzeugung von Mandelbrot-Grafiken                                                                                                                                                                                                                                                                | G                                                                   |
| Fish 22   | Lemarcs<br>Pemarcs                                                                                                                                       | Microemacs Version 3.6<br>Microemacs mit erweiterten Funktionen                                                                                                                                                                                                                                                                   | U<br>U                                                              |
| Fish 23   | Emacs V30                                                                                                                                                | Microemacs, Version 30                                                                                                                                                                                                                                                                                                            | U                                                                   |
| Fish 24   | Conquest<br>Csh<br>Modula-2                                                                                                                              | Weltraum-Adventure<br>erweitertes CLI<br>Vorversion eines ursprünglich für den Apple Macintosh erstellten Compilers                                                                                                                                                                                                               | S<br>U<br>P                                                         |
| Fish 25   | Hack                                                                                                                                                     | Version 1.0.1E. des Unix-Spieles<br>»Hack« mit verbesserter Grafikausgabe                                                                                                                                                                                                                                                         | S                                                                   |
| Fish 26   | UnHunk<br><br>C-Kermit<br>Ps<br>Archx                                                                                                                    | Manipulation der Lade-Dateien<br>DFÜ-Programm zeigt Prozeß-Prioritäten an<br>bündelt Textdateien                                                                                                                                                                                                                                  | T<br><br>A<br>T<br>T                                                |
| Fish 27   | ABdemos<br><br>Disassem<br>DvorakKeymap<br><br>Hypocycloids<br>LinesDemo<br>MemExpansion<br><br>SafeMalloc<br><br>ScienceDemos                           | Demo-Programme in Amiga-Basic<br>einfacher Disassembler<br>Beispiel einer »Keymap«-Struktur<br>elektronischer Spirograph<br>Grafik-Demo<br>Anleitung für den Bau einer 1 MByte-Speichererweiterung mit Schaltplan<br>Tool zur Speicherorganisation<br>einige Astronomie-Demos                                                     | T<br><br>U<br>T<br><br>A<br>G<br>T<br><br>T<br><br>A                |
| Fish 28   | Backgammon<br>cpp<br>Cribbage<br>Shar<br><br>SuperBitMap                                                                                                 | Brettspiel in Amiga-Basic<br>Vorübersetzer für C-Progr.<br>Spiel in Amiga-Basic<br>Packer, kompatibel mit Unix »shar«<br>Scroll-Demo                                                                                                                                                                                              | S<br>T<br>S<br>U<br><br>G                                           |
| Fish 29   | AegisDrawDemo<br><br>»AegisDraw«                                                                                                                         | vollständige Version des kommerziellen Malprog.<br>jedoch ohne »SAVE«-Funktion                                                                                                                                                                                                                                                    | G                                                                   |



| Diskette: | Programm:  | Bemerkung:                                                                    | Art: |
|-----------|------------|-------------------------------------------------------------------------------|------|
|           | cc         | C-Compiler-Controller, (Manx-Version)                                         | T    |
|           | Enough     | CLI-Kommando, Speicher-anzeige                                                | T    |
|           | Player     | spielt Aegis-Animator-Animationen                                             | G    |
|           | Rubik      | Zauberwürfel mit grafischer Animation                                         | S    |
|           | StringLib  | Implementation der Unix String Funktions-Library                              | T    |
|           | VT100      | Terminal-Emulation                                                            | A    |
| Fish 30   | BBS        | Mailbox-System, Version 1.1                                                   | A    |
|           | FineArt    | Sammlung toller Amiga-Bilder                                                  | G    |
|           | FontEditor | Zeichensatz-Editor zum Erstellen von Menüs, speichert als C-Source oder binär | U    |
|           | MenuEditor |                                                                               | U    |
|           | StarTerm   | Terminal-Programm, Version 3.0                                                | A    |
| Fish 31   | Life       | schnelle Version der bekannten Lebenssim.                                     | A    |
|           | Mandelbrot | Mandelbrot-Programm, Version 3.00                                             | G    |
|           | RamSpeed   | mißt Speicher-Geschwindigkeit                                                 | T    |
|           | Set        | ersetzt das Manx->set-<-Kommando                                              | T    |
|           | Tree       | Grafik-Demo                                                                   | G    |
|           | TxED       | verbesserte Version des Editors                                               | U    |
|           | Vdraw      | Malprogramm, Version 1.08                                                     | G    |
|           | Xicon      | Anzeige von Texten per Icons                                                  | T    |
| Fish 32   | Address    | Adressen-Kartei in Amiga-Basic                                                | A    |
|           | Calendar   | Tagebuch-Programm in Amiga-Basic                                              | A    |
|           | DosPlus1   | CLI-orientiertes Entwickler-Tool                                              | T    |
|           | DosPlus2   | wie DosPlus1                                                                  | T    |
|           | MacView    | dient zur Anzeige von Macintosh-Bildern auf dem Amiga                         | U    |
|           | Puzzle     | Verschiebe-Puzzle                                                             | S    |
|           | ShowHAM    | dient zur Anzeige von HAM-Bildern                                             | T    |
|           | Solitaire  | Spiel in Amiga-Basic                                                          | S    |
|           | Spin3      | Grafik-Demo                                                                   | G    |
|           | Sword      | Text-Adventure in Amiga-Basic                                                 | S    |
|           | Trails     | Grafik-Demo, Mauszeiger hinterläßt Spur                                       | G    |
| Fish 33   | 3dStars    | Grafik-Demo                                                                   | G    |
|           | BigMap     | Scroll-Demo                                                                   | G    |
|           | Dbuf.gels  | Grafik-Demo, Bobs und Sprites                                                 | G    |
|           | DiskMapper | zeigt die Belegung von Disketten-Sektoren                                     | U    |
|           | MemView    | ständige Speicheranzeige                                                      | U    |
|           | Oing       | Grafik-Demo                                                                   | G    |
|           | ScreenDump | Hardcopy-Routine                                                              | U    |
|           | Sdb        | einfache Datenbank                                                            | A    |
|           | Sproing    | Grafik-Demo                                                                   | G    |
|           | Stars      | Grafik-Demo                                                                   | G    |
|           | TermPlus   | Terminal-Emulation                                                            | A    |
|           | VT100      | Terminal-Emulation, Version 2.0                                               | A    |
| Fish 34   | Alint      | unterstützt »Gimpel-lint«                                                     | T    |
|           | Blink      | Linker als Ersatz für »Alink«                                                 | T    |
|           | Browser    | Update von Fish 18                                                            | U    |
|           | Btree      | Implementation eines »B-tree«-Algorithmus                                     | T    |
|           | Btree2     | Original-Version von »Btree«                                                  | T    |
|           | Calendar   | Kalender-Programm                                                             | A    |

| Diskette: | Programm:     | Bemerkung:                                                      | Art: |
|-----------|---------------|-----------------------------------------------------------------|------|
|           | Less          | dient zur Anzeige von Text-Files                                | U    |
|           | NewFonts      | 28 neue Amiga-Fonts                                             | T    |
|           | Pr            | Druck-Utility                                                   | U    |
|           | Requester     | File-Requester                                                  | T    |
| Fish 35   | ASendPacket   | DOS-Handling-Tool                                               | T    |
|           | ConsoleWindow | Finden des Intuition-Zeigers auf CON:                           | T    |
|           | DirUtil       | Utility à la CLI-Mate                                           | U    |
|           | Dirutil2      | Variante von »DirUtil«                                          | U    |
|           | FileRequester | Requester-Modul für Lattice C                                   | T    |
|           | MacView       | dient zur Anzeige von Macintosh-Bildern auf d. Amiga            | G    |
|           | Plop          | einfache IFF-Leseroutine                                        | T    |
|           | PopCLI        | CLI-Fenster öffnen per Hotkey, Screen-Blanker                   | U    |
|           | QuickCopy     | Kopierprogramm, drei Versionen                                  | U    |
|           | ScrollPf      | Scroll-Demo                                                     | G    |
|           | SendPacket    | DOS-Handling-Tool                                               | T    |
|           | SpriteMaker   | Sprite-Editor, speichert C-Source                               | U    |
|           | Tracker       | verwandelt Daten auf der Kickstart-Diskette in einzelne Files   | U    |
|           | TriClops      | Spiel mit 3D-Grafik                                             | S    |
|           | Tsize         | gibt Disk-Verzeichnisgrößen aus                                 | T    |
|           | Unlfddef      | Tool für Source-Code                                            | T    |
|           | VTtest        | testet Kompatibilität von VT100-Terminals                       | T    |
| Fish 36   | Acp           | Kopieroutine à la Unix »cp«                                     | T    |
|           | Clock         | Update von Fish15                                               | U    |
|           | Csh           | CLI-Verbesserung, Version 2.01A                                 | U    |
|           | DietAid       | elektronischer Diät-Plan                                        | A    |
|           | Echo          | verbessertes »echo«                                             | T    |
|           | FixHunk       | adaptiert Programme für Zusatzspeicher                          | T    |
|           | Fm            | zeigt an, welche Sektoren ein File auf Diskette belegt          | T    |
|           | KickBench     | Tips zur Erstellung einer biniierten Kickstart-Workbench-Disk   | T    |
|           | Lex           | prüft Lesbarkeit von Texten                                     | U    |
|           | TunnelVision  | 3D-Labyrinth-Spiel                                              | S    |
|           | Vc            | Tabellenkalkulation                                             | A    |
|           | VT100         | Terminal-Emulation, Version 2.2                                 | A    |
|           | YaBoing       | Spiel zur Sprite-Demonstr.                                      | S    |
| Diskette: | Programm:     | Bemerkung:                                                      | Art: |
| Fish 37   | Smalltalk     | Programmiersprache                                              | P    |
| Fish 38   | CSquared      | Grafik-Algorithmus                                              | G    |
|           | FixObj        | kürzt XModem-Files                                              | T    |
|           | Handler       | Erstellung eines Amiga-DOS-Handlers                             | T    |
|           | HP-10c        | simuliert einen Hewlett-Packard HP-10c                          | U    |
|           | IffDump       | zwei Tools für IFF-Files                                        | T    |
|           | Jsh           | einfache CLI-Alternative                                        | U    |
|           | NewStat       | Alternative zu »status«                                         | T    |
|           | Reversi       | Strategiespiel                                                  | S    |
|           | UUdecode      | dient zum Codieren und Decodieren von Binär-Files               | U    |
|           | Vdraw         | Malprogramm, Version 1.14                                       | G    |
|           | VoiceFiler    | Soundverwaltung für Yamaha DX-Synthesizer über Midi             | M    |
|           | Window        | Window-Demo                                                     | T    |
| Fish 39   | AsiEcho       | Kommandos wie »cls« etc.                                        | T    |
|           | Display       | dient zur Anzeige von HAM-Bildern mit einigen tollen Beispielen | G    |



| Diskette: | Programm:                                                                                                                                                                 | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Art:                                                                                    |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|           | Driver<br>Xlisp                                                                                                                                                           | Demo-Device-Treiber<br>kleiner Lisp-Interpreter<br>(V1.7)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | T<br>P                                                                                  |
| Fish 40   | Ahost<br><br>AmigaMonitor<br><br>Arc<br>AreaCode<br>Blink<br><br>Cosmo<br>Dg210<br><br>DirUtil<br><br>DOSHelper<br><br>PagePrint<br><br>PopCLI<br><br>SpriteEd<br>X-Spell | Terminal-Emulation,<br>Version 0.9<br>Status-Anzeige aller<br>Tasks etc.<br>Packer, Version 0.16<br>Decodier-Tool<br>Linker als Ersatz für<br>»Alink«, Version 6.5<br>Action-Spiel la »Asteroids«<br>Terminal-Emulation<br>eines D-210<br>CLI-Benutzeroberfläche<br>(V1.4)<br>»Help« für DOS-<br>Kommandos (V1.6)<br>druckt Text-Files mit<br>Seitennummer etc.<br>CLI-Fenster per »Hotkey«<br>öffnen, Screen-Blanker,<br>Version II<br>Sprite-Editor<br>Spell-Checker                                                                         | A<br><br>U<br><br>U<br>T<br>T<br><br>S<br>A<br><br>U<br><br>U<br><br>U<br><br>U<br>U    |
| Fish 41   | AmigaVenture<br><br><br>Csh<br>Dbug<br><br>DualPlayField<br>GetFile<br>LatticeXref<br><br>Lines<br>SetFont<br>VT100                                                       | damit sind eigene<br>Adventure im Infocom-Stil<br>zu erstellen<br>Shell, Version 2.03<br>C-Debugger auf<br>Makro-Basis<br>Grafik-Demo<br>File-Requeser<br>Cross-Referenz für<br>Lattice 3.10<br>Grafik-Demo<br>ändert Workbench-Font<br>Terminal-Emulation                                                                                                                                                                                                                                                                                     | S<br><br><br>U<br>U<br><br>G<br>T<br>T<br><br>G<br>T<br>A                               |
| Fish 42   | Micro<br>GNUEMACS                                                                                                                                                         | Amiga-Version des Editors                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | U                                                                                       |
| Fish 43   | BasicBoing<br><br>Bbm<br><br>BbsList<br><br>cc<br><br>Copper<br><br>InstrIFF<br><br>PopColours<br><br>SpriteClock<br>STEMulator<br><br>WBrun<br><br>Wild                  | Grafik-Demo in<br>Amiga-Basic<br>Demo-Version von<br>»B.E.S.T Software's<br>Business Management<br>System«<br>Liste von Mailboxen, die<br>den Amiga unterstützen<br>Compiler-Controller für<br>Manx und Lattice C<br>Disassembler für<br>Copper-Liste<br>konvertiert digital. Sounds<br>in das IFF-Format<br>erlaubt jederzeit die<br>Veränderung der<br>Workbench-Farben (V1.0)<br>Uhr in Form eines Sprites<br>Gag-Simulation eines<br>Atari ST<br>Aufruf von Workbench-<br>Programmen vom CLI<br>zwei Versionen einer<br>Unix-artigen Shell | G<br><br>A<br><br>T<br><br>T<br><br>U<br><br>M<br><br>U<br><br>U<br>G<br><br>U<br><br>U |
| Fish 44   | Icons<br>NewIFF<br>RayTracePics<br>ViewLBM                                                                                                                                | verschiedene Icons<br>Toll für IFF Sound-Files<br>verschiedene HAM-Bilder<br>zeigt IFF-Bilder an<br>(auch HAM)                                                                                                                                                                                                                                                                                                                                                                                                                                 | G<br>M<br>G<br><br>G                                                                    |
| Fish 45   | Clue<br>Make<br><br>Pictures<br>Update                                                                                                                                    | Brettspiel<br>Implementation des<br>Unix-Kommandos »make«<br>einige tolle Bilder<br>Hilfe beim Erneuern<br>von Dateien                                                                                                                                                                                                                                                                                                                                                                                                                         | S<br>T<br><br>G<br>U                                                                    |

| Diskette: | Programm:                                                                                                  | Bemerkung:                                                                                                                                                                                                                                                         | Art:                                                          |
|-----------|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
|           | Wherels                                                                                                    | sucht nach einer Datei                                                                                                                                                                                                                                             | U                                                             |
| Fish 46   | Asm<br>CheckModem<br>Egad<br>Jive<br><br>My.lib<br>ProffMacros<br><br>ValSpeak                             | Makro-Assembler<br>prüft Modem-Verbindung<br>Gadet-Editor<br>wandelt Eingaben in<br>»Jive-Slang«<br>Runtime-Library<br>Implementation der<br>Makro-Packages für<br>»Proff«<br>wandelt Eingaben in<br>»Val-Slang«                                                   | P<br>U<br>U<br>A<br><br>T<br>T<br><br>A                       |
| Fish 47   | 3D-Arm<br>Juggler<br><br>VT100                                                                             | Grafik-Demo<br>Grafik-Demo im<br>HAM-Modus<br>Terminal-Emulation (V2.4)                                                                                                                                                                                            | G<br>G<br><br>A                                               |
| Fish 48   | Bru<br>Comm<br><br>Csh2.04<br>Csh2.04M<br>Diskperf<br><br>Du<br>MemWatch<br>Profiler                       | Harddisk-File-Archivierung<br>Terminal-Emulation,<br>Version 1.30<br>Shell<br>Shell, modifiziert f. Manx C<br>Benchmark-Test,<br>Diskettenzugriff<br>zeigt Disketten-Belegung an<br>Speicher-Wächter<br>bestimmt Laufzeiten<br>unter Manx 3.30E                    | T<br>A<br><br>U<br>U<br>U<br><br>U<br>U<br>T                  |
| Fish 49   | Cycloids<br>DirUtil<br><br>Mutidef<br><br>MyUpdate<br>Plot<br>Polygon<br>QMouse<br><br>Touch<br><br>Trees  | Update von Fish 27<br>CLI-Benutzeroberfläche,<br>Update von Fish 35<br>sucht mehrfach definierte<br>Symbole<br>Update-Utility<br>zeichnet 3D-Funktionen<br>Grafik-Demo<br>Abfrage der linken<br>Maustaste<br>Aktualisierung des Datums<br>von Files<br>Grafik-Demo | A<br>U<br><br><br><br>T<br>U<br>G<br>G<br>T<br><br>U<br><br>G |
| Fish 50   | Asm<br><br>BreakOut<br>DiskZap<br>FirstSiliCon<br>Missile<br>PerfectSound<br>Sizzlers<br>UnicArc<br>Wombat | Shareware-Makro-<br>Assembler<br>3D-Spiel für Rot-Blau-Brille<br>Disk-Editor, sektorweise<br>verbessertes CLI<br>Action-Spiel<br>Sound-Editor mit Beispielen<br>Grafik-Demo<br>Packer<br>Terminal-Emulation, (V3.01)                                               | P<br><br>S<br>U<br>U<br>G<br>M<br>G<br>U                      |
| Fish 51   | Bison<br><br>Compress<br>Cos<br>DifSsed<br>Sq.Usq                                                          | Implementation von<br>Unix »yacc«<br>Packer<br>Spiel in Amiga-Basic<br>Vergleich von Files<br>Packer                                                                                                                                                               | T<br><br>U<br>S<br>U<br>U                                     |
| Fish 52   | assign<br>Fractal<br><br>HAMpoly<br>MxGads<br>Poly<br>Tek4010<br>Vdraw1.16<br>Vdraw1.19                    | Ersatz für »assign«<br>erzeugt fraktale<br>Landschaften<br>Grafik-Demo in HAM<br>Gadget-Tool<br>Grafik-Demo<br>Terminal-Emulation<br>Malprogramm<br>Malprogramm, erweiterte<br>Version                                                                             | T<br>G<br><br>G<br>T<br>G<br>A<br>G<br>G                      |
| Fish 53   | Animations<br><br>ARCre<br><br>Arp<br><br>Compiler<br>SpreadSheet                                          | Aegis-Animator-<br>Animationen mit Player<br>Umbenennungsroutine<br>für »Arc«<br>ersetzt CLI-Befehle durch<br>schnellere Assembler-<br>Äquivalente<br>C-Compiler<br>Tabellenkalkulation                                                                            | G<br><br>T<br><br>T<br><br>P<br>A                             |



| Diskette: | Programm:                                                                                                                          | Bemerkung:                                                                                                                                                                                                                                                                                                                   | Art:                                                      |
|-----------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
|           | TarSplit<br>UUencode                                                                                                               | extrahiert Dateien<br>codiert/decodiert binäre<br>Dateien                                                                                                                                                                                                                                                                    | T<br>T                                                    |
| Fish 54   | Hanoi<br>ISpell<br>Ing<br>Lav<br>MidiTools<br>MoreRows<br><br>Tilt                                                                 | Grafik-Demo/Rekursion<br>Spell-Checker<br>Grafik-Demo<br>zeigt laufende Tasks an<br>einfacher Midi-Sequencer<br>vergrößert Workbench-<br>Screen<br>Grafik-Demo                                                                                                                                                               | G<br>U<br>G<br>U<br>M<br>T<br><br>G                       |
| Fish 55   | Csh<br>NewStartups<br><br>Palette<br><br>PipeDevice<br><br>ScreenSave<br><br>ShanghaiDemo<br><br>SoundExample<br>Vsprites<br>VT100 | Shell, Version 2.05<br>»Startups« für C-<br>Programmierer<br>dient zur Farbveränderung<br>in Programmen<br>Datenaustausch zwischen<br>Prozessen<br>speichert Screen als<br>IFF-File, auch HAM<br>Demo-Version des<br>kommerziellen Brettspiels<br>Sound-Demo für Manx C<br>Grafik-Demo<br>Terminal-Emulation,<br>Version 2.6 | U<br>T<br><br>U<br><br>T<br>T<br>S<br>M<br>G<br>A         |
| Fish 56   | ClipBoard<br>ConPackets<br><br>GetDisks<br><br>GetVolume<br>Icon2C<br><br>MergeMem<br><br>mCAD                                     | Clipboard-Gerätetreiber<br>zeigt Benutzung von<br>»DOS-Packets«<br>gibt alle Namen der<br>verfügbaren Diskettenlauf-<br>werke als Exec-Liste aus<br>gibt Diskettennamen aus<br>wandelt Icon-File in<br>C-Sourcecode<br>verschmilzt Speicher-<br>bereiche<br>einfaches CAD-Programm<br>(V1.1)                                 | T<br>T<br><br>T<br><br>T<br>T<br>T<br>G                   |
| Fish 57   | CutAndPaste<br><br>GraphIt<br>Juggler<br><br>MouseReader<br><br>Ogre<br>Splines                                                    | Implementation von<br>Unix »cut/paste«<br>Funktionsplotter<br>Grafik-Demo in HAM,<br>Update von Fish 47,<br>Version 1.2<br>dient zur Anzeige von Text-<br>und IFF-Files<br>Weltraumspiel<br>Grafik-Demo                                                                                                                      | T<br><br>G<br>G<br><br>U<br><br>S<br>G                    |
| Fish 58   | ASDG-rdd<br>BigView<br>EGraph<br><br>HyperBase<br>MemClear<br><br>NewZap<br><br>RainBow<br>View<br>WBdump                          | resetfeste RAM-Disk<br>zeigt jedes IFF-Bild an<br>zeichnet Kurven aus<br>Datenpaaren<br>Datenbank, Version 1.5<br>setzt Speicherbereiche<br>auf Null<br>Editor für File-Sektoren<br>(V3.0)<br>Grafik-Demo<br>zeigt IFF-Bilder an<br>Hardcopy für JX-80                                                                       | U<br>G<br>G<br><br>A<br>T<br><br>U<br><br>G<br>G<br>U     |
| Fish 59   | Browser<br>Browser2<br><br>Clock<br>Dme<br>DropCloth<br><br>DropShadow<br><br>mCAD<br>Robotroff<br><br>Supermort                   | Update von Fish 18, 34<br>zeigt Dateien und<br>Verzeichnisse<br>ständig sichtbare Uhr<br>Editor, Version 1.22<br>ändert Workbench-<br>Hintergrund<br>setzt Schatten hinter<br>Windows<br>Update von 56 (V1.2.2)<br>Grafik-Demo, kleine<br>Roboter jagen hinter dem<br>Mauszeiger her<br>Finanzprogramm                       | U<br>U<br><br>U<br>U<br><br>T<br>G<br><br>G<br>G<br><br>A |

| Diskette: | Programm:                                                                                                                                                                             | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Art:                                                                                                    |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Fish 60   | Blitz<br><br>BlitzFonts<br><br>HandShake<br><br>Med<br>PrtDrvGen<br><br>Show<br>Uedit<br>Ueturbo                                                                                      | zeigt Text-Files an,<br>wird per »Hotkey« gestartet<br>erhöht die Geschwindigkeit<br>der Textausgabe<br>Terminal-Emulation,<br>Version 1.20a<br>Editor, Version 2.1<br>erzeugt individuelle<br>Druckertreiber<br>Diashow-Programm<br>Editor, Version 2.0<br>Unterstützung für »Uedit«                                                                                                                                                                                                                                     | U<br><br>T<br><br>A<br><br>U<br>T<br><br>G<br>U<br>T                                                    |
| Fish 61   | ATPatch<br><br>FillDisk<br><br>LPatch<br><br>MicroEmacs<br>PearlFont<br>Terrain<br>Vsprites                                                                                           | paßt alte Version des<br>MS-DOS-Emulators an<br>Workbench 1.2 an<br>»nullt« freie Blocks von<br>Disketten aus<br>paßt alte Programme an<br>Workbench 1.2 an<br>Editor, Version 3.8b<br>Zeichensatz<br>Grafik-Demo<br>Sprite-Demo                                                                                                                                                                                                                                                                                          | T<br><br>U<br><br>T<br><br>U<br>T<br>G<br>G                                                             |
| Fish 62   | Hack                                                                                                                                                                                  | Unix-Spiel, Version 1.0.3D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | S                                                                                                       |
| Fish 63   | Larn                                                                                                                                                                                  | Unix-Spiel, Version 12.0B                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | S                                                                                                       |
| Fish 64   | Developer's IFF                                                                                                                                                                       | Update von Fish 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | T                                                                                                       |
| Fish 65   | Bawk<br><br>CloseWB<br>Cookie<br>JTime<br><br>MenuBuilder<br><br>MWB<br><br>NewPackets<br><br>PascalToC<br><br>um Prep<br><br>RunBack<br>SunMouse                                     | Implementation des Unix<br>Text-Prozessors »awk«<br>schließt Workbench-Screen<br>Sprücheklopfer<br>Bauanleitung für eine<br>batteriegepufferte Echt-<br>zeituhr<br>generiert Menüs in<br>C-Source<br>generiert neuen<br>Workbench-Screen<br>beschreibt Strukturen von<br>Amiga-DOS 1.2<br>wandelt Pascal-Programme<br>in C<br>FORTRAN-Vorübersetzer<br>(V2.1)<br>verbessertes »run«<br>simuliert »Sunwindows-<br>Maus«                                                                                                    | U<br><br>T<br>A<br>T<br><br>T<br><br>T<br>T<br><br>P<br><br>P<br>T<br>U                                 |
| Fish 66   | AmScsi<br><br>Asm68k<br><br>Assigned<br><br>Dk<br>Flip<br>Foogol<br><br>Free<br><br>MallocTest<br><br>Melt<br>Nart<br>Purty<br><br>RayTracer<br><br>SendPackets<br>SnapShot<br>TagBBS | Beschreibung zum Bau<br>eines SCSI-Controllers<br>Makro-Assembler,<br>Version 1.0.1<br>zeigt an, ob ein Name<br>mittels »assign«<br>umgelenkt wurde<br>Grafik-Demo<br>Grafik-Demo<br>Foogol-Compiler<br>(Algol-ähnlich)<br>zeigt freien Platz auf<br>Diskette<br>erlaubt Speicher-<br>überprüfung<br>Grafik-Demo<br>Grafik-Demo<br>erlaubt Veränderung der<br>Drucker-Einstellungen per<br>Gadgets<br>einfaches Raytracing-<br>Programm<br>Update von Fish-Disk 35<br>Hardcopy-Routine<br>Mailbox-System,<br>Version 1.02 | T<br><br>P<br><br>T<br><br>G<br>G<br>P<br><br>T<br><br>T<br><br>G<br>G<br>U<br><br>G<br><br>T<br>U<br>A |
| Fish 67   | AmCat<br>AmigaSpell<br>Bouncer                                                                                                                                                        | Diskettenverwaltung<br>Spell-Checker, Version 2.0<br>Grafik-Demo in Forth                                                                                                                                                                                                                                                                                                                                                                                                                                                 | A<br>U<br>G                                                                                             |



| Diskette: | Programm:     | Bemerkung:                                                                                      | Art: |
|-----------|---------------|-------------------------------------------------------------------------------------------------|------|
|           | Comm          | Terminal-Emulation, Version 1.33                                                                | A    |
|           | Dux5          | CLI-Benutzeroberfläche                                                                          | U    |
|           | HexCalc       | Rechner für Hex-, Oktal-, Dezimal- und Dualzahlen                                               | U    |
|           | Icons         | verschiedene Icons                                                                              | T    |
|           | Mandala       | Grafik-Demo mit Musik                                                                           | G    |
|           | PersMait      | Dateiverwaltung, Demo                                                                           | A    |
|           | RSLClock      | ständig sichtbare Uhr                                                                           | U    |
|           | RTCubes       | Grafik-Demo                                                                                     | G    |
|           | Wheel         | Glücksrad-Spiel                                                                                 | S    |
| Fish 68   | MicroGNUEmacs | leistungsfähiger Editor, läuft in dieser Version auch auf Unix-Systemen, unter VMS oder Sun 3.2 | U    |
| Fish 69   | Asm68k        | Makro-Assembler, Version 1.0.3                                                                  | P    |
|           | BlitLab       | dient für Experimente mit dem Blitter                                                           | U    |
|           | ConMan        | verbessert Editierung etc. unter CON:                                                           | T    |
|           | Console       | verschiedene Routinen                                                                           | T    |
|           | Dk            | Grafik-Demo                                                                                     | G    |
|           | FragS         | zeigt freie Speicherbereiche an                                                                 | T    |
|           | IconType      | ändert Typ von Icons                                                                            | U    |
|           | Make          | »make« für Manx                                                                                 | T    |
|           | MonProc       | zeigt Prozeß-Aktivitäten an                                                                     | T    |
|           | MouseClock    | verwandelt Mauszeiger in Uhr                                                                    | U    |
|           | Sb            | zeigt System-Strukturen an                                                                      | T    |
|           | Spew          | erzeugt Überschriften wie »National Enquirer«                                                   | G    |
|           | Spool         | Drucker-Spooler                                                                                 | U    |
|           | Wc            | Wortzähler                                                                                      | U    |
| Fish 70   | AmigaMonitor  | zeigt dynamisch den Status des Amiga an: geöffnete Files, aktive Tasks etc.                     | U    |
|           | Arc           | Packer, Version 0.23                                                                            | U    |
|           | BlackBook     | Adressenverwaltung                                                                              | A    |
|           | DoTil         | erweitertes CLI-Kommando »list«                                                                 | T    |
|           | GravityWars   | Action-Spiel                                                                                    | S    |
|           | Jobs          | Benutzeroberfläche, Version 2.1                                                                 | U    |
|           | Lens          | vergrößert Bereich um den Mauszeiger                                                            | U    |
|           | Life-3D       | 3D-Version des Klassikers                                                                       | A    |
|           | Logo          | Logo-Interpreter, ähnlich Apple II-Logo                                                         | P    |
|           | SetKey        | Keymap-Editor, Demo-Version 1.0                                                                 | U    |
|           | Vpg           | Testbild-Generator, Version 1.0                                                                 | U    |
| Fish 71   | AirFoil       | berechnet Tragflügel und Stromlinien                                                            | A    |
|           | AmiBas        | verschiedene Basic-Programme                                                                    | U    |
|           | Blocks        | Grafik-Demo                                                                                     | G    |
|           | Comm          | Terminal-Emulation, Version 1.34                                                                | A    |
|           | DiskX         | Disketten-Editor                                                                                | U    |
|           | Fpic          | Manipulation von IFF-Bildern                                                                    | G    |
|           | IconMk        | erzeugt Icons, Version 1.2a                                                                     | U    |
|           | NewFonts      | zwei neue Fonts: »shalt18«, »ibm5«                                                              | T    |
|           | PetCLI        | CLI-Oberfläche in Amiga-Basic                                                                   | U    |
|           | PWDEmo        | Demo-Version von Power-Windows                                                                  | P    |
|           | Rot           | generiert 3D-Objekte                                                                            | G    |
|           | TimeSet       | erlaubt Einstellen der Systemzeit                                                               | U    |
| Fish 72   | Pictures      | jede Menge Bilder,                                                                              |      |

| Diskette: | Programm:     | Bemerkung:                                                                                       | Art: |
|-----------|---------------|--------------------------------------------------------------------------------------------------|------|
|           |               | auch HAM                                                                                         |      |
| Fish 73   | Add           | generiert »Short-Cuts« für beliebige Programme                                                   | U    |
|           | AutolconOpen  | täuscht der Workbench Mouseingaben vor                                                           | T    |
|           | Dio           | Gebrauch von »exec.devices«                                                                      | T    |
|           | Dissolve      | blendet IFF-Bilder ein                                                                           | G    |
|           | DTerm         | Terminal-Emulation                                                                               | A    |
|           | Expose        | legt Fensterfreie frei                                                                           | A    |
|           | Lit           | File-Filter                                                                                      | T    |
|           | Lmv           | Diashow-Programm                                                                                 | U    |
|           | MouseOff      | läßt Mauszeiger bis zur nächsten Bewegung der Maus verschwinden                                  | G    |
|           | ParOut        | Zugriff auf Parallel-Port                                                                        | T    |
|           | PenPalFonts   | Zeichensatz, ähnlich einer Handschrift                                                           | T    |
|           | RunBackGround | verbessertes CLI-Kommando »run«                                                                  | T    |
|           | SnapShot      | Hardcopy-Routine, Update Fish 66                                                                 | U    |
|           | TypeAndTell   | demonstriert die Installation eines Device-Treibers vor dem der Intuition zeigt System-Listen an | T    |
|           | Xplor         |                                                                                                  | U    |
| Fish 74   | Cled          | Kommandozeilen-Editor (V1.3)                                                                     | U    |
|           | Control       | erlaubt besseren Zugriff auf die »printer-device« Editor, Update zu Fish 59 (V1.25)              | U    |
|           | Dme           | Update zu Fish 59 (V2.0)                                                                         | U    |
|           | DropShadow    | verwaltet Börsenkurse                                                                            | G    |
|           | Funds         | Update zu Fish 34 (V1.1)                                                                         | A    |
|           | Less          | erstellt »make«-Datei                                                                            | U    |
|           | Makemake      | Update zu Fish 59 (V1.2.4)                                                                       | T    |
|           | mCAD          | Zufallszahlen-Generator in C                                                                     | G    |
|           | Random        | I/O-Monitor                                                                                      | T    |
|           | TDebug        | rechnet verschiedene Einheiten um                                                                | U    |
|           | Units         | verbessertes »copy«                                                                              | U    |
|           | XCOPY         |                                                                                                  | T    |
| Fish 75   | Bezier        | Grafik-Demo                                                                                      | G    |
|           | BSplines      | Grafik-Demo                                                                                      | G    |
|           | Comm          | Source-Code zu Fish 71                                                                           | A    |
|           | Copy          | verbessertes »copy«                                                                              | T    |
|           | Diff          | vergleicht zwei Dateien                                                                          | T    |
|           | DuM2          | CLI-Benutzeroberfläche in Modula                                                                 | U    |
|           | Eless         | verbessertes CLI-Kommando »dir«                                                                  | T    |
|           | Fd            | verbessertes CLI-Kommando »dir«                                                                  | T    |
|           | Hardcopy      | erzeugt Hardcopy einer »CLI-Session«                                                             | U    |
|           | MouseOff      | Update zu Fish 73                                                                                | T    |
|           | SetFont       | ändert Workbench-Font (V2.0)                                                                     | T    |
|           | SpeedDir      | verbessertes »dir«                                                                               | T    |
| Fish 76   | Draco 1       | Programmiersprache                                                                               | P    |
| Fish 77   | Draco 2       | Programmiersprache                                                                               | P    |
| Fish 78   | Cycles        | Spiel à la »Iron« (V1.0)                                                                         | S    |
|           | EOMS          | »Experts Only Mercenary Simulator«, schwieriges Spiel                                            | S    |
|           | MandelVroom   | Mandelbrot-Generator (V1.50)                                                                     | G    |
| Fish 79   | AsmTools      | CLI-ähnliche Befehle                                                                             | T    |
|           | AssignDev     | erlaubt einem Device mehrere Namen zuzuweisen                                                    | T    |
|           | AuxHandler    | Terminal-Datentransfer                                                                           | T    |
|           | Cmd           | lenkt die Daten, die an den seriellen oder parallelen                                            | T    |



| Diskette: | Programm:                                             | Bemerkung:                                                                                                                             | Art:                  |
|-----------|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
|           | Info<br>Kill<br>M2Error                               | Port geschickt werden, auf ein File um verbessertes »info« entfernt Tasks zeigt Errors beim Compilieren mit dem TDI Modula-Compiler an | T<br>T<br>T           |
|           | MonProc<br>Mounted                                    | Update von Fish 69 zeigt an, ob eine Diskette »gemountet« ist                                                                          | T<br>T                |
|           | Nro<br>PaTask<br>QueryAny                             | Text-Formatier-Routine sucht den »Parent«-Task erlaubt Abfragen in CLI-Scripts                                                         | T<br>T<br>T           |
|           | ScnSizer                                              | ändert Maximalwerte für Screens                                                                                                        | T                     |
|           | SharedLib                                             | Beispiel einer geteilten Library                                                                                                       | T                     |
|           | Task                                                  | Beispiel des Gebrauchs von »CreateTask«                                                                                                | T                     |
|           | Uw<br>Who                                             | Windows-Tool listet Tasks in Warteschlangen                                                                                            | T<br>U                |
| Fish 80   | zurückgezogen                                         | Fish 80 wurde aus Copyright-Gründen zurückgezogen                                                                                      |                       |
| Fish 81   | Asm68k                                                | Makro-Assembler, Version 1.1                                                                                                           | P                     |
|           | AutoFacc                                              | entfernt das Window von »FACC«                                                                                                         | T                     |
|           | Brushes                                               | DPaint-Brushes für Schaltpläne                                                                                                         | G                     |
|           | CheckIFF                                              | prüft IFF-Strukturen                                                                                                                   | T                     |
|           | Cled                                                  | einfacher Editor                                                                                                                       | U                     |
|           | Conman                                                | Update von Fish 69 (V0.99B)                                                                                                            | U                     |
|           | Fonts                                                 | verschiedene Zeichensätze                                                                                                              | T                     |
|           | Icon                                                  | Programmiersprache Icon (V6.0)                                                                                                         | P                     |
|           | KeyLock                                               | Paßwort-Abfrage                                                                                                                        | U                     |
|           | Scat                                                  | Grafik-Demo                                                                                                                            | G                     |
|           | Smush                                                 | Grafik-Demo                                                                                                                            | G                     |
|           | Target                                                | Grafik-Demo                                                                                                                            | G                     |
| Fish 82   | Adventure<br>AmiCTerm<br>D2D-Demo                     | Abenteuerspiel Terminal-Programm, (V0.50) Demo-Version von »Disk-2-Disk« dient zum Datenaustausch zwischen Amiga-DOS und MS-DOS        | S<br>A<br>U           |
|           | DX-Synth<br>DiskMan<br>Icons<br>Panl                  | Update von Fish 38 CLI-Benutzeroberfläche verschiedene Icons universelles Utility zur Veränderung von Parametern bei Midi-Geräten      | M<br>U<br>T<br>M      |
|           | Rocket<br>Sand                                        | Grafik-Demo<br>Grafik-Demo                                                                                                             | G<br>G                |
| Fish 83   | TeX-Demo                                              | enthält eine Demo-Version des professionellen Satzprogramms »AmigaTeX«                                                                 | A                     |
| Fish 84   | AudioTools<br>BlitLab<br>Ed<br>GravityWars<br>HunkPad | Zugriff auf Audio-Device Update zu Fish 69 (V1.2) Editor Update zu Fish 70 (V1.04) ändert »Hunks« einer Ladedatei                      | M<br>U<br>U<br>S<br>T |
|           | PipeHandler                                           | Austausch zwischen Prozessen                                                                                                           | T                     |
|           | PopCLI                                                | Update zu Fish 40                                                                                                                      | U                     |
|           | Requester                                             | Update zu Fish 34                                                                                                                      | T                     |
|           | ScottDevice                                           | SCSI-Treiber                                                                                                                           | T                     |
|           | Viacom                                                | Grafik-Demo                                                                                                                            | G                     |
| Fish 85   | Csh<br>FileReq                                        | Shell, Version 2.06 File-Requester                                                                                                     | U<br>T                |

| Diskette: | Programm:                                          | Bemerkung:                                                                                    | Art:             |
|-----------|----------------------------------------------------|-----------------------------------------------------------------------------------------------|------------------|
|           | Hide                                               | läßt »schlecht« programmierte Software mit Expansion-RAM laufen                               | T                |
|           | ImageTools                                         | Tools zur Manipulation von IFF-Bildern                                                        | G                |
|           | LowMem                                             | gibt Programmen Auskunft bei knappem Speicher                                                 | T                |
|           | Plot6<br>RawIO                                     | Grafik-Demo Implementation von Unix »CBREAK«                                                  | G<br>T           |
|           | Rocket<br>CMore                                    | Grafik-Demo dient zum Betrachten von Text-Files                                               | G<br>U           |
|           | Vnews                                              | zeigt »News«-Files an                                                                         | T                |
| Fish 86   | AutoPoint                                          | wählt automatisch das Window unter dem Mauszeiger, zugleich »ScreenSaver«                     | U                |
|           | ClickToFront                                       | bringt durch Doppelklick auf ein Window dieses in den Vordergrund                             | U                |
|           | Cmd                                                | lenkt Ausgaben an den seriellen oder parallelen Port auf File um (Version 3)                  | T                |
|           | FileISG-Demo                                       | Datenbank-Demo                                                                                | A                |
| Fish 87   | AdvSys                                             | Software zum Erstellen eigener Text-Adventures, Version 1.2                                   | S                |
|           | AutolconOpen<br>Claz                               | Update zu Fish 73 (V1.2) wandelt IFF-Bilder in Postscript-Format                              | T<br>U           |
|           | Commodities                                        | »exec.library« für Anwender-Programme                                                         | T                |
|           | Diff                                               | Update zu Fish 75                                                                             | T                |
|           | Dme                                                | Update zu Fish 74 (V1.27)                                                                     | U                |
|           | DropShadow                                         | wie Fish 74, Version 2.0                                                                      | G                |
|           | Elib                                               | Erstellen einer aufgeteilten Library unter Manx C                                             | T                |
|           | ID-Handler                                         | erzeugt eindeutige Namen                                                                      | T                |
|           | Install                                            | verbessertes »install«                                                                        | T                |
|           | MemWatch                                           | Speicherwächter, Version II                                                                   | U                |
|           | MovePointer                                        | bewegt Mauszeiger zu Position x/y                                                             | T                |
|           | MoveWindow                                         | bewegt Window zu Position x/y                                                                 | T                |
|           | PalTest                                            | stellt fest, ob ein PAL-System vorhanden ist                                                  | T                |
|           | Sc                                                 | erstellt Zufalls-Landschaft                                                                   | G                |
|           | Tek4695                                            | Druckertreiber                                                                                | T                |
|           | WBDualPF                                           | zeigt »Dual-Playfield«-Modus                                                                  | T                |
|           | WarpText                                           | schnelle Textausgabe                                                                          | T                |
|           | YaIFFr                                             | liest IFF-Bilder, auch HAM                                                                    | G                |
|           | Zoo                                                | Packer                                                                                        | U                |
| Fish 89   | AutoEnquirer<br>DEMOfition<br>DirMaster<br>FuncKey | Requester Grafik-Demo Diskettenverwaltung dient zur Tastenbelegung mit Kommandos etc. (V1.01) | T<br>G<br>A<br>U |
|           | MFF-Demo                                           | »MicroFiche-Filer«-Demo, Datenbank                                                            | A                |
|           | Screenshift                                        | erlaubt Veränderung der Bild-Position                                                         | U                |
|           | Snake                                              | Grafik-Demo                                                                                   | G                |
| Fish 90   | AmiGazer                                           | bringt Sternenhimmel auf den Bildschirm, kennt über 1500 Sterne                               | A                |
|           | CardFile                                           | Dateiverwaltung speziell zur Datensammlung für Studienzwecke                                  | A                |
|           | Conman                                             | verbessert Editierung etc. unter CON: (V0.98B)                                                | T                |
|           | IMandelVroom                                       | Update zu Fish 78                                                                             | G                |
|           | NewDemos                                           | verschiedene Grafik-Demos                                                                     | G                |
|           | Othello                                            | Brettspiel                                                                                    | S                |
|           | PrintText                                          | zeigt ASCII-Files an                                                                          | U                |



| Diskette: | Programm:                                                                                | Bemerkung:                                                                                                                                                                                                        | Art:                                 |
|-----------|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
|           | PrtDrvGen                                                                                | generiert individuelle Druckertreiber (V2.2b)                                                                                                                                                                     | U                                    |
|           | Rainbench<br>ShortCut                                                                    | Grafik-Demo<br>Ersetzen von Tastenkombinationen durch einzelnen Tastendruck                                                                                                                                       | G<br>T                               |
|           | ShowPrint<br>Sizzlers<br>SpaceAceDemo<br>Timer                                           | Hardcopy von IFF-Screens<br>verschiedene Grafik-Demos<br>Schießspiel<br>erzeugt Window mit einem Timer                                                                                                            | U<br>G<br>S<br>U                     |
| Fish 91   | ADL                                                                                      | »Adventure Definition Language«, erlaubt die Programmierung von Adventures                                                                                                                                        | P                                    |
| Fish 92   | As6502<br>Bawk                                                                           | 6502-Assembler<br>Textprozessor à la Unix<br>»awk«                                                                                                                                                                | P<br>U                               |
|           | HunkPad<br>Less<br>Ndir                                                                  | Update zu Fish 84, Vers. 2<br>Update zu Fish 74, Vers. 1.2<br>Implementation von Unix<br>Directory-Zugriff                                                                                                        | T<br>U<br>T                          |
|           | Parse<br>Shar<br>SmallLib<br>UUencode                                                    | rekursiver Parser<br>Packer und Entpacker<br>ersetzt »amiga.lib«<br>Update zu Fish 53                                                                                                                             | T<br>U<br>T<br>T                     |
| Fish 93   | Dme<br>MicroEmacs                                                                        | Update zu Fish 87 (V1.27)<br>Update zu Fish 61 (V3.8i)                                                                                                                                                            | U<br>U                               |
| Fish 94   | AudioTools<br>ClickUpFront                                                               | Update zu Fish 84, Vers. 2<br>bringt Windows durch<br>Doppelklick mit der Maus<br>in den Vordergrund (V1.0)                                                                                                       | M<br>U                               |
|           | HeliosMouse<br>Iff2Ps                                                                    | simuliert »SunMouse« (V1.0)<br>konvertiert IFF-File in<br>Postscript-Format (V1.2)                                                                                                                                | T<br>U                               |
|           | ModulaTools                                                                              | verschiedene Routinen<br>für Modula                                                                                                                                                                               | P                                    |
|           | Terrain3D                                                                                | erzeugt 3D-Landschaften                                                                                                                                                                                           | G                                    |
| Fish 95   | Cmd                                                                                      | lenkt die Daten, die an den<br>seriellen oder parallelen<br>Port geschickt werden, auf<br>ein File um (Version 4)                                                                                                 | T                                    |
|           | CygnusEdDemo<br>Gomf<br>Journal                                                          | kommerzieller Editor, Demo<br>verhindert »Guru« (V1.0)<br>zeichnet Mausbewegungen<br>in einem File auf, ideal für<br>Demos                                                                                        | U<br>U<br>U                          |
|           | MergeMem<br>PrinterStealer                                                               | Update zu Fish 56 (V2.0)<br>leitet Druckausgabe in<br>Datei um                                                                                                                                                    | T<br>T                               |
|           | Record-Play                                                                              | ähnlich »Journal« (s.o.)                                                                                                                                                                                          | U                                    |
| Fish 96   | AnimPlayer                                                                               | spielt »anim«-Files (von<br>Videospace, Sculpt3D,<br>Silver etc.)                                                                                                                                                 | G                                    |
|           | Chess<br>HackBench<br>Label<br>LineDrawer<br>PopUpMenu<br>Tek4695<br>TimeRam<br>WarpText | Schachprogramm, Vers. 1.0<br>ähnlich Workbench<br>druckt Etiketten, Version 1.3<br>malt Linien nach einer Datei<br>generiert »PopUp«-Menüs<br>Druckertreiber<br>mißt Speicherzugriffszeit<br>schnelle Textausgabe | S<br>U<br>U<br>G<br>T<br>T<br>U<br>T |
| Fish 97   | CutAndPaste                                                                              | Implementation von Unix<br>»cut/paste«                                                                                                                                                                            | T                                    |
|           | GraphIt<br>Juggler<br>MouseReader<br>Shm<br>Splines                                      | Funktionsplotter<br>Update zu Fish 47 (V1.2)<br>zeigt Text- und IFF-Files<br>Grafik-Demo<br>Grafik-Demo                                                                                                           | G<br>G<br>U<br>G<br>G                |
| Fish 98   | Access<br>BackUp<br>DCDemo<br>HdDriver<br>QBase<br>Thai                                  | Terminal-Emulation (V1.34)<br>Kopierprogramm<br>Diskettenverwaltung<br>Festplatten-Treiber<br>spezielle Dateiverwaltung<br>Quiz in Thai                                                                           | A<br>U<br>A<br>U<br>A<br>A           |

| Diskette: | Programm:                                                                                                         | Bemerkung:                                                                                                                                                                           | Art:                                      |
|-----------|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Fish 99   | A-Render                                                                                                          | Ray-Tracing mit Demos<br>(V0.3)                                                                                                                                                      | G                                         |
| Fish 100  | Berserk<br>Conman                                                                                                 | Animation, sehr sehenswert<br>verbessert Editierung etc.<br>unter CON: (V1.0)                                                                                                        | G<br>U                                    |
|           | WBLander                                                                                                          | Grafik-Demo                                                                                                                                                                          | G                                         |
| Fish 101  | CirPlane                                                                                                          | Flugzeug-Generator für<br>Videospace, (V1.0)                                                                                                                                         | G                                         |
|           | IconAssebler<br>MicroSPELL<br>Midi<br>Pslntpr                                                                     | Verbinden von Icons<br>Spell-Checker<br>Midi-Paket<br>Postscript-Interpreter mit<br>Bildschirmausgabe                                                                                | T<br>U<br>M<br>U                          |
|           | StartUps                                                                                                          | ersetzt C-Start-Routinen                                                                                                                                                             | T                                         |
| Fish 102  | Dbug<br>Match_stuff<br>Sectorama<br>SiliCon<br>Xicon                                                              | C-Debugger<br>Tool<br>repariert Disketten<br>Zeilen-Interpreter<br>Update zu Fish 31 (V2.00)                                                                                         | U<br>T<br>U<br>U<br>T                     |
| Fish 103  | AvlTrees                                                                                                          | Implementation von<br>AVL-Bäumen                                                                                                                                                     | T                                         |
|           | Calc<br>Cref<br>DosKwik<br>IntuiDOS<br>MFF-Update<br>Pack-It<br>Sol                                               | programmierbarer Rechner<br>C-Cross-Referenz<br>Schnell-Lader<br>CLI-Benutzeroberfläche<br>Utility für »MicroFicheFiler«<br>legt Directory in File<br>Brettspiel                     | U<br>U<br>U<br>U<br>U<br>U<br>S           |
| Fish 104  | AnalyticCalc                                                                                                      | Tabellenkalkulation                                                                                                                                                                  | A                                         |
| Fish 105  | AsmProgs<br>BasicProgs<br>Bison<br>DrunkenMouse<br>FlamKey<br>GravityWars<br>IPo2C<br>Pere-et-Fils<br>Record-Play | Assembler-Tools<br>verschiedene Basic-Progr.<br>Update zu Fish 51<br>Grafik-Demo<br>Paßwort-Abfrage<br>Update zu Fish 84 (V2.0)<br>C-Tool<br>Prozeß-Tool<br>Update zu Fish 95 (V2.0) | T<br>T<br>T<br>G<br>U<br>S<br>T<br>T<br>U |
| Fish 106  | FuncKey<br>MoreArt<br>QuickFlix                                                                                   | Update zu Fish 89 (V1.1)<br>verschiedene Bilder<br>Diashow-Programm mit<br>Animationsfunktion (V0.13)                                                                                | U<br>G<br>G                               |
|           | RistiNolla                                                                                                        | Spiel                                                                                                                                                                                | S                                         |
| Fish 107  | Csh<br>Diff<br>ProSuite                                                                                           | Shell, Version 2.07<br>Vergleich von Files<br>Informationen für<br>Programmierer                                                                                                     | U<br>T<br>T                               |
|           | SYTools                                                                                                           | verschiedene Tools                                                                                                                                                                   | T                                         |
| Fish 108  | AList<br>DirMaster                                                                                                | verbessertes »list«<br>Update zu Fish 89,<br>Version 1.0b                                                                                                                            | T<br>U                                    |
|           | Dots-Perfect                                                                                                      | Druckertreiber für<br>Epson MX80                                                                                                                                                     | T                                         |
|           | MonIDCMP                                                                                                          | zeigt Intuition-Meldungen<br>an                                                                                                                                                      | T                                         |
|           | PrintPop                                                                                                          | Senden von Einstellungen<br>an PRT:                                                                                                                                                  | U                                         |
|           | Sectorama<br>Tek<br>Zoo                                                                                           | Update zu Fish 101 (V1.1)<br>Terminal-Emulation<br>Update zu Fish 87 (V1.42B)                                                                                                        | U<br>A<br>U                               |
| Fish 109  | Machine<br>SimCPM<br>UUpc                                                                                         | Animation<br>CP/M-Simulator<br>Netzwerk-Tool                                                                                                                                         | G<br>U<br>T                               |
| Fish 110  | A68k<br>Pdc                                                                                                       | Assembler, in Modula-2<br>geschrieben<br>C-Compiler, Update zu<br>Fish 53                                                                                                            | P<br>P                                    |
| Fish 111  | AmyLoad<br>AssignDev<br>Gauge<br>HeliosMouse                                                                      | zeigt Auslastung des<br>Systems<br>Update zu Fish 79<br>zeigt Speicherbelegung<br>grafisch<br>Update zu Fish 94 (V1.1)                                                               | U<br>T<br>U<br>T                          |



| Diskette: | Programm:   | Bemerkung:                                                         | Art: |
|-----------|-------------|--------------------------------------------------------------------|------|
|           | Labels      | Cross-Referenz von System-Konstanten                               | T    |
|           | Mandel      | Mandelbrot-Generator                                               | G    |
|           | PopLife     | Lebens-Simulation                                                  | A    |
| Fish 112  | BeachBirds  | Grafik-Demo                                                        | G    |
|           | Bully       | Grafik-Demo                                                        | G    |
|           | DropShadow  | setzt Schatten hinter Windows                                      | G    |
|           | Focus       | Grafik-Demo                                                        | G    |
|           | RGB         | Grafik-Demo, Wettbewerbssieger                                     | G    |
|           | Viacom      | Grafik-Demo                                                        | G    |
|           | WaveBench   | Grafik-Demo                                                        | G    |
| Fish 113  | AmiCron     | Implementation von Unix »cron«                                     | T    |
|           | Dme         | Update zu Fish 93 (V1.28f)                                         | U    |
|           | DosDev      | Modula-Compiler, Demo-Version von »M2Amiga«                        | P    |
|           | NolconPos   | löscht Icon-Positionen                                             | T    |
| Fish 114  | CDec1       | hilft beim Erlernen von C-Deklarationen                            | T    |
|           | VT100       | Update zu Fish 55 (V2.7)                                           | A    |
|           | WBLander    | Grafik-Demo                                                        | G    |
| Fish 115  | Killer      | Grafik-Demo                                                        | G    |
|           | Marketroid  | Grafik-Demo                                                        | G    |
| Fish 116  | Movies      | zum Erstellen von Animations-Filmen, mit drei Demos                | G    |
| Fish 117  | AMUC_Demo   | Grafik-Demo, 2400 x 200 Punkte großes IFF-Bild                     | G    |
|           | ExeP_Demo   | Demo-Version Express Paint 1.1                                     | G    |
| Fish 118  | Empire      | komplexes Spiel                                                    | S    |
|           | HAMmmm      | Grafik-Demo, HAM                                                   | G    |
|           | Stars       | Grafik-Demo                                                        | G    |
|           | WireDemo    | Grafik-Demo                                                        | G    |
| Fish 119  | MicreEMACS  | Update zu Fish 93 (V3.9e)                                          | U    |
| Fish 120  | Amoeba      | Space-Invaders                                                     | S    |
|           | BackGammon  | Brettspiel                                                         | G    |
|           | Bankn       | Haushaltsbuch, Version 1.3                                         | A    |
|           | EgyptianRun | Spiel                                                              | S    |
|           | IconImage   | ersetzt Icons                                                      | U    |
| Fish 121  | BasicStrip  | hilft beim Konvertieren von anderen Basic-Dialekten in Amiga-Basic | U    |
|           | DataPlot    | Funktionsplotter                                                   | G    |
|           | Plot        | 3D-Plotter                                                         | G    |
|           | Stairs      | Musik-Demo, erzeugt Ton-Illusion                                   | M    |
|           | Uedit       | Update zu Fish 60, Version 2.3                                     | U    |
|           | WBColors    | ändert Workbench-Farben                                            | U    |
| Fish 122  | Asteroids   | Ballerspiel                                                        | S    |
|           | IffPcs      | Puzzle-Spiel, arbeitet mit beliebigen IFF-Bildern                  | S    |
|           | Names       | erstellt Versandlisten                                             | A    |
|           | Pr          | druckt Listings                                                    | U    |
|           | PushOver    | Strategie-Brettspiel                                               | S    |
|           | PuzzlePro   | Puzzle-Spiel                                                       | S    |
| Fish 123  | Arp         | ersetzt CLI-Befehle durch schnellere Assembler-Äquivalente         | T    |
|           | Car         | Grafik-Demo                                                        | G    |
| Fish 124  | Icons       | animierte Icons                                                    | T    |
|           | Tarot       | Spiel                                                              | S    |
| Fish 125  | EIGato      | Grafik-Demo mit Musik                                              | G    |
| Fish 126  | Colour      | dient zur Farbmanipulation                                         | T    |
|           | Dance       | Grafik-Demo                                                        | G    |
|           | HBHill      | Grafik-Demo, Animation                                             | G    |
|           | Iconify     | erstellt Icon                                                      | T    |

| Diskette: | Programm:   | Bemerkung:                                                                 | Art: |
|-----------|-------------|----------------------------------------------------------------------------|------|
|           | OnlyAmiga   | Grafik-Demo, Animation                                                     | G    |
|           | Suplib      | Library mit verschiedenen Funktionen                                       | T    |
|           | VCHeck      | Virus-Checker, Version 1.2                                                 | U    |
| Fish 127  | Bounce      | Grafik-Demo                                                                | G    |
|           | Nemesis     | Grafik-Demo                                                                | G    |
|           | Ripples     | Grafik-Demo, Animation                                                     | G    |
| Fish 128  | Dis         | Disassembler                                                               | U    |
|           | DropCloth   | ändert Workbench-Hintergrund, Version 2.2                                  | T    |
|           | LedClock    | Uhr                                                                        | U    |
|           | MRBackUp    | Festplatten-Backup, Files einfaches Malprogramm in »web«                   | U    |
|           | Paint       | Druckertreiber                                                             | G    |
|           | PrtDriver   | Druckertreiber                                                             | T    |
|           | SDBackUp    | Festplatten-Backup (V1.1)                                                  | U    |
|           | Sed         | Implementation von Unix-Editor                                             | U    |
|           | wKeys       | »Hotkey«-Tool                                                              | T    |
| Fish 129  | DosKwik     | Update zu Fish 103, Version 2.0                                            | U    |
|           | MRBackUp    | Festplatten-Backup, File für File, Version 2.0 und 2.1                     | U    |
|           | PaintJet    | Druckertreiber für HP-Paintjet                                             | T    |
|           | Patch       | Implementation von Unix »patch«                                            | T    |
| Fish 130  | DirMaster   | Update zu Fish 108, Version 1.1                                            | U    |
|           | Evo         | Lernprogramm zur Evolution                                                 | A    |
|           | HP          | Rechner, Version 1.0                                                       | U    |
|           | Mach        | Mausunterstützung, Hotkeys                                                 | T    |
|           | PatEdit     | zum Erstellen von Patterns                                                 | U    |
|           | QMan        | Mandelbrot-Generator                                                       | G    |
| Fish 131  | Dfc         | Kopierprogramm                                                             | U    |
|           | HyperBase   | Datenbank, Update zu Fish 58, Version 1.6                                  | A    |
|           | Life        | Lebenssimulation                                                           | A    |
|           | Mackie      | Monitor-Schoner mit Grafik                                                 | U    |
|           | Mg1b        | Editor                                                                     | U    |
|           | WFrags      | zeigt Speicherbelegung                                                     | U    |
| Fish 132  | Berserk     | Animation, sehenswert                                                      | G    |
| Fish 133  | Conman      | Update zu Fish 100, Version 1.1                                            | U    |
|           | Crc         | bildet Checksumme von Disketten                                            | U    |
|           | CrcLists    | Checksummenliste                                                           | T    |
|           | Overscan    | Fish-Disks 1-128 zur Aktivierung des Overscan-Modus                        | T    |
| Fish 134  | BoingThrows | HAM-Animation                                                              | G    |
|           | Browser     | »Workbench für Programmierer«, Vers. 1.2                                   | U    |
|           | Dme         | Update zu Fish 113 (V1.29)                                                 | U    |
|           | Find        | Suchroutine, Version 1.0                                                   | T    |
|           | Library     | speichert Textinformationen, komplexe Suchfunktionen                       | A    |
|           | SmartIcon   | Icon-Tool                                                                  | T    |
| Fish 135  | TeXF        | enthält 78 Fonts von TeX, mit Programm zur Konvertierung in AmigaDOS-Fonts | T    |
| Fish 136  | AsmToolBox  | Interface Assembler-AmigaDOS                                               | T    |
|           | Bison       | Implementation von Unix » yacc «                                           | T    |
|           | Iff2Pcs     | Update zu Fish 122, Version 1.1                                            | G    |
|           | Paste       | Implementation von Unix »paste«                                            | T    |
|           | YaBoingll   | Update zu Fish 36                                                          | G    |



| Diskette: | Programm:     | Bemerkung:                                                                 | Art: |
|-----------|---------------|----------------------------------------------------------------------------|------|
|           | Zoo           | Update zu Fish 108 (V1.71)                                                 | U    |
| Fish 137  | CT            | zum Anzeigen von CT-Scanner-Bildern                                        | U    |
|           | JeansIcons    | Icons                                                                      | G    |
|           | Muncho        | spielt Sound beim Entnehmen der Diskette                                   | M    |
|           | Sit           | Update zu Fish 107 (Tools), Version 1.10                                   | T    |
|           | VGad          | Gadget-Editor, Ausgabe von C-Source                                        | U    |
|           | VirusX        | Virus-Checker                                                              | U    |
|           | Vlabel        | druckt Disk-Etiketten mit IFF-Bild                                         | U    |
| Fish 138  | AmigaLine     | technische Informationen für Programmierer                                 | T    |
|           | Diff          | Implementation von Unix »diff«                                             | T    |
|           | ForEach       | File-Tool                                                                  | T    |
|           | MacFont       | konvertiert Apple Macintosh-Fonts in das Amiga-Format                      | U    |
|           | ModulaTools   | Update zu Fish 94                                                          | P    |
|           | VT100         | Terminal-Emulation (V2.6//2.8)                                             | A    |
| Fish 139  | AmiCron       | Update zu Fish 113, Version 2.3                                            | T    |
|           | ListScanner   | zeigt Exec-Listen an                                                       | U    |
|           | ProCalc       | simuliert HP-11C, Rechner                                                  | U    |
|           | RemLib        | Library-Tool                                                               | T    |
|           | TurboBackup   | Kopierprogramm (V1.00)                                                     | U    |
|           | Warranger     | setzt Windows in den Vordergrund                                           | T    |
|           | WheelChairSim | Rollstuhl-Simulator, Projekt für behinderte Kinder                         | A    |
| Fish 140  | SBProlog      | Prolog-Compiler Vol. 1 (V2.3.2)                                            | P    |
| Fish 141  | SBProlog      | Prolog-Compiler Vol. 2 (V2.3.2)                                            | P    |
| Fish 142  | Diff          | wie Fish 138, mit Source-Code                                              | T    |
|           | FracGen       | Fraktal-Generator                                                          | G    |
|           | SciSubr       | Fortran-Module                                                             | P    |
| Fish 143  | Rim           | anspruchsvolle relationale Datenbank                                       | A    |
| Fish 144  | AnalytiCalc   | Tabellenkalkulation, Version 22-3D                                         | A    |
| Fish 145  | Csh           | Shell, benötigt ARP1.1 (Fish 123)                                          | U    |
|           | DMouse        | zugleich Screen-Blanker, Mouse-Blanker, automatischer Window-Selektor etc. | U    |
|           | DNet          | erlaubt Prozeßaustausch Amiga-Unix                                         | T    |
|           | Tab           | Tabulatur-Programm für Banjo und Gitarre                                   | M    |
|           | TinyProlog    | einfacher Prolog-Interpreter, Version 1.1                                  | P    |
| Fish 146  | Blanker2      | Screen-Blanker (V1.27.88)                                                  | U    |
|           | C-Light       | Demo eines Ray-Tracing-Programms                                           | G    |
|           | CrcLists      | Checksummenliste von Fish 129-145                                          | T    |
|           | DmeMacros     | erweitert Dme (Fish 113)                                                   | T    |
|           | MemoPad       | Notizblock                                                                 | A    |
| Fish 147  | MicroGNUEmacs | Editor, Version 2b                                                         | U    |
| Fish 148  | EFJ           | »Escape from Jovi«                                                         | S    |
|           | Fme           | Editor für »Fire-Power« (Spiel)                                            | U    |
|           | Handylcons    | erweitert Workbench-Menü                                                   | T    |
|           | Scrambler     | Codier-Routine                                                             | T    |
| Fish 149  | AnimalSounds  | digitalisierte Tierstimmen mit Player                                      | M    |

| Diskette: | Programm:                      | Bemerkung:                                                                            | Art: |
|-----------|--------------------------------|---------------------------------------------------------------------------------------|------|
|           | DX-VoiceSorter                 | arbeit zusammen mit »DXSynth«                                                         | M    |
|           | KeepLess                       | Netzwerk-Utility                                                                      | U    |
|           | Update zu Fish 93, Version 1.3 |                                                                                       | U    |
|           | Schema                         | Lisp-Dialekt                                                                          | P    |
| Fish 150  | AirFoil                        | Update zu Fish 71                                                                     | A    |
|           | DC10                           | Flugsimulator                                                                         | S    |
|           | ExecLib                        | Beispiel für Library                                                                  | T    |
| Fish 150  | AirFoil                        | Update zu Fish 71                                                                     | A    |
|           | DC10                           | Flugsimulator                                                                         | S    |
|           | ExecLib                        | Beispiel für Library                                                                  | T    |
|           | Iconizer                       | Mauszeiger-Utility                                                                    | U    |
|           | Pilot                          | Pilot-Implementation                                                                  | P    |
|           | StealMemBoot                   | baut in Bootbock »NoFastMem« ein                                                      | U    |
| Fish 151  | GlobeDemo                      | Grafik-Demo                                                                           | G    |
|           | Icons                          | verschiedene Icons                                                                    | T    |
|           | Pcopy                          | Kopierprogramm                                                                        | U    |
|           | SCT                            | ermöglicht Farbveränderung                                                            | U    |
|           | SlideShow                      | Diashow-Programm                                                                      | G    |
|           | Surveyor                       | gibt Pointer-Information aus                                                          | U    |
| Fish 152  | Blk                            | Requester                                                                             | T    |
|           | RunBack                        | Variante von Fish 73                                                                  | T    |
|           | UUCP                           | Implementation von Unix »uucp«                                                        | U    |
| Fish 153  | Dme                            | Update zu Fish 134 (V1.3)                                                             | U    |
|           | HP11                           | emuliert HP-11C, Rechner                                                              | U    |
|           | HPMan                          | erlaubt Einstellungen bei Drucker HP Laserjet+                                        | U    |
|           | Synthmania                     | Musik-Demo                                                                            | M    |
| Fish 154  | Ada                            | Syntax-Prüfer                                                                         | T    |
|           | AssemblyDemos                  | Demos in Assembler                                                                    | G    |
|           | DiskLib                        | Disketten-Utilities                                                                   | U    |
|           | Guardian                       | Virus-Checker                                                                         | U    |
|           | PrintSpool                     | Drucker-Spooler (V1.0.0)                                                              | U    |
|           | Utilities                      | verschiedene Utilities                                                                | U    |
|           | VirusX                         | Update zu Fish 137 (V1.21)                                                            | U    |
|           | Virus_Alert                    | Virus-Checker (V1.01/2.01)                                                            | U    |
|           | Wicon                          | verwandelt Windows in Icons                                                           | U    |
| Fish 155  | AsmExamples                    | verschiedene Assembler-Programme                                                      | U    |
|           | Bison                          | Update zu Fish 136                                                                    | T    |
|           | NoSmoking                      | zeigt Umgang mit »Recoverable-Alerts«                                                 | T    |
|           | Scenery                        | generiert Landschaft                                                                  | G    |
| Fish 156  | Blocks2                        | Grafik-Demo                                                                           | G    |
|           | Flex                           | Implementation von Unix »Lexical Analyzer Generator«                                  | U    |
|           | Go64                           | Grafik-Demo                                                                           | G    |
|           | Grammars                       | Files für Ada, C und Pascal zu »Flex«                                                 | T    |
|           | OOPS!                          | Grafik-Demo                                                                           | G    |
| Fish 157  | 60or80                         | Umschaltung 60-/80-Zeichenmodus                                                       | U    |
|           | AmicForm                       | Telefonverzeichnis                                                                    | A    |
|           | AnimBalls                      | Animations-Programm                                                                   | G    |
|           | BootBack                       | kopiert Bootblocks                                                                    | U    |
|           | ECPM                           | Update zu Fish 109                                                                    | U    |
|           | KeyFiler                       | sortiert Files                                                                        | U    |
|           | ScreenZap                      | Screen-Killer                                                                         | U    |
|           | SetPrefs                       | erlaubt Anlegen von mehreren Preference-Einstellungen und Umschaltung zwischen diesen | U    |
|           | Xicon                          | Update zu Fish 102 (V2.01)                                                            | T    |
| Fish 158  | DiskX                          | Disk-Editor                                                                           | U    |
|           | MemBoardTest                   | testet Amiga 1000-Speichererweiterungen                                               | U    |
|           | MSDOS                          | liest File im MS-DOS oder ST-Format                                                   | U    |



| Diskette: | Programm:                                                                         | Bemerkung:                                                                                                                    | Art:                            |
|-----------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
|           | PCBTool<br>ScreenX                                                                | PC Board Layout-Programm<br>Zeit-, Speicheranzeige und<br>Screen-Manipulator                                                  | U<br>U                          |
|           | TaskX<br>VirusX<br>YachtC3                                                        | Echtzeit-Task-Editor<br>Update zu Fish 154 (V1.6)<br>Update zu Fish 10 (V3)                                                   | U<br>U<br>S                     |
| Fish 159  | Free                                                                              | gibt Task- und Speicher-<br>Status aus                                                                                        | T                               |
|           | MidiTools<br>StarChart                                                            | verschiedene Tools für Midi<br>Astronomieprogramm,<br>kennt 600 Sterne                                                        | M<br>A                          |
|           | TaskControl<br>TUC                                                                | Task-Manipulator<br>»The Ultimate Clock«, Uhr                                                                                 | U<br>U                          |
| Fish 160  | Calls<br>Check                                                                    | C-Tool<br>findet Strukturfehler<br>in C-Source                                                                                | T<br>T                          |
|           | Dis                                                                               | Update zu Fish 128                                                                                                            | U                               |
|           | DMouse                                                                            | Update zu Fish 145 (V1.09)                                                                                                    | U                               |
|           | DWIP                                                                              | Hardcopy für Daisy<br>Wheel-Drucker                                                                                           | U                               |
|           | M4                                                                                | Makroprozessor                                                                                                                | T                               |
|           | MemoPad                                                                           | Update zu Fish 146 (V1.2)                                                                                                     | U                               |
|           | NeuralNets                                                                        | Netzwerk-Beispiel                                                                                                             | T                               |
| Fish 161  | Friends<br>Getsprite                                                              | Grafik-Demo<br>konvertiert DPaint-Brushes<br>in C-Source                                                                      | G<br>U                          |
|           | IncRev                                                                            | Compiler-Tool                                                                                                                 | T                               |
|           | LGZ                                                                               | Editor für LGZ (Spiel)                                                                                                        | U                               |
|           | Mackie                                                                            | Screen-Blanker mit Grafik                                                                                                     | U                               |
|           | Nag                                                                               | Kalender, Version 1,6                                                                                                         | U                               |
|           | Perl                                                                              | Perl-Interpreter                                                                                                              | P                               |
|           | VRTest                                                                            | Antivirus-Utility                                                                                                             | U                               |
|           | XBoot                                                                             | verwandelt Bootblock in<br>ausführbares File                                                                                  | T                               |
| Fish 162  | Avi<br>CLI_Uilities<br>Dark<br>Flow2Troff<br>Labyrinthll<br>Iffar<br>SetPALorNTSC | Editor wie Unix »vi«<br>verschiedene Utilities<br>Grafik-Demo, Animation<br>File-Konverter<br>Text-Adventure<br>IFF-Tool      | U<br>U<br>G<br>U<br>S<br>T<br>U |
|           | TES                                                                               | verschiedene Utilities, die<br>testen, ob Programme auf<br>PAL und/oder NTSC laufen<br>erweitert CLI mit<br>Funktions-Gadgets | U                               |
|           | UnknownGirl                                                                       | Musik-Demo                                                                                                                    | M                               |
| Fish 163  | Bankn<br>FiveInLine<br>Machll                                                     | Update zu Fish 120 (V1.5)<br>Brettspiel<br>Update zu Fish 130 (V2.4c)<br>stark erweitert                                      | A<br>S<br>U                     |
|           | MemTrace<br>PcPatch                                                               | Speicher-Debugger<br>Patches für »PCCopy« und<br>»PCFormat« auf der Amiga<br>2000 »Extras 1.2«-Disk                           | T<br>T                          |
|           | ReadmeMaster                                                                      | verwaltet Programme der<br>Fish-Disk (bis 154<br>enthalten)                                                                   | A                               |
|           | View                                                                              | zeigt Text-Files an                                                                                                           | U                               |
| Fish 164  | C-Functions<br>DiskSalv                                                           | verschiedene C-Module<br>Update zu Fish 20 (V1.3)<br>stark erweitert                                                          | T<br>U                          |
|           | Hed                                                                               | verbessertes CLI-<br>Kommando »ed«                                                                                            | T                               |
|           | Newton<br>NewZAP<br>PcView                                                        | Newton-Algorithmus<br>Update zu Fish 58 (V3.18)<br>zeigt IFF-Bilder auf<br>EGA-PCs an                                         | T<br>U<br>U                     |
|           | PolyRoot<br>PrtDrivers                                                            | Newton-Algorithmus<br>Druckertreiber für<br>Digital LN03+<br>Laser-Drucker und<br>Mannesmann MT420d                           | T<br>T                          |
|           | Zoo                                                                               | Update zu Fish 136 (V2.00)                                                                                                    | U                               |
| Fish 165  | Conman<br>CPM                                                                     | Update zu Fish 133 (V1.3)<br>CP/M-Emulator                                                                                    | U<br>U                          |

| Diskette: | Programm:                                                                 | Bemerkung:                                                                                                                                              | Art:                            |
|-----------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
|           | Parsnag<br>PlotView                                                       | Druckhilfe für Epson JX-80<br>dient zum Betrachten von<br>Unix Plot-Files                                                                               | U<br>U                          |
|           | RamCopy                                                                   | Diskettenkopie in einem<br>Durchgang bei Einzel-<br>laufwerk                                                                                            | U1                              |
|           | SPUDclock                                                                 | akustische Zeitangabe                                                                                                                                   | U                               |
| Fish 166  | AutoGraf<br>Cref<br>MultiCalc<br>Stevie                                   | Autokostenberechnung<br>C-Cross-Referenz<br>Rechner<br>Implementation von Unix<br>»vi« (Editor)                                                         | A<br>T<br>U<br>U                |
| Fish 167  | CDecl<br>CLIcon                                                           | Update zu Fish 114<br>startet CLI-Programme von<br>der Workbench                                                                                        | T<br>T                          |
|           | CloseMe<br>DSM                                                            | Grafik-Demo<br>»Dynamic Sound<br>Machine«-Demo                                                                                                          | G<br>M                          |
|           | MRPrint<br>Smus3.6a<br>Sounddemos                                         | druckt Text-Files<br>spielt Musik-Files<br>verschiedene Sound-<br>Beispiele                                                                             | U<br>M<br>M                     |
| Fish 168  | MattDillon1                                                               | Volume 1, verschiedene<br>Programme von Matt Dillon                                                                                                     | U                               |
| Fish 169  | MattDillon2                                                               | Volume 2                                                                                                                                                | U                               |
| Fish 170  | AftTerm<br>Dis6502                                                        | Terminal-Emulation<br>6502-Disassembler,<br>unterstützt C64-Files                                                                                       | A<br>U                          |
|           | FastText<br>MRBackUp<br>PtrAnim                                           | Textausgabe mit Blitter<br>Update zu Fish 129 (V2.4)<br>Pointer-Animator<br>mit Beispielen                                                              | T<br>U<br>G                     |
|           | Surf                                                                      | generiert Gegenstände mit<br>realistisch wirkenden<br>Oberflächen                                                                                       | G                               |
|           | Turbo                                                                     | Ausschalten von Bitplanes,<br>Sprites Copper und<br>Audio-DMA per Mausclick                                                                             | U                               |
| Fish 171  | AZComm<br>Maze<br>Sozobon-C                                               | Terminal-Emulation<br>Labyrinth-Generator<br>C-Compiler/Linker,<br>Assembler, von Atari ST<br>zeigt System-Aktivitäten an                               | A<br>S<br>P                     |
|           | Xoper                                                                     |                                                                                                                                                         | U                               |
| Fish 172  | DataToObj                                                                 | konvertiert »Rohdaten«-<br>Files, (Sprites, Text etc.)<br>in Objekt-Code                                                                                | U                               |
|           | Handshake<br>MFix<br>PopInfo<br>ProCalc<br>Spiff                          | Terminal-Emulation<br>Patch für Maurauder II<br>Workbench-Utility<br>Update zu Fish 139 (V1.2)<br>File-Tool                                             | A<br>T<br>U<br>U<br>T           |
| RPD 1     | c (dir)                                                                   | enthält verschiedene Hilfs-<br>programme, die teilweise<br>nur unter Kickstart 1.1<br>laufen (gilt auch für andere<br>Programme auf dieser<br>Diskette) | T                               |
|           | Dazzle<br>DirUtil 1.31<br>DPSlide<br>Popcli.c                             | Grafik-Demo<br>CLI-Benutzeroberfläche<br>fehlerhaft, Kickstart 1.1?<br>CLI-Aufruf per Hotkey,<br>Screen-Blanker                                         | G<br>U<br>G<br>U                |
|           | Rubik<br>StarTerm 3.0<br>Wallpaper                                        | fehlerhaft, Kickstart 1.1?<br>fehlerhaft, Kickstart 1.1?<br>Grafik-Demo                                                                                 | S<br>A<br>G                     |
| RPD 2     | AniPlayer 1.01                                                            | spielt Aegis-Animator-<br>Animationen, mit Demos<br>fehlerhaft, Kickstart 1.1?                                                                          | G                               |
|           | Cosmo<br>CSquared<br>Klock<br>MandFxp<br>MenuEd 1.0<br>Oing<br>PSound 1.9 | Grafik-Demo<br>Uhr, Vorversion<br>Mandelbrot-Generator<br>fehlerhaft, Kickstart 1.1?<br>Grafik-Demo<br>Digitizer-Programm<br>mit Demo-Sounds            | S<br>G<br>U<br>G<br>T<br>G<br>M |



| Diskette: | Programm:                                                                                                                                                                                                                                                                        | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Art:                                                                                                                                             |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|           | Yaboing<br>Sproing<br>Zoing<br>Sizzlers<br>Star10<br>Triclops 1.6<br>Zoomlens                                                                                                                                                                                                    | Grafik-Demo<br>Grafik-Demo<br>Grafik-Demo<br>Grafik-Demo<br>Grafik-Demo<br>Spiel mit 3D-Grafik<br>BildschirmLupe                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | G<br>G<br>G<br>G<br>G<br>S<br>U                                                                                                                  |
| RPD 3     | Diashow                                                                                                                                                                                                                                                                          | fehlerhaft, Kickstart 1.1?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | G                                                                                                                                                |
| RPD 4     | Dpaintx_source<br><br>DVideo-Player<br><br>MacView 1.01<br><br>ST.vs.Amy<br>Star<br>Title<br><br>VaxRayTracing                                                                                                                                                                   | Source-Code für Diashow-Programm<br>Animations-Player mit Demos<br>zeigt Macintosh-Bilder, mit Demos<br>Grafik-Demo<br>Grafik-Demo<br>Vorversion eines Video-Titelgenerators<br>Ray-Tracer in C                                                                                                                                                                                                                                                                                                                                                                                                                                                      | G<br><br>G<br><br>G<br>G<br>G<br>G                                                                                                               |
| RPD 5     | AniPlayer 1.01<br><br>Dpaintx<br>Show                                                                                                                                                                                                                                            | spielt Aegis-Animationen, mit Demos<br>Diashow-Programm<br>Dia-Show (Defender of the Crown)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | G<br><br>G<br>G                                                                                                                                  |
| RPD 6     | APAINT                                                                                                                                                                                                                                                                           | Malprogramm in ABasiC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | G                                                                                                                                                |
| RPD 7     | Settime 1.0<br><br>MCat<br>Xicon<br><br>Ticon<br>VT100 2.2<br>XLisp 1.7<br>Ternacs<br><br>DiskMapper<br><br>A-Copier<br>AmCat<br>Amt1<br>Browser<br><br>Cle<br><br>DiskCat<br>HamEditor<br>Lace<br>NewSysDisk<br><br>Reboot<br>RSLClock 1.2<br>Runback<br>See<br><br>Setfont 2.0 | dient zum Stellen der Uhrzeit<br>Diskettenverwaltung<br>Start von DOS-Files von der Workbench<br>Anzeige eines Text-Files<br>Terminal-Emulation<br>Programmiersprache<br>Version von Micro-Emacs, Editor<br>zeigt Belegung einer Diskette<br>Kopierprogramm (V1.0)<br>Diskettenverwaltung<br>Terminal-Emulation<br>zeigt Disk-Verzeichnisse an, Vorversion erlaubt das Editieren von CLI-Zeilen<br>Diskettenverwaltung<br>Editor für HAM-Bilder<br>aktiviert Interlace-Modus<br>»Assign« für alle Dircetories auf df0:<br>Reset<br>Uhrzeit- Speicheranzeige verbessertes »run«<br>Diashow-Programm, auch HAM<br>Veränderung des Zeichensatzes im CLI | U<br><br>A<br>U<br><br>U<br>A<br>A<br>P<br>U<br><br>U<br><br>U<br>A<br>A<br>A<br>U<br><br>U<br>A<br>G<br>U<br>T<br><br>T<br>U<br>T<br>G<br><br>T |
| RPD 8     | 3DStars<br><br>NewFonts<br><br>FontEditor<br>Raw2Iibm<br>ColorBars<br>PopColours<br>echo<br>GadgetEditor<br><br>Reversi<br>3DBreakkout<br>Clue<br>MouseClock                                                                                                                     | Grafik-Demo für Rot-Blau-Brille<br>28 verschiedene Zeichensätze<br>Zeichensatz-Editor (V1.19)<br>IFF-Tool<br>Testbild-Generator erlaubt Farbveränderung verbessertes »echo« erzeugt Riesengadgets, fehlerhaft<br>Brettspiel<br>3D-Spiel für Rot-Blau-Brille<br>fehlerhaft, Kickstart 1.1?<br>verwandelt Mauszeiger in Uhr                                                                                                                                                                                                                                                                                                                            | G<br><br>T<br><br>U<br>G<br>G<br>U<br>T<br>U<br><br>S<br>S<br>S<br>U                                                                             |
| RPD 9     | AniPlayer 1.01<br><br>Pictures                                                                                                                                                                                                                                                   | spielt Aegis-Animationen, mit Demo<br>verschiedene IFF-Bilder                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | G<br><br>G                                                                                                                                       |

| Diskette: | Programm:                                                                                                                                                                                                   | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Art:                                                                                                        |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
|           | SlideShow                                                                                                                                                                                                   | Diashow-Programm mit Demos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | G                                                                                                           |
| RPD 10    | Canfield.bas<br><br>Klondike<br>Millebornes<br>Monopoly<br>Othello<br>Backgammon<br>Tvisions                                                                                                                | Kartenspiel, wie alle anderen Programme auf dieser Diskette ein Spiel in ABasiC<br>Kartenspiel<br>Kartenspiel<br>Brettspiel<br>Brettspiel<br>Brettspiel<br>Irrgartenspiel                                                                                                                                                                                                                                                                                                                                          | S<br><br>S<br>S<br>S<br>S<br>S                                                                              |
| RPD 11    | Draw<br>AmigUtil<br><br>Brush2Icon<br><br>Filezap<br>FixHunk<br><br>Kickbench<br><br>LibDir<br>SaveLMB<br><br>N.D.V.R.<br><br>EDLP 0.01<br><br>TCWindow<br>ToolWrapper<br><br>WBrun<br><br>Mondrian<br>Tree | Malprogramm, Version 1.08 erlaubt Saven des aktuellen Bildschirms<br>verwandelt DPaint-Brushes in IFF<br>Patch-Programm für Files adaptiert Programme für Zusatzspeicher<br>dient zur Erzeugung einer Diskette, die Kickstart und Workbench zugleich enthält<br>File-Tool speichert Bilder im IFF-Format<br>synchronisiert Bilder mit Ton<br>Utility für Musik- und Videoanwendungen<br>Digitalstoppuhr<br>Aufruf von CLI-Scripts per Icon<br>Start von Workbench-Programmen vom CLI<br>Grafik-Demo<br>Grafik-Demo | G<br>U<br><br>U<br><br>T<br>T<br><br>T<br><br>T<br>U<br><br>G<br><br>A<br><br>U<br>T<br><br>T<br><br>G<br>G |
| RPD 12    | Pictures                                                                                                                                                                                                    | digitalisierte Bilder                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | G                                                                                                           |
| RPD 13    | Fish<br>Calc<br><br>MacView 1.2<br><br>Asm68k 1.0.1<br>Blitz<br>Newzap 2.8<br>STEmulator<br><br>Tek4010                                                                                                     | Grafik-Demo, Animation<br>Emulation eines HP-Rechners<br>dient zur Anzeige von Macintosh-Bildern auf dem Amiga<br>Makro-Assembler<br>Anzeige von Text-Files<br>File-Sektor-Editor<br>Gag-Simulation eines Atari ST<br>Terminal-Emulation                                                                                                                                                                                                                                                                           | G<br>U<br><br>U<br><br>P<br>U<br>U<br>G<br><br>A                                                            |
| RPD 14    | InstantMusic<br><br>Marble                                                                                                                                                                                  | Demo eines Musikprogrammes<br>Demo »Marble-Madness« (Spiel)                                                                                                                                                                                                                                                                                                                                                                                                                                                        | M<br><br>S                                                                                                  |
| RPD 15    | FScapePlus1.0<br>Tracer<br>Missile<br>StarChart1.0<br>Stoneage 1.1                                                                                                                                          | Fraktal-Generator<br>Ray-Tracer mit Source-Code<br>Action-Spiel<br>elektronische Sternkarte<br>Geschicklichkeits-Spiel                                                                                                                                                                                                                                                                                                                                                                                             | G<br>G<br>S<br>A<br>S                                                                                       |
| RPD 16    | BardDemo                                                                                                                                                                                                    | Demo »The Bards Tale« (Spiel)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | S                                                                                                           |
| RPD 17a   | CAO3D1.0<br>Queen1                                                                                                                                                                                          | einfaches Malprogramm<br>Teil 1 eines digitalisierten Musikstückes                                                                                                                                                                                                                                                                                                                                                                                                                                                 | G<br>M                                                                                                      |
| RPD 17b   | Queen2                                                                                                                                                                                                      | Teil 2 (s. RPD 17a)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | M                                                                                                           |
| RPD 18    | Smalltalk                                                                                                                                                                                                   | Objekt-Programmiersprache                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | P                                                                                                           |
| RPD 19    | Citydesk 0.2<br><br>Aterm 7.3<br>Ing<br>AmigaMonitor<br><br>BCS_VFC<br>FileMap                                                                                                                              | Vorversion eines DTP-Programmes<br>Terminal-Emulation<br>Grafik-Demo<br>zeigt den System-Zustand an<br>Uhr<br>zeigt Diskettenbelegung an                                                                                                                                                                                                                                                                                                                                                                           | A<br><br>A<br>G<br>U<br><br>U<br>U                                                                          |











| Diskette: | Programm:        | Bemerkung:                                                           | Art: |
|-----------|------------------|----------------------------------------------------------------------|------|
|           | ACalc 1.1b       | verschiedene mathematische Funktionen                                | T    |
| RPD 50    | SlideShow        | Diashow                                                              | G    |
| RPD 51    | Journal          | zeichnet Mausbewegungen in einem File auf, ideal für Demonstrationen | U    |
|           | PlayBack         | spielt Journal-Dateien ab                                            | U    |
|           | AmicTerm 0.55    | Terminal-Emulation                                                   | A    |
|           | GraphIt II       | Funktionsplotter                                                     | G    |
|           | MCad 1.2.5       | Cad-Programm                                                         | G    |
|           | ShowPrint 0.3    | druckt IFF- und Pagesetter-Bilder                                    | U    |
| RPD 52    | SlideShow        | Diashow                                                              | G    |
| RPD 53    | HackBench        | neue Funktionen für die Menüleiste der Workbench                     | U    |
|           | Esuom            | Grafik-Demo                                                          | G    |
|           | Prefs/Setprefs   | Preference-Einstellungen können verändert werden, ohne neu zu booten | U    |
|           | Hyperbase        | Datenbank                                                            | A    |
|           | Egraph           | zeigt Daten grafisch an                                              | G    |
|           | Blur             | Grafik-Demo                                                          | G    |
|           | DirUtilDeLuxe    | CLI-Benutzeroberfläche                                               | U    |
|           | Help             | erklärt Befehle von Amiga-DOS, online                                | T    |
|           | PlaceWindow      | Veränderung der Parameter eines Windows                              | T    |
|           | Purty            | Änderung der Druckerparameter                                        | U    |
|           | Scat!            | Grafik-Demo                                                          | G    |
|           | Strobe           | Grafik-Demo                                                          | G    |
|           | WarpText         | beschleunigt Textausgabe                                             | T    |
|           | Wave!            | spielt Musikstücke, mit Demos                                        | M    |
| RPD 54    | Zeus.anim        | Videoscape-Animation mit Player                                      | G    |
| RPD 55    | SlideShow        | Diashow mit berechneten Bildern                                      | G    |
| RPD 56    | PlayBack         | spielt Dateien von Journal (RPD 51) ab                               | U    |
|           | Suck             | Grafik-Demo                                                          | G    |
|           | Access           | Terminal-Emulation                                                   | A    |
|           | Four             | Knobelspiel                                                          | S    |
|           | Kalah            | Knobelspiel                                                          | S    |
|           | TTT              | Strategiespiel »Tic-Tac-Toe«                                         | S    |
|           | ChessClock       | Schachuhr                                                            | A    |
|           | SimpAnim         | Grafik-Demo                                                          | A    |
|           | DLCS             | druckt Diskettenetiketten                                            | G    |
|           | Show 2.1         | Diashow-Programm                                                     | G    |
|           | MandelVroom      | Mandelbrot-Generator (V1.5)                                          | G    |
|           | NewZap           | Diskettenmonitor                                                     | U    |
|           | Sectorama        | Diskettenmonitor, auch für Hard-Disks                                | U    |
|           | FuelGauge        | zeigt Speicherbelegung an                                            | U    |
|           | Mach             | Mausunterstützung, Hotkeys                                           | U    |
|           | ScreenShift      | erlaubt Veränderung der Bildposition                                 | U    |
|           | CLIRun           | Start von Batch-Dateien per Icon                                     | T    |
|           | Icon+Ed          | Icon-Editor                                                          | U    |
|           | Labeler 1.0      | druckt Disketten-Etiketten                                           | U    |
| RPD 57    | Berserk          | Videoscape-Animation                                                 | G    |
|           | Lmv              | animiert mehrere IFF-Bilder                                          | G    |
| RPD 58    | 3D-Plot          | zeichnet 3D-Funktionen, Amiga-Basic                                  | G    |
|           | Almazar          | Text-Adventure in Amiga-Basic                                        | S    |
|           | Amortice         | Finanzprogramm in Amiga-Basic                                        | A    |
|           | Classics         | verschiedene Melodien                                                | M    |
|           | CrissCross       | Adventure, Amiga-Basic                                               | S    |
|           | Enigma           | Knobelspiel, Amiga-Basic                                             | S    |
|           | Fractal-Mountain | generiert fraktale Landschaften                                      | G    |

| Diskette: | Programm:         | Bemerkung:                                                            | Art: |
|-----------|-------------------|-----------------------------------------------------------------------|------|
|           | Grids             | Musikprogramm, Amiga-Basic (AB)                                       | M    |
|           | IFFtoPat          | wandelt IFF-Bilder in »pat«-Files                                     | T    |
|           | LabelPrint        | druckt Disketten-Etiketten                                            | U    |
|           | MailTalk          | Adressenverwaltung (AB)                                               | A    |
|           | MasterMind        | Knobelspiel (AB)                                                      | S    |
|           | MathGraphic       | Grafik-Demo (AB)                                                      | G    |
|           | Maze              | Labyrinth-Spiel (AB)                                                  | S    |
|           | Pyramid           | Geschicklichkeitsspiel (AB)                                           | S    |
|           | Solitaire         | Spiel (AB)                                                            | S    |
|           | Space             | Geschicklichkeitsspiel (AB)                                           | S    |
|           | Stars             | zeigt Sternenhimmel (AB)                                              | S    |
|           | StarTrek          | Weltraumspiel (AB)                                                    | S    |
|           | WallPaper         | Grafik-Demo (AB)                                                      | G    |
|           | WheelOfFortune    | Glücksrad (AB)                                                        | S    |
| RPD 59    | Balls             | Ray-Tracing-Animation, mit Silver generiert                           | G    |
|           | Merlin3D.anim     | Videoscape-Animation                                                  | G    |
|           | Spinner2          | Ray-Tracing-Animation                                                 | G    |
|           | ShowAnim 3.5      | spielt Videoscape-Animationen                                         | G    |
| RPD 60    | WBLander          | Grafik-Demo                                                           | G    |
|           | ShowAnim 4.0      | spielt Videoscape-Animationen                                         | G    |
|           | Scher4.anim       | Videoscape-Animation                                                  | G    |
|           | AniPlayer1.04     | spielt Animationen vom Aegis-Animator, mit Demos                      | G    |
| RPD 61    | SlideShow         | Diashow mit verschiedenen Bildern                                     | G    |
| RPD 62    | Scheme            | CAD-Programm, Demo-Version                                            | G    |
|           | ProWrite          | Textverarbeitung, Demo-Version                                        | A    |
|           | Microfiche-Filter | Dateiverwaltung, simuliert Microfiches, Demo-Version                  | A    |
| RPD 63    | Fonts             | über 20 verschiedene Zeichensätze                                     | T    |
|           | Ps                | Postscript-Emulator                                                   | U    |
|           | FontAssign        | weist neues Font-Verzeichnis zu                                       | T    |
|           | FontFixer         | Zeichensatz-Tool                                                      | T    |
| RPD 64    | AmiGazer          | bringt Sternenhimmel auf den Bildschirm, kennt über 1500 Sterne       | A    |
|           | J-Clock           | Uhr-, Datums- und Speicheranzeige                                     | U    |
|           | Amigaplot         | Funktionsplotter, 3D                                                  | G    |
|           | Geocycle          | Grafik-Demo                                                           | G    |
|           | Disk-Man          | CLI-Benutzeroberfläche                                                | U    |
|           | All4096           | Grafik-Demo                                                           | G    |
|           | Lightning-Logger  | Diskettenverwaltung, Demo                                             | A    |
|           | Audiomaster-      | tolles Musikprogramm, Demo                                            | M    |
|           | MSE               | Mandelbrot-Generator                                                  | G    |
| RPD 65    | Multiscope        | Mandelbrotgenerator mit Musik                                         | G    |
|           | Spindel-Animation | Videoscape-Animation                                                  | G    |
|           | Marble-Animation  | Videoscape-Animation                                                  | G    |
| RPD 66    | Filetype          | gibt Typen etc. von Files an                                          | T    |
|           | Sumkick           | berechnet neue Prüfsumme bei Veränderungen auf der Kickstart-Diskette | T    |
|           | TCB               | erklärt Amiga-DOS                                                     | T    |
|           | Overscan          | Grafik-Demo                                                           | G    |
|           | Ttasks            | Datenaustausch zwischen Tasks                                         | T    |
|           | MidiUtil          | verschiedene Utilities für Midi, unter anderem ein Midi-Monitor       | M    |
|           | Arp               | ersetzt CLI-Befehle durch                                             | T    |



| Diskette: | Programm:                                                                                                                                                                                   | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                  | Art:                                                        |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
|           | Finish<br>NoKlickStart                                                                                                                                                                      | schnellere Assembler-Äquivalente<br>stoppt Multitasking<br>verändert Kickstart 1.2                                                                                                                                                                                                                                                                                                          | T<br>T                                                      |
| RPD 67    | Car<br><br>Control<br><br>Wizard<br>Snipit<br><br>WordWright<br>SpretemasterII<br>DRW<br>Pointerz<br>Access25                                                                               | Rennspiel auf der Workbench<br>Manipulation der Druckerausgabe<br>erleichtert Dateioperationen<br>Textaustausch zwischen Windows<br>einfache Textverarbeitung<br>Sprite-Editor<br>Musikprogramm<br>Grafik-Demo<br>Terminal-Emulation, verbesserte Version                                                                                                                                   | G<br><br>T<br><br>T<br>T<br><br>A<br>G<br>M<br>G<br>A       |
| RPD 68    | SlideShow                                                                                                                                                                                   | Diashow                                                                                                                                                                                                                                                                                                                                                                                     | G                                                           |
| RPD 69    | Movie<br>Pllbm<br><br>Dllbm<br><br>Vllbm<br>Rocker<br><br>Kahnakas                                                                                                                          | Programm für Animationen packt IFF-Files zu einer Animation<br>errechnet Differenz von zwei IFF-Files<br>Anzeige von Grafiken<br>Animation eines Schaukelstuhls<br>Animation mit Kugeln, Ray-Tracing                                                                                                                                                                                        | G<br>G<br><br>G<br>G<br>G<br>G                              |
| RPD 70    | Sythia-Demo<br><br>UD<br>Tracker<br><br>Spiral<br>Squeek                                                                                                                                    | tolles Musikprogramm, Demo<br>Grafik-Demo zum Animieren von Vektorgrafiken<br>Grafik-Demo die Maus piepst                                                                                                                                                                                                                                                                                   | M<br><br>G<br>G<br><br>G<br>M                               |
| RPD 71    | Gypsy                                                                                                                                                                                       | Kartenspiel in Amiga-Basic                                                                                                                                                                                                                                                                                                                                                                  | S                                                           |
| RPD 72    | ArkDemo<br><br>HBHill.anim<br>StarRun.anim<br>ShowAnim 4.00                                                                                                                                 | Demo-Version von »Arkanoid«<br>Animation, Halfbrite-Modus<br>Videoscope-Animation spielt Videoscope-Animationen                                                                                                                                                                                                                                                                             | S<br><br>G<br>G<br>G                                        |
| RPD 73    | Arp-Commandos<br>CD0<br>FileIODemo<br>XTextDemo                                                                                                                                             | Ergänzung zu RPD 66 verbessertes »cd«<br>File-Abfrage<br>Programmierung von Bildschirmausgaben                                                                                                                                                                                                                                                                                              | T<br>T<br>T<br>T                                            |
| RPD 74    | DirMaster<br>SCA-Virus-Ch.<br>SpTil<br>KillSpl<br>WindowKeys<br><br>AssignDevice<br><br>CShell<br>Commodities<br>MonIDCMP<br><br>ListDevs<br>ListInts<br>ListLibs<br>ListPorts<br>ListTasks | Diskettenverwaltung<br>Virus-Checker<br>installiert Drucker-Spooler entfernt Druckerspooler<br>erleichtert Umgang mit Windows<br>weist einem Gerät mehrere Namen zu<br>verbessertes CLI<br>Library<br>Anzeigen von Intuition-Meldungen<br>zeigt die aktiven Devices an<br>zeigt alle Interrupts an<br>zeigt geöffnete Libraries an<br>listet alle belegten Ports auf<br>zeigt alle Tasks an | A<br>U<br>T<br>T<br>U<br><br>T<br><br>U<br>T<br>T<br>T<br>T |
| RPD 75    | Nemesis<br><br>TheCity.anim<br>ShowAnim 4.00                                                                                                                                                | Grafik-Demo, Flug durch das All<br>Videoscope-Animation spielt Videoscope-Animationen                                                                                                                                                                                                                                                                                                       | G<br><br>G<br>G                                             |
| RPD 76    | Marketroid<br>PlayRiff<br><br>Display                                                                                                                                                       | Grafik-Demo<br>Abspielroutine für R-IFF-Bilder, Demo<br>Grafik-Demo, Animation                                                                                                                                                                                                                                                                                                              | G<br>G<br><br>G                                             |
| RPD 77    | A68k                                                                                                                                                                                        | Makro-Assembler                                                                                                                                                                                                                                                                                                                                                                             | P                                                           |

| Diskette: | Programm:                                                                                                                                                                                  | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                             | Art:                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|           | Logo<br>Postscript                                                                                                                                                                         | Logo-Interpreter<br>Postscript-Emulator                                                                                                                                                                                                                                                                                                                                                                | P<br>U                                                                                |
| RPD 78    | SlideShow                                                                                                                                                                                  | Diashow mit Bildern aus der Fabelwelt                                                                                                                                                                                                                                                                                                                                                                  | G                                                                                     |
| RPD 79    | Amoebas<br><br>ArkDemo<br>Introduction<br>Gravity<br>2Balls<br>Moose<br><br>Dllbm<br><br>Pllbm<br><br>Movie 1.01<br>SpiroGraph                                                             | Spiel, Prinzip Space-Invaders<br>Demo-Version von Arkanoid<br>Grafik-Demo<br>Spiel<br>Animation, Ray-Tracing<br>modulierbare Sprachausgabe<br>errechnet Differenz zweier IFF-Bilder<br>packt IFF-Files zu einer Animation<br>spielt Animationen<br>Zeichenprogramm                                                                                                                                     | S<br><br>S<br>G<br>S<br>G<br>M<br><br>G<br><br>G<br>G                                 |
| RPD 80    | Pak 1.0<br>Access! jr.<br>AmicTerm<br>Nag<br>RSL-Clock 1.4<br>BrushIcon<br><br>Icon2Brush<br><br>Iconize<br><br>QuickFlix0.13<br>ReColor 1.01<br><br>UtilMaster<br><br>Ug<br><br>MFFImport | Packer<br>Terminal-Emulation<br>Terminal-Emulation<br>Kalender<br>Uhr mit Zusatzfunktionen<br>wandelt IFF-Brushes in Icons um<br>wandelt Icons in IFF-Brushes um<br>wandelt IFF-Bilder in Icons um<br>Diashow-Programm<br>Veränderung von Farbpaletten<br>CLI-Befehle per Maus ausführbar<br>belegbare Gadgets in der Menüleiste<br>wandelt ASCII-Dateien in für Microfiche-Filer verwertbare Daten um | U<br>A<br>A<br>U<br>U<br>U<br><br>U<br><br>U<br><br>G<br>G<br><br>U<br><br>T<br><br>T |
| RPD 81    | Boing<br>Killer                                                                                                                                                                            | Animations-Demo<br>Grafik-Demo                                                                                                                                                                                                                                                                                                                                                                         | G<br>G                                                                                |
| RPD 82    | SlideShow                                                                                                                                                                                  | Diashow, HAM-Bilder                                                                                                                                                                                                                                                                                                                                                                                    | G                                                                                     |
| RPD 83    | WaveBench<br>QDC<br>Walk 0.1<br>Morse<br>Keep 1.1<br>PopColours<br><br>XE<br>TWM<br><br>Handshake1.50a<br>PrefChange<br><br>MasterWeave                                                    | Grafik-Demo<br>Kopierprogramm<br>erweitert CLI-Befehle<br>Morse-Trainer<br>DFÜ-Tool<br>erlaubt jederzeit Veränderung der Workbench-Farben (V1.2)<br>einfacher Rechner<br>einfaches Starten von CLI-Programmen<br>Terminal-Emulation<br>Ändern der Preference-Einstellungen<br>Grafik-Demo                                                                                                              | G<br>U<br>T<br>A<br>T<br>U<br><br>U<br>T<br><br>A<br>U<br><br>G                       |
| RPD 84a   | Rot.anim<br><br>Movie 1.01                                                                                                                                                                 | Animation von Sculpt/Animate3D<br>spielt Animationen                                                                                                                                                                                                                                                                                                                                                   | G<br><br>G                                                                            |
| RPD 84b   | RotAmiga.anim                                                                                                                                                                              | Animation von Sculpt/Animate3D                                                                                                                                                                                                                                                                                                                                                                         | G                                                                                     |
| RPD 85    | BankN<br>Lit<br>Sectorama<br><br>UEdit                                                                                                                                                     | Verwaltung eines Kontos<br>Text-Tool<br>Diskettenmonitor, auch für Hard-Disks<br>Text-Editor                                                                                                                                                                                                                                                                                                           | A<br>T<br>U<br><br>U                                                                  |
| RPD 86a   | DemoReel1a                                                                                                                                                                                 | Animationsdemo                                                                                                                                                                                                                                                                                                                                                                                         | G                                                                                     |
| RPD 86b   | DemoReel1b                                                                                                                                                                                 | Animationsdemo                                                                                                                                                                                                                                                                                                                                                                                         | G                                                                                     |
| RPD 87    | ShowAnim 4.00<br><br>Machine.anim<br>FlyBy.anim<br>SonixPlayer1.0                                                                                                                          | spielt Videoscope-Animationen<br>Videoscope-Animation<br>Videoscope-Animation<br>spielt Sonix-Stücke                                                                                                                                                                                                                                                                                                   | G<br><br>G<br>G<br>M                                                                  |



| Diskette: | Programm:      | Bemerkung:                                                       | Art: |
|-----------|----------------|------------------------------------------------------------------|------|
|           | SonixScores    | Stücke und Instrumente für Sonix                                 | M    |
| RPD 88    | ShowAnim 4.00  | spielt Videoscape-Animationen                                    | G    |
|           | Movie.anim     | Videoscape-Animation                                             | G    |
|           | Faug.anim      | Videoscape-Animation                                             | G    |
|           | Movie 1.01     | spielt Animationen                                               | G    |
|           | Birds          | Animation von Sculpt/Animate 3D                                  | G    |
| RPD 89    | Icon 6.0       | Programmiersprache                                               | P    |
|           | XLisp 2.0      | Programmiersprache                                               | P    |
| RPD 90    | PMate 3.0      | Telefonverzeichnis                                               | A    |
|           | Keep 1.1       | DFÜ-Utility                                                      | U    |
|           | VCheck 1.2     | Virus-Checker                                                    | U    |
|           | VirusOff       | sucht Viren im Speicher                                          | U    |
|           | Quick-Base     | Dateiverwaltung                                                  | A    |
|           | YourMoney      | Finanzprogramm                                                   | A    |
|           | AmiCron        | Implementation von Unix »cron«                                   | T    |
|           | Mach 1.2       | Mausunterstützung, Hotkeys                                       | T    |
|           | EHBarGraph     | erstellt Diagramme                                               | G    |
|           | RWkWik         | beschleunigt Diskettenzugriff                                    | T    |
|           | DiskWipe       | löscht Disketten                                                 | U    |
|           | Dme 1.28f      | Text-Editor                                                      | U    |
|           | VisiCalc       | Tabellenkalkulation                                              | A    |
| RPD 91    | SlideShow      | Diashow mit Ray-Tracing-Bildern                                  | G    |
| RPD 92    | Dis            | Disassembler                                                     | U    |
|           | PKax           | Archivierungsprogramm für Dateien                                | U    |
|           | ASDG-rdd       | resettefeste RAM-Disk                                            | U    |
|           | LowMemory-     | Library für Speicherfreigabe                                     | T    |
|           | Brush          | Server verwandelt IFF-Brushes in Sprites                         | T    |
|           | ClickUpFront   | bringt Windows durch Doppelklick mit der Maus in den Vordergrund | U    |
|           | Palette        | Verändern der Bildschirmfarben                                   | U    |
|           | Getfile        | File-Requester                                                   | T    |
|           | SimCPM         | CP/M-Emulator                                                    | U    |
|           | Make           | Implementation von Unix »make«                                   | T    |
|           | Small          | Tool für Lattice C                                               | T    |
|           | JayCalc        | Blitter-Tool                                                     | T    |
| RPD 93    | Hilbert        | Grafik-Demo                                                      | G    |
|           | C-Light 1.0    | Ray-Tracing-Programm, Demo                                       | G    |
|           | Chaos          | generiert Apfelmännchen                                          | G    |
|           | MandelZoom     | generiert Apfelmännchen                                          | G    |
| RPD 94    | SuperBrickout  | Breakout-Variante                                                | S    |
|           | ZicZacZoe      | Knobelspiel                                                      | S    |
|           | GravAttack     | Schießspiel                                                      | S    |
|           | Daleks         | Schießspiel                                                      | S    |
|           | Interceptor-   | Flugsimulator, Demo-Vers.                                        | S    |
| RPD 95    | AniPlayer1.02  | spielt Aegis-Animationen, mit Demos                              | G    |
| RPD 96    | SlideShow      | Diashow                                                          | G    |
| RPD 97    | Vac            | Grafik-Demo                                                      | G    |
|           | Apong          | Ballspiel                                                        | S    |
|           | FSpots         | Grafik-Demo                                                      | G    |
|           | FOT            | wandelt Sulpt3D- und Videoscape-Objekte                          | T    |
|           | Leach 1.0      | Bearbeitung von digitalisierten Bildern                          | U    |
|           | Freelcon       | löscht Icon-Positionen                                           | T    |
| RPD 98    | Access 0.28    | Terminal-Emulation                                               | A    |
|           | CoComm 4.0     | Terminal-Emulation                                               | A    |
|           | Handshake1.60b | Terminal-Emulation                                               | A    |
|           | Wombat 3.1     | Terminal-Emulation                                               | A    |
|           | EMit           | erhöht Baud-Rate am                                              | T    |

| Diskette: | Programm:     | Bemerkung:                                                     | Art: |
|-----------|---------------|----------------------------------------------------------------|------|
|           |               | seriellen Port                                                 |      |
| RPD 99    | SlideShow     | Diashow mit Bildern von Tieren                                 | G    |
| RPD 100   | Introducer    | erstellt Laufschriften                                         | G    |
|           | 3DStars       | Grafik-Demo                                                    | G    |
|           | MoreColors0.9 | mehr Farben auf der Workbench                                  | T    |
|           | ChaosKit      | Mandelbrot-Generator                                           | G    |
|           | IffsDraw      | Grafik-Demo                                                    | G    |
|           | MSet 1.1      | Mandelbrot-Generator                                           | G    |
|           | Paletta       | erzeugt Farbpaletten                                           | U    |
|           | Planet 1.0    | Mandelbrot-Generator                                           | G    |
|           | Mot           | Midi-Tool                                                      | M    |
|           | DropCloth 2.2 | ändert Workbench-Hintergrund                                   | T    |
| RPD 101a  | Movie 1.01    | spielt Animationen                                             | G    |
|           | Hyper3.anim   | Videoscape-Animation                                           | G    |
| RPD 101b  | BThrows       | Animation von Sculpt/Animate3D                                 | G    |
|           | PMM           | Animation von Sculpt/Animate3D                                 | G    |
|           | Tellstar-One  | Videoscape-Animation                                           | G    |
| RPD 102   | Ferrari       | Demo-Version von »Ferrari-Formula-One«                         | S    |
|           | Asteroids     | Schießspiel                                                    | S    |
|           | Galactic Worm | Geschicklichkeitsspiel                                         | S    |
|           | SpaceAce-Demo | Schießspiel, Demo-Version                                      | S    |
|           | FPEdit 2.0    | Editor für »Fire Power« (Spiel)                                | U    |
| RPD 103   | Undelete      | rettet versehentlich gelöschte Dateien                         | T    |
|           | Dme 1.29      | Text-Editor                                                    | U    |
|           | FixText       | ersetzt Tabulatoren durch Leerzeichen                          | T    |
|           | IDev          | Device-Tool                                                    | T    |
|           | SetPri 2.1    | Ändern von Task-Prioritäten                                    | U    |
|           | TaskX 2.0     | Echtzeit-Task-Editor                                           | U    |
|           | WBDepth       | verändert Anzahl der Workbench-Bitplanes                       | T    |
|           | Browser 1.2   | »Workbench für Programmierer«                                  | U    |
|           | Mach 1.6      | Mausunterstützung, Hotkeys                                     | U    |
|           | AmiCron 2.4a  | Implementation von Unix »cron«                                 | T    |
|           | ScreenX 2.0   | Uhr, weitere Funktionen                                        | U    |
|           | Select        | freie Wahl der Batch-Datei                                     | T    |
|           | B             | gibt Eingaben an CON:                                          | T    |
|           | ColourReg     | Requester                                                      | T    |
|           | FileInjector  | File-Requester                                                 | T    |
|           | WW            | wandelt Windows in C-Source                                    | T    |
|           | PrettyPrint   | verändert C-Sourcecode                                         | T    |
| RPD 104   | Muncho        | spielt Sound beim Entnehmen der Diskette                       | M    |
|           | DBWiz         | Adreßverwaltung                                                | A    |
|           | Nag1.2        | Kalender                                                       | U    |
|           | LU            | spielt »smus«-Files von DMCS                                   | M    |
|           | Du-Hermes2.5  | vereinfacht Dateioperationen                                   | U    |
|           | DirMaster     | Diskettenverwaltung                                            | A    |
|           | VCheck 1.9    | Virus-Checker                                                  | U    |
|           | TextReader    | zeigt Text-Files an                                            | U    |
|           | WIconify      | bringt Workbench-Fenster auf Icon-Größe                        | T    |
|           | BlitzFonts    | beschleunigt Textausgabe                                       | T    |
|           | Help          | Rechner, ASCII-Tabelle                                         | U    |
|           | WP-PS         | konvertiert Files von »WordPerfect« in das »PageSetter«-Format | T    |
| RPD 105   | Superview     | Diashow-Programm                                               | G    |
|           | NoCycle       | entfernt »Cycles« aus IFF-Bildern                              | T    |
|           | Author        | Schrifteinbindung in Bilder                                    | T    |



| Diskette: | Programm:                                                                                                                                                                                                                    | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Art:                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|           | Pictures                                                                                                                                                                                                                     | verschiedene IFF-Bilder                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | G                                                                                                   |
| RPD 106a  | Movie 1.01<br>Henry.anim                                                                                                                                                                                                     | spielt Animationen<br>Animation von<br>Sculpt/Animate3D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | G<br>G                                                                                              |
|           | LorMovie                                                                                                                                                                                                                     | Animation von Sculpt/<br>Animate3D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | G                                                                                                   |
| RPD 106b  | PlaySilver<br>Fist2.ani<br>Head.anim                                                                                                                                                                                         | spielt Silver-Animationen<br>Animation von Silver<br>Animation von Sculpt/<br>Animate3D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | G<br>G<br>G                                                                                         |
|           | Rock.anim                                                                                                                                                                                                                    | Animation von Sculpt/<br>Animate3D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | G                                                                                                   |
| RPD 107a  | SonixPlayer<br>SonixScores                                                                                                                                                                                                   | spielt Sonix-Stücke<br>Instrumente und Stücke<br>für Sonix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | M<br>M                                                                                              |
| RPD 107b  | SonixScores                                                                                                                                                                                                                  | Instrumente und Stücke<br>für Sonix                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | M                                                                                                   |
| RPD 108   | Life<br>R2D2<br>Chaos<br>Labyrinthe<br>Copilot<br>Fragments<br>Headlines<br>SportsText<br>TIcon<br>VirusX 1.0<br>Maze1/2                                                                                                     | Lebens-Simulation<br>erzeugt zufällige Töne<br>stellt Töne optisch dar<br>erzeugt Labyrinthe<br>Utility, Demo-Version<br>Grafik-Demo<br>generiert zufällige Sätze<br>Titelgenerator<br>Anzeige von Textfiles<br>Virus-Checker<br>Grafik-Demos                                                                                                                                                                                                                                                                                                                                                                                                        | A<br>M<br>M<br>G<br>U<br>G<br>U<br>U<br>G<br>U<br>U<br>G                                            |
| RPD 109   | PDC<br><br>A68k                                                                                                                                                                                                              | Pre-Prozessor für<br>C-Programme<br>Makro-Assembler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | P<br><br>P                                                                                          |
| RPD 110   | Movie 1.01<br>Boing3D.anim<br><br>Flight.anim<br><br>Guitar<br><br>Jet.movie<br>WeirdNird                                                                                                                                    | spielt Animationen<br>Animation von Sculpt/<br>Animate3D<br><br>Animation von Sculpt/<br>Animate3D<br><br>Animation von Sculpt/<br>Animate3D<br><br>Animation<br>Animation                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | G<br>G<br><br>G<br><br>G<br>G                                                                       |
| RPD 111   | SlideShow                                                                                                                                                                                                                    | Diashow mit Ray-Tracing-<br>Bildern                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | G                                                                                                   |
| RPD 112   | Calcula<br>BModem<br>Sound.mod<br>Projector1.2<br><br>MultiView 2.0<br><br>NGI 1.3<br><br>ConMan 1.1<br>BootBack<br>DFC<br>EnTab<br>DeTab<br>FlickFix<br>Slice<br>Title<br>InovaTools<br>FileRequest<br>Match 2.0<br>PtrData | mathematische Funktionen<br>DFÜ-Programm<br>Sound-Modul für »Director«<br>Abspielen eines<br>»Director«-Scripts<br>Betrachten und<br>Konvertieren von<br>IFF-Bildern<br>verwandelt Brushes<br>in Icons, Images oder<br>C-Source<br>verbesserte Editierung etc.<br>unter CON:<br>kopiert Bootblocks<br>in Dateien<br>Kopierprogramm<br>verwandelt Tabulatoren<br>in Leerzeichen<br>Leerzeichen in Tabulatoren<br>Such-/Ersetz-Routine für<br>ASCII-Files<br>Such-Routine für<br>ASCII-Files<br>ändert Fenstertitel<br>Requester-Demo<br>File-Requester,<br>ARP (RPD 66)-Kommando<br>Benutzerabfragen für<br>Batch-Dateien<br>erstellt Pointer-Sources | T<br>A<br>M<br>G<br><br>G<br><br>T<br>T<br>U<br>T<br>T<br>T<br>T<br>T<br>T<br>T<br>T<br>T<br>T<br>T |
| RPD 113   | BattleMech<br>Descender                                                                                                                                                                                                      | Strategiespiel<br>Action-Spiel                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | S<br>S                                                                                              |

| Diskette: | Programm:                                                                                                                                                                                                              | Bemerkung:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Art:                                                                                                           |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|           | Hearts<br>BullRun<br>CryED                                                                                                                                                                                             | Kartenspiel<br>Strategiespiel<br>Level-Editor für<br>»Crystal-Hammer« (Spiel)                                                                                                                                                                                                                                                                                                                                                                                                                                | S<br>S<br>U                                                                                                    |
| RPD 114   | SlideShow                                                                                                                                                                                                              | Diashow mit Bildern von<br>Patrick Nagel                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | G                                                                                                              |
| RPD 115   | MacFontCvrt<br><br>SuperView 1.0<br>Fonts<br>Icons<br>Planes                                                                                                                                                           | wandelt Macintosh-<br>Zeichensätze in<br>Amiga-Fonts<br>zeigt IFF-Bilder an<br>verschiedene Zeichensätze<br>verschiedene Icons<br>ändert Anzahl der<br>Workbench-Bitplanes                                                                                                                                                                                                                                                                                                                                   | T<br><br>U<br>T<br>T<br>T                                                                                      |
| RPD 116   | AdySys 1.0<br><br>Pilot<br>Smalltalk 2.0<br>Web                                                                                                                                                                        | Programmiersprache<br>für Adventures<br>Programmiersprache<br>Programmiersprache<br>Pre-Prozessor für<br>Assembler-Programme                                                                                                                                                                                                                                                                                                                                                                                 | P<br><br>P<br>P<br>P                                                                                           |
| RPD 117   | Bubbles<br>GhostPool<br>Movie 1.01<br>Lumiere                                                                                                                                                                          | Grafik-Demo, Animation<br>Grafik-Demo<br>spielt Animationen<br>Tool für Farbeinstellung                                                                                                                                                                                                                                                                                                                                                                                                                      | G<br>G<br>G<br>T                                                                                               |
| RPD 118   | MathPaint<br>PhoneMate<br><br>SCalc<br>Simplica<br><br>MemoPad<br>BFormat<br>DefDisk<br>ViewBoot<br>Print<br>Bell<br>Big<br>MenuRunner<br><br>NW<br><br>Inf<br>MouseUtil<br><br>Read&Print<br><br>Rez<br><br>ShowLocks | Funktionsplotter<br>Telefonverzeichnis mit<br>Wählfunktion<br>Tabellenkalkulation<br>berechnet mathematische<br>Funktionen<br>Terminplaner<br>formatiert defekte Disketten<br>lenkt System-Directories um<br>Virus-Checker<br>Ausdrucken von Dateien<br>Bildschirmblitzer<br>Workbench-Lupe<br>Programm-Aufrufe<br>per Menü<br>Durchschalten von<br>Windows<br>verbessertes »info«<br>verändert Maus-<br>geschwindigkeit<br>Anzeigen/Ausdrucken<br>von Texten<br>macht Programme<br>speicherresident<br>Tool | G<br>A<br><br>A<br>A<br><br>A<br>U<br>T<br>U<br>U<br>T<br>U<br>T<br><br>T<br><br>T<br>T<br><br>U<br><br>T<br>T |
| RPD 119   | SlideShow                                                                                                                                                                                                              | Diashow                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | G                                                                                                              |
| RPD 120   | Muncho 2<br><br>BattleShip1.6<br><br>DeLux-<br>Hamburger<br>AnimBalls<br>Wave<br><br>ChangeColors<br>SpeechToy                                                                                                         | spielt Sound beim<br>Entnehmen der<br>Diskette<br>Spiel, Typ »Schiffe<br>versenken«<br>Schießspiel<br><br>Grafik-Demos, 3D<br>Konvertierprogramm für<br>Sound-Files<br>Veränderung der<br>Workbench-Farben<br>Sprachausgabe                                                                                                                                                                                                                                                                                  | M<br><br>S<br>S<br><br>G<br>M<br><br>U<br>M                                                                    |
| RPD 121   | ShowAnim 4.2<br><br>Movie<br>ThunderCat<br>Car.anim                                                                                                                                                                    | spielt Videoscape-<br>Animationen<br>spielt Animationen<br>Animation<br>Videoscape-Animation                                                                                                                                                                                                                                                                                                                                                                                                                 | G<br><br>G<br>G<br>G                                                                                           |
| RPD 122   | Arp 1.1                                                                                                                                                                                                                | Update zu RPD 66                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | T                                                                                                              |
| RPD 123a  | ShowAnim4.2<br><br>VideoFilm                                                                                                                                                                                           | spielt Videoscape-<br>Animationen<br>Videoscape-Animation                                                                                                                                                                                                                                                                                                                                                                                                                                                    | G<br><br>G                                                                                                     |
| RPD 123b  | ComingAt.anim<br>Ripples                                                                                                                                                                                               | Videoscape-Animation<br>Grafik-Demo, Animation                                                                                                                                                                                                                                                                                                                                                                                                                                                               | G<br>G                                                                                                         |



Von Klaus Sonnenleiter

**B**ereits der erste Kontakt mit dem PD-Programm »AnalytiCalc« ist für jeden Freak eine Herausforderung. Nach Durchforsten der Fish-Disk 104 stellt sich heraus, daß alle entscheidenden Files gepackt sind. Also schnell noch Fish 58 (mit »Arc«) besorgt, damit man endlich loslegen kann.

Der erste Start des entpackten Programms ist wenig motivierend: Der Guru schickt letzte Grüße vom System (0000 00 08: Privilegverletzung). Meine erste Reaktion (Wühlen im Doc-File) bringt keinen Fortschritt, aber viele Informationen über die offensichtlich fantastischen Fähigkeiten dieses Programms. Sollten diese wirklich nur für den Guru gedacht sein?

Das Fleddern aller erreichbaren »doc«- und »readme«-Files

meisten Programmierern der Weg über die Brechstange (das Neucompilieren) verwehrt bleibt.

Ist das schon das Ende meiner Experimente mit AnalytiCalc? Ein weiterer Blick ins Handbuch läßt mich hartnäckig bleiben. Von Leistungen, die gängige Programme wie Lotus bei weitem in den Schatten stellen, ist da die Rede. Offensichtlich hat sich der Programmierer, Glenn Everhart, doch einiges gedacht. Das macht dann doch neugierig auf die neue Version auf Fish 144. Auch diese Version trägt zwar noch den Vermerk, sie sei nur für den Interlace-Screen geeignet (diesmal sogar auffallend über mehrere »readme«-Files gestreut), aber Besitzer eines PAL-Amiga (mit 256 Punkten für die senkrechte Auflösung) können dies getrost ignorieren. AnalytiCalc öffnet

# Kraftp rauher C

**Tabellenkalkulationen für  
mit unzähligen Spielereien  
»AnalytiCalc« beschre  
Dessen Oberfläche ist s  
Die Anleitung spricht**

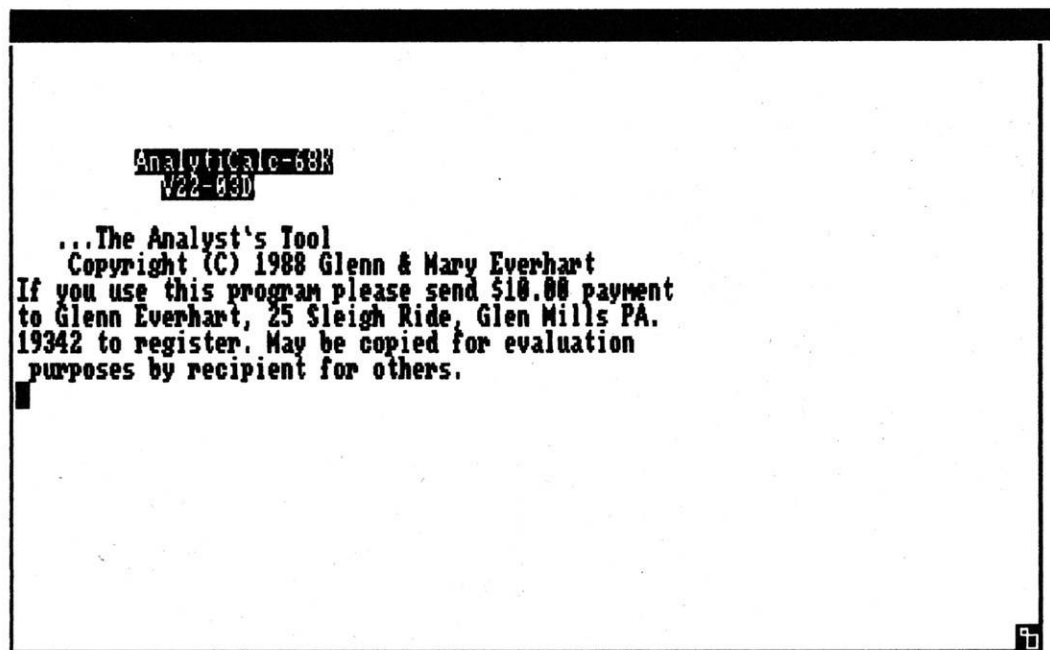


Bild 1. »AnalytiCalc« zeigt eine Einschaltmeldung, bevor alle benötigten Fenster geöffnet werden. Zwei Fenster sind aktiv.

bringt endlich eine erste Spur: Die lapidare Bemerkung, AnalytiCalc benötige einen Interlace-Screen, ist gut versteckt. Ein erster Versuch mit einer entsprechend eingerichteten Start-Diskette bringt tatsächlich eine Tabellenkalkulation auf den Bildschirm. Aber diese Form der Darstellung erfreut höchstens die Brieftasche des Optikers, nicht aber den Anwender einer Tabellenkalkulation. Noch dazu sind die Sourcecodes in Fortran, so daß den

ein Fenster im Bereich zwischen Zeile 200 und 256. Dadurch benötigen (amerikanische) NTSC-Amigas weiterhin die Interlace-Auflösung. Wer die (europäische) PAL-Norm benutzt, kann jedoch getrost einen augenschonenden Workbench-Screen verwenden. Allerdings sollten Sie wenigstens 1 MByte zur Verfügung haben. Mit 512 KByte ist AnalytiCalc nur mit größeren Einschränkungen zum Laufen zu bringen. Damit öffnet sich dem Benut-

zer endlich die ganze Bandbreite der AnalytiCalc-Leistungen auf einem erträglichen Bildschirm. Dieses Programm kann bis zu 18000 Spalten und 18000 Zeilen verwalten, arbeitet mit virtueller Speicherverwaltung, läßt alle nur erdenklichen Manipulationen zu und macht eine enorme Anzahl Formeln direkt zugänglich. Weitere sind über eine Fortran-Schnittstelle erreichbar. Sogar Schnittstellen zu fast beliebigen anderen Programmen (beispielswei-

se Textverarbeitung, Datenbank oder ähnliches) sind eingebaut.

Der Preis für diese Flexibilität ist die eher mittelmäßige Geschwindigkeit beim Bildschirmaufbau und eine auf den ersten Blick etwas umständliche Bedienung. Beide Nachteile sind Folge des Portabilitäts-Konzepts von AnalytiCalc. Das Programm wurde in Fortran entwickelt, und der Sourcecode ist so gestaltet, daß er möglichst ohne große Änderungen an jedem Computer angepaßt werden kann. Dadurch werden nicht die vom Amiga gewohnten Bedienungselemente verwendet – diese sind ja auf anderen Computern nicht vorhanden.

## Kalkulierbare Tabellen?

Allerdings sind die Befehle kurz und einprägsam gewählt, so daß man bereits nach kurzer Einarbeitungszeit kaum noch im Handbuch blättern muß. Dieses befindet sich auf der Diskette und ist mit 105 Druckseiten für ein PD-Programm ungewöhnlich umfangreich. Daneben wird eine Referenztabelle mitgeliefert. In einem Readme-File finden Sie die Amiga-typischen Besonderheiten von AnalytiCalc, und während der Arbeit sind die wichtigsten Befehle ständig per Tastendruck (<HELP>) verfügbar.

Nachdem Tabellenkalkulationen (oder neudeutsch: »Spreadsheets«) beim Amiga noch nicht die Tradition haben, wie auf anderen Computern, werden sich viele Anwender fragen, wozu



## Lotus mit Oberfläche

nigagelten als schön bunt,  
hen und intuitiv zu bedienen.  
ien völlig anderen Weg.  
iktional, fast spröde.  
istungen wie bei Lotus.

diese Programme nützlich sind? Zunächst ist eine Tabellenkalkulation einfach nur eine rechteckige Datenmatrix. Das allerdings gilt für eine Dateiverwaltung auch. Das besondere an einem Spreadsheet ist die einfache Art der Auswertung. Es ist aus diesem Grund besonders dafür geeignet, Daten zu analysieren. Dazu werden Formeln verwendet, die aus eingegebenen Daten ein Ergebnis berechnen und dieses an der Stelle anzeigen, an der die Formel steht. Dieses Ergebnis kann von anderen Formeln wiederum zur Eingabe verwendet werden. Das Besondere an diesem Vorgehen ist, daß alle Formeln (ein Datenblatt besteht oft bis zu 95 Prozent aus Formeln) bei jeder Eingabe neu berechnet werden. Wenn Sie also die Daten Ihrer Tabelle ändern, wird automatisch auch das Ergebnis aktualisiert.

Besorgen Sie sich die Fish-Disk 144 mit dem Programm »AnalytiCalc«. Dieses ist nicht auf unserer Programmservice-Diskette enthalten. Zum Entpacken benötigen Sie das Programm »Arc« von Fish 58. Dieses funktioniert im wesentlichen wie Zoo, das auf Seite 95 beschrieben wird. Sie können mit ARC gepackte Files jedoch nicht mit Zoo entpacken. Richten Sie sich eine Startdiskette ein, die nur die nötigsten Dateien enthält. Das sind vor allem das C-Verzeichnis mit den Dateien »assign«, »Copy«, »delete«, »cd«, »run«, »path«, »setmap«, »Stack« und »endcli«. Wenn Sie »runback« haben, kopieren Sie es ebenfalls in den C-Ordner. Der I-Ordner sollte die drei Dateien »Disk-Valida-

tor«, »Port-Handler« und »RAM-Handler« enthalten, der libs-Ordner die Libraries. Im devs-Ordner brauchen Sie neben den Devices (»printer.device« und »parallel.device«) die »system-configuration« und zwei weitere Unterverzeichnisse: »printers« enthält den Treiber für Ihren Drucker, und in »keymaps« benötigen Sie die Datei »d«, um den deutschen Zeichensatz einzustellen.

Schreiben Sie für diese Diskette eine Startup-Sequence, die mindestens folgende Einträge aufweist:

```
setmap d
stack 80000
assign dk: df0:
run df0:analy
loadwb
endcli
```

### Kurzreferenz

AnalytiCalc wird ausschließlich über Kommandos gesteuert. Als Kommando werden im »Command«-Modus die ersten Zeichen bis zur ersten Leerstelle interpretiert. Im »Enter«-Modus gilt das gleiche für die Zeichen nach dem Schrägstrich. Sie wechseln die beiden Modi mit den Befehlen »/;« und »//«. Die wichtigsten Befehle:

|    |                                                                       |
|----|-----------------------------------------------------------------------|
| e  | Standard-Eingabe                                                      |
| e" | Eingabe von Text, der unverändert übernommen werden soll              |
| ed | Editieren einer Zelle                                                 |
| ca | Kopieren aller Zell-Eigenschaften                                     |
| k  | Starten des integrierten Taschenrechners                              |
| l  | Springt eine Zelle direkt an                                          |
| oa | Setzt den Nullpunkt neu (normalerweise a1)                            |
| p  | Speichern des kompletten Spreadsheets                                 |
| r  | Neuberechnen des kompletten Spreadsheets (wichtig für einige Formeln) |
| s  | Startet die Anzeige mit neuen Parametern (Setup)                      |
| v  | Erneuern den Bildschirminhalt                                         |
| w  | Schreibt das File auf ein Ausgabegerät (Diskette oder Drucker)        |
| x  | Beendet AnalytiCalc                                                   |
| ze | Löscht angegebenen Bereich                                            |
| za | Löscht das ganze Spreadsheet                                          |

Bevor Sie von dieser Diskette booten, erledigen Sie erst die restlichen Vorbereitungen, sonst erhalten Sie die Fehlermeldung »unknown command df0:analy«. Entpacken Sie das File »Analyticalc.Arc« von der Fish-Disk 144 und kopieren Sie alle Dateien ins Stammverzeichnis Ihrer Startdiskette. Das gleiche machen Sie mit »Analy-Keypad.Arc«. Entpacken Sie außerdem die Dokumentation (»AnalyDocuments.Arc«) und kopieren Sie diese mit den »readme«, ».txt«- und ».doc«-Files auf eine andere Diskette. Außerdem sollten Sie die Dokumentation ausdrucken und wenigstens kurz überfliegen (das »Hauptwerk« besteht ausgedruckt aus 105 Seiten, daneben sind noch einige andere Files enthalten).

Wenn Sie zwei Laufwerke haben, bietet es sich an, auf die

Dokumentations-Diskette einen Editor zu kopieren und diesen nach AnalytiCalc von der Start-up-Sequence aus zu starten. Dabei ist der »DME« eine gute Lösung (siehe Seite 61). Er benötigt nur sehr wenig Speicher und kann auf Icon-Größe verkleinert werden. In diesem Fall könnten Sie die Dokumentation sozusagen als »Online-Hilfe« ständig als eigene griffbereit halten. Dieses Vorgehen ist wesentlich angenehmer als das ständige Blättern in den ausgedruckten Seiten.

### Durchblick durch zwei Fenster

Booten Sie jetzt von Ihrer neu eingerichteten Diskette. Sie startet AnalytiCalc automatisch. Auf dem Bildschirm sehen Sie ein Fenster mit der Ein-

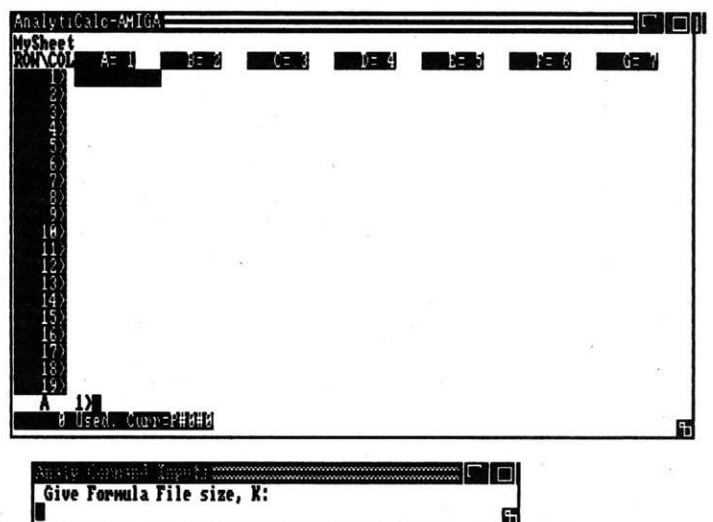


Bild 2. Das Spreadsheet besteht aus einer rechteckigen Datenmatrix. Die Spalten sprechen Sie mit den Buchstaben an, die Zeilen mit Zahlen.

schaltmeldung (Bild 1). AnalytiCalc öffnet in der aktuellen Version noch zwei Fenster: Ein Ausgabefenster, das für die Tabellenkalkulation verwendet wird, und ein Fenster zur Eingabe von Kommandos. Dies mag etwas verwirrend erscheinen, Sie erhalten aber jedesmal eine eindeutige Meldung, wenn von Ihnen eine Eingabe in dem unteren Fenster erwartet wird. Natürlich ist es störend, wenn Sie – um auf diese Anforderung reagieren zu können – jedesmal das entsprechende Fenster mit der Maus aktivieren müssen. Dieses Problem dürfte jedoch in der nächsten Version von AnalytiCalc beseitigt sein.

Beim Start des Programms werden Ihnen von AnalytiCalc einige Fragen im unteren Fenster gestellt. Diese betreffen das Format des Files und sollten so genau wie möglich be-



antwortet werden, wenn auch die Werte später jederzeit geändert werden können.

Die erste Frage betrifft das Format der Fließpunktzahlen. Dies ist voreingestellt als Fortran-Format »9.2«. Das bedeutet, daß jede Zahl mit maximal neun Stellen dargestellt wird, wobei die letzten beiden hinter dem Dezimalpunkt stehen. Beantworten Sie diese Frage stets mit »N«, es sei denn, Sie wollen

## Fließkomma mit Format

ein anderes Format verwenden. Danach können Sie einen beliebigen Text als Titel für das Spreadsheet angeben. Die nächsten vier Fragen betreffen den intern zu reservierenden Platz. Geben Sie für den Anfang die folgenden Werte an: 22, 22, 1, 1. Sie können natürlich jederzeit geändert werden. Wenn Sie alle Werte angegeben haben, wird die leere Tabelle aufgebaut (Bild 2).

Natürlich brauchen Sie nicht jedesmal sechs Werte eingeben, bevor Sie mit Ihrem Spreadsheet starten können. AnalytiCalc sucht beim Start zunächst im aktuellen Directory nach einem File »ACINIT.PRM«. Wenn ein solches vorhanden ist, werden die Parameter aus diesem gelesen. Um dieses File anzulegen, starten Sie Ihren Editor und geben Sie folgende

Zeilen ein:

```
N
beliebiger Name
22
22
1
1
```

Speichern Sie diesen Text unter dem Namen »ACINIT.PRM« und starten Sie AnalytiCalc neu. Die Fragen werden diesmal von dem neu angelegten File beantwortet.

Bewegen Sie den Cursor mit den Pfeiltasten über die Anzeige. Beim ersten Tastendruck dauert es einen kleinen Moment, bevor der Cursor reagiert – das entsprechende Kommando wird erst nachgeladen. AnalytiCalc hat zwei Eingabemodi – den »Enter«- und den »Command«-Modus. Beim Start des Programms befinden Sie sich im Command-Modus. Sie erkennen dies an der Fußzeile, die folgendes Aussehen hat:

```
A 1>
```

Das bedeutet aber nicht, daß Sie in diesem Modus nichts eingeben können. Sie müssen nur für jede Eingabe das entsprechende Kommando voranstellen, beispielsweise schreibt

```
e 555.33 <RETURN>
```

die Zahl 555.33 in das Feld, auf dem der Cursor steht. Wenn diese Art der Eingabe zu mühsam ist, der kann mit dem Befehl

```
;/
```

in den Enter-Modus wechseln. In diesem geben Sie die Werte direkt (ohne vorangestelltes »e«) ein. Dafür müssen Sie jetzt für jedes Kommando einen Schrägstrich voranstellen. Um wieder zurückzuwechseln in den Command-Modus, geben Sie ein:

```
//
```

Wir gehen im folgenden bei allen Kommandos davon aus, daß Sie sich im Command-Modus befinden. Geben Sie zunächst einige Werte mit »e Wert« ein. Beachten Sie dabei, daß alle Werte im »Floating-Point-Format« angegeben wer-

den müssen, also als Gleitpunktzahl (siehe Bild 3). Sie können dieses Format mit dem »F«-Befehl zu Integer- oder Text-Format ändern, wenn es Ihnen nötig erscheint. Dieser Befehl kann selbstverständlich auch nur für einen Bereich angewendet werden. AnalytiCalc interpretiert in diesem Modus grundsätzlich das erste Zeichen als Kommando. Wird kein entsprechendes Kommando in der Liste gefunden, erhalten Sie eine Fehlermeldung. Für den Anfänger ist diese Eigenschaft leicht verwirrend, denn oft wird das »e« vergessen, das Ergebnis ist eine Aktion von

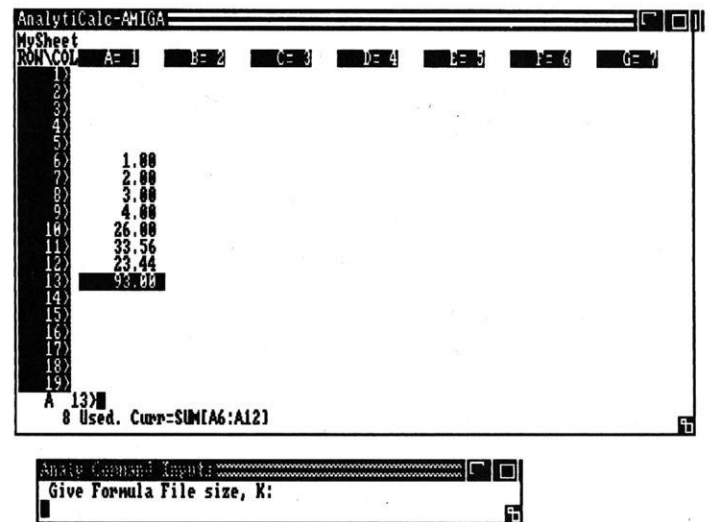


Bild 3. Alle Werte werden standardmäßig im Fortran-Format »9.2« dargestellt (maximal 9 Stellen)

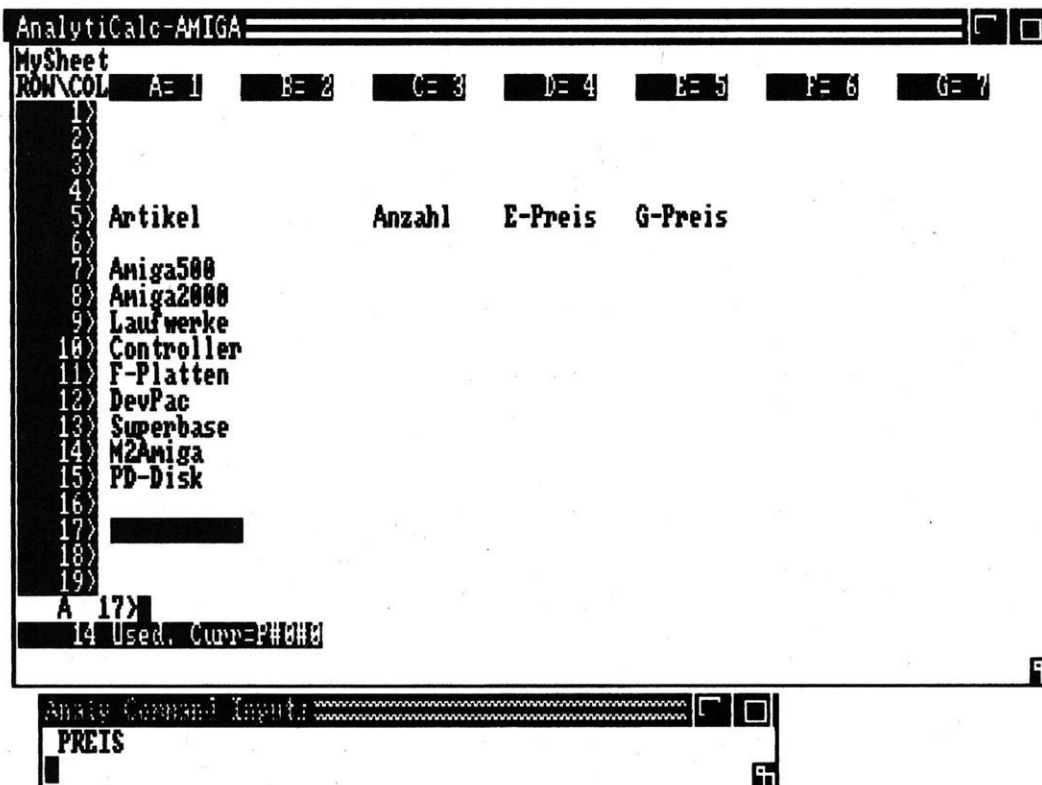


Bild 4. Texte in gemischter Schreibweise werden mit dem »e«-Kommando eingegeben

AnalytiCalc, die durch das erste Zeichen hervorgerufen wurde. Lassen Sie sich davon nicht beirren, es gibt immer einen Weg, dieses Problem zu umgehen. Meist genügt ein Blick ins Handbuch.

Geben Sie in einige Felder Text ein. Sie werden feststellen, daß alle eingegebenen Zeichen zu Großbuchstaben konvertiert werden. Setzen Sie den Cursor auf ein freies Feld und geben Sie dort ein:

```
e" kleinGROSS
```

Dieser Text erscheint in der angegebenen Schreibweise. Wenn Sie ein Feld mit einem anderen Wert überschreiben, funktioniert dies nicht immer fehlerfrei. Daher ist es oft sinnvoll, Felder, die überschrieben werden sollen, vorher mit dem »Z«-Befehl (»Zero«) zu löschen. Sie können mit »za« (»Zero All«) das ganze Fenster löschen oder mit »ze« einen Bereich. Diesen geben Sie mit Hilfe seiner Koordinaten an, beispielsweise »a2« oder »a7:b13«. Der Doppelpunkt hat die Bedeutung »bis«. Der Befehl



# ANWENDUNGEN

| Artikel        | Anzahl         | E-Preis | G-Preis |
|----------------|----------------|---------|---------|
| Amiga500       | 567.00         |         |         |
| Amiga2000      | 288.00         |         |         |
| Laufwerke      | 377.00         |         |         |
| Controller     | 43.00          |         |         |
| F-Platten      | 56.00          |         |         |
| DevPac         | 297.00         |         |         |
| Superbase      | 314.00         |         |         |
| M2Amiga        | 512.00         |         |         |
| PD-Disk        | 3178.00        |         |         |
| <b>Gesamt:</b> | <b>5632.00</b> |         |         |

C 17) 25 Used. Curr=SUM(C7:C15)

Bild 5. Die volle Leistungsfähigkeit von Spreadsheets zeigt sich erst bei der Verwendung von Formeln

ze [a3:c9]

würde also die Spalten a bis c im Bereich 3 bis 9 löschen. Bereiche müssen in eckigen Klammern eingeschlossen sein.

Geben Sie zur Übung die Felder ein, die Sie in Bild 4 sehen. Erweitern Sie darauf die Tabelle, indem Sie für jeden Artikel eine Stückzahl eintragen. Alle bisherigen Aktionen könnten Sie mit einer Textverarbeitung wesentlich komfortabler durchführen. Die Stärke einer Tabellenkalkulation liegt in anderen Bereichen. Geben Sie im Feld c17 ein:

e sum[c7:c15]

Sie erhalten in der Anzeige sofort die Gesamtzahl der verkauften Artikel (Bild 5). In der Zeile unter der Eingabezeile lesen Sie jederzeit ab, welchen tatsächlichen Inhalt das aktuelle Feld hat. Damit können Sie unterscheiden, ob es sich um eine Konstante oder um eine Formel handelt. Geben Sie im Feld e7 ein:

e (c7\*d7)

Sie erhalten auch hier das Ergebnis der Formel. Der Befehl

cr e7 e8

kopiert die Formel aus der Zelle e7 nach e8. Dabei werden aber die absoluten Adressen der Formel mitkopiert. Sie haben also in e8 den gleichen Wert wie in e7. Der Sinn dieser Vorgehensweise ist natürlich nicht sofort zu erkennen. Die Sache wird erst dann wirklich interessant, wenn Sie keine absoluten Adressen verwenden, sondern relative. Dazu geben Sie statt der oben genannten Formel dieselbe in relativer Darstellung ein (»P#n#m« für Werte, die

| Artikel        | Anzahl         | E-Preis | G-Preis   |
|----------------|----------------|---------|-----------|
| Amiga500       | 567.00         | 998.00  | 565866.00 |
| Amiga2000      | 288.00         | 2166.00 | 623888.00 |
| Laufwerke      | 377.00         | 215.00  | 81055.00  |
| Controller     | 43.00          | 455.00  | 19565.00  |
| F-Platten      | 56.00          | 1678.00 | 93968.00  |
| DevPac         | 297.00         | 149.00  | 44253.00  |
| Superbase      | 314.00         | 399.00  | 125286.00 |
| M2Amiga        | 512.00         | 349.00  | 178688.00 |
| PD-Disk        | 3178.00        | 11.90   | 37818.20  |
| <b>Gesamt:</b> | <b>5632.00</b> |         | *****     |

E 17) 44 Used. Curr=SUM(E7:E15)

#11\*P#D#11)

d:\analy.doc

Bild 6. Zahlen, die breiter als neun Stellen sind, können nicht im Standard-Format dargestellt werden, das Ausgabeformat ist überstrapaziert

relativ zum gesamten Datenblatt sind; »D#n#m« bezieht sich auf den dargestellten Bereich). Wenn Sie diese Formel kopieren, erhalten Sie die korrekten Werte.

Kopieren Sie die Formeln in jede Zeile und berechnen Sie die Summe in der letzten Zeile. Ihr Datenblatt sollte jetzt aussehen wie das von Bild 6 (eventuell mit anderen Werten). Die Summe sehen Sie dargestellt als »\*\*\*\*\*«. Das braucht Sie nicht zu beunruhigen, es soll Sie nur darauf hinweisen, daß

die benötigten Ziffern das eingestellte Darstellungsformat überstrapazieren. Ändern Sie es mit dem »S«-Kommando (beispielsweise auf 11.2) und Sie erhalten den korrekten Wert.

Natürlich kann dieser Artikel nur einen sehr kleinen Ausschnitt aus den möglichen Leistungen von AnalytiCalc zeigen. Wir wollten Sie hiermit auch eher zu eigenen Experimenten ermuntern. AnalytiCalc hat sicher etwas mehr Beachtung verdient als es im Moment erfährt. Die sehr ausführliche Programmdokumentation hilft bei Schwierigkeiten weiter. Es werden alle Bereiche klar und deutlich erklärt.

Lassen Sie sich nicht entmutigen, wenn der Bildschirm ein-

ten Paket aufwerten. Sowohl eigene Makros wie Schnittstellen zu anderen Programmen sind problemlos möglich, sobald man sich etwas an das Prinzip gewöhnt hat (Ihr erstes Makro finden Sie übrigens in »ACINIT.PRM«).

Für den Fortran-Programmierer sind die mitgelieferten Sourcecodes interessant. Vielleicht gelingt es Ihnen, das Programm so weiterzuentwickeln, daß es die speziellen Fähigkeiten des Amiga besser nutzt, und so der Grafikaufbau beschleunigt und die Benutzerfreundlichkeit erhöht wird. Es dürfte nur wenig Probleme bereiten, wichtige Funktionen wie Laden und Speichern auf Menüs zu legen und damit für den Anwender übersichtlicher zu

mal nicht das gewohnte Erscheinungsbild hat. Mit dem »V«-Kommando können Sie ihn reorganisieren. Wenn Sie einen

## Reorganisation des Bildschirms

falschen Wert eingegeben haben, helfen die »Z«-Kommandos weiter.

Für den fortgeschrittenen Anwender sehr interessant sind die Schnittstellen von AnalytiCalc, die es fast zum integrier-

machen. Daneben kann es auch nicht sonderlich schwierig sein, die zusätzlich benötigten Fenster als Requester zu installieren, die nur auftreten, wenn sie wirklich gebraucht werden.

In jedem Fall erhalten Sie mit AnalytiCalc ein Programm, mit dem zwar anfangs nicht leicht umzugehen ist. Dafür entpuppt es sich nach einiger Gewöhnung als äußerst leistungsfähig und mit vielen ungewöhnlichen Funktionen versehen. Sie sollten diesen Fish-Zug also keinesfalls auslassen.



In den Vereinigten Staaten ist die DFÜ schon lange ein absoluter Renner. Immer mehr Computer-Anwender nutzen die Datenfernübertragung (DFÜ). Dieser Trend wird auch in Europa immer deutlicher. Mailboxen schießen wie Pilze aus dem Boden, Modems zum Anschluß des Computers an das Telefonnetz werden häufiger verkauft denn je. Ein immer beliebteres Anwendungsgebiet ist die Verbreitung von Public Domain-Programmen. So finden Amiga-Benutzer beispielsweise in der Mailbox HOST eine Reihe von nützlichen Hilfsprogrammen.

Für den Amiga gibt es einige sehr interessante Terminal-Programme. Ein kommerzieller Vertreter ist beispielsweise »Diga!«. Aus dem PD-Bereich stellen wir Ihnen hier ein ausgereiftes und leistungsfähiges Produkt vor: »AZComm« in der Version 1.0 (Bild 1). Dieses Programm von dem Amerikaner S.S. Patel ist eine Weiterentwicklung von »Comm 1.34« von D.J. James. Neu sind bei »AZComm« gegenüber »Comm« die eingebauten ZModem-Optionen, die Datenübertragung mit rasanter Geschwindigkeit unterstützen. Alle bekannten Vorteile von »Comm« wurden beibehalten. Ein kleiner Fehler in der Routine zum XModem-Empfang wurde gleich mit behoben, die Version auf der Fish 171 ist eindrucksvoll.

Die wichtigsten Features von »AZComm«:

- im ASCII-Modus senden und empfangen
- XModem-Protokoll mit CRC (Cyclic Redundancy Check, 16 Bit) und Checksummen-Prüfung (nach Christensen)
- ZModem-Protokoll
- Bildschirm Ein-/Ausgabe größer 2400 Baud
- Telefonbuch mit maximal 44 Einträgen, höchstens zehn Einträge über Tastenkombinationen (Hot Keys) abrufbar
- bis zu 20 Tastatur-Makros
- Geteilter Bildschirm (Ein-/Ausgabe) für Konferenzen
- Parity, Bit-Länge, Stop-Bit-Länge einstellbar
- Full- oder Half-Duplex

Bevor Sie mit »AZComm« so richtig loslegen, sind einige kleine Punkte zu erledigen. Das Programm nutzt die Voreinstellung der Baud-Rate, die in den Preferences eingestellt ist. Passen Sie diese vor dem Start an Ihre Erfordernisse an.

Ein Telefonbuch mit den wichtigsten Nummern Ihrer Stamm-

Mailbox oder einiger Freunde ist anzulegen. Wählen Sie als Namen für das File »Comm.Phone«. Kopieren Sie den ED (Editor) in das c-Directory Ihrer Startup-Diskette. Diesen benötigen Sie später, wenn Sie Einträge in das Telefonbuch anfügen und die Modem-Kommandos (z.B. für automatisches Anwählen) bestimmen. Die Einträge in die Datei »Comm.Phone« sind auch mit einem beliebigen anderen Editor vorzunehmen. Allerdings sind Änderungen vom gestarteten Programm nur dann möglich, wenn der Editor im c-Directory mit »ED« bezeichnet ist. Zu den Einträgen später mehr.

Wenn Sie »AZComm« von der Workbench aus starten wollen, sind im Info-File die Dateien »Comm.Phone« und »Comm.Keys« als Tool-Types einzutragen. Klicken Sie dazu das »AZ-

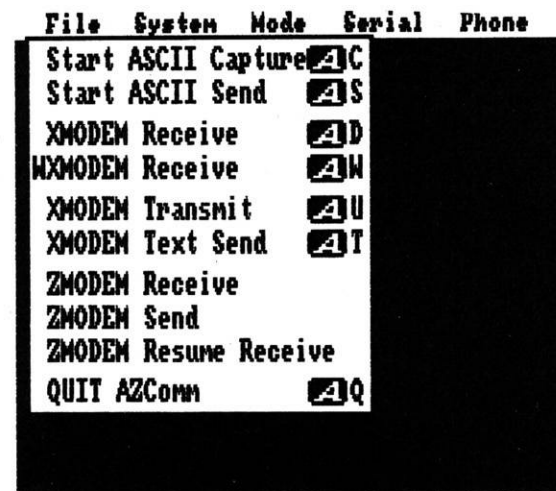


Bild 1. Das »ZModem«-Protokoll bringt dem »AZComm«-Anwender hohe Geschwindigkeit und große Sicherheit bei der Datenübertragung

Comm«-Icon einmal an und wählen im Workbench-Menü den Punkt Info. Bewegen Sie den Mauszeiger auf den Schalter »ADD« in der Zeile »TOOL TYPES«, klicken einmal mit der linken Maustaste und schreiben in die Eingabezeile:

PHONE=Comm.Phone  
<RETURN>

Klicken Sie erneut auf »ADD« und geben folgendes ein:

KEYS=Comm.Keys <RETURN>

Sollten schon Einträge unter »Tool Types« zu sehen sein, überprüfen und ändern Sie diese gegebenenfalls.

Setzen Sie zum Schluß noch den Wert für den Stack auf 8000 und speichern alle Einstellungen, indem Sie »SAVE« anklicken.

Die Vorbereitungen für den Start sind damit abgeschlossen. Jetzt können Sie »AZComm« aufrufen. Von der Workbench genügt ein Doppelklick auf das entsprechende Symbol (Icon).

# Der Daten

Ein Spediteur verpackt und versendet fähig sein. Ähnliches leistet ein Terminalreibrungslosen und möglichst schneller finden Sie auf der Fish-Disk 171 bestechender

Wenn Sie das Programm vom CLI aus aufrufen, geben Sie folgendes ein:

azcomm [filename]

Der optionale, also nicht unbedingt benötigte Parameter »Filename« bezeichnet das Telefonbuch, daß beim Start geladen werden soll.

Hat alles bisher geklappt?

Bildschirmrand finden Sie fünf Menü-Bezeichnungen.

»File« (Bild 1) ist das Übertragungs-Menü. Hier geben Sie das Übertragungs-Protokoll (ASCII, XModem, WModem, ZModem) an, bestimmen, ob Files gesendet oder empfangen werden, und legen die Datei fest, die übertragen werden soll. Dazu erscheint ein Requester in der Bildschirmmitte (Bild 2), in dem Sie Pfade und Dateinamen festlegen.

Im File-Menü verlassen Sie nach der Reise durch das Datennetz »AZComm«, klicken Sie dazu »QUIT AZCOMM« an.

Die meisten Befehle in diesem Menü sind zusätzlich mit Hot-Keys aufzurufen. Folgende Tasten zusammen mit der linken AMIGA-Taste sind belegt:

- |   |                          |
|---|--------------------------|
| C | Capture ein/aus          |
| S | Senden einer ASCII-Datei |
| D | Empfang mit XModem       |
| W | Empfang mit WModem       |
| U | Senden mit XModem        |
| T | Texte senden mit XModem  |
| Q | AZComm verlassen         |

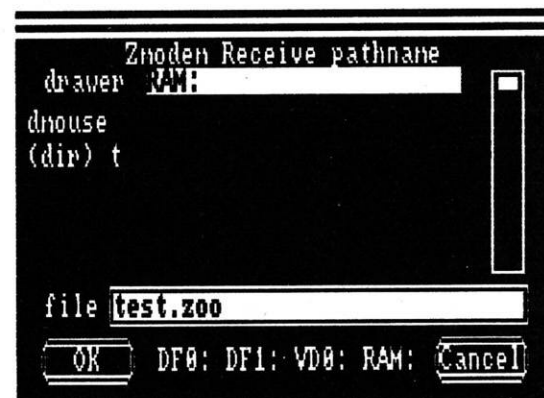


Bild 2. Komfortabel: File-Requester ersparen lästiges Tippen

Dann freuen Sie sich auf ein leistungsfähiges Terminalprogramm. Nach dem Laden erscheint eine nicht sehr eindrucksvolle, einsame Zeile am oberen Rand des Bildschirms. Dort steht lapidar »AZComm 1.00«. Natürlich ist ein vom Amiga verwöhnter Anwender Menü gewohnt. Und diese erscheinen dann auch, wenn man die rechte Maustaste drückt und festhält. Am oberen

Wählen Sie mit <linke-Amiga C> den Capture-Modus. Es erscheint die Meldung »Default CAPTURE file RAM:Capture.txt opened«. Es wurde ein File in der RAM-Floppy mit dem Namen »Capture.txt« angelegt. Drücken Sie erneut <linke-Amiga C>, erscheint die Meldung »End File Capture«, das File wurde geschlossen.

Wenn Sie den Punkt »Start ASCII Capture« im Menü wäh-



## spediteur

**Güter. Er muß dabei schnell und sorg-  
Programm: Es bereitet Daten für den  
Transport auf. Im Public Domain-Pool  
ein Terminal-Programm mit  
Leistungen.**

len, erscheint ein Requester, in dem Sie den Pfad und Namen für die Capture-Datei (in die alle empfangenen Zeichen geschrieben werden) festlegen. Geben Sie in den Balken für den Filenamen ein:

CAP.txt

und drücken <RETURN>. In der rechten unteren Ecke des Bildschirms erscheint die Meldung »ON«. Drücken Sie erneut <linke-Amiga C>, wechselt die Meldung in »SUS«. Dies steht für »Suspended«, die Daten werden nun nicht mehr in das File »CAP.txt« übertragen. Ein weiteres <linke-Amiga C> schaltet wieder auf »ON«. Sie können so Daten gezielt in der Capture-Datei speichern. Verlassen Sie den Capture-Modus, indem Sie im File-Menü den Punkt »STOP ASCII Capture« wählen.

Ein erneuter Aufruf der Funktion »START ASCII Capture« zeigt wieder den oben beschriebenen Requester. Wäh-

auf »CONTINUE« und »APPEND«. Mit <linke-Amiga B> sparen Sie die Mausbewegung auf »CANCEL« und »DELETE«.

Eine weitere Funktion ist auf der Tastatur versteckt: Die HELP-Taste zeigt einen Statusbericht. Dieser zeigt den Capture-Zustand, den freien Speicher und ob der Drucker aktiviert ist. Mit dem Schieberegler

| File System             | Mode | Serial                                                       | Phone    |
|-------------------------|------|--------------------------------------------------------------|----------|
| Markt&Technik Verlag AG |      | Baud Rate                                                    | 300      |
| 1200 8nl login: halifa  |      | Parity                                                       | ✓ 1200   |
| Passwort:               |      | Data Length                                                  | 2400     |
|                         |      | Stop Bits                                                    | 4800     |
|                         |      | Duplex                                                       | 9600     |
|                         |      | XON / XOFF                                                   | 19200    |
| Letztes Login an : Sa.  |      | Send Break                                                   | 1/23/83. |
|                         |      | Markt & Technik Verlag AG<br>Online - Informations - Systeme |          |

**Bild 4. Rasant: Bis zu 19200 Bit/s sind einstellbar – für Übertragungen von Amiga zu Amiga durchaus sinnvoll**

| File System | Mode | Serial         | Phone          |
|-------------|------|----------------|----------------|
|             |      | Split Screen   |                |
|             |      | View Transfers |                |
|             |      | ✓ Chop Files   |                |
|             |      | End of Line    | ✓ TX EOL = CR  |
|             |      | Key Macro      | TX EOL = CR/LF |
|             |      | Phone Dir      | ✓ RX EOL = CR  |
|             |      | Title Bar      | RX EOL = CR/LF |

**Bild 3. Im System-Menü stellen Sie unter anderem die Kennung für Zeilenende ein**

len Sie durch Anklicken mit der linken Maustaste das File »CAP.txt« und klicken auf »O.K.«. Die Warnung »File already exists!« erscheint. Klicken Sie auf »Continue«, erscheint die Frage »Delete existing file?«. Wählen Sie »APPEND«, werden die folgenden Daten dem File angehängt. »DELETE« löscht den Inhalt des Files. Die Tastenkombination <linke-Amiga V> hat die gleiche Wirkung wie ein Klick

»Send Delay« stellen Sie die Verzögerung zwischen der Übertragung eines einzelnen Zeichens ein. Steht der Schieberegler ganz links, ist dieser Verzögerungswert 10 Millisekunden, ganz rechts sind es zirka 0,75 Sekunden.

Soviel zu einigen wichtigen Feinheiten von »AZComm«. Die Funktionen des File-Menüs sind weitgehend selbsterklärend.

Aber wie wähle ich denn nun einen anderen Computer an? Wählen Sie im System-Menü (Bild 2) den Punkt »Split Screen«. Daraufhin erscheint im unteren Drittel des Bildschirms ein durch eine Linie abgetrennter Screen. Auf diesem sehen Sie jetzt Ihre Eingaben. Wählen Sie nun eine beliebige Telefonnummer. Dabei ist zu berücksichtigen, daß die eingegebenen Daten zum Modem geschickt werden. Dieses muß in den Kommando-Modus umgeschaltet werden, damit gewählt werden kann. Eine typische Folge sieht so aus:

```
+++ATDT 0895606031
```

Damit teilen Sie zum Beispiel dem »Lightspeed«-Modem mit, daß die Nummer 089/4606031 gewählt werden soll. Komfortabler ist das Wählen natürlich, wenn Sie ein Telefonbuch an-

Ist das Telefonbuch geladen, sind die ersten zehn Einträge mit der Tastenkombination <rechte-Amiga Zahl> aufzurufen. Für Zahl steht die Nummer des Eintrags, bei 1 beginnend – für den zehnten Eintrag steht die 0.

Vor der Anwahl eines Partners im Datennetz müssen natürlich die Übertragungsparameter stimmen. Unter dem Menü »Serial« (Bild 2) ändern Sie die Einstellungen. Hier ist auch ein Break-Signal zu senden.

Ein weiterer Vorteil von »AZComm« sind die frei definierbaren Tastatur-Makros. Beim Starten sucht das Programm nach der Datei »comm.keys«. Diese ist vorher mit einem Editor anzulegen oder zu ändern. Die Funktionstasten <F1> bis <F10> sind zu belegen, zusätzlich die gleichen Tasten in Kombination mit <SHIFT>. Das Format im »comm.keys«-File muß folgendermaßen aussehen:

Fx/...

oder

Sx/...

Mit F oder S (für SHIFT) beginnt der Eintrag. Es folgt die Nummer der Funktionstaste und durch einen Schrägstrich abgetrennt der Inhalt des Makros, der sogenannte Körper.

Der Körper kann Control-Anweisungen beinhalten. Diese sind in der C-Konvention einzugeben. Zum Beispiel steht »\n« für einen Zeilenvorschub. Die unterstützten Befehle sind:

\n, \r, \t, \f und \\

Die Ausdrücke »\n« und »\r« bewirken beide einen Zeilenvorschub. Alternativ dazu ist mit dem Hochdach und folgendem »J« ein Zeilenvorschub zu bewirken.

Zusätzlich gibt es noch die Option »\w«. Damit bewirken Sie eine kurze Pause von zirka einer halben Sekunde. Mehrere »\w«-Sequenzen nacheinander sind in einem Makro gültig. Ein Makro darf allerdings nicht länger als 127 Zeichen sein.

### Versteckte Schalter

Das Close-Gadget (Schließ-Schalter) und der Schalter, mit dem das »AZComm«-Fenster in den Hintergrund geklickt wird, sind versteckt angebracht. Das Close-Gadget schließt das Fenster. Es befindet sich wie allgemein üblich in der linken oberen Ecke. Finden Sie's?

Das Close-Gadget hat eine weitere Funktion. Haben Sie die Option »Split Screen« aus



dem System-Menü aktiviert, können Sie das kleine Dialogfenster am unteren Bildschirmrand mit dem Gadget schließen.

Nun noch ein paar Sätze zum Format des Telefonbuchs. Wie auch die Tastatur-Makros erstellen und ändern Sie dieses mit einem Editor, beispielsweise dem SuperED. Ein Beispiel für ein Telefonbuch ist zur Version 1.0 mitgeliefert. Die Einträge der einzelnen Teilnehmer sind in einer Spalte festzulegen. Es sind dabei in der Spalte reserviert:

- die ersten 17 Zeichen für den Namen
- die folgenden 17 Zeichen für die Nummer
- 9 Zeichen für die Baud-Rate
- 32 Zeichen für einen beliebigen Kommentar

Die Wählkommandos für das Modem sind ebenfalls hier festzulegen. Benutzen Sie die Sequenzen, die Sie im Handbuch des Modems finden. Im mitgelieferten »comm.phone« sind die Werte für Hayes-Kompatible eingestellt. Die Erkennungscodes für die Unterschiede der Sequenzen:

```
<D hier Sequenz zum
Wählen eintragen
<T Sequenz zum
Unterbrechen
<H Sequenz zum Auflegen
<I Initialisieren des
Modems
<E Exit String
```

Zum Schluß möchten wir die bisher nicht oder nur kurz angesprochenen Menüpunkte von »AZComm« kurz aufführen. Die einzelnen Punkte sind auch im »doc«-File des DFÜ-Programmes beschrieben. Hier finden Sie zusätzliche Tips.

HOST

Privater - Kommunikations - Service

V 1.1

Copyright (C) 1988 Markt&Technik Verlag Aktiengesellschaft

\* Foren  
\* Spiele  
\* Konferenz  
\* Dialog  
\* Mail  
\* Parameter  
\* Ablage

Information und Diskussion  
High-Score-Jagd Online  
Unterhaltung mit mehreren Usern  
Privatgespräch mit einem User  
Elektronische Post  
Einstellen der Systemparameter  
Persoenliches Archiv/Bibliotheken

\* Hilfe

\* Logoff

==>■

Bild 5. Die kommerzielle Mailbox HOST bietet aktuelle Themen. Wählen Sie die unter der Vorwahl München die zu erreichende Box doch einmal als Gast an.

### MENÜ SYSTEM

**Split Screen:** Öffnet ein kleines Fenster am unteren Rand. Alle getippten Zeichen erscheinen jetzt hier. Die vom Partner ankommenden Meldungen sehen Sie im oberen Fenster. Das betrifft auch Paßwörter, die beispielsweise beim Zugang zu einer Mailbox eingegeben werden. Lassen Sie sich dabei nicht über die Schulter sehen, oder wählen Sie »Split Screen« erst nach dem Einloggen. Schließen des Screens mit erneuter Anwahl des Menüpunktes oder dem Betätigen des versteckten Close-Gadgets.

**View Transfers:** Während der Übertragung werden die Daten in einem Fenster angezeigt. Nur bei ASCII-Texten sinnvoll.

**Chop Files:** Muß unbedingt eingestellt sein, wenn binäre Daten übertragen werden. Zu erkennen ist die Einstellung an dem Haken hinter dem Menü-

punkt. Bei übertragenen »arc«-Files schaltet »AZComm« automatisch in den Chop-Modus.

**End of Line:** Die Ende-Kennung einer Zeile ist hier einzustellen. Gilt für Bildschirmdarstellung und für die Übertragung auf den seriellen Port.

**Key Macro:** Makro-Aufrufe sind zu laden und zu verändern.

**Phone Dir:** Telefonbuch laden. Das Editieren ist nur dann möglich, wenn »AZComm« vom CLI aus aufgerufen wurde.

### MODE

**Checksum:** Im Zweifelsfall diese Fehlerkorrektur wählen. Arbeitet nach dem Christensen-Protokoll. Mit allen XModem-Varianten übertragen Sie auf diese Art problemlos.

**CRC:** Sehr effektive Fehlerkorrektur und -überprüfung. Nicht alle Übertragungsprotokolle unterstützen diese Variante.

### SERIAL

**Baud:** Einstellen der Übertragungsrate (Bild 3). Die Höhe der maximalen Geschwindigkeit hängt davon ab, ob ein Capture-File oder Druckerausgaben aktiviert sind. Bei ausgeschaltetem Capture und ohne Druckerprotokoll liegt die maximale Geschwindigkeit bei 2400 bis 4800 Baud.

**Parity:** Wählen Sie ODD, EVEN oder NO parity.

**Data Length:** 7 oder 8 Bit.

**Stop-Bits:** 1 oder 2 sind einstellbar.

**Duplex:** FULL oder HALF Duplex. Bei HALF sehen Sie die Zeichen, die Sie eingeben. Wenn der Partner am anderen Ende eine »Echo«-Funktion eingestellt hat, ist das natürlich nicht nötig.

**XON/XOFF:** Check für den seriellen Treiber. Wählen Sie »Automatic«, wenn der Treiber den Check benutzen soll. Bei Baud-Raten über 2400 sinnvoll.

Dir: PD (1)

|         |                |        |        |               |        |
|---------|----------------|--------|--------|---------------|--------|
| * 1     | Infofile       | 5248   | * 2    | ASDG.ARC      | 14080  |
| * 3     | Blitz.PAK      | 16768  | * 4    | BobEd.PAK     | 21248  |
| * 5     | Cruncher       | 8704   | * 6    | DMouse.ARC    | 8448   |
| * 7     | DIV.ZOO        | 24320  | * 8    | Handshake.PAK | 71296  |
| * 9     | Cled.ARC       | 11520  | * 10   | Journal.ARC   | 56320  |
| * 11    | KV.ZOO         | 6016   | * 12   | Mindlink.PAK  | 29184  |
| * 13    | MouseOff.PAK   | 6784   | * 14   | Ping.PAK      | 3456   |
| * 15    | ShowFonts.PAK  | 8192   | * 16   | Display.PAK   | 10880  |
| * 17    | TurboBackup.AR | 33024  | * 18   | ViewBoot.PAK  | 8576   |
| * 19    | VirusX32.ARC   | 17664  | * 20   | Warp2.0       | 17792  |
| * 21    | A86K.ZOO       | 68864  | * 22   | MCAD.ARC      | 88704  |
| * 23    | Dsalvl.3.ZOO   | 34304  | * 24   | UUPC.ZOO      | 125952 |
| * 25    | DME.ARC        | 40320  | * 26   | AZComm.ARC    | 53760  |
| * 27    | Xoper.ARC      | 21120  | * 28   | Popinfo.ZOO   | 20224  |
| * 29    | Mfix.ZOO       | 8960   | * 30   | HDSK2.12.ZOO  | 109824 |
| * 31    | Funckey.ZOO    | 23424  | * 32   | SmallLib.ARC  | 6144   |
| * 33    | Lisa.ZOO       | 7168   | * 34   | Menu.ZOO      | 4480   |
| * 35    | Access.ZOO     | 132096 | * 36   | LaserHelp.ARC | 1664   |
| * 37    | Wicon.ZOO      | 22400  | * 38   | Conman.ARC    | 19456  |
| * 39    | SpriteEd.ZOO   | 30976  | * 40   | MachII.ZOO    | 35456  |
| * Hilfe |                |        | * Vorw |               | * Empf |

==>■

### PHONE

**Hang up:** Auflegen

**Alt Service:** Stellt der im folgenden gewählten Nummer einen »long distance dial code« voran. Sinnvoll für Netze in den USA.

Sie kennen nun die meisten Funktionen, die »AZComm« bietet. Schnell werden Ihnen dessen vielseitige Möglichkeiten vertraut sein. Am Schluß möchten wir kurz auf die Mailbox HOST vom Markt & Technik Verlag eingehen (Bild 5). In dieser kommerziellen Box finden Sie unter anderem ein PD-Menü für den Amiga (Bild 6). Sie können Programme einfach und schnell aus der Box abrufen. Sehen Sie sich die Mailbox einmal als Gast an. Die Nummer: 089/460 60 31 (1200 Bit/s, 8N1).

Bild 6. PD-Software offeriert die Mailbox HOST für den Amiga-Fan. Das Abrufen der Programme ist kostenlos.



Von Martin Jobst

**D**er Bereich Musik wurde in den Besprechungen von PD-Software bisher stark vernachlässigt. Dabei bieten Fred Fish & Co. auch hier Programme, die sich hören lassen. Vor allem mit speziellen MIDI-Utilities helfen sie aus. Was hätten Sie denn gern? Einen kleinen Sequenzer, eine Sound-Verwaltung, oder wie wär's mit einer MIDI-Library? Vielleicht suchen Sie zusätzlich ein paar neue Instrumente für Sonix? Kein Problem - PD macht's möglich.

Eine Auswahl dessen, was Public-Domain in Sachen Musik so alles zu bieten hat, stellen wir Ihnen hier vor (siehe auch Tabelle 1). Beginnen wollen wir mit allem, was mit MIDI zusammenhängt.

Warum kompliziert, wenn es auch einfach geht? Dies könnte das Motto des Midi-Sequenzer »D. J. Joe« sein, der sich auf »ACS 80« befindet (Bild 1). Mittels seiner Funktions-Gadgets läßt er sich so einfach bedienen wie ein Kassettenrecorder. Ein Klick auf »REC« genügt, und die Aufzeichnung aller Daten, die bei »MIDI-IN« anliegen, startet. »Play« spielt das Meisterwerk anschließend originalgetreu ab. Besonders für kurze Sequenzen ist dieses Programm bestens geeignet. Es erspart dem Anwender, der etwa nur die Begleitung per MIDI gespielt haben will, Komplikationen mit aufwendigeren Sequenzern und vor allem eine Menge Geld.

Ein weiterer - wenn auch etwas ungewöhnlicher - Sequenzer befindet sich auf der »CC 48 - Midi # 2«. Unter dem Namen »LyrHarp« verbirgt sich dort eine Art elektronischer Harfe (Bild 2). Der Amiga stellt die

**Seine bestechenden Sound-Fähigkeiten haben dem Amiga unter Musikern viele Freunde eingebracht. Das Angebot an Musik- und MIDI-Software wächst ständig.**

## Der Ton macht die Musik

Instrumente. Aber auch mit anderen MIDI-Geräten arbeitet dieses tolle Programm zusammen. Wird anstelle von »Live« die Funktion »Sequenced« gewählt, werden alle Mausbewegungen aufgezeichnet. Probieren Sie dieses Programm aus, Sie werden nicht glauben, wie gut sich eine Maus eignet, Synthesizern Töne zu entlocken.

Weitere einfache Sequenzer-Routinen finden Sie auf Fish 54. »Record« zeichnet alle an MIDI-IN anliegenden Daten auf und legt diese in ein ASCII-File ab. Mit »Play« können die Daten - natürlich auch verändert - abgespielt werden. Diese beiden Programme eignen sich besonders zum Einbau in eigene MIDI-Anwendungen.

Sie besitzen zwar einen Synthesizer, aber kein MIDI-Inter-

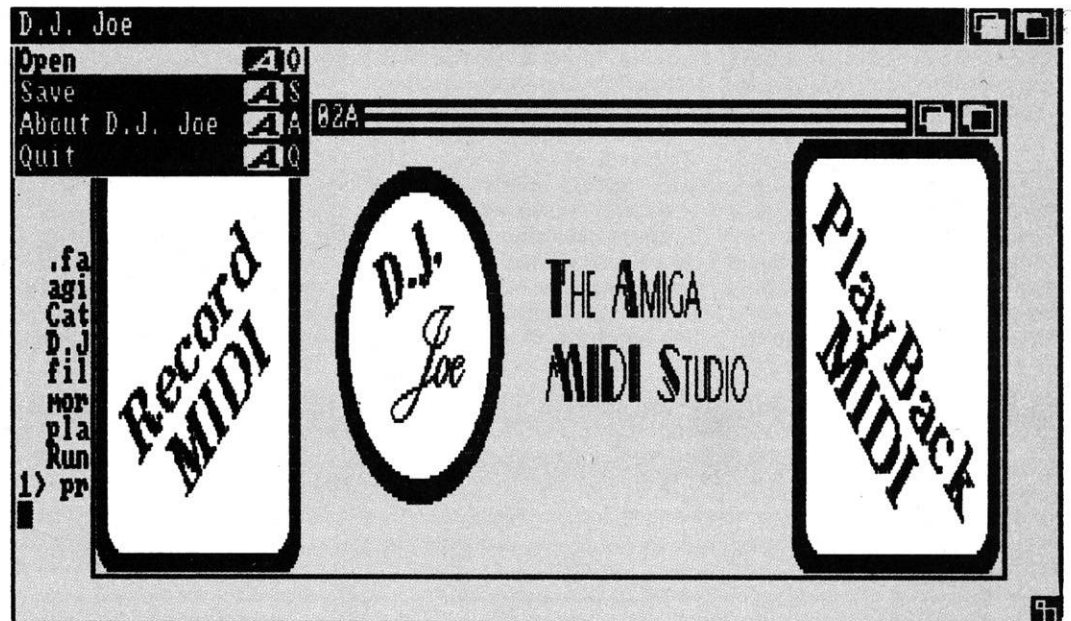


Bild 1. Einfach und funktionell - der MIDI-Sequenzer auf ACS 80 läßt sich leicht bedienen

| Programm                | Diskette                 |
|-------------------------|--------------------------|
| D.J. Joe                | ACS 80                   |
| LyrHarp                 | CC 48                    |
| Record, Play            | Fish 54                  |
| MIDI-Interface          | CC 8, Amicus 13          |
| CZL                     | CC 8                     |
| CZ-101                  | CC 8                     |
| MIDIUtility             | CC 8                     |
| Panl                    | Fish 82, RPD 20, CC 8    |
| DX-VoiceFiler           | Fish 82                  |
| DX-VoiceSorter          | Fish 149                 |
| MIDI-Library            | Fish 101                 |
| MIDI-Utilities          | Fish 101                 |
| div. Musikstücke/Sounds | Musik 1 bis 23           |
| InstantMusic-Demo       | RPD 14                   |
| AudioMaster-Demo        | RPD 54                   |
| PerfectSound            | Fish 50, RPD 23, CC 49   |
| Sonix-Files             | CC 47                    |
| DMCS-Files              | CC 5, CC 6, CC 45, CC 46 |
| Synthmania              | Fish 153                 |
| Visionsounds            | Fish 167                 |
| div. Musikstücke/Sounds | Tornados 29, Taifun 47   |

Tabelle 1. Übersicht über alle beschriebenen Programme

Saiten einer Harfe, einer Gitarre oder beispielsweise eines Banjos samt Tabulaturen grafisch dar. Mit der Maus können nun die Saiten gezupft werden und lassen ein angeschlossenes MIDI-Gerät erklingen. Per Tastendruck können Sie zudem die Saiten gleich in bestimmten Tonleitern ertönen lassen. Spezielle Effekt-Funktionen machen den Klang runder. Bei senkrechter Bewegung der Maus werden die entsprechenden Saiten gezupft. Wird die Maus zusätzlich noch nach rechts oder links bewegt, läßt sich damit nach Wunsch die Lautstärke oder das »Pitch-Bending« beeinflussen. »Lyr-Harp« arbeitet mit Yamaha DX-Synthesizern zusammen. Das Programm erlaubt bei diesen auch eine direkte Anwahl der

face? Für die richtige Verbindung zum Amiga steht Hilfe bereit: Auf der »CC 8 - Midi # 1« und auf Amicus 13 befinden sich als IFF-Bild vollständige Schaltpläne für ein MIDI-Interface (Bild 3). Das Interface besitzt einen Eingang, drei Ausgänge und eine Buchse für MIDI-Through. Die Anleitung für den Aufbau der Schnittstelle liefert das zugehörige Readme-File.

Wenn Sie Besitzer eines Synthesizers der Casio CZ-Serie sind, finden Sie auf »CC 8 - Midi # 1« einen besonderen Leckerbissen. Diese Diskette enthält fast ausschließlich Programme und Files für diese Synthesizer: »CZL« ist eine Soundverwaltung mit vielen Demos. Damit können Sie die Sounds des Synthesizers auf



Amiga-Disketten verwalten. Außerdem können mit »CZL« Einstellungen der CZ-Synthesizer direkt vom Amiga aus vorgenommen werden. In den Directories »CZ-101« und »CZ-Instruments« finden sich außerdem noch Sound-Banks und jede Menge neuer Instrumente für die Casio-Synthesizer.

Auch für andere Synthesizer geeignet ist das »MidiUtility«, das sich ebenfalls auf dieser Diskette befindet. In erster Linie ist das Programm ein Sequenzer, mit dem MIDI-Daten von beliebigen MIDI-fähigen Synthesizern aufgezeichnet, abgespielt und auf Diskette gespeichert werden. Die weiteren Funktionen dieses Utility sind

### Ein Editor für alle MIDI-Geräte

aber wieder für ein Casio-Instrument bestimmt: für den CZ-101. So lassen sich sowohl alle internen Daten des Synthesizers (also die frei definierbaren Soundbanks, Parameter etc.) als auch Daten von RAM-Cartridges zum Amiga übertragen. Diese sind zusätzlich auf Diskette zu verwalten.

Eine Menge weiterer Sounds für die Synthesizer von Casio finden Sie auf der »CC 48 - Midi#2«. Diese bietet außer-

dem einige Musikstücke für das »DeLuxe Music Construction-Set«, welche speziell für diese Synthesizer angepaßt wurden. So etwa das Stück »Memory« aus dem Musical »Cats« oder eine Toccata von Bach.

Das Programm »Panl« (CC 8, Fish 82, RPD 20) ist ein einfacher Editor, der an alle MIDI-Geräte angepaßt werden kann. Auf einem »Panel« können über 16 Schieberegler und acht Schalter beliebige Parameter verändert und Funktionen geschaltet werden (Bild 4). Man kann also den Synthesizer bequem mit der Maus bedienen und muß sich nicht mit dem - meist recht dürftigen - eingebauten Display herumschlagen. Besonders nützlich ist dieses Utility für jene Anwender, die ein MIDI-Gerät besitzen, für welches noch kein spezieller Editor erhältlich ist. Die Anpassung ist relativ einfach vorzunehmen: Man wählt den Setup-Modus, klickt einen Schieberegler/Schalter an, bestimmt dessen Funktion und gibt die Parameternummer des Midi-Geräts für diese Option an. Hat man dies mit allen verfügbaren Reglern gemacht, ist der Sound-Editor »fertig« und man kann die Einstellungen auf Diskette speichern.

Der »DX-Synth/DX-VoiceFiler« von der Fish 82 unterstützt die

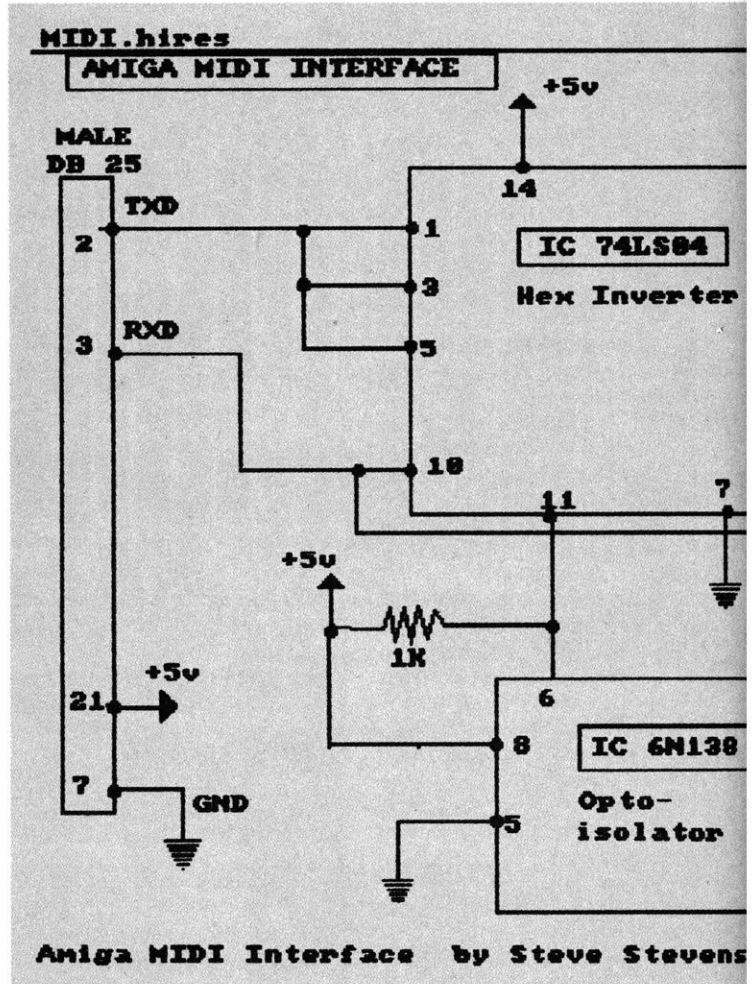


Bild 3. Der Schaltplan für ein MIDI-Interface - zu finden auf

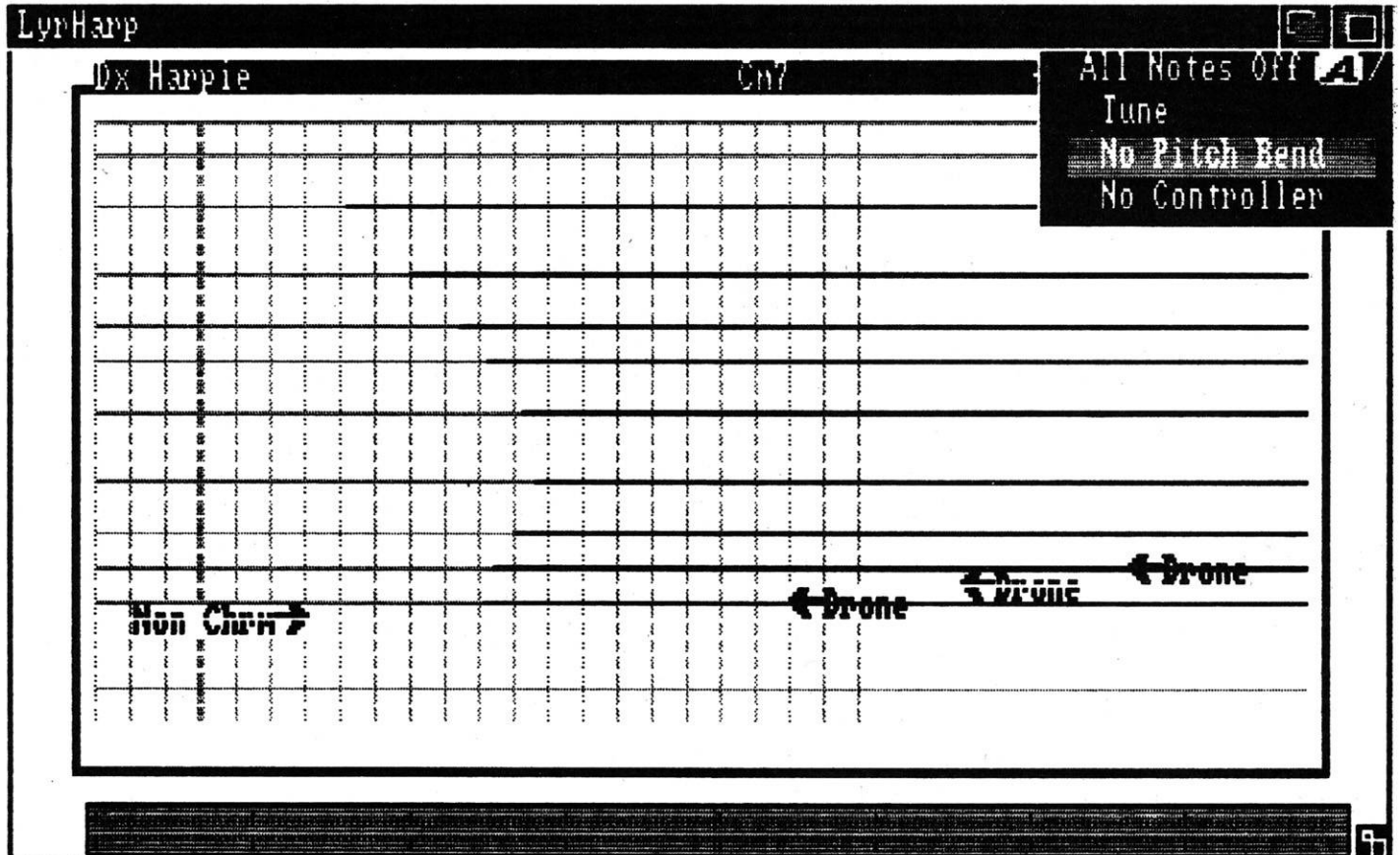
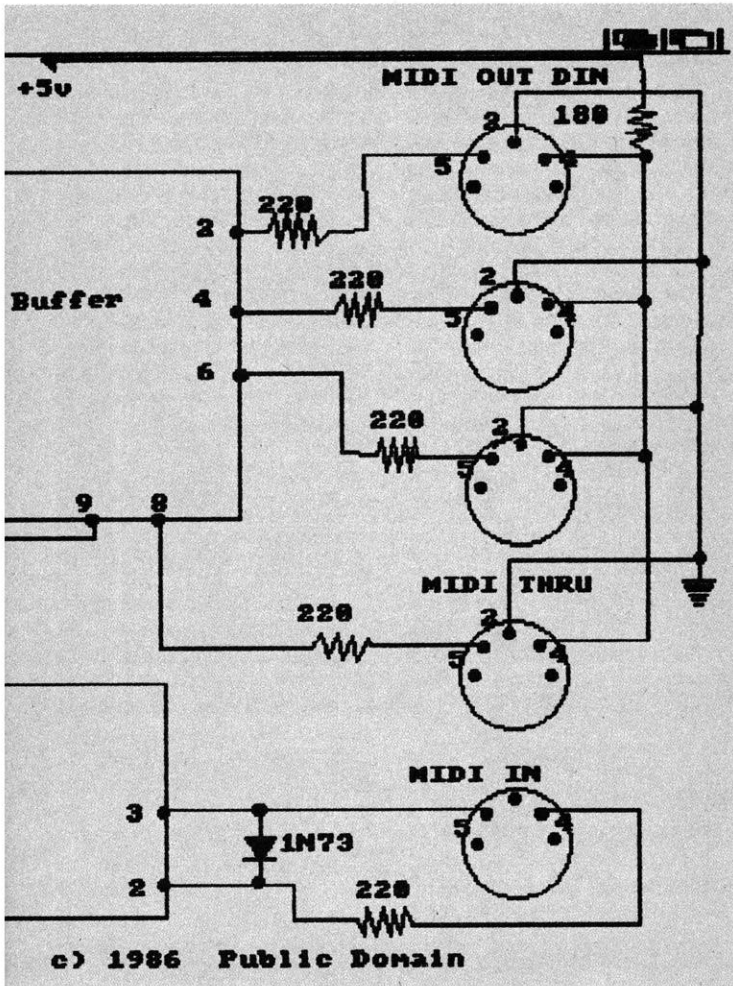


Bild 2. Harfe spielen mit der Maus - »LyrHarp« macht's möglich. Der Amiga stellt die Saiten dar.



# ANWENDUNGEN



CC 8 und Amicus 13. Bauen Sie sich die Brücke zur Musik.

Synthesizer Yamaha DX7, DX21 und DX100. Damit lassen sich die Klänge dieser weit verbreiteten Geräte auf Amiga-Disketten verwalten. Neben einigen Beispiel-Klängen gibt es eine Option für das Erzeugen von Sounds per Zufallsgenerator. Ein Zusatzprogramm zum Voice-Filter ist der »DX-VoiceSorter« von Fish 149. Dieses hilft bei der Verwaltung der Klänge und erlaubt das Zusammenfassen mehrerer Sound-Files zu einem einzelnen.

## Wie wär's mit einer Library?

Für Programmierer von MIDI-Software ist die Fish 101 von besonderem Interesse. Sie finden auf dieser eine komplette MIDI-Library mit jeder Menge Funktionen sowie viele nützliche Utilities für die Arbeit mit MIDI-Daten. Die vollständig dokumentierte Library bietet unter anderem folgende Funktionen: Übertragen von MIDI-Daten über den seriellen Port, Verarbeitung von MIDI-Message, auch »System-Exclusive«. Die Library setzt die anfallenden Datenströme also gleich in entsprechende »Midi-Events« um, welche vom Programm weiterverarbeitet werden können. Außerdem erlaubt die Library den

störungsfreien Zugriff von mehreren MIDI-Applikationen zugleich auf den seriellen Port.

Bei den MIDI-Utilities auf Fish 101, die nur zusammen mit der MIDI-Library laufen, handelt es sich um folgende Programme: »HexTransmit« bringt MIDI-Meldungen in das Hexadezimal-Format, der »MidiMonitor« zeigt alle eingehenden »Midi-Message« an, »Receive System Exclusive« empfängt systemspezifische Kommandos von MIDI-Geräten und »Stat« liefert einen Status-Report.

Soweit zu den uns bekannten MIDI-Programmen im Meer der Public Domain-Disketten. Wenn Ihnen weitere MIDI-Software aus dem PD-Pool bekannt ist, schreiben Sie uns. Wir werden gern in einem unserer nächsten PD-Artikel darauf eingehen.

Auch Musikfreunde, die mit MIDI »nichts am Hut haben«, finden auf PD-Disketten viele nützliche Hilfen. Vor allem mit digitalisierten Sounds und ganzen Musikstücken wird der Liebhaber geradezu verwöhnt. Empfehlenswert ist in diesem Zusammenhang eine eigene Serie, die unter dem Namen »Musik« läuft und bis jetzt 23 Disketten umfaßt. Vertrieben wird dieses Label unter anderem von der österreichischen Firma S.A.F.E. (siehe Anbieter-

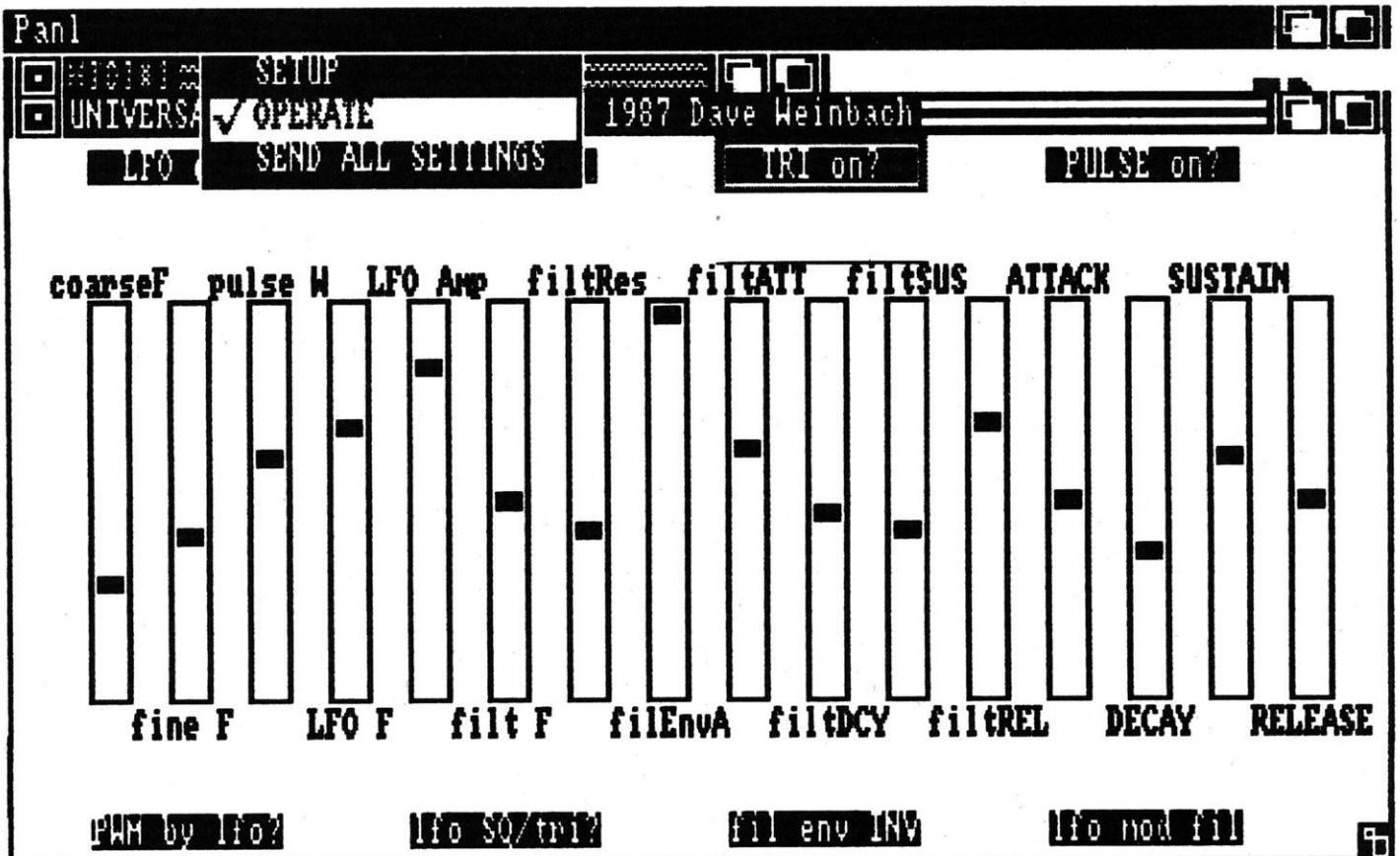


Bild 4. Erstellen Sie mit »Panl« einen Editor für Ihren Synthesizer. Über 16 Schieberegler warten auf Veränderung.



# ANWENDUNGEN

verzeichnis auf dieser Seite). Diese Reihe enthält vor allem digitalisierte Musikstücke. So befindet sich auf den Disketten 1 bis 4 das Thema der TV-Serie »Miami Vice«. Abgesehen davon, daß dabei zwei Laufwerke empfehlenswert sind, ist das digitale Werk absolut hörensenswert.

Fred Fish trumpft auf Diskette 149 mit »AnimalSounds« auf. Hühnergegacker hört man dort ebenso wie Hundebellen oder Vogelgezitscher. Ein Muß für Tierfreunde, die diese Klänge für eigene Video-Produktionen oder ähnliches verwenden wollen! Setzen Sie die Sounds in eigenen Programmen ein – die Wirkung auf den Zuhörer ist immer ein voller Erfolg.

Musikalischer geht es auf der RPD 14 zu. Dort finden Sie eine Demo-Version des Programms »Instant Music«. Damit sind auf einfache Weise Musikstücke und Rhythmen zu erstellen. Instant Music hilft dem angehenden Musiker mit Auto-Chord-Funktionen, die von Bontempi-Orgeln her bekannt sind.

Wer einen Sounddigitizer sein Eigen nennt, dem ist mit »AudioMaster« auf RPD 54 weitergeholfen. Mit AudioMaster – es handelt sich allerdings nur um eine Demo-Version – lassen sich digitalisierte Klänge mit der Maus einfach bearbeiten. Das Schneiden oder das Verän-

dern der Tonhöhe von Samples ist genauso möglich, wie das Versehen der Klänge mit einem Echo-Effekt. Da es sich jedoch nur um eine Demo-Version handelt, besitzt das Programm in der PD-Version keine Save-Funktion. Die kommerzielle Version des SoundMasters kostet zirka 80 Mark. Wenn Ihnen die Demo-Version zusagt, ist diese Summe sicher vertretbar.

## PD ist Musik in meinen Ohren

Wenn Sie jedoch keine zu hohen Ansprüche stellen, finden Sie auf Fish 50, RPD 24 oder CC 49 Versionen von »PerfectSound« (Bild 5). Damit lassen sich zwar keine Echo-Effekte erzielen, und die Bedienung ist auch nicht ganz so komfortabel wie bei AudioMaster, ansonsten steht das Programm aber diesem in nichts nach. Als besonderes »Zuckerl« erlaubt es mit einem entsprechenden Sampler sogar die Digitalisierung und Bearbeitung von Stereo-Samples. Auf CC 49 befinden sich außerdem noch eine ganze Menge Beispiel-Klänge zu PerfectSound.

Wer für »Sonix« neue Instrumente und Stücke sucht, sollte ein Auge auf Disk 47 von Chiron Conceptions werfen. Dort fin-

den Sie neben einer Vielzahl von Instrumenten auch einige »Score«-Files. »Fall over« und »Thriller« sind dabei besonders hörensenswert. Auch für das »DeLuxe Music Construction-Set« (DMCS) von Electronic Arts sind wohlklingende Beispiele vorhanden. Auf CC 5, CC 6, CC 45 und CC 46 befinden sich einige Musikstücke für das DMCS. Dabei möchten wir Ihnen besonders das Stück »Pictures at an Exhibition« von CC 46 ans Herz legen. Es enthält die vollständige Klavierversion (fast 1000 Takte!) des berühmten Ton-Bilder-Zyklus von Modest Mussorgsky. Allerdings benötigt dieses Werk mehr als 1 MByte Speicher. Ohne Übertreibung: ein absolutes Muß für Freunde klassischer Musik. Fans der schwedischen Gruppe Abba – sofern es noch welche gibt – finden auf CC 6 eine Version des einstigen Hits »Super Trouper«.

## Die Welle macht den Ton

Was Sie schon immer über Wellenformen wissen wollten – oder warum schwingt ein Ton? Ein kleiner Waveform-Workshop als Basic-Programm geht dem Klang auf die Spur. Optisch und akustisch wird Ihnen

der Unterschied zwischen den verschiedenen Wellenformen nahegelegt. Sie können diese auch selber zeichnen und erfahren, außerdem, wie Lärm (»Noise«) optisch aussieht. Zu finden ist dieser Grundlagenkurs auf Disk 19 von Amicus.

Zuletzt sei noch auf einige sehr gute, teilweise digitalisierte Musikstücke verwiesen. »Synthemia« von Fish 153 reißt ebenso vom Hocker wie die Stücke »VisionSound 1 bis 4« auf Fish 167. Ebenfalls nicht zu verachten sind die Sounds auf Taifun 46 und 47.

Haben Sie Lust auf die hörenswerten Demos bekommen? Dann wünschen wir Ihnen viel Spaß beim Ausprobieren. Halten Sie die Ohren offen, es werden sicher auch in Zukunft zahlreiche interessante PD-Disketten für diesen Bereich erscheinen. Stöbern Sie doch in der langen Liste ab Seite 80 in diesem Sonderheft. Dort sind die Programme in einzelne Rubriken aufgeteilt, Sie finden also schnell die gewünschte Software.

Perfekte Klänge mit dem Amiga sind dann auch für Sie kein Fremdwort mehr. rs

PD-Anbieter in Österreich:

SAFE, Bernd Küppers, Felberstraße 7, A-5730 Mittersill, Tel. 06562/282

m.a.r.-Computershop, Weidengasse 41, A-1100 Wien, Tel. 0222/621535

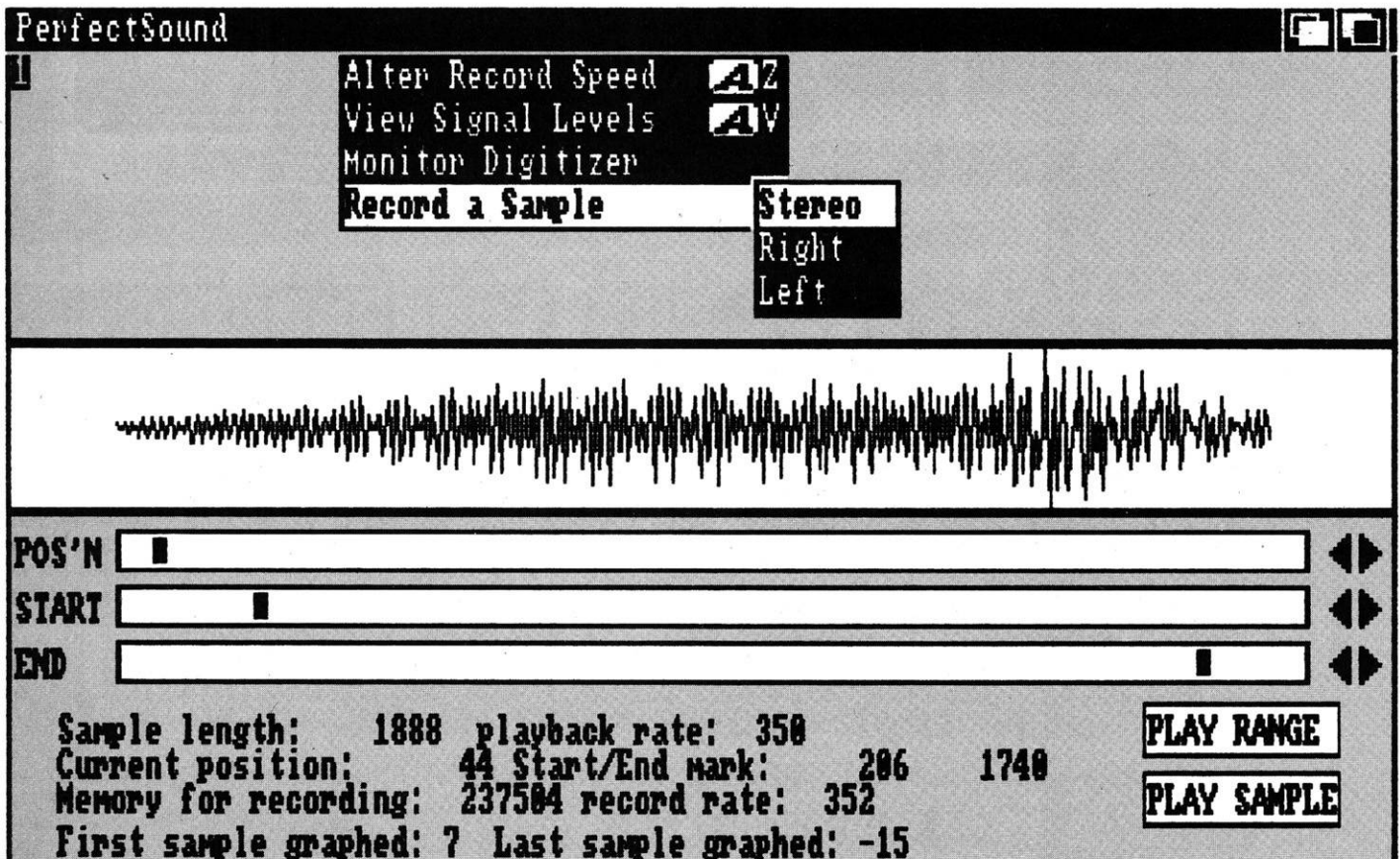
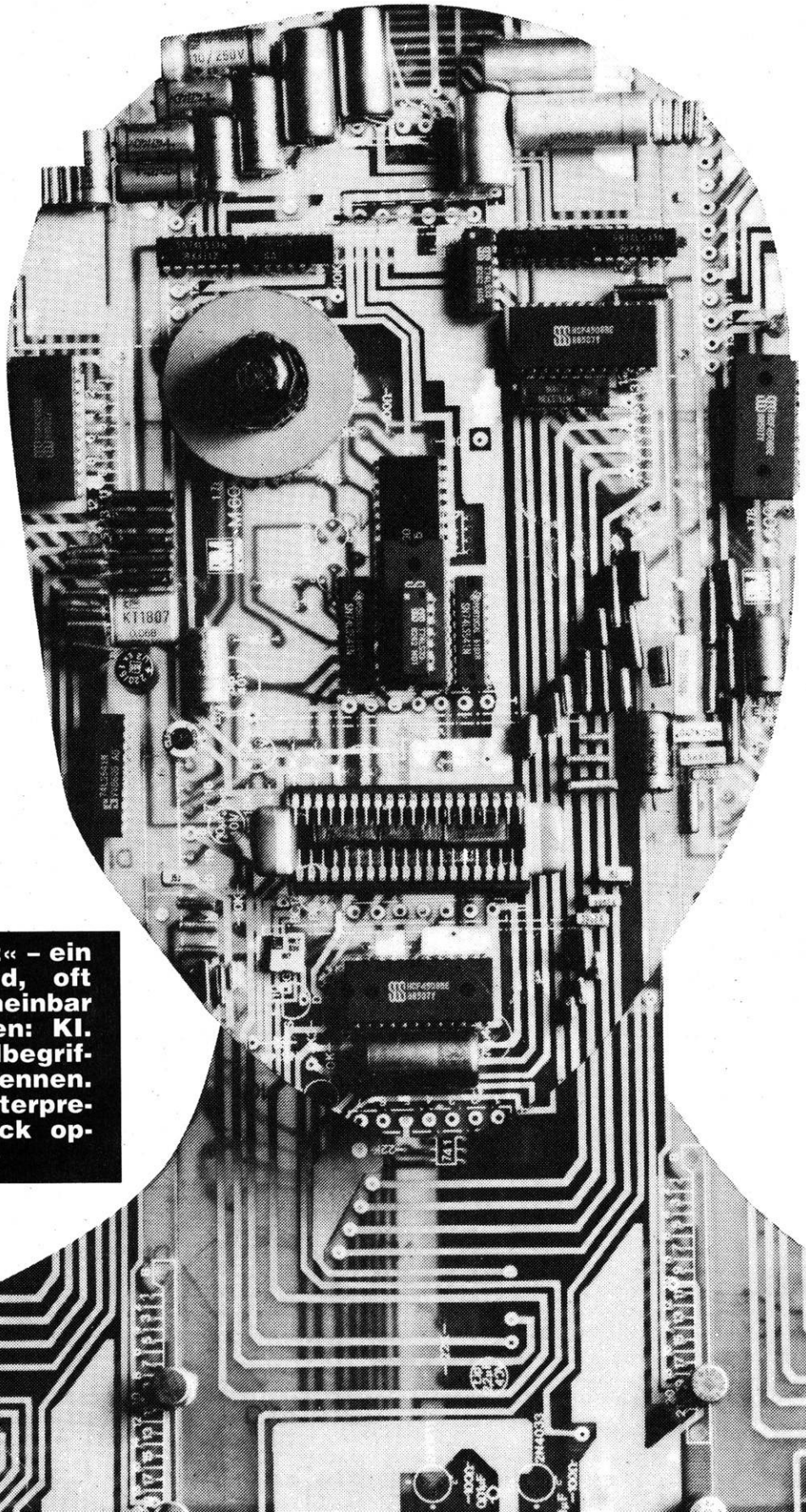


Bild 5. Perfekt digitalisieren mit »PerfectSound«. Das Programm bietet fast die Leistungen von AudioMaster.



# KÜNSTLICHE

# INTELLIGENZ



»Künstliche Intelligenz« – ein Reizwort. Faszinierend, oft mißverstanden und scheinbar schwer zu beschreiben: KI. Lernen Sie die Grundbegriffe des KI-Konzepts kennen. »X-LISP« ist ein PD-Interpreter, der für diesen Zweck optimal geeignet ist.



**C**omputer sind doof«. Stimmt diese Aussage eigentlich? Kann beispielsweise der Amiga, richtig programmiert, denken? Nein. Aber der Begriff »Künstliche Intelligenz« (KI) verleitet doch zu der Annahme, Computer könnten zum Denken animiert werden. Das ist (leider, glücklicherweise?) auch falsch. Richtig ist, daß verschiedene Programmiersprachen, wie Prolog, Scheme oder Lisp, den Computer befähigen, Querschlüsse aus vorgegebenen Fakten zu ziehen. Daraus kann das »Elektronengehirn« eventuell Entscheidungshilfen ableiten. Die endgültige Entscheidung bleibt auch bei diesem Verfahren immer noch beim Anwender. Denn was ist, wenn sich in das Programm ein Fehler eingeschlichen hat? Trotzdem sind die KI-Sprachen als Aufbruch zu bezeichnen. Ein Beispiel für deren Anwendung sind sogenannte Expertensysteme.

Auf der Fish-Disk Nummer 37 befindet sich der Interpreter XLISP (Version 1.7). Dieser ist eine leistungsfähige Implementation der Sprache LISP, die zum Bereich der Künstlichen Intelligenz (kurz: KI) gerechnet wird. Wir möchten hier nicht ausführlich alle Befehle des Interpreters vorstellen. Diese sind in der umfangreichen Dokumentation (auf der PD-Diskette) erklärt. Dieser Artikel behandelt vielmehr das Konzept und die vielen Vorteile der Sprache LISP. Nebenbei erfahren Sie, welche »Philosophie« der KI zugrundeliegt.

## Ein neuer Anfang

Wer in XLISP programmieren will, kann erst einmal das meiste vergessen, was er bisher an Programmieretechnik kennengelernt hat. Dafür wird ein umfassender Einstieg in die Welt des objektorientierten Programmierens geboten. Diese Annäherung an die Struktur der menschlichen Denkweise ist zwar ungewohnt, aber nach kurzer Einarbeitung faszinierend.

Drei grundlegende Dinge müssen hier erst einmal klargelegt werden: Erstens muß man sich zum Teil völlig neue Denkschemata zur Programmerstellung angewöhnen. Zweitens ist die Ausführungsgeschwindigkeit nicht im mindesten konkurrenzfähig mit den übrigen Spra-

chen für den Amiga. Und zudem muß man auch noch auf Spezialitäten wie Grafik oder Sound völlig verzichten.

Das oben erwähnte Prinzip der objektorientierten Programmierung ist übrigens in keiner Weise neu, sondern reicht in die Anfänge der sechziger Jahre zurück. Entsprechend kann auch die Sprache LISP auf eine lange Vergangenheit zurückblicken. Doch was ist eigentlich das Wesen der objektorientierten Programmierung, die gerade in den letzten Jahren durch die Verwirklichbarkeit auf Mikrocomputern stark an Popularität gewonnen hat?

Diese Frage soll an einem relativ einfachen Beispiel beantwortet werden. In konventionellen Programmiersystemen existiert die Operation »Addition«. Dem System wird nun folgender Befehl gegeben: »operiere mit den Integerzahlen 2 und 3«. Daraufhin verzweigt die Operation »Addition« in die Unteroperation »Integer-Addition« und wertet die Botschaft damit aus.

Entsprechend verhält es sich mit der Botschaft »operiere mit den Realzahlen 3.141 und 2.718«. Es wird in die Unteroperation »Real-Addition« verzweigt. Es wird also für jede Klasse (Integerzahlen, Realzahlen, Strings und so weiter) der Operation »Addition« jeweils eine spezielle Operationsvorschrift zugeordnet.

Beim objektorientierten Ansatz verhält es sich genau andersherum. Wir definieren ein Objekt »Integerzahlen«, indem wir ihr alle Operationsvorschriften für Addition, Multiplikation und so weiter zuweisen. Die Botschaft »operiere mit den Integerzahlen 2 und 3« wird nun also umgewandelt in »operiere in der Klasse Integer mit Addition, 2 und 3«.

Somit ist ein Objekt also eine Gruppe von Informationen zusammen mit den Methoden derer Verknüpfung. Dies hat radikale Auswirkungen bei der Erweiterung von Datenverarbeitungssystemen. Als Beispiel soll ein Mathematikprogramm dienen, welches um die Verarbeitung von komplexen Zahlen erweitert werden soll. Auf konventionellen Systemen müßte nun jede einzelne Verarbeitungsroutine um eine entsprechende Unteroutine erweitert werden. Beim objektorientierten Ansatz dagegen fügt man das Objekt »komplexe Zahl« mit sämtlichen Verarbeitungsvorschriften einfach hinzu.

Doch der Einsteiger bemerkt von diesem neuen programmtechnischen Ansatz erst einmal

recht wenig. Wer mit einer Sprache wie Basic oder C vertraut ist und nun LISP lernen will, der hat mit ganz anderen Problemen zu kämpfen.

Dieses Problem hat auch einen Namen: »Liste«. In LISP funktioniert alles, aber auch wirklich alles mit Listen. Programme sind Listen, Daten sind Listen und so weiter. Doch wie sieht eine solche Liste überhaupt aus?

## Listen als Basis aller Operationen

Die einfachste Liste überhaupt hat diese Form:

( )

Alle Listen beginnen mit einer öffnenden runden Klammer und enden mit einer schließenden. Dazwischen können unterschiedlich viele sogenannte »Elemente« stehen – bei der Liste »()« ist deren Anzahl Null. Es

könnten aber genauso gut auch 1, 10 oder 300 sein.

Das erste Element einer Liste ist normalerweise der Name einer Prozedur. Genau genommen sind in LISP fast alle Prozeduren Funktionen, da sie alle einen bestimmten Wert »zurückgeben« (siehe unten).

Die darauffolgenden Elemente der Liste sind dann meistens Daten, welche die Prozedur als Parameter benötigt. Dies können nun Zahlen oder Zeichenketten sein, aber wiederum auch Listen.

Eine solche Schachtelung – wenn eine Liste weitere Listen als Elemente enthält – scheint auf den ersten Blick verwirrend. Tatsächlich aber ist es eine eindeutige und übersichtliche Art der Strukturierung. Deshalb wird sie ja auch in der Mathematik verwendet. Ein Beispiel:

$$(5*(2+5*(3-1)))=60$$

Soll dieser Term berechnet werden, so geht man von der in-

|            |             |
|------------|-------------|
| ABS        | FLATSIZE    |
| ADDRESS-OF | FLOAT       |
| ALLOC      | FUNCAL      |
| AND        | FUNCTION    |
| APEND      | GC          |
| APPLY      | GENSYM      |
| AREF       | GET         |
| ASSOC      | GO          |
| ATOM       | HASH        |
| BACKQUOTE  | IF          |
| BAKTRACE   | INTERN      |
| BOUNDP     | LAMBDA      |
| BREAK      | LAST        |
| CAR        | LENGTH      |
| CASE       | LET         |
| CATCH      | LET*        |
| CDR        | LIST        |
| CERROR     | LISTP       |
| CHAR       | LOAD        |
| CLEAN-UP   | LOGAND      |
| CLOSE      | LOGIOR      |
| COND       | LOGNOT      |
| CONS       | LOGXOR      |
| CONSP      | MAKE-ARRAY  |
| CONTINUE   | MAKE-SYMBOL |
| COS        | MAPC        |
| DEFMACRO   | MAPCAR      |
| DEFUN      | MAPL        |
| DELETE     | MAPLIST     |
| DO         | MAX         |
| DOLIST     | MEM         |
| DOTIMES    | MEMBER      |
| DO*        | MIN         |
| ERROR      | MINUSP      |
| ERRSET     | NCONC       |
| EQ         | NOT         |
| EQL        | NTH         |
| EQUAL      | NTHCDR      |
| EVAL       | NUMBERP     |
| EVALHOOK   | NULL        |
| EVENP      | ODDP        |
| EXIT       | OPENI       |
| EXP        | OPENO       |
| EXPAND     | OR          |
| EXPT       | PEEK        |
| FLATC      | PEEK-CHAR   |

Tabelle 1. Alle Schlüsselwörter von »XLISP V1.7« auf einen Blick



nersten Klammer aus und berechnet ihren Wert (hier 3-1, also 2). Mit diesem Ergebnis kann man dann die nächste Klammer ausrechnen:

$2+5*2=12$

Erst zuletzt kommt dann die äußerste Klammer zum Zug und man berechnet

$5*12=60$

Nicht anders tut dies auch ein LISP-Interpreter. Er würde diesen Term als Liste verstehen. Die Rechenzeichen wären Funktionsaufrufe, Zahlen, die dazugehören, Argumente.

Einen Unterschied gibt es allerdings zu dem obenstehenden Term. Schon oben wurde erwähnt, daß das erste Element jeder Liste immer der Prozedurname ist. Die Argumente folgen dann. Im mathematischen Term dagegen ist das erste Element eine Zahl: »-5«.

Daher werden solche Rechnungen in LISP anders ge-

schrieben. Die Liste beginnt immer mit dem Prozedurnamen, und dann kommen die Argumente. Zudem werden zwischen alle Elemente Leerzeichen eingefügt, um jene voneinander trennen zu können.

In LISP würde der obenstehende Term also folgendermaßen aussehen:

```
(* 5 (+ 2 (* 5
(- 3 1))))
```

Sehen Sie sich diese Liste einmal ganz genau an. Sie werden feststellen, daß diese Schreibweise in Wirklichkeit wesentlich übersichtlicher ist – man muß sich nur daran gewöhnen. Übrigens wird diese Schreibweise in der Mathematik als »umgekehrte polnische Notation« bezeichnet.

Wenn man eine solche Liste unter XLISP eingibt, so wird sie automatisch vom Interpreter »evaluiert«. Evaluieren bedeutet nichts anderes als auswerten, also den Wert des Ausdrucks berechnen.

Bei der Evaluation sucht der Interpreter zunächst die innerste Klammerebene. Diese Ebene nennt man »Top Level«. Das erste Top-Level-Element ist also (- 3 1). Der Interpreter weiß nun, es soll subtrahiert werden (»-«). Diese Prozedur führt er dann auch mit den darauf folgenden Argumenten aus.

Das Ergebnis setzt er statt (- 3 1) in die Liste ein. Damit hätte er also die neue Liste:

```
(* 5 (+ 2(* 5 2)))
```

In dieser sucht er sich dann wieder das Top-Level-Element, berechnet seinen Wert, setzt ihn ein, und so weiter. Dieser Vorgang wiederholt sich, bis nur noch eine Klammer übrig ist. Das Ergebnis dieser Berechnung gibt der Interpreter dann als Evaluation zurück (also erscheint am Bildschirm das Ergebnis 60).

## Prozeduren: CAR, CDR, CONS

Typisch für XLISP sind aber nicht Rechen-Prozeduren, sondern solche, die mit Listen arbeiten. Dies sind reine Datenlisten, die auf verschiedene Weise manipuliert werden können.

Die drei wichtigsten Prozeduren heißen CAR, CDR und CONS. Sie sind grundlegend für die meisten Listen-Manipulationen. CAR übergibt als Ergebnis das erste Element einer Liste. Dies sieht so aus:

```
(CAR '(A B C))
```

Diese Liste besteht also aus zwei Elementen. Das erste Ele-

ment (CAR) ist der Name der Prozedur. Der Interpreter weiß also, daß er aus der als nächstes folgenden Liste das erste Element herausziehen soll.

Nun werden Sie sich wahrscheinlich über das Hochkomma vor der Liste (A B C) wundern. Was soll das? Oben haben wir festgestellt, daß der Interpreter zuerst das Top-Level-Element einer Liste bestimmt und dieses evaluiert.

Doch wie sieht es in unserem Beispiel aus? Das Top-Level-Element ist natürlich (A B C). Allerdings läßt es sich nicht evaluieren, denn »A« ist kein gültiger Name einer Prozedur. Der Interpreter würde uns also eine Fehlermeldung ausgeben.

Das Hochkomma (man erreicht es übrigens mit <Alt + ä>) ist in diesem Fall ein sogenannter Evaluationsschutz. Es hindert den Interpreter daran, die nun folgende Liste zu evaluieren. Nur so kann das Beispiel funktionieren.

Den Evaluationsschutz nennt man auch Quotierung. Deshalb kann man auch wahlweise anstatt des Hochkommats QUOTE schreiben. Das Beispiel zu CAR könnte also auch so aussehen:  
(CAR QUOTE(A B C))

Beachten Sie, daß zwischen QUOTE und der Liste (A B C) kein Leerzeichen stehen darf, sonst wären es ja zwei verschiedene Elemente.

Ob man jetzt QUOTE oder das Hochkomma benutzt, das Ergebnis ist in beiden Fällen das gleiche: Der Computer gibt den Buchstaben A aus, das erste Element der Liste.

Die Prozedur CDR ist genau das Gegenteil von CAR. Als Ergebnis bekommt man hier die folgende Liste ohne das erste Element. In unserem Beispiel müßte demnach der Übergabewert (A B C) ohne das A, also (B C) sein. Und dies ist auch tatsächlich der Fall, wenn Sie eingeben:

```
(CDR '(A B C))
```

Die dritte Prozedur im Bunde heißt CONS. Hier wird nichts aus der Liste herausgeholt, sondern ein Element eingefügt. Daher benötigt CONS auch zwei Argumente: das einzufügende Element selbst, sowie die Liste, in die das Element eingefügt werden soll. Nach der Ausführung von CONS ist das angegebene Element das erste der neuen Liste.

```
(CONS 'A '(B C))
```

Dieses Beispiel fügt also das Element A in die Liste (B C) an erster Stelle ein. Das Ergebnis ist also wieder (A B C).

Zum Schluß dieses Abschnittes wollen wir noch ein etwas komplizierteres Beispiel ansehen, welches alle drei Prozeduren auf einmal benutzt.

Angenommen, wir haben eine Liste (A B C). Wenn wir nun mit CAR das erste Element A herausziehen, und in den CDR der Liste (das ist ja (B C)) einfügen, so erhalten wir am Schluß wieder die ursprüngliche Liste!

## Schutz vor Evaluation

Probieren wir es gleich einmal aus:

```
(CONS (CAR '(A B C))
(CDR '(A B C)))
```

Der Interpreter sucht sich zunächst das Top-Level-Element der Liste und findet zwei. Diese beiden Elemente (die Datenlisten) sind allerdings evaluationsgeschützt. Also läßt er sie, wie sie sind, und bearbeitet den nächsten Level.

Dies bedeutet, daß er den CAR und den CDR von (A B C) feststellt. Die Ergebnisse setzt er dann wieder in die Liste ein (natürlich quotiert, um eine nochmalige Evaluation zu verhindern). Die Liste sieht danach also so aus:

```
(CONS 'A '(B C))
```

Das Ergebnis dieser Liste ist dann natürlich (A B C) – wie wir es auch vorgehabt haben. Sie sehen, man muß sich schon etwas an diesen Programmierstil gewöhnen. Allerdings sehen die Listen oft »wilder« aus, als sie sind.

Natürlich kann man auch unter LISP Variablen verwenden. Diese können – wie sollte es auch anders sein – neben Zahlenwerten auch Listen darstellen. Somit können wir uns das Leben sehr vereinfachen.

Die dazu nötige Prozedur heißt SETQ. Dies ist die Abkürzung von »set equal with«, also »setze gleich mit«. Was wird hier gleich gesetzt? Das erste Element nach SETQ ist der Name der Variablen. Dieser kann beliebig gewählt werden, darf aber keinem XLISP-Schlüsselwort entsprechen (alle XLISP-Schlüsselwörter finden Sie in Tabelle 1).

Das zweite Element nach SETQ ist der Wert, den die Variable annehmen kann; also eine Zahl oder eine Liste (natürlich gibt es noch einige andere Möglichkeiten, die wollen wir hier allerdings nicht beachten). Das zweite Element nach SETQ muß quotiert werden. Hier einige Beispiele:

```
PLUSP
POKE
PRINC
PRINT
PRIN1
PROG
PROG1
PROG2
PROGN
PROG*
PUTPROP
QUOTE
RANDOM
READ
READ-CHAR
READ-LINE
REM
REMOVE
REMPROP
RETURN
REVERSE
RPLACA
RPLACD
SET
SETF
SETQ
SIN
STRCAT
STRING
SQRT
SUBLIS
SUBST
SUBSTR
SYMBOLP
SYMBOL-NAME
SYMBOL-PLIST
SYMBOL-VALUE
TAN
TERPRI
THROW
TOP-LEVEL
TRANSCRIPT
TRUNCATE
TYPE-OF
WRITE-CHAR
ZEROP
```



```
(SETQ EINS 1)
(SETQ ZEHN 10)
(SETQ LISTE1 '(A B C))
(SETQ LISTE2 '(1 2 3))
```

Wenn Sie diese vier Listen eingeben, so hat die Variable EINS den Wert 1, die Variable ZEHN den Wert 10, die Variable LISTE1 den Wert (A B C) und die Variable LISTE2 den Wert (1 2 3). Als Rückgabewert erhält man von SETQ immer den Wert der Variablen, also in unseren Beispielen 1, 10, (A B C) und (1 2 3).

Nun können wir immer statt der Zahlen oder Listen die Namen der Variablen angeben. Diese dürfen nicht quotiert werden. Denn wenn nun der Interpreter auf den Namen einer Variablen stößt, dann setzt er automatisch ihren Wert ein:

```
(+ EINS ZEHN)
```

ergibt: 11

```
(- ZEHN EINS)
```

ergibt: 9

```
(CAR LISTE1)
```

ergibt: A

```
(CDR LISTE2)
```

ergibt: (2 3).

Grundlegend und sehr wichtig ist die Definition eigener Prozeduren unter LISP. Mit der Prozedur DEFUN können Sie beliebige Kombinationen aus Prozeduren unter einem Namen zusammenfassen.

## Selbstdefinierte Prozeduren

Nach DEFUN muß in der Liste zunächst der Name folgen, welche die neudefinierte Funktion tragen soll. Für ihn gilt dasselbe wie für Variablennamen. Das zweite Argument nach DEFUN ist eine Liste. Diese Liste enthält so viele Variablennamen, wie Argumente an die neue Funktion übergeben werden sollen.

Arbeitet die neue Funktion parameterlos (benötigt sie also keine Argumente), ist diese Liste leer. Wird genau ein Argument übergeben, so enthält die Liste einen Namen, bei zwei Argumenten zwei Namen und so weiter. Unter den in dieser Liste genannten Namen sind die übergebenen Argumente in der Prozedur verwendbar.

Nach der Argumenten-Liste folgen eine oder mehrere Listen, welche Prozeduren enthalten. Beim Aufruf der Prozedur werden diese Listen dann evaluiert, und die Prozedur gibt den Evaluationswert der letzten Liste zurück. Am besten probie-

ren wir gleich einmal ein einfaches Beispiel aus:

```
(DEFUN /2 (X) (/ X 2))
```

Wir definieren also eine Prozedur mit dem Namen »/2«. Die Prozedur benötigt genau ein Argument. Wird sie mit mehr Argumenten aufgerufen, so meldet der Interpreter einen »too many arguments error«. Bei weniger Argumenten erhalten Sie einen »too few arguments error«.

## Spezieller Aufruf: Rekursion

Das übergebene Argument soll also X heißen. Diese Variable X wird nun einfach durch zwei geteilt. Diese Liste mit der Prozedur »/« ist zugleich auch die letzte Liste, daher übergibt die Prozedur »/2« ihren Evaluationswert. Probieren Sie es aus:

```
(/2 4)
```

ergibt: 2

```
(/2 10)
```

ergibt: 5

```
(/2 2222)
```

ergibt: 1111.

Ein oft verwendetes Beispiel für eine Funktion mit zwei Argumenten ist die Währungsumrechnung. Wir wollen also einen Betrag und den Umrechnungskurs eingeben. Anschließend soll der Betrag in eine fremde Währung umgerechnet werden. Die Lösung des Problems ist sehr einfach:

```
(DEFUN WAHRUNG
  (BETRAG KURS)
  (* BETRAG KURS))
```

Es müssen lediglich Betrag und Kurs multipliziert werden, und schon ist in die andere Währung umgerechnet worden. Angenommen, man bekäme für eine Mark 715 italienische Lire. Dann wäre der Umrechnungskurs 715. Um beispielsweise 40 Mark in Lire umzurechnen geben Sie ein:

```
(WAHRUNG 40 715)
```

Das Ergebnis ist 28600. Bei einem Kurs von 1 zu 715 erhält man also für 40 Mark 28600 Lire. Natürlich gibt es noch unendlich viele Möglichkeiten, eigene Funktionen unter LISP zu definieren. Allerdings soll dieser Artikel ja nur eine kurze Einführung in LISP bieten, und nicht alles erklären.

Eine interessante Spezialität von LISP soll noch zur Sprache kommen: die Rekursion. Einfach ausgedrückt könnte man rekursive Prozeduren als Prozeduren definieren, die sich selbst aufrufen.

Zur näheren Erklärung wollen wir gleich »medias in res« gehen und uns ein Beispiel ansehen. EXPONENT berechnet das erste Argument in der Potenz des zweiten Argumentes, also X hoch Y. Dies ist ja bekanntlich Y mal X mit sich selbst multipliziert. Folgende Funktion berechnet diesen Wert:

```
(DEFUN EXPONENT (X Y)
  (COND ((ZEROP Y) 1)
        (T (* X
              (EXPONENT X
                        (- Y 1))))))
```

Bitte schreiben Sie für eigene Experimente die dritte bis fünfte Zeile gemeinsam in eine. Sicher werden Sie erst einmal ratlos diese Liste betrachten. Die Funktion heißt EXPONENT und benötigt zwei Parameter, nämlich X und Y.

Angenommen, wir geben nun ein (EXPONENT 2 3), dann hat X also den Wert 2 und Y den Wert 3. Als Ergebnis soll 2 hoch 3 (also 8) berechnet werden.

In der zweiten Zeile prüft zunächst die Prozedur COND, ob die Variable Y gleich null ist (dies funktioniert mit ZEROP). Gleich zu Anfang ist dies nicht der Fall. Also wenden wir uns der dritten bis fünften Zeile zu. Der Buchstabe T zu Anfang ist zum Verständnis nicht weiter wichtig.

Nun sucht der Interpreter erst einmal das Top-Level-Element. Dieses ist (- Y 1). Also zieht er von Y eins ab. Y erhält so den Wert 2. Als nächstes Top-Level-Element finden wir EXPONENT. Also fängt die Prozedur von vorne an. Hier wird zunächst wieder geprüft, ob Y gleich null ist. Da dies nicht der Fall ist, wird wieder die dritte Zeile bearbeitet. Dort wird wiederum Y um eins erniedrigt und EXPONENT aufgerufen.

Dies geschieht so oft, bis Y gleich null ist. Jetzt tritt die zweite Zeile in Aktion. Wenn nämlich Y gleich null ist, dann ist der Rückgabewert der Funktion gleich 1. Die unterste Ebene unserer Schachtelung gibt nun also den Wert eins zurück.

In der nächsthöheren Ebene wird dieser Rückgabewert mit X multipliziert. Das Ergebnis ist X. Dieser Rückgabewert wird wiederum auf der nächsthöheren Ebene mit X multipliziert und so weiter. Wenn auf diese Weise die gesamte Schachtelung wieder zurückverfolgt wurde, wurde X insgesamt Y-mal mit sich selbst multipliziert – wir erhalten den Wert von X hoch Y.

Zugegeben, die Rekursion ist nicht ganz einfach zu verstehen. In unserem Beispiel der

Potenz-Rechnung gibt es sicher einfachere Wege der Berechnung. Viele komplexe Probleme lassen sich aber nur auf dem Weg der Rekursion zufriedenstellend lösen, und gerade hier zeigt LISP seine Stärken.

Noch eine letzte Besonderheit von LISP sollten Sie kennenlernen, um ein Gefühl für diese Sprache zu bekommen. Die Assoziations-Listen lassen sich am besten am Beispiel erklären. Dazu definieren wir zwei Listen, die als Elemente wiederum Listen enthalten:

```
(SETQ LISTE1
  '( (NAME DUCK)
    (VORNAME DONALD)
    (KINDER TICK TRICK TRACK)
    (FREUNDIN DAISY)))
(SETQ LISTE2
  '( (NAME GANS)
    (VORNAME GUSTAV)
    (KINDER KEINE)
    (FREUNDIN FORTUNA)))
```

Nun können wir diese Listen als Assoziations-Liste benutzen. Dadurch bekommt das jeweils erste Element der Top-Level-Elemente die Funktion eines Schlüssels. Mit diesen Schlüsseln können wir uns Daten aus der Assoziations-Liste suchen lassen:

```
(ASSOC 'NAME LISTE1)
ergibt: (NAME DUCK)
(ASSOC 'NAME LISTE2)
ergibt: (NAME GANS)
(ASSOC 'KINDER LISTE1)
ergibt: (KINDER TICK TRICK TRACK).
```

Die Prozedur ASSOC benötigt also zwei Argumente. Das erste ist das Schlüssel-Element, nach dem gesucht werden soll. Das zweite Argument ist eine Assoziations-Liste beziehungsweise ein entsprechender Variablenname.

Nun durchsucht ASSOC die angegebene Assoziations-Liste nach einem Element, dessen CAR den Schlüssel ergibt. Als Rückgabewert erhält man die gefundene Liste.

## Speichern und Laden

Natürlich gibt es wie in allen anderen Sprachen auch die Möglichkeit, Programmtext abzuspeichern und später wieder zu laden. So gehen die wertvollen selbstdefinierten Listen nicht verloren und stehen für weitere Projekte zur Verfügung.

Die einfachste Möglichkeit besteht darin, längere Listen



mit einem Editor (zum Beispiel ED aus dem C-Verzeichnis der Workbench-Diskette oder EMACS von der Extras-Diskette) einzugeben und auf Diskette (oder kurzfristig in die RAM-Disk) zu speichern. Dabei ist lediglich zu beachten, daß der Dateiname das Suffix »LSP« tragen muß. Ihre ersten Versuche könnten Sie zum Beispiel unter dem Namen »DF0:TEST1.LSP« auf Diskette abspeichern.

Unter XLISP wird dieser Programmtext dann folgendermaßen wieder eingeladen:

```
(LOAD "DF0:TEST1.LSP")
```

Als Reaktion lädt der Interpreter die Datei und evaluiert die Listen genauso, als hätte man Sie mit der Hand eingegeben. Wem der Umweg über den Editor zu unpraktisch ist, der

kann auch einen anderen Weg beschreiben:

```
(TRANSCRIPT "DF0:TEST1.LSP")
```

Von nun an werden alle Ein- und Ausgaben in der angegebenen Datei mitprotokolliert. Sie finden in der Datei später alles, was Sie eingetippt haben, und alles, was der Computer er-

## Ausführliche Dokumentation

widert hat. Das Protokoll beendet man, indem man TRANSCRIPT ohne Argumente aufruft:

```
(TRANSCRIPT)
```

Nun wird die Datei wieder geschlossen. Sie wird dann wie

oben besprochen mit LOAD geladen. Wundern Sie sich aber nicht, wenn dabei des öfteren Fehlermeldungen erscheinen. Denn da sowohl Ein- als auch Ausgaben protokolliert werden, versucht der Interpreter, seine eigenen Ausgaben zu evaluieren. Dies muß zwangsläufig zu Fehlern führen.

Hoffentlich haben Sie nun einen kleinen Einblick in die auf Heimcomputern bisher unbekannt Sprache LISP bekommen. Natürlich können im Rahmen dieses Artikels bei weitem nicht alle Spezialitäten aufgezeigt werden. Vielleicht sind Sie nun aber doch neugierig geworden und wollen selbst etwas experimentieren – oder auch LISP vollständig erlernen.

Eine große Hilfe ist dabei die ausführliche Dokumentation,

die der Autor von XLISP V1.7 dem Programmierer mitliefert. Sie befindet sich neben dem Programm im Ordner »XLISP V1.7« und trägt den Namen »XLISP.DOC«. Die Dokumentation umfaßt mächtige 43 Seiten und erklärt den gesamten implementierten Sprachumfang.

Allerdings finden Sie in der Dokumentation lediglich Syntax-Erklärungen und einige Beispiele. Lernen können Sie LISP mit dieser Anleitung allerdings nicht. Wer LISP auf dem Amiga kennenlernen will, der sollte sich mit weiterführender Literatur zu diesem Thema befassen. Wir können Ihnen »LISP« von P. Winston und B. Horn empfehlen (Verlag: Addison-Wesley). Dieses Standardwerk erklärt leicht verständlich die Sprache LISP. rs

# Der Public Domain-Assembler

Von Jan Ippen

Viele Leser stehen vor dem Problem, mit abgedruckten Assembler-Listings nichts anfangen zu können. Der A68k hilft weiter, die meisten Programme können mit diesem übersetzt werden. Aber auch Gelegenheitsprogrammierer werden an diesem Public Domain-Programm ihre Freude haben.

Der A68k ist ein Public Domain-Assembler. Sie finden ihn auf der Fish-Disk 110. Auf der gleichen Diskette ist der Linker »BLINK« enthalten, mit dem assemblierte Files endgültig in die lauffähige Version gebracht werden. Inzwischen ist der Assembler auch in der Mailbox HOST vom Markt & Technik Verlag im Amiga-File-Directory abrufbar.

Der A68k wurde ursprünglich in Modula geschrieben und 1986 in Dr. Dobb's Journal veröffentlicht. Später wurde er in C umgeschrieben und auf den Amiga portiert.

Der Assembler selbst hält sich strikt an die von der Firma Motorola vorgegebenen Richtlinien für 68000-Assembler. Der Baustein 68000 wurde von Motorola entwickelt.

Um ein lauffähiges Programm zu erzeugen, wird neben dem A68k noch ein Linker benötigt. Besonders empfehlenswert ist zum Beispiel »BLINK«. Zu diesem wird noch die Library »amiga.lib« benötigt. Es sollten auch die Assembler-Includes vorhanden sein.

**Das Bonbon auf der Fish-Disk 110 ist der A68k. Der Zwei-Paß-Assembler ist gut geeignet, kleine Projekte zu realisieren. Aber auch die meisten veröffentlichten Assembler-Listings im AMIGA-Magazin können Sie damit übersetzen.**

Nun zum Assembler selbst. Dieser arbeitet nur vom CLI aus und hat keinen integrierten Editor. Zur Programmeingabe muß deswegen ein beliebiger Editor verwendet werden. Der Sourcecode sollte in einer Datei mit der Endung »asm« gespeichert werden. Dadurch sind die entsprechenden Files schnell zu finden.

Bei der Erstellung des Sourcecodes bietet der Assembler eine Vielzahl von Direktiven (Anweisungen), die das Programmieren flexibel und einfach gestalten. Die Direktiven lauten:

**EQU** ordnet einem Symbol einen Wert zu. Das erfolgt in folgender Form:

```
<label>: EQU <Wert>
```

Im Verlauf des Programms kann nun der Wert durch das Label ersetzt werden. Es ist aber zu beachten, daß das Label nur ein Symbol ist, das vom A68k beim Assemblieren durch seinen zugeordneten Wert ersetzt wird. Es ist also ausdrücklich keine Variable! Diese Zuordnung hat trotzdem ihre Bedeutung, wenn man beispielsweise einen Sourcecode leicht änderbar gestalten möchte.

Beispielsweise auf folgende Weise:

```
XResolution: EQU 640
```

Möchte man einen anderen Wert, ändert man diesen nur noch an einer einzigen Stelle im Code.

**EQUR** arbeitet ähnlich wie EQU. Der Unterschied ist dabei, daß hier den Prozessor-Registern ein Name zugeordnet werden kann. Es dürfen allerdings nur die Adreß- und Datenregister verwendet werden. Formal sieht das so aus:

```
<label>: EQUR  
<register>
```

Man könnte beispielsweise schreiben:

```
execbase: EQUR a6
```

Im Verlauf kann man nun anstelle von »a6« den Ausdruck »execbase« schreiben. Der Assembler ersetzt beim Assemblieren dann jeweils »execbase« durch »a6«.

**REG** hat den gleichen Effekt wie EQUR, nur werden hier Registerlisten angegeben. Dies ist im Zusammenhang mit dem »Movem«-Befehl wichtig.

Beispiel:  
reglist: REG d0/d1/d2/  
a1/a3

**DC** definiert eine Konstante. Diese wird im Speicher abgelegt und kann verändert werden (im Gegensatz zu EQU). Die Form ist

```
<label>: DC[.size]  
<Wert>
```

Die Option size gibt an, ob der Wert ein Byte (.b), Word (.w) oder Longword (.l) ist. »Wert« ist der eigentliche Wert dieser »Variable«. Es können auch mehrere Werte durch Kommata getrennt hintereinander gesetzt werden.

Diese Konstanten können im Programm nach Belieben modifiziert werden, denn der Assembler setzt einen Zeiger auf die Speicherstelle, an der der <Wert> steht. Der Inhalt der Speicherstelle kann verändert werden.

**DS** hat die Form

```
<label>: DS[.size]  
<Anzahl>
```

Diese Anweisung reserviert eine Anzahl von Speicherstellen mit der Größe size (siehe DC).

**DCB** bietet die Möglichkeit, einen Block zu reservieren und ihn mit einem angegebenen Wert zu füllen. Das Anweisungsformat ist:

```
<label>: DCB[.size]  
<Anzahl>,  
<Wert>
```

Das bedeutet, DCB füllt eine angegebene Anzahl Speicherstellen der Größe size mit einem Wert. Man könnte diese Anweisung auch ersetzen, indem man der <Anzahl> ent-



sprechend oft ein »DC[.size] <Wert>« verwendet.

**EVEN** entspricht der Anweisung CNOP 0,2. Siehe dazu die CNOP-Direktive.

**END** ist das Endstatement des Assemblers. Der Assembler erkennt hieran, daß er das Ende des Sourcecodes erreicht hat. Es muß unbedingt ans Ende des Codes!

Alles, was nach dem Endstatement folgt, wird vom Assembler ignoriert.

**XDEF** definiert ein im Sourcecode vorhandenes Label als ein externes Symbol.

Dieses Symbol kann jetzt von anderen Programm-Modulen mit »XREF« benutzt werden.

**XREF** bietet die Möglichkeit, sich auf ein Symbol zu beziehen, das außerhalb des Programms definiert wurde. Beispiel:

```
XREF _AbsExecBase
```

zeigt dem Assembler an, daß das Symbol außerhalb dieses Sourcecodes definiert ist. In diesem Fall ist es im »amiga.lib« definiert. Im Code der amiga.lib könnte stehen:

```
_AbsExecBase: EQU $4
```

## Listingfiles mit Format

Die nun folgenden Direktiven sind nur im Zusammenhang mit der »-l«- beziehungsweise »-x«-Option wirkungsvoll, die ein Listingfile generieren.

**PAGE** veranlaßt den Assembler, einen Seitenvorschub auszuführen.

**LIST** bedeutet, daß der nachfolgende Code ausgegeben wird. Dies ist der Normalfall.

**NOLIST** bringt den Assembler dazu, keinen Code auszugeben, bis er entweder auf eine »LIST«- oder »END«-Anweisung trifft.

**SPC** generiert eine gewünschte Anzahl von Leerzeilen im Listing. Das Format dieser Anweisung ist

```
SPC <Anzahl der  
Leerzeilen>
```

Mit **TTL** kann ein Kopfzeilenkommentar festgelegt werden. Der Text sollte eine Länge von 40 Zeichen nicht überschreiten. Er erscheint neben dem Programmnamen und der Seitenzahl in der ersten Zeile jeder neuen Seite.

Diese Assembleranweisungen beziehen sich wie gesagt nur auf das Listingfile und haben sonst auf nichts anderes eine Wirkung. Die Anweisungen selbst erscheinen nicht als Text im Listing.

**CNOP** setzt den Code an eine vom Prozessor ansprechbare Adresse. Aufruf:

```
CNOP <nummer1> ,  
<nummer2>
```

Im Parameter nummer2 ist zu bestimmen, ob sich der Code auf words oder longwords bezieht. Hier ist entsprechend 2 (word) oder 4 (longword) einzusetzen.

Der Parameter nummer1 wird in Byte angegeben und bedeutet eine weitere Verschiebung um die hier angegebene Zahl.

**INCLUDE** »filename« muß geschrieben werden, wenn das angegebene File in das Programm eingebunden werden soll. Dies muß geschehen, bevor auf Teile aus dem Includefile zurückgegriffen wird, also am besten am Anfang des Quellcodes.

**SET** hat die gleiche Funktion wie EQU. Der einzige Unterschied besteht darin, daß die mit SET definierten Symbole später noch mal mit SET umdefiniert werden können.

**MACRO** definiert eine Makro-Anweisung. Diese werden verwendet, wenn gleiche Anweisungsblöcke häufig geschrieben werden müssen. Es werden in dem Makro mehrere Assemblerbefehle unter einem Kürzel zusammengefaßt. Erstellt werden Makros auf diese Weise:

```
<label> : MACRO  
<Anweisungs-  
block>  
ENDM
```

Dabei ist zu beachten, daß die Anweisung **MACRO** in der gleichen Zeile wie <label> steht. Das Makro wird dann mit dem Namen <label> versehen. Ein Makro wird nach der Namenszuweisung wie folgt aufgerufen:

```
<label> P1 P2
```

P1 und P2 sind Parameter, die dem Makro übergeben werden. Im Makro selbst können diese in der Form \n eingebunden werden. n steht hier für die Nummer des Parameters. Das Ganze sieht dann praktisch so aus:

```
add: MACRO  
move.l \1,d0  
add.l \2,d0  
ENDM
```

Aufgerufen wird das Makro dann mit

```
add #3,d1
```

Jetzt wird die 3 dem \1 und das d1 \2 zugeordnet. Es ist aber wichtig zu beachten, daß ein Makro keine Unterroutine

darstellt, sondern der Assembler an den Stellen, wo das Makro aufgerufen wird, den im Makro definierten Anweisungsblock einsetzt.

**ENDM** beendet ein Makro.

**SECTION** zeigt dem Assembler, in welchem Bereich des Programms er sich befindet. Der Aufruf lautet:

```
SECTION <Name> ,  
<Typ> ,  
<Option>
```

Es gibt drei Typen: CODE, DATA und BSS. CODE zeigt an, daß diese Sektion das Programm enthält. Mit DATA muß die Sektion bezeichnet werden, in der beispielsweise mit DC definierte Daten stehen.

Die Sektion BSS hat keine initialisierten Daten, es wird hier nur Platz für Daten reserviert. Beispiel: DS.

Beim Schreiben des Programms ist zu beachten, daß die Labels und Sprungmarken immer am Beginn einer Zeile stehen, also nicht eingerückt sind. Andernfalls würde der Assembler jene als Befehle interpretieren. Genauso ist deswegen zu beachten, daß Assemblerbefehle immer mindestens ein Leerzeichen eingerückt sind, da sie sonst als Label angenommen werden.

Der fertige Sourcecode kann nun mit dem A68k assembliert werden. Hierzu stellt der Assembler eine Vielzahl von Op-

```
SECTION print,CODE  
  
print: MACRO  
move.l dosbase,a6  
move.l \1,d1 * Handlnummer  
move.l #\2,d2 * Zeiger auf Daten  
move.l #\3-\2,d3 * Laenge der Daten  
jsr _LVOwrite(a6)  
ENDM  
  
move.l _AbsExecBase,a6  
move.l #dosname,a1  
move.l #0,d0  
jsr _LVOOpenLibrary(a6) * Doslibrary oeffnen  
move.l d0,dosbase  
  
move.l dosbase,a6  
jsr _LVOOutput(a6) * CLI-Outputhandle  
move.l d0,wdhandle  
  
jsr _LVOInput(a6)  
move.l d0,inphandle * CLI-Inputhandle  
  
print wdhandle,text1,textlend  
  
print wdhandle,text2,text2end  
  
move.l wdhandle,d1  
move.l inphandle,d2  
move.l #1,d3  
jsr _LVORead(a6) * Auf Return warten  
  
rts * Ruecksprung ins CLI  
  
SECTION data,DATA  
  
cnop 0,2  
dosname: dc.b 'dos.library',0  
cnop 0,2  
  
text1: dc.b $0c,$9b,'12;31H'  
textlend:  
  
text2: dc.b 'Press <RETURN> '  
text2end:  
  
SECTION vars,BSS  
  
dosbase:  
dc.l 0  
wdhandle:  
dc.l 0  
inphandle:  
dc.l 0  
inphandle:  
dc.l 0  
inphandle:  
dc.l 0  
  
END
```

Listing 1. Ein kurzer Text wird im CLI-Fenster ausgegeben



tionen zur Verfügung:

**-e <filename>**

Diese Option bietet die Möglichkeit, eine Datei zu erstellen, in die sämtliche Definitionen gespeichert werden, die über die Assemblerdirektive EQU erzeugt wurden. Falls kein eingetragener Filename angegeben wird, werden die Daten in einer Datei mit der Endung ».equ« gespeichert.

Die Option **-i** veranlaßt den Assembler, in den hinter »-i« angegebenen Directories nach Dateien zu suchen, die in den Sourcecode eingebunden werden sollen. Hinter dem »-i« wird direkt, ohne Leerzeichen, das Directory angegeben, in dem sich das oder die Includefiles befinden. Es ist auch möglich, mehrere Directories anzugeben. Hierzu trennt man die Directories nur durch ein Komma, wieder ohne Leerzeichen.

Das hat also beispielsweise die Form:

```
-idf1:include,ram:
myincludes
```

Hier wird der Assembler erst im Directory »df1:include« und dann im Directory »ram:myincludes« nach den verlangten Dateien suchen. Im Sourcecode

selbst wird dann nur noch über die INCLUDE-Anweisung der Name des Includefiles angegeben.

**-h<filename>** hat die gleiche Wirkung wie die Anweisung »Include in Source«.

Man gibt hier einfach den Namen des Files an, das eingebunden werden soll.

Auch hier werden die in der Option »-i« angegebenen Directories durchsucht.

Die letzten beiden Optionen sind natürlich hinfällig, wenn der volle Suchpfad in den INCLUDE-Anweisungen oder der »-h«-Option angegeben wird.

**-l<filename>** erstellt ein Listingfile. Dieses File hat die Endung ».lst«, wenn nicht ein eigener Name angegeben wurde. In dieser Datei werden die aus dem Assemblersource erzeugten Maschinenbefehle in Form von Hexadezimalwerten dargestellt. Neben diesen steht zum Vergleich noch einmal die Assemblerzeile, aus der der Code erzeugt wurde.

**-x** hat die gleiche Wirkung wie **-l**, hängt aber ans Ende des Listings noch einen »Symboltable«. In dieser Tabelle werden alle Symbole dargestellt, also Sprungmarken, Definitio-

nen etc. Es wird aufgeführt, an welcher Stelle im Sourcecode diese definiert wurden und wo sie benutzt werden.

**-p<Wert>** verlangt die Anzahl von Zeilen, die eine Seite haben soll, bevor ein »Formatfeed«, also ein Vorschub bis zur nächsten Seite gegeben wird. Diese Option nimmt aber nur Einfluß auf die Parameter »-l« und »-x«. Sie hat nur Wirkung auf die Druckerausgabe.

## Zwei kleine Beispiele

**-o<filename>** verlangt einen beliebigen Objektfilenamen. Das Objektfile ist die Datei, in die der Assembler das übersetzte Programm schreibt und aus welcher der Linker dann das ausführbare Programm zusammenfügt. Wird diese Option weggelassen, erstellt der Assembler eine Datei mit der Endung ».o« (für Objektfile). Dies ist der übliche Weg.

**-d** veranlaßt den Assembler, »symbol table entries« in das Objektfile zu schreiben. Diese werden benötigt, wenn man mit symbolischen Debuggern arbeitet.

**-s** erzeugt ein File im »Motorola S-record«-Format mit der Endung ».s«.

Diese Files haben für die Programmierung auf dem Amiga keine praktische Bedeutung.

**-w** bietet die Möglichkeit, den internen Heap des Assemblers entweder zu vergrößern oder zu verkleinern. Das Format dieser Option ist

```
-w<Size Primary Heap> ,
<Size Secondary Heap>
```

Die Standardeinstellungen sind \$8000 für den »Primaryheap« und \$400 für den »Secondaryheap«. In diesen Heaps werden vom Assembler verschiedene Informationen, wie Symboltabellen, Includefilenamen etc. gespeichert.

Nach beendeter Assemblierung zeigt der A68k den benötigten Platz in den Heaps an.

Sollten Fehler beim Assemblieren auftauchen, bietet der A68k eine ausreichende Anzahl von Fehlermeldungen, die das Korrigieren vereinfachen.

Zur praktischen Anwendung noch zwei Programme, die einige der Fähigkeiten des Assemblers demonstrieren sollen.

Das Programm in Listing 1 gibt einen kurzen Text auf dem CLI-Window aus. Hierzu wird zuerst die Handlenummer des Fensters über die Funktion »\_LVOutput« geholt. Mit dieser Identifikationsnummer und einem Zeiger auf einen Daten-

buffer (hier der Text) kann nun ein beliebiger Text ausgegeben werden.

Das Programm muß wie folgt assembliert werden:

```
A68K Listing1.o
```

Der Assembler gibt zuerst einige Texte in eigener Sache an und dann die Information, in welchem Durchlauf der Assemblierung und in welcher Zeile er sich gerade befindet. An dieser Stelle können auch oben erwähnte Optionen eingesetzt werden, die dann zu den besprochenen Resultaten führen. Der Assembler erzeugt nach dem Durchlauf ein File namens »Listing.o«. Dieses Objektfile, welches so nicht ausführbar ist, ist nun noch mit dem »amiga.lib« gelinkt worden. Hierzu kann man beispielsweise den Linker »BLink« benutzen. Der Aufruf sieht dann wie folgt aus:

```
BLink Listing1 LIB
amiga.lib
```

Damit entsteht dann endlich ein ausführbares Programm.

In Listing 2 werden dem Diskettenlaufwerk amüsante Geräusche entlockt. Es wird wieder mit

```
a68k Listing2.asm
assembliert. Gelinkt wird wieder mit »amiga.lib«:
```

```
BLink Listing2.o LIB
amiga.lib
```

Wenn man im Sourcecode des Listing 2 den Begriff »\_AbsExecBase« gegen »\$4« austauscht und die Zeile »XREF\_AbsExecBase« löscht, kann man das Programm auch ohne »amiga.lib« linken. Der Aufruf lautet dann:

```
BLink Listing2.asm
```

Sie werden feststellen, daß der Assembler A68k für erste Schritte und die Eingabe von veröffentlichten Assembler-Listings durchaus ausreichend ist. Man kann ihn zwar nicht zu den schnellsten rechnen, aber das wird dadurch wieder wettgemacht, daß er sich an den 68000-Assembler-Standard hält (im Gegensatz zu vielen kommerziellen Assemblern). Die aktuelle Version beherrscht noch keine Befehle der Prozessoren 68010 oder 68020. Diese könnten aber mit den nötigen Programmierkenntnissen hinzugefügt werden.

Für die Einführung in die Assembler-Programmierung möchten wir auf den ausführlichen und leichtverständlichen Kurs im AMIGA-Sonderheft 4 verweisen. Dort finden Sie auch weitere Beispiel-Listings.

rs

```
*
*   Diskmusik - Ping Pong im Laufwerk
*
XREF   _AbsExecBase

SECTION prog,CODE

move.l  _AbsExecBase,a6
move.b  #$0,295(a6)          *Interrupts aus
or.b    #$7d,$bfd100        *Alle Drives aus
or.b    #$78,$bfd100
bclr   #1,$bfd100          *Track nach innen fahren
bclr   #3,$bfd100          *Df0:

sepp:  bclr   #0,$bfd100      *Steppen
        nop
        nop
        bset  #0,$bfd100

        move.l fred,d2        *Pause
mouse:  btst  #6,$bfe001
        beq  mouse
        dbf  d2,mouse

        bchg  #1,$bfd100      *Track innen/ausßen
sub.l   #100,fred
add.l   #1,count
cmp.l   #40,count
beq     ende
bra     sepp
ende:  move.l  _AbsExecBase,a6 *Interrupts an
        move.b  #$ff,295(a6)
        rts

SECTION vars,DATA

fred:  dc.l  $4000
count: dc.l  0

END

Listing 2. Amüsante Geräusche entlockt dieses Listing dem Diskettenlaufwerk
```



# PDC

## ein totgeborenes Kind?

Von Christian Wolf

Jeder Programmierer, der einen Einblick in den internen Aufbau eines komplexen Projektes haben möchte, wird vom PDC (Fish 110) begeistert sein. Mit genauer Analyse des Source-Codes lernen Sie wichtige Fakten des Compiler-Baus kennen.

Wir können Ihnen natürlich keinen Einführungskurs in Compilerbau geben, da dieses Thema schon über ganze Bücher hinweg ausgeweitet werden müßte. Vielmehr ist die Eigeninitiative eines Amiga-Freaks und dessen Experimentierfreudigkeit gefragt, um dem Compiler mit geschickter Programmierung aus seinem »Mauerblümchendasein« herauszuhelfen.

Genau diese Strategie wird auch von dem ursprünglichen Autor, Jeff Lydiatt, propagiert. Er verschweigt keineswegs, daß seine Version 0.2 des C-Compilers in einigen Bereichen recht verbesserungswürdig ist.

### Der Source mit dem makefile neu übersetzt

Wie von vielen PD-Sources gewohnt, wird auch PDC mit einem »makefile« geliefert. Der Vorgang der Erstellung des Compilers selbst wirft so keine unlösbaren Probleme auf. Einer potentiellen Weiterentwicklung steht somit nichts im Wege, wobei für ein solches Projekt fundierte Kenntnisse in C unbedingt erforderlich sind. Es kommt allerdings erleichternd hinzu, daß hier spezielle Kenntnisse über den Amiga ausnahmsweise nicht benötigt werden. Ein schneller Blick über den Sourcecode offenbart nämlich ein in Standard-C gehaltener Programmierstil. Wer

**Schon seit längerer Zeit findet man unter den Fish-Disks einen C-Compiler. Natürlich reichen dessen Leistungen bei weitem nicht an die eines Aztec- oder Lattice-Compilers heran. Der mitgelieferte Quelltext stellt jedoch eine faszinierende Schatzgrube für Tüftler dar.**

C schon von einem anderen Rechner her kennt, dürfte sich auch hier schnell zurechtfinden.

Der PDC-Compiler erlaubt die direkte Übersetzung des C-Quelltextes in ein Assemblerlisting, genau so wie man es unter Aztec gewohnt ist. Mittels »pdc test.c« würde also beispielsweise »test.s« kreiert werden, das man sich dann mit einem Editor anschauen könnte. Der Assembler-Source hält sich an den Metacomco-Standard und kann damit problemlos von jedem Assembler übersetzt werden.

PDC besitzt genau zwei Optionen; es läßt sich nur sehr wenig indirekter Einfluß auf das Compilat ausüben. Die Optionen lauten wie folgt:

-l: Dem Assembler-Quelltext werden die entsprechenden Zeilen des C-Sources hinzugefügt. Man bekommt so recht schnell eine Vorstellung davon, wie die C-Anweisungen in Maschinencode aussehen.

-n: Mit dieser Option wird PDC angewiesen, den Pro-

grammcode zu optimieren. Es wird versucht, überflüssige Anweisungen zu entfernen.

Mit »A68k« wird ein entsprechender PD-Assembler gleich mitgeliefert, so daß man nicht schon in einen Engpaß gerät. Es handelt sich hierbei um einen MetaComco-kompatiblen Cross- und Makroassembler, der also auch von reinen Assembler-Programmierern benutzt werden kann. Er ist in seinen Funktionen soweit ausgereift, daß er sich von kommerziellen Produkten nicht so ohne weiteres in den Schatten stellen läßt. Die wenigen Einschränkungen sind so geringfügig, daß sie im normalen Programmierbetrieb fast nicht auffallen (es handelt sich hierbei zum Beispiel um fehlende Direktiven, die in bestehenden Assemblerlistings fast nie verwendet werden). »A68k« ist also für die Erstellung von AmigaDOS-kompatiblen Objektmodulen bestens geeignet, zumal man so der PD-Linie treu bleibt. Weitere Informationen über den A68k finden Sie im Artikel auf Seite 113 in diesem Sonderheft.

Kommen wir nun zur letzten Phase der Erstellung eines ablauffähigen C-Programms, dem Linkvorgang: Zur Bewältigung dieser Aufgabe bietet sich »blink« an, ein auf den Fish-Disks vorgestellter, voll funktionsfähiger Linker. Die Frage nach dessen Leistungsfähigkeit entfällt, wenn man bedenkt, daß der Lattice C-Compiler in neueren Versionen auch schon auf diesen, allerdings bereits verbesserten Linker zurückgreift. Der bei Lattice zuvor verwendete »alink« konnte einem Vergleich mit »blink« nicht standhalten, so daß man diesen kurzerhand mit in das Compiler-Paket implementierte.

Jetzt stellt sich nur noch die Frage nach der zu verwenden-

den Library, da auf der PDC-Diskette leider keine mitgeliefert wird. Hier wird der Benutzer wohl doch auf die »amiga.lib« zurückgreifen müssen, da es leider noch kein entsprechendes Public Domain-Äquivalent gibt. Lediglich ein passendes Startup-Modul, das die Anpassung der CLI-Argumente an die typische C-Form »main(argc, argv)« unternimmt und das vernünftige Starten eines Programms überhaupt erst ermöglicht, wird zur Verfügung gestellt. Steht dem Programmierer die »amiga.lib« nicht zur Verfügung, dann muß wohl oder

### Anwendungsgebiet: Standard C

übel jede noch so kleine Funktion selbst geschrieben werden.

Wie es vielleicht der Quelltext des C-Compilers schon demonstriert, so ist er hauptsächlich für die Übersetzung von C-Programmen geeignet, die keinerlei Amiga-spezifische Funktionen oder Strukturen verwenden. Der Grund dafür liegt ganz klar auf der Hand: PDC verwendet in seiner momentanen Version einen sehr verkümmerten Pre-Prozessor, der allerdings als fast schon unerlässlich bezeichnet werden kann. Des weiteren ist die für C sehr wichtige »typedef«-Anweisung noch nicht implementiert, so daß mit diesen beiden Einschränkungen die Amiga-Include-Files nicht mehr vernünftig eingelesen werden können. Allein das Einbinden kann schon Probleme bereiten. Diejenigen unter Ihnen, die vielleicht mal verbessernd in den Quelltext eingreifen möchten, sollten genau hier ansetzen, da die Beseitigung dieser Mankos den Compiler



wahrscheinlich schon in einen akzeptablen Zustand versetzen würde. Übrigens soll sich auf Fish-Disk 28 ein leistungsfähiger Pre-Prozessor befinden, der bei diesem Compiler Anwendung finden könnte.

Sollten Sie nur an der experimentellen Benutzung des Compilers Interesse haben, dann sind folgende Mängel beim Programmieren unbedingt im Hinterkopf zu behalten: 1. Der Compiler unterstützt keine Fließkommaarithmetik. Variablen, die im Quelltext als »float« deklariert sind, werden zwar akzeptiert, bei der Generierung des Codes aber nicht beachtet.

2. PDC behandelt alle Integerkonstanten als Langworte. Dies würde zum Beispiel bei der Anweisung

```
printf( "Here is the
```

```
character %c\n",  
'x' );
```

zu Problemen führen, da hierbei der Funktion »printf« eine Charakterkonstante als Argument erwartet. Das Code-Fragment

```
char x;  
x = 'x';  
printf( "Here is the  
character %c\n", x );
```

würde hingegen mit der Übergabe einer Charaktervariablen korrekt übersetzt werden. Bei Char- und Wordkonstanten ist also Vorsicht geboten, im Zweifelsfall sollten immer Variablen des entsprechenden Typs verwendet werden.

3. PDC erlaubt noch nicht alle unter C gültigen Konstrukte, wie etwa die wiederholte Anwendung des Zuweisungsoperators.

Ein Beispiel: Die Anweisung  
`x=y=0`

würde zur Ausgabe eines Fehlers führen. Ebenso ist die direkte Initialisierung einer Autovariablen bei deren Deklaration, wie etwa

```
int x=0
```

noch zu vermeiden.

4. Der Aufzählungstyp »enum« ist noch nicht zulässig und ist

## Kleine Probleme mit Zeigern auf Zeiger

bisher nicht im Keyword-Verzeichnis des Compilers vermerkt.

5. Bei der Behandlung von Zeigern auf Zeiger gibt es noch Probleme. So würde die Variable »char \*\*argv« zwar noch

richtig behandelt werden, »char \*argv[]« jedoch zu Fehlern bei der Übersetzung führen.

Sicher lassen sich diese Einschränkungen mit etwas Geschick beim Programmieren umgehen, so daß Sie der Willkür des Compilers doch noch entgegen können, um sich an kleinere Programmierprojekte heranzuwagen. Die Fehlermeldungen sind zwar manchmal nicht besonders vielsagend, der versiertere C-Programmierer hat aber bestimmt die nötige Intuition, um auch diese Probleme zu meistern.

Auf den neuesten Fish-Disks erscheinen neue C-Compiler, die vielleicht über das Leistungsspektrum des PDC hinausgehen. Wir sind auf diese brandneuen Produkte natürlich genauso gespannt wie Sie. Schau'n wir mal... rs

# Von Wanzen und ähnlichem Ungeziefer...

Von Klaus Sonnenleiter

**A**ls Computer noch raumfüllende Ungetüme waren und mit Röhren statt integrierten Schaltkreisen betrieben wurden, entstand eine Legende: Es hieß, alle unerklärlichen Programmfehler rührten von Wanzen und ähnlichem Ungeziefer (englisch: »Bugs«), die sich im Rechnergehäuse breit machten und die Elektronik lahmlegten.

Ob dies der Wahrheit entsprach oder nur eine Ausrede der Perfektionsbesessenen Programmierer war, mag dahingestellt bleiben. Tatsache ist, daß sich der Ausdruck bis heute erhalten hat.

Als »Bug« bezeichnet man auch heute noch einen besonders perfiden (oder unerklärli-

## Aus der Urzeit der Elektronik

chen) Programmfehler. Folglich versteht man unter »Debugging« auch heute noch die Fehlersuche, auch wenn die Ursache sicher nicht mehr bei irgendwelchen kleinen Insekten zu suchen ist.

Ein hohes Ansehen bei allen Programmierern genießen daher sogenannte Debugging-Hil-

**Sie sind der Alptraum eines jeden Programmierers: Diese hinterhältigen kleinen, gut versteckten Programmfehler, die erst nach endlosen Testläufen (wenn überhaupt) gefunden werden. Abhilfe schafft ein guter Debugger.**

fen. Diese versprechen Unterstützung bei der Suche nach Fehlerursachen. Der letzte Schrei sind im Moment aufwendige »Source-Level-Debugger«. Wir stellen Ihnen zwei einfache aber effektive Hilfsmittel vor, die Ihnen die Arbeit enorm erleichtern.

Ein Utility stammt aus der Feder von »Mister PD« persönlich. Fred Fish schrieb diesen Debugger für C-Programmierer (»DBug« von der Fish-Disk 102), der einen ungewöhnlichen, aber dennoch naheliegenden Weg geht. Wir gehen auf diesen weiter unten intensiv ein.

»GOMF« steht für »Get Outa My Face«. Dieser unwillige Ausspruch eines entnervten Programmierers (hier Christian Johnsen) bezieht sich auf die von allen geliebten Gurus. GOMF verbannt den Guru dahin, wo er hingehört: ins Nirwana.

Fast alle abgestürzten Programme können sicher verlassen werden. Sie können Ihre wertvolle Arbeit speichern, bevor Sie neu booten (oder eventuell den Debugger hinzuziehen).

Daher ist GOMF nicht zu den Debuggern zu zählen. Das Programm hilft Ihnen lediglich, die schlimmsten Abstürze zu vermeiden. Daneben werden einige Informationen über die möglichen Ursachen gegeben. Die Version 1.0 finden Sie auf der Fish-Disk 95.

GOMF installieren Sie per Doppelklick auf das Icon oder mit der Zeile

```
run Gomf
```

im CLI oder in der Startup-Sequenz. Testen Sie danach die mitgelieferten Demoprogramme. Diese bestehen aus »fehlerhaften« Programmen, die

normalerweise einen Guru zur Folge hätten. Installieren Sie GOMF. Sie sehen für einen kurzen Moment ein Fenster mit Text in der linken oberen Ecke des Bildschirms. Dieses verschwindet automatisch, nachdem GOMF sich selbst als Hintergrundtask installiert hat. Starten Sie darauf das Programm »Error1«. Sie sehen ein Auswahlfeld, das die Wahl zwischen verschiedenen Gurus erlaubt. Wählen Sie beispielsweise das Gadget mit der Aufschrift »Address Error«. Falls Sie GOMF nicht installiert haben, stürzt der Amiga in diesem Moment ab. GOMF dagegen

## Dem Guru auf die Finger geklopft

meldet sich mit einem Requester (»Task held - finish all disk activity«). Klicken Sie diesen mit Cancel weg, und Sie erhalten das GOMF-Menü. Sie werden in der ersten Zeile über die Art des Fehlers informiert. Dies geschieht mit der Nummer der drohenden Gurumeldung. Außerdem finden Sie darunter einen Text, der Ihnen Hinweise auf die Fehlerursache gibt.

Am unteren Rand sehen Sie vier Gadgets. Klicken Sie auf das mit der Aufschrift »Whap« und danach auf das Ausgabe-



fenster von »Err1«. Gornf löscht darauf das fehlerhafte Programm sowie das angeklickte Fenster und meldet sich wieder mit dem Requester. Klicken Sie diesmal auf das »Gornf«-Gadget. Dieses kehrt wieder zum Ausgangszustand zurück. Falls Sie lieber booten oder eine Audienz beim Guru wollen, verwenden Sie eines der beiden anderen Gadgets.

Testen Sie das gleiche bei dem Programm »err2«. Sie werden feststellen, daß Gornf diesmal nur das letzte geöffnete Fenster verschwinden läßt. Wollen Sie auch die anderen wegklicken, erreichen Sie dies durch mehrmaliges Klicken auf »Whap«. »Err3« demonstriert ein Programm, das ohne Öffnen eines Windows oder Screens sofort abstürzt.

Eine sehr gute Möglichkeit für Programmierer demonstriert »Err4«. Dieses Programm wurde mit Gornf gelinkt. Dadurch braucht Gornf nicht initialisiert zu werden, es ist automatisch beim Start von Err4 aktiv. Testen Sie dies, indem Sie neu booten und ohne Gornf zu starten, dieses Demo aufrufen. Sie sehen die »Einschaltmeldung« von Gornf und danach ein ähnliches Fenster wie bei Err1. Wählen Sie einen beliebigen Guru. Alles weitere läuft ab wie beim ersten Beispiel.

Sie können daher als Programmierer die Möglichkeiten von Gornf mitbenutzen. Sicher ist dies ganz nett, aber viele Fehler sind sehr viel subtiler und besser versteckt. Es ent-

steht vielleicht kein Guru, sondern nur eine Ausgabe, die Sie nicht erwartet haben. Gornf hilft hier nicht weiter, also was ist zu tun? Hier gibt es nur die klassische Variante.

## Der klassische Debugger

Wer bei »DBug« einen Source-Level-Debugger mit komfortabler Maussteuerung und ähnlichem Schnick-Schnack erwartet, wird enttäuscht. Dbug unterstützt eher die traditionelle Methode der Fehlersuche.

Sicher haben Sie diese in der einen oder anderen Form schon einmal angewandt: An einer wichtigen Stelle Ihres Programms lassen Sie die entscheidenden Werte ausgeben, um zu überprüfen, ob diese in einem kritischen Bereich liegen.

Konsequenterweise handelt es sich bei Dbug nicht um ein alleinstehendes Programm, sondern um Routinen, die Sie in eigene Programme einbinden können. Der Sourcecode ist in Standard-C gehalten und unter Unix und AmigaDOS compilierbar. Es sollte aber wenig Mühe bereiten, den Debugger für eine beliebige andere Umgebung anzupassen.

Dbug bietet eine große Anzahl Funktionen zur Kontrolle eigener Programme, die mit dem Einbinden von »dbug.h« zur Verfügung stehen:

- Ein »Trace«-Modus zeigt die Verschachtelungstiefe der aufgerufenen Funktionen in einer halb-grafischen Darstellungs-

Argument	Wirkung
d[,schlüssel]	aktiviert die Ausgabe über Makros, welche die angegebenen Schlüssel verlangen. Sind keine Schlüssel genannt, so gelten alle als aktiviert.
D[,zeit]	hält das Programm nach jeder Ausgabe für den Wert in »zeit« an (voreingestellt ist 0).
f[,Funktionen]	begrenzt die Aktivitäten des Debuggers auf die angegebenen Funktionen. Werden keine übergeben, so gelten alle als aktiviert.
F	gibt bei jeder Ausgabezeile das Sourcefile mit an, aus dem diese stammt.
L	markiert jede Ausgabezeile mit der Zeilennummer, welche die Ausgabe verursacht hat.
n	markiert jede Ausgabezeile mit der aktuellen Verschachtelungstiefe der Funktionen.
N	numeriert alle Ausgabezeilen des Debuggers. Dies ist besonders hilfreich, wenn Debugger- und Programm-Ausgabe nicht getrennt sind.
o[,file]	lenkt die Ausgabe des Debuggers auf »file« um. Voreingestellt ist »stderr«. Ohne die Angabe eines Files wird »stdout« zum Ausgabefile.
p[,prozesse]	begrenzt die Aktivitäten des Debuggers auf die angegebenen Prozesse. Wird keiner angegeben, gelten alle als aktiviert. Der Prozeßname (wenn angegeben) muß mit dem ersten Parameter von »DBUG_PROCESS« übereinstimmen.
P	markiert jede Ausgabe mit dem Namen des Prozesses, aus dem die Ausgabe stammt.

Alle Parameter in eckigen Klammern dürfen auch weggelassen werden. In diesem Fall werden die Defaultwerte eingesetzt.

Tabelle 1. Die Argumente für »DBug« können aus dieser Liste gewählt werden

weise. Jede neue Ebene wird eingerückt.

- Ausgabe des Wertes aller oder eines Teils der benutzten Variablen.

- Ausschließen von kritischen Funktionen.

- Beschränken der Verschachtelungstiefe auf einen vorher definierten Wert (nicht größer als 255).

- Versehen jeder Ausgabe mit dem Namen des zugehörigen Sourcefiles und der Zeilennummer.

- Der aktuelle Status kann auf den Stack gelegt und wieder von dort geholt werden, während das Programm in der Zwischenzeit mit Defaultwerten fortfährt.

- Die Ergebnisse des Debug-

# \*AMICUS\*

DIE ABENTEUER EINER FLOPPY-DISC





gers können auf ein beliebiges File umgeleitet werden. Dies ist unabhängig von der Standardausgabe des zugrundeliegenden Programms. Auch eine eventuelle Umleitung beim Programmaufruf beeinflusst die für den Debugger eingestellten Werte nicht.

Die hier verwendete »interne« Debuggingtechnik ist den meisten Programmierern vertraut, da sie meist die naheliegendste darstellt. Der erste Weg führt über die Kontrolle der Programmausgabe und »print«-Anweisungen an den kritischen Stellen. Liefert dies noch kein zufriedenstellendes Ergebnis, wird der Vorgang mit neuen Werten wiederholt, solange bis entweder der Fehler gefunden oder die anfallenden Daten wegen zu großer Masse unkontrollierbar werden.

Sind alle Fehler gefunden, müssen die eingefügten »print«-Anweisungen wieder mühsam entfernt werden. Dbug automatisiert dieses Vorgehen. Sie können beliebige Funktionen in Abhängigkeit von der Existenz eines Symbols (»DEBUG«) ausführen lassen. Ist das Symbol nicht definiert, werden auch keine Debugging-Anweisungen ausgeführt.

## DBug spart Speicher

Haben Sie Dbug in Ihr Programm eingebunden, sind folgende Makros verfügbar:

**DEBUG\_ENTER** übergibt

dem Laufzeitsystem den Namen einer Funktion, die ausgeführt werden soll, als String. Beispielsweise:

```
DEBUG_ENTER ("main");
```

Für jedes **DEBUG\_ENTER** muß ein **DEBUG\_RETURN** oder **DEBUG\_VOID\_RETURN** definiert sein (siehe unten). Fehlt dies, erhalten Sie eine Warnung.

**DEBUG\_RETURN** verwenden Sie, um eine mit **DEBUG\_ENTER** aufgerufene Funktion zu verlassen. Funktionen ohne Rückgabewert werden mit **DEBUG\_VOID\_RETURN** verlassen:

```
DEBUG_RETURN (wert);
```

```
DEBUG_VOID_RETURN;
```

Wenn Sie Funktionen mit **DEBUG\_RETURN** verlassen, die nicht mit **DEBUG\_ENTER** aufgerufen wurden, erhalten Sie eine Fehlermeldung.

**DEBUG\_PROCESS** verwenden Sie, um den momentan ausgeführten Prozeß zu benennen.

```
DEBUG_PROCESS (argv[0]);
```

ist ein typischer Aufruf. Sie können jedoch auch jeden anderen String angeben.

**DEBUG\_PUSH** setzt einen neuen Debugger-Status, indem der aktuelle auf einen internen Stack geschoben wird. Der neue Status hängt ab vom Argument:

```
DEBUG_PUSH ("");
```

```
DEBUG_PUSH ("d:t");
```

```
DEBUG_PUSH ((argv[1] [2]));
```

Die letzte Art des Aufrufs ist die gebräuchlichste. Dabei wird ein Argument aus der Kommandozeile geholt. Die ersten beiden Zeichen (bei Dbug-Argumenten immer »-#«) werden abgeschnitten.

**DEBUG\_POP** restauriert einen alten Debugger-Status durch »Ziehen« vom internen Stack. Falls **DEBUG\_POP** häufiger aufgerufen wird als **DEBUG\_PUSH**, erhalten Sie keine Fehlermeldung, der Aufruf wird einfach ignoriert. **DEBUG\_POP** benötigt keinen Parameter:

```
DEBUG_POP ();
```

**DEBUG\_FILE** stellt den Ein-/Ausgabekanal für »**DEBUG\_PRINT**« ein. Es wird auf die gleiche Weise verwendet wie die beiden Symbole »**stdout**« und »**stderr**«.

**DEBUG\_EXECUTE** führt ein beliebiges Stück C-Code aus. Das erste Argument ist dabei das Dbug-Schlüsselwort, bei dem der als zweites Argument übergebene Programmteil gestartet werden soll. Dabei ist allerdings Vorsicht geboten, denn beim Fehlen der Argumentliste für das »d«-Flag wird dieses Makro automatisch angesprochen.

```
DEBUG_EXECUTE ("abort", abort ());
```

**DEBUG\_PRINT** schreibt beliebige Argumente in den Ein-/Ausgabekanal, der mit **DEBUG\_FILE** umgelenkt werden kann. Das erste Argument ist ein Dbug-Schlüsselwort, das zwei-

te der Format-String und die zugehörige Argumentliste:

```
DEBUG_PRINT ("eof", ("end of file found"));
DEBUG_PRINT ("type", ("type is %x", type));
DEBUG_PRINT ("stp", ("%x -> %s", stp, stp -> name));
```

Die Schlüsselwörter definieren Sie in der Argumentliste hinter der »d«-Option (siehe unten). Wenn dort keine Schlüsselwörter übergeben werden, gelten alle verwendeten Schlüsselwörter als aktiviert.

## Kontrolle ist Ihre Sache

Sie kontrollieren den Debugger über den sogenannten Kontrollstring in der Argumentliste. Dieser besteht aus der Zeichenfolge »-#« sowie den durch Doppelpunkt getrennten Argumenten. Diese entnehmen Sie der Liste in Tabelle 1.

Natürlich kann in diesen wenigen Zeilen nur ein kurzer Überblick der Leistungen von Dbug gegeben werden. Wenn Sie diese voll ausreizen wollen, sollten Sie sich auf jeden Fall mit dem mitgelieferten Doc-File (»user.t«) beschäftigen. Es enthält viele wertvolle Hinweise (beispielsweise auch zu »Cross-Development« und Einbau in Libraries). In jedem Fall erhalten Sie mit »Gomf« und »Dbug« zwei Werkzeuge, die Gurus und ähnliches Ungeziefer zuverlässig im Zaum halten.





Programm:	Art:	Bemerkungen:
Fish 173 FileSelect	T	File-Selector Version 1.0 von Fabrice Lienhardt. Komplet in Assembler geschrieben, Sourcecode enthalten.
IffLib	T	Library zur einfacheren Handhabung von IFF-Files von Christian A. Weber. Mitgeliefert werden einige Demo-Anwendungen, beispielsweise zum Speichern des vorderen Screens als IFF-File.
ILBM2C	T	Macht aus IFF-Bildern C-Sourcecode, den Sie direkt in eigene Programme einbinden können. Die Bilder werden dann mit der »DrawImage«-Funktion angezeigt. Die Sources und ein Beispielprogramm werden mitgeliefert. Von Tim Kemp.
Tetrix	S	Eine faszinierende Tetris-Variante von David Corbin. Version 1.1 ohne Sourcecode.
Uedit	U	Shareware-Editor Verson 2.4g (Beta) von Rick Stiles. Läßt sich in hohem Maß den Bedürfnissen des Anwenders anpassen. Sourcecode wird nicht mitgeliefert. Update zu der Version auf Fish 121.
Fish 174 Castle	S	Einfaches Adventure (für Anfänger geeignet) von H. James Fox. Sourcecode wird nicht mitgeliefert, ist aber vom Autor erhältlich.
Iff2Sun	U	Von Steve Berry und Mark Thompson. Konvertiert Amiga-IFF-Files ins »Rasterfile«-Format der SUN-Workstation. Hilfreich für alle, die Zugang zu einer SUN haben. Nur Sourcecode, das Programm muß auf der SUN neu compiliert und von dort gestartet werden.
Print1.2	U	Druck-Utility von John F. Zacharias. Steuerung aus dem CLI, Files können in ASCII oder HEX gedruckt werden. Zeilen können auf Wunsch numeriert werden. Abbruch mit <CTRL c>. Einschließlich Sourcecode.
Sh	U	Utility von Jim Guilford zur Arbeit mit Archivfiles. Source enthalten.

# In letzter Minute

Kurz nach Redaktionsschluß dieser Ausgabe erreichten uns die Fish-Disks 173 bis 200. Wir wollen Ihnen diese natürlich nicht vorenthalten. In Ergänzung zu der Liste auf Seite 80 finden Sie hier die neuesten »Fische«.

(so)

Programm:	Art:	Bemerkungen:
Strings	U	Durchsucht Textfiles nach einem String. Wird über die Kommandozeilen per Optionen gesteuert. Version 1.0 von Joel Swank. Sourcecode wird mitgeliefert.
TitlePage	U	Utility zur Manipulation der Titelseite von Listings von Joel Swank. Steuerung über die Kommandozeile. Sourcecode eingeschlossen.
Tunnel	G	Grafic-Demo von Garth Thornton. Sourcecode in TDI-Modula wird mitgeliefert.
Fish 175 Elements	A	Eine Darstellung des Periodensystems der chemischen Elemente von Paul Thomas Müller. Eine Menge zusätzlicher Informationen werden angezeigt.
Furnish	A	Ein »Wohnzimmereinrichtungsprogramm« von Terry Gintz. Das Programm, nach dem Sie schon immer gesucht haben. Ohne Sourcecode.
Plot	A	Programm zum Berechnen und Ausdrucken von dreidimensionalen

# \*AMICUS

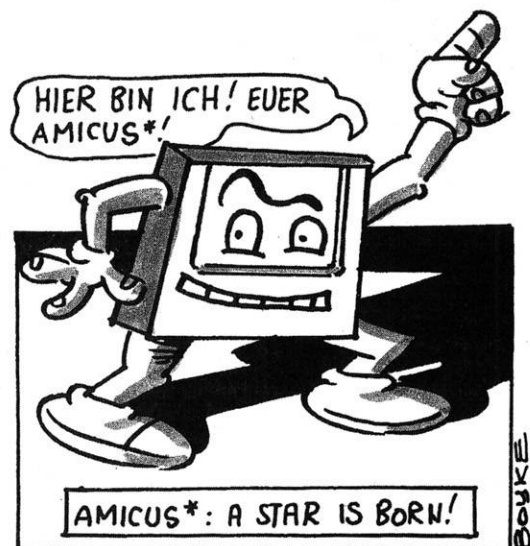
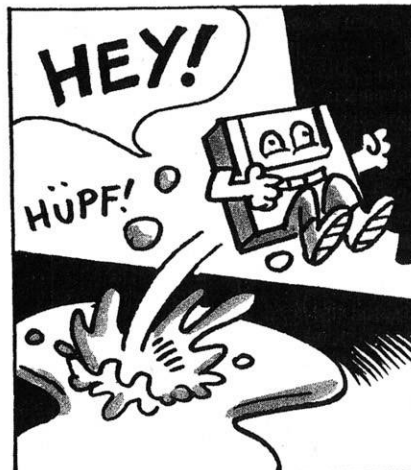
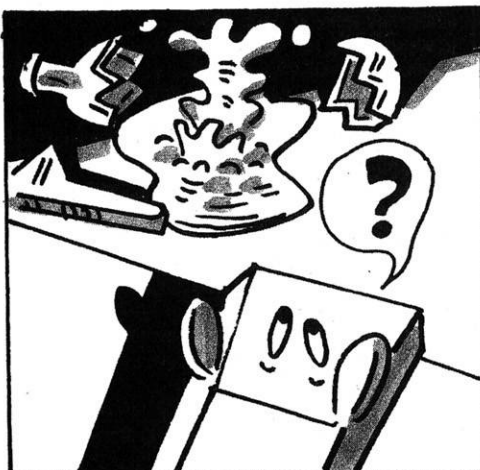




Programm:	Art:	Bemerkungen:
SafeBoot	U	Funktionen von Terry Gintz. Version 4.1, Update der Version von Fish 49. Liest und speichert beliebige Bootblöcke. Diese können später restauriert werden, falls eine Diskette mit einem Virus infiziert ist. Version 2.2 von Mark Lanoux.
SendMorse	A	Liest ein Textfile und gibt die Zeichen in Morsetönen aus. Die Geschwindigkeit kann eingestellt werden. Von Joe Larson.
VirusX	U	Version 3.10 des Virus-Erkennungs-Programms von Steve Tibbett. Erkennt und entfernt alle derzeit bekannten Viren (einschließlich des IRQ-Virus). Sourcecode wird mitgeliefert.
WBDepth	T	Verändert die Anzahl der BitPlanes für die Workbench. Von Andry Rachmat, ohne Sourcecode.
Zippy	U	Eine »grafische Shell« von Michael Weiblen. Legt alle wichtigen CLI-Funktionen auf Menüs. Eigene Menüs sind definierbar und Sie haben Zugriff auf die Daten des FileInfo-Blocks. Version 2.5, ohne Sourcecode.
<b>Fish 176</b> AnalytiCalc	A	Äußerst leistungsfähiges Spreadsheet-Programm von Glenn Everhart (Version 23-2A). Update der Version von Fish 144. Es wurden nur die veränderten Files aufgenommen. Für ein komplettes Paket ist auch die Fish 144 nötig. Fragt beim Start nach, ob ein Interlace-Screen verwendet werden soll.
HyperNet	A	Implementiert eine Hypertext Shell auf dem Amiga. Diese gibt AmigaDOS-Prozessen die Möglichkeit, eigene »Unter-Prozesse« zu starten. Von Glenn Everhart, Sourcecode wird mitgeliefert.
<b>Fish 177</b> ASpice	A	Schaltkreisanalyseprogramm, das ursprünglich für eine VAX entwickelt und hier auf den Amiga übertragen wurde. Benötigt mindestens 1,5 MByte

Programm:	Art:	Bemerkungen:
Disksalv	U	Speicherkapazität. Ohne Sourcecode. Restauriert versehentlich gelöschte Files.
Jask	U	Verbesserte Version 1.32 von Dave Haynie. Ersatz für das Ask-Kommando des CLI von John Barshinger. Läuft unter der Workbench, Version 1.0 mit Sourcecode.
Marge	U	Einfaches Programm von Joel Swank, das auf der linken Seite von Textfiles eine anzugebende Zahl Tabs oder Spaces einfügt. Sourcecode wird mitgeliefert.
<b>Fish 178</b> AmicForm	A	Version 1.4 des Telefonverzeichnisses von John Motsinger. Update zu Fish 157, ohne Sourcecode.
BlackBox	S	Spiel von Tim Kemp, bei dem Sie mit Hilfe eines Röntgenstrahls Atome in einer undurchsichtigen Schachtel finden müssen. Ohne Sourcecode.
CIATimer	T	Zwei Versionen einer Routine von Karl Lehenbauer (basierend auf einem Programm von Paul Higginbottom), welche die CIA-Zeit mit hoher Genauigkeit ausliest. Mit Sourcecode.
Cosmic	S	Intergalaktisches Spiel von Krieg und Frieden. Version 1.1 von Carl Edman mit Sourcecode.
Ls	U	Directory-Lister im Unix-Stil von Justin V. McCormick. Version 2.0 wurde an Lattice 5.0 und die Workbench 1.3 angepaßt. Mit Sourcecode.
RemLib	T	Entfernt eine unbenutzte Bibliothek aus dem Speicher. Zeigt auch Informationen über alle verfügbaren Libraries an. Version 1.11 von Heiko Rath ist ein Update zur Fish 139. Assembler-Sourcecodes werden mitgeliefert.
RexxArpLib	T	Version 2.0 der bekannten Library von W. G. J. Langeveld. ARexx-Makros werden mitgeliefert.
DietAid	A	Diät-Planer von Terry Gintz. Berechnet automatisch aus der Zutatenliste eines Rezepts den Kaloriengehalt. Version 3.1 ist

Tabelle 1. Der neueste Streich von Fred Fish. Seine Amiga-PD-Bibliothek ist inzwischen auf 200 Disketten angewachsen.





Programm:	Art:	Bemerkungen:
Dmake	U	ein Update zu Fish 36. Ohne Sourcecode. Beta-Version von Matthew Dillons Make-Utility für C-Programmierer. Sourcecode enthalten.
Exception	T	Routinen zur Fehlerbehandlung, die oft schwierig in einem Anwendungsprogramm zu implementieren sind (»read/write-error«, »no more memory«, etc.), sollen damit vereinfacht werden. Version 0.6 von Gerald T. Hewes mit Sourcecode.
KickFont	U	Ersetzt den Standardfont (»Topaz«) auf der Kickstart-Diskette mit einem neuen. Nur für Besitzer eines Amiga 1000. Version 3.0 von Greg Browne.
Launch	U	Beispielprogramm von Peter da Silva, das zeigt, wie ein Programm von der Workbench geladen werden und nach Beendigung ins CLI zurückkehren kann. Mit Sourcecode.
TSnip	T	Unterroutine für Editoren oder ähnliche Programme, läßt komfortable Blockoperationen über ein Intuition-Steuerfeld zu. Version 1.4a von John Russell. Geliefert wird nur der Sourcecode zum Einbinden in eigene Programme.
UnixUtil	U	Implementation einiger Unix-Utilities (»Wc«, »Head«, »Tail«, »Tee«, »Detab«, »Entab« und »Trunc«) als CLI-Befehle. Von Gary Brant.
<b>Fish 180</b> Browser	U	Die »Programmierer-Werkbank« von Peter da Silva. Einfaches Kopieren, Umbenennen, Löschen und Versetzen von Files und Directories aus einer CLI-Umgebung. Version 1.6 ohne Sourcecodes.
GeoTime	G	Einige interessante Uhr-Programme. Version 1.0 von Mike Smithwick. Ohne Sourcecode.
GPrint	U	Schwarz-Weiß-Druckroutine für Epson-kompatible Drucker von Peter Cherna. Verschiedene Optionen erlauben das Einstellen von Druckqualität und -dichte. Version 2.03 wird ohne Sourcecode aber mit einigen Beispielen geliefert.
Jed	U	Version 1.0 eines Shareware-Editors von Dan Burris. Dieser hat ein sehr benutzerfreundliches Design mit Word-wrap, Auto-Indent, zweitem Textpuffer und -fenster, benutzerdefinierbare Makros, und einer CLI-Schnittstelle.
NoVirus	U	Version 1.56 eines Virus-Checkers von Luciano Bertato. Erkennt alle bekannten Bootblock-Viren und macht auf Nicht-Standard-Bootblöcke aufmerksam. Bei Virusbefall kann die Diskette mit verschiedenen Optionen »installiert« werden. Geschrieben in Assembler, ohne Sourcecode.
RepString	U	CLI-Hilfsprogramm, das in beliebigen Files beliebige Strings durch andere ersetzt. Version 1.0 von Luciano Bertato, ohne Sourcecode.
TrekTrivia	S	Star-Trek-Fans werden ihre reine Freude haben, für alle anderen - na ja. 100 Fragen werden mitgeliefert, weitere Disketten sind vom Autor zu erhalten (Shareware). Von George Broussard, ohne Sourcecode.
<b>Fish 181</b> AMXLisp	P	Speziell an den Amiga angepaßte Version von XLisp. Das Original ist in C geschrieben und stammt von David Michael Betz. Amigapezifische Erweiterungen sind ebenfalls in C und wurden von Francois Rouaix programmiert. Sourcecodes und viele Beispiele werden mitgeliefert. AMXLisp ist ein sehr leistungsfähiger Lisp-Interpreter, der sich

Programm:	Art:	Bemerkungen:
Bally	S	weitgehend an den Common-Lisp-Standard hält wie er in dem Buch »Lisp« von Winston und Horn (Addison-Wesley-Verlag) beschrieben wird. Amiga-Version des Spiels »Click«. Sound-Effekte sollen in späteren Versionen eingebaut werden. Version 0.1 von Oliver Wagner ohne Sourcecode.
Tracker	T	Einige hilfreiche Debugging-Routinen zum Überwachen und Aufzeichnen aller Aufrufe von »AllocMem()« und »FreeMem()«. Version 0.0a mit Sourcecode.
<b>Fish 182</b> AMC	U	Das »Amiga Message Center« von Foster Hall. Scrollt ein Textfile vor farbigem Hintergrund über den Bildschirm. Version 1.0 ohne Sourcecode.
Edimap	U	Ein Keymap-Editor von Gilles Gamesh. Liest existierende Tastaturtreiber, die verändert und zurückgeschrieben werden können. Sourcecode wird mitgeliefert.
HR136	G	IFF-Grafik mit allen möglichen Mischungen basierend auf 16 Grundfarben. Von Dick Bourne.
Iconmerger	G	Mischt ein Icon aus zwei beliebigen Brushes. Version 2.0 ohne Sourcecodes von Terry Gintz.
Sam	U	Abspielprogramm für Sounds, die im IFF-Format gespeichert sind, von Nic Wilson. Verschiedene Optionen können über die Kommandozeile übergeben werden. Einige Beispiele werden mitgeliefert. Version 1.0 ohne Sourcecode.
SetFont	U	Erlaubt das Wechseln des Systemfonts. Version 2.5 von David Haynie enthält die Sourcecodes in »C++«. Die Fehler der Version 2.0 von Fish 75 sind beseitigt.
<b>Fish 183</b> FixFd	U	Macht die »fd«-Dateien für Assembler-Programmierer zugänglich. Die FD-Dateien werden zu Sprungtabellen konvertiert, die per »Include« eingebunden werden statt die »Amiga.lib« zu verwenden. Das Programm von Peter Wyspianski ist im Sourcecode enthalten.
MkLib	T	Diese Library stellt eine Weiterentwicklung der »Elib« von Fish 87 dar. Außerdem wird Edlib mitgeliefert, die nicht in den Manx Standard-Libraries enthalten ist. Einschließlich Sourcecode. Von Edwin Hoogerbeets, sowie einigen anderen.
PCQ	P	Ein kleiner Pascal-Compiler von Patrick Quaid, der den Pascal-Sprachumfang größtenteils unterstützt. Im Moment sind Sets, Unterbereiche und die »With«-Anweisung nicht implementiert. Dafür werden Includes, externe Referenzen, Records, Aufzählungstypen, Zeiger, Arrays und alle wichtigen Pascal-Standards unterstützt. Mit Sourcecode und Beispielen.
<b>Fish 184</b> BI	T	Ein einfacher Konverter von Terry Gintz, der Brushes zu Image-Strukturen in C-Code wandelt. Version 1.0 ohne Sourcecode.
CardMaker	T	Routinen von Terry Gintz, die Kartengrafiken für ein 52er Spielkartenset erzeugen. Version 1.0 ohne Sourcecode.
DPS	U	Demo-Version eines Programms von Foster Hall, das IFF-Files als aufrufbares Programm speichert. Bei Aufruf wird das Bild angezeigt. Version 1.0 ohne Sourcecode
MouseUtil	U	Erlaubt das Verändern der Mausgeschwindigkeit, ohne den Umweg über



Programm:	Art:	Bemerkungen:
VacBench World	U S	Preferences zu gehen. Version 1.1 mit Assembler-Sourcecode. Räumt den Bildschirm auf, wenn er zu voll wird. Ohne Sourcecode von Randy Jouett. Text-Adventure von Doug McDonald und Eric Kennedy. Hat Ähnlichkeiten mit den Infocom-Adventures »Planetfall« und »Starcross«. Erstaunliche Vielfalt an Antworten. Version 1.02 mit Sourcecode.
Fish 185		Diese Diskette ist eine Kopie der offiziellen Commodore-IFF-Disk von November 1988. Die Dokumente sind mit Zoo gepackt.
Fish 186 A68k	P	Dieser 68000er-Assembler wurde ursprünglich von Brian Anderson in Modula geschrieben und im »Dr. Dobb's Journal« veröffentlicht. Charlie Gibb konvertierte den Source nach C und für den Amiga. A68K verarbeitet Metacomco-kompatiblen Sourcecode. Version 2.42 ist ein Update der Version von Fish 110 und enthält den Sourcecode.
Card's O'Rama	S	Einfaches Spiel von Werther Pirani, wird mit 32 Karten gespielt, die in 16 Paaren vorliegen. Sie müssen möglichst viele Paare erhalten, bevor der Screen leer ist. Version 1.0 einschließlich Sourcecode.
Qt2	U	Eine etwas andere Uhr von Charlie Gibb. Sie gibt Ihnen die Zeit in der Form an, wie es die meisten Leute tun würden (statt »16 Uhr 50« heißt das »Es ist 10 vor fünf«).
SimCPM	A	Sourcecode in Assembler wird mitgeliefert. CP/M-Simulator von Jim Cathey (Amiga-Übertragung von Charlie Gibbs und Willi Kusche). Simuliert einen 8080-Prozessor und ein H19-Terminal. Version 2.3 ist ein Update von Fish 109.
Fish 187 Diskperf	U	Dieses Benchmarkprogramm von Rick Spanbauer und Joanne Dow läuft sowohl auf dem Amiga als auch auf Unix. Diese Version ist ein Update von Fish 48 mit einigen Fehlerkorrekturen und einer größeren Meßgenauigkeit. Arbeitet mit dem Fast File System zusammen.
Hacklite	S	Eine Übertragung des bekannten Spiels »Hack« auf den Amiga, die dessen besondere Fähigkeiten besser nutzt als ältere Versionen. Version 1.0.0 ohne Sourcecode. Von der Software-Destillery.
Mackie	U	Ein anpassungsfähiger CLI-Ersatz von Thomas Rokicki, ähnlich »PopCLI«. Als »Dreingabe« ist ein »Screen-Blanker« enthalten. Hier ist die Version 1.13 enthalten, ein Update von Fish 161.
SetCPU	U	Erlaubt dem Benutzer, die systemnahe Umgebung seines Prozessors nach eigenen Wünschen mit Parametern zu versorgen, wenn er eine andere als die 68000-CPU verwendet (beispielsweise den 68030-Chip). Version 1.4 von Dave Haynie mit Sourcecode.
Fish 188 BootIntro	U	Ermöglicht Ihnen, einen eigenen Vorspann in den Bootblock zu schreiben. Die Überschrift kann bis zu 20 Buchstaben lang sein, der Text mit bis zu 255 Zeichen wird gescrollt. Von Roger Fischlin.
DiffDir	U	Vergleicht den Inhalt zweier Directories und zeichnet die Unterschiede auf. Erkennt und angezeigt werden Unterschiede in den Einträgen der Filenamen. Version 1.0 mit Sourcecode von Mark Rinfret.

Programm:	Art:	Bemerkungen:
ExecDis	U	Generiert ein disassembliertes ROM-Listing der Exec-Library und versieht es mit Kommentaren. Version 1.0 ohne Sourcecode von Markus Wendel.
FastGro	G	Simulationsprogramm von Doug Houck basierend auf einem Artikel in der Zeitschrift »Scientific American« (»Computer Recreations«). Verwendet Fraktale zur Darstellung und ist angeblich wesentlich schneller als das Original.
FracGen	G	Fraktal-Generator von Doug Houck, der die fraktalen Bilder aus vom Anwender vorgegebenen Wurzeln berechnet. Version 1.23 wird ohne Sourcecode geliefert und ist ein Update der Version auf Fish 142.
Memory-Clock	U	Zeigt neben der aktuellen Uhrzeit das verfügbare Fast- und Chip-RAM an. Sourcecode in Assembler wird mitgeliefert. Von Roger Fischlin.
MinRexx	P	Dieses einfache ARexx-Interface kann sehr einfach in nahezu jedes Programm eingebunden werden. Als Beispiel ist ein Programm von Fish 1 enthalten. Daneben wird der Sourcecode mitgeliefert. Von Tomas Rokicki.
Null	T	Dieses neue DOS-Device verhält sich im wesentlichen wie »NIL:«, allerdings ist es im Gegensatz zu NIL ein echter Handler. Dadurch kann es auch in Situationen angewendet werden, in denen NIL versagt. Sourcecodes werden mitgeliefert. Von Gunnar Nordmark.
TextDisplay	A	Dieses Text-Anzeigeprogramm von Roger Fischlin arbeitet ähnlich »more« oder »less«. Es benötigt nur halb so viel Platz und erkennt alle Screen-Formate.
Fish 189 Mackie	U	Ein anpassungsfähiger CLI-Ersatz von Thomas Rokicki, ähnlich »PopCLI«. Als »Dreingabe« ist ein »Screen-Blanker« enthalten. Hier ist die Version 1.20 enthalten, ein Update der Version 1.13 von Fish 187.
NetHack	S	Teil 1 des zweiteiligen NetHack Version 2.3 von Olaf Seibert und anderen. Teil 2 finden Sie auf Fish 190. Beide Teile zusammen waren – selbst in gepacktem Zustand – zu groß für eine Diskette.
Uedit	U	Version 2.4g des Shareware-Editors von Rick Stiles. Läßt sich in hohem Maß den Bedürfnissen des Anwenders anpassen. Es wird kein Sourcecode geliefert. Diese Version ersetzt die Beta-Version von Fish 173.
Fish 190 Garylcons	G	Eine Zusammenstellung von interessanten und nützlichen Icons von Gary Roseman.
ILBM2Image	T	Macht aus IFF-Bildern C-Sourcecode, den Sie direkt in eigene Programme einbinden können. Die Bilder werden dann mit der »DrawImage«-Funktion angezeigt. Von Denis Green.
NetHack	S	Teil 2 von NetHack Version 2.3 von Olaf Seibert und anderen. Den ersten Teil finden Sie auf Fish 189 (mit Sourcecode).
Fish 191 BlitLab	T	Mit diesem Programm von Thomas Rokicki können Sie relativ gefahrlos mit dem Blitter experimentieren, ohne dafür erst ein Programm schreiben zu müssen. BlitLab öffnet ein Workbench-Fenster mit einem Gadget für jedes Blitter-Register. Sie können alle erdenklichen Manipulationen durchführen und die Ergebnisse beobachten – fast ohne Absturzgefahr. Sourcecode wird mitgeliefert.



Programm:	Art:	Bemerkungen:
Blk	T	Requester-Konstruktions-Set von Stuart Ferguson. Dieses Programm konvertiert Textfiles zu C-Sourcecode, der in eigene Programme eingebunden werden kann. Interessant ist vor allem der verwendete rekursive Algorithmus. Mit Sourcecode.
FileBoot-Block	T	Dieses Programm von John Veldthuis liest Block 0 und 1 einer bootfähigen Diskette und legt sie in einem Programmfile ab. Dieses können Sie starten (wenn Sie den Guru aufschrecken wollen) oder mit einem Disassembler untersuchen. Der Assembler-Sourcecode ist enthalten.
ISpell	A	Spell-Checker, der von Unix übertragen wurde. Das Programm von Pace Willisson wurde von Thomas Rokicki erweitert. Diese Version 2.0.02 ist ein Update der Version von Fish 54. Der Sourcecode ist enthalten.
Pz15	S	Sehr einfaches Computerspiel von Mike Hall. Imitation der 15teiligen, quadratischen Plastik-Puzzles, mit Sourcecode.
<b>Fish 192</b> Eval	A	Paket von David Gay zur Auswertung von mathematischen Ausdrücken. Daneben werden einige grundlegende Vereinfachungen durchgeführt, die die Ergebnisse anschaulicher machen. Sourcecode wird mitgeliefert.
PacMan87	U	Ein Pacman-ähnliches Spiel von Steve Jacobs und Jim Boyd mit vielen zusätzlichen Hindernissen, die überwunden werden müssen. Abschaltbarer Sound, drei Schwierigkeitsstufen, High-Score-Liste. Das Programm ist Shareware. Ohne Sourcecode.
ReSource-Demo	P	Demo-Version (0.36) eines interaktiven Disassemblers von Glen McDiarmid. Vollständig lauffähige Version, nur die »Save«-Funktion ist abgeschaltet.
<b>Fish 193</b> keyMapEd	U	Voll lauffähiger Keymap-Editor von Tim Friest (Version 1.02). Sie können beliebige Tasten(-kombinationen) mit beliebigen Funktionen oder Texten belegen. Die editierten Keymaps können mit dem Setmap-Befehl eingestellt werden. Einschließlich Sourcecode.
Zc	P	Modifizierte Version des Sozobon C-Compilers von der Fish-Disk 171 (Version 1.01) von Johann Ruegg und Joe Montgomery. Der generierte Code ist jetzt kompatibel und mit dem A58k-Assembler von der Fish 186. Außerdem erleichtert ein neues Frontend den Gebrauch des Compilers. Sourcecode wird mitgeliefert.
<b>Fish 194</b> Moria	S	Höhlen-»Simulation« von Alan Koeneke, Richard Henderson und anderen für einen Spieler. Arbeitet mit verschiedenen Modi, viele neue Features, die die Besonderheiten des Amiga nutzen. Version 3.0, benötigt mindestens 1 MByte Speicher. Ohne Sourcecode.
<b>Fish 195</b> MicroEmacs	U	Version 3.10 dieses bekannten Editors. Das Programm stammt von Dave Conroy, Daniel Lawrence hat viele Erweiterungen programmiert. Auf dieser Diskette finden Sie ein Update zu der Version von Fish 119. Die neuesten Erweiterungen bestehen aus mehrfachen Marken, besserer Funktions-tastenunterstützung und viele andere Erweiterungen. Mit Sourcecode und einer ausführlichen Online-Hilfe.

Programm:	Art:	Bemerkungen:
<b>Fish 196</b> HamPics	G	Einige interessante IFF-Grafiken, die mit einem Scanner bei einer Auflösung von 4096 x 2800 Pixel eingelesen, nachbearbeitet und im IFF-HAM-Format gespeichert wurden. Von Jonathan Hue (mit Sourcecode).
<b>Fish 197</b> CTags	U	Unterstützt den Programmierer durch das Extrahieren eines sogenannten »Tag«-Files aus dem Sourcecode, die in einem damit zusammenarbeitenden Editor weiterverarbeitet werden können. Dadurch kann schnell an eine korrespondierende Stelle beispielsweise in einem C-Include-File gesprungen werden. Von Ken Arnold, Jim Kleckner und Bill Joy (V4.7 mit Source).
Find	U	Sucht ein Device nach einem angegebenen Filenamen an. Hat viele Ähnlichkeiten mit dem Unix-»Find«. Version 1.2 wird mit Sourcecode geliefert. Von Rodney Lewis.
FixHunk	T	Modifiziert ausführbare Files, um sie im Fast-Memory laufen zu lassen. Es legt alle »DATA«- und »BSS«-Hunks ins CHIP-Memory. Alle »CODE«-Hunks werden dagegen ins FAST-RAM geladen (falls verfügbar). Version 2.1 wird ohne Sourcecode ausgeliefert. Update zu Fish 36 von D.J. James.
Nro	U	Version 1.5 eines Textformatierers von einem unbekanntem Autoren.
Stevie	U	Erweiterungen stammen von Olaf Seibert. Ein Editor mit ähnlichen Eigenschaften wie der Unix-Editor »vi«. Erlaubt das Verändern der Fenstergröße, den Gebrauch der Cursorstasten und der Helptaste. Version 3.35a ist ein Update der Version von Fish 166. Sourcecode wird mitgeliefert.
<b>Fish 198</b> Charon	G	»The First Annual Badge Killer Demo Contest«. Der Text stammt von Lord Dunsany (geschrieben 1915), das Programm von Bradley Schenk (1988).
<b>Fish 199</b> ASimplex	T	Implementiert den »Simplex«-Algorithmus zur Lösung linearer Probleme. Version 1.2 von Stefan Forster. Der Sourcecode wird mitgeliefert.
Csh	U	Version 3.02a einer Shell von Matthew Dillon, Steve Drew, Carlo Borreo und Cesare Dieni. Eine Weiterentwicklung von Matthew's Csh-Shell Version 2.07. Sourcecode wird mitgeliefert.
MIDIsoft	A	Dieses Programm transferiert Sound-Files vom Amiga auf eine Roland S-220. Ohne Sourcecode von Dieter Bruns.
Pyro	U	Screenblanker, der nicht nur einfach verdunkelt. Sehen Sie sich's an! Von Steve Jacobs und Jim Boyd.
SnipDemo	T	Demo-Version eines Signal-Prozessors, der von Digital Dynamics verkauft wird. Ohne Sourcecode von John Hodgson.
Viewer	U	Kleines Programm zur Anzeige von IFF-Bildern eines beliebigen Formats. Das Programm ist etwas weniger als 1 KByte lang und wurde in Assembler geschrieben. Ohne Sourcecode von Mike McKittrick.
<b>Fish 200</b> NotBoing-Again	G	»Dr. Gandalf's entry for the First Annual Badge Killer Demo Contest«. Animation im Interlace-Modus mit HAM-Bildern. Am besten sehen Sie sich's an. Benötigt mindestens ein MByte.



## Jetzt können Sie einpacken

**W**er hat sich noch nicht geärgert über die Datenflut, die schon nach ersten Gehversuchen mit dem Amiga entsteht. Verschiedene Workbenchversionen, Anwendungsprogramme, endlose Datenfiles und so manches Public Domain-Programm machen sich den Platz auf den anfangs meist knapp bemessenen Disketten streitig. Daß diese Datenflut mit zunehmendem Alter des Amiga keineswegs abnimmt, ist kein Geheimnis. Die Frage ist also, was tun?

Die einzig vernünftige Lösung ist das Archivieren der nicht ständig benötigten Files mit einem Packer. »Zoo« archiviert alle Arten von Files. Auf diese Weise sparen Sie durchschnittlich 50 Prozent Speicherplatz.

Auf der Fish-Disk 164 finden Sie die neueste Version von Zoo (2.00). Diese ist auch auf der Programmservice-Diskette (siehe Seite 33) enthalten. Um sich einen Überblick zu verschaffen, kopieren Sie zunächst Ihre Diskette mit allen Zoo-Dateien. Darauf starten Sie das CLI und kopieren das File »Zoo« in die RAM-Disk. Sollten Sie Probleme mit den dafür benötigten Befehlen haben, schlagen Sie bitte auf Seite 76 nach.

Wechseln Sie jetzt in das Verzeichnis Zoo Ihrer Programmservice- oder Fish-Diskette. Es enthält neben den drei Programmen Zoo, »fiz« und »uncd« einige Textfiles und Info-Dateien. Starten Sie Zoo mit der Option h, um die Hilfsseite auf den Bildschirm zu bringen. Geben Sie dazu ein:

```
zoo h
```

Die jetzt erscheinende Bildschirmseite (Bild 1) hat ein etwas verwirrendes Aussehen. Lassen Sie sich davon nicht durcheinanderbringen. Solange Sie mit Zoo noch nicht viel Übung haben, benutzen Sie am besten nur die Kommandos der letzten beiden Zeilen. Diese sind für Zoo-Anfänger gedacht und sind daher so gewählt, daß sie sich leichter einprägen. Je weiter Sie fortgeschritten sind, desto mehr Kommandos kürzen Sie ab und um so mehr erweiterte Optionen können Sie

**Bringen Sie Ihre Daten in den »Zoo«. Packen Sie sie auf engstem Raum zusammen, anstatt sie frei in Ihrem Disketten-Speicher wildern zu lassen.**

nutzen. Wenn Sie nicht mehr weiter wissen oder wenn Ihnen etwas entfallen ist, gibt es immer noch den Griff zur Hilfsseite (mit dem Kommando »Zoo h«).

Zur Übung packen Sie am besten Ihr erstes File mit der Zeile

```
cd ram:
zoo -add test df0:
zoo/zoo.man
```

Diese Befehlszeile hat bei Zoo immer dasselbe Format. Am Anfang steht der Programmname. Darauf folgt ein Schlüsselwort, das festlegt, was ausgeführt werden soll. Der nächste Name bezeichnet das Archiv-File und der letzte das Original. Das Schlüsselwort »-add« fügt einem bestehenden Archiv-File (»test«) einen

weiteren Eintrag (»zoo.man«) hinzu. Wenn kein Archiv-File dieses Namens besteht, wird es angelegt. Auf dem Bildschirm sehen Sie die Ausgabe

```
Zoo: zoo.man -- (64%)
added
```

Sie besagt, daß das File »zoo.man« dem gewünschten Archiv-File aus dem aktuellen Directory, das in der Eingabezeile angegeben wird, angefügt wurde. Die Zahl in Klammern bedeutet, daß »zoo.man« um 64 Prozent verkleinert wurde. Überzeugen Sie sich mit Hilfe des List-Befehls, daß dies tatsächlich der Fall ist. Das File zoo.man hat eine Länge von 60508 Byte, während »test.zoo« (die Endung »zoo« wird beim Packen automatisch angehängt) gerade

noch 21661 Byte Länge aufweist. Bevor Sie in Ihrer Euphorie alle erreichbaren Files packen, beachten Sie, daß das File sich ziemlich verändert hat. Laden Sie zoo.man in einen beliebigen Editor. Versuchen Sie das gleiche mit test.zoo. Welches Ergebnis erhalten Sie? Werfen Sie Zoo deswegen nicht gleich auf den Müll! Es gibt auch eine Möglichkeit, die archivierten Files wieder in den Urzustand zu bringen.

Geben Sie dazu ein:

```
cd ram:
zoo -extract test.zoo
zoo.man
```

Sie finden in der RAM-Disk jetzt beide Files (»test.zoo« und »zoo.man«). Zoo.man ist identisch mit dem ursprünglichen File. Aber ein gepacktes File kann nicht nur einen Eintrag enthalten. Geben Sie ein:

```
zoo -add test df0:zoo/
fiz.man
```

Damit fügen Sie dem Archiv »test« ein weiteres File hinzu.

**AmigaDOS**

Usage: zoo {acDeglLPTuUvX}[aAcCdEfInMNoOpPqul:/.@n] archive file  
 ("zoo h" for help)

Choose a command from within {} and zero or more modifiers from within [].  
 E.g.: 'zoo a save /bin/\*' will archive all files in /bin into save.zoo.  
 (Please see the user manual for a complete description of commands.)

<b>Commands in {} mean:</b>	<b>Modifiers in [] mean:</b>
a add files	A show archive name(s) in listing
c update comments	a apply g or c to archive
D delete stored files	c add/list comments
e,x extract files	d extract/list deleted files too
g adj. gen. limit/count	dd extract/list only deleted files
l,L,v,V list filenames	E erase backup after packing
P pack archive	f fast add (no compression) or list
T fix archive datestamp	M move when adding (erase original)
u add only newer files	n add only files not already in archive
U undelete stored files	N send extracted data to Neverland
f act as filter	c/u compress/uncompress as filter
q be quiet	o don't ask "Overwrite?"
: don't store dir names	p pipe extracted data to standard output
. pack to current dir	/,// extract full pathnames
C show file CRC value	I add filenames read from stdin
S overwrite newer files	+/- enable/disable generations
P pack after adding	g list generation limits
	en start extract/list at position n

Novice usage: zoo -cmd archive[.zoo] file... where -cmd is one of these:  
 -add -extract -move -test -print -delete -list -update -freshen -comment  
 1)

Bild 1. Die Hilfsseite von »Zoo« befreit Sie davon, alle Schlüsselwörter im Kopf zu behalten



Length	CF	Size	Now	Date	Time	
60508	64%	21492	22 Nov 88	18:31:14	f12f	zoo.man
5293	57%	2253	22 Nov 88	18:31:18	a6e2	fiz.man
10072	23%	7752	22 Nov 88	18:30:42	f394	fiz
2566	73%	682	22 Nov 88	18:31:22	db0d	fiz.man.info
2566	73%	680	22 Nov 88	18:30:46	e8e1	zoo.man.info
41428	26%	30555	22 Nov 88	18:30:34	4919	zoo
6011	50%	3034	22 Nov 88	18:31:26	7bab	Amiga.notes
2566	73%	680	22 Nov 88	18:30:14	1582	Amiga.notes.info
2566	73%	684	22 Nov 88	18:30:18	d2c7	Announcement.info
224	17%	185	22 Nov 88	18:30:48	df5b	Announcement
646	32%	437	22 Nov 88	18:30:12	28fc	changes
2566	73%	682	22 Nov 88	18:30:54	0770	changes.info
2566	73%	682	22 Nov 88	18:30:38	41b3	README.info
464	29%	330	22 Nov 88	18:30:10	a278	README
92	14%	79	22 Nov 88	18:30:52	48c4	uncd
703	17%	647	22 Nov 88	18:31:30	1463	POSTER
140917	50%	70054	16 files			

Bild 2. So sollte Ihr gepacktes Directory nach dem Bearbeiten mit »Zoo« aussehen

Allerdings läßt sich jetzt im Inhaltsverzeichnis nicht mehr erkennen, welche Files im Archiv liegen. Mit dem »dir«-Befehl erhalten Sie nur den Namen »test.zoo«. Geben Sie folgende Zeile ein:

```
zoo -list test
```

Sie erhalten ein »Inhaltsverzeichnis« aller enthaltenen Files. Im Moment sind dies zoo.man und fiz.man. In der ersten Spalte sehen Sie die Länge des entpackten Files in Byte, in der zweiten den gepackten Anteil in Prozent. Die dritte Spalte informiert über die aktuelle Größe, die vierte und fünfte über das Datum und die Zeit der letzten Veränderung des Original-Files. In der letzten Spalte finden Sie den Namen des Originals, der wiederverwendet wird, wenn die Datei restauriert wird.

Fügen Sie jetzt dem Archiv einige Files hinzu. Packen Sie

Listen Sie nochmals das Archiv-File, um zu überprüfen, ob alle Dateien enthalten sind. Sie sollten jetzt die Ausgabe von Bild 2 auf dem Bildschirm sehen. In der letzten Spalte sehen Sie die Namen aller Files.

Sicher haben Sie sich beim Archivieren der Files gewundert, was diese enthalten. Alle Dateien mit der Endung ».info« sind nur für die Workbench. Sie enthalten alle Daten der Icons für das jeweilige Programm. In den beiden Files mit der Endung ».man« finden Sie die Dokumentation zu den Programmen »fiz« und »zoo«. Die »Amiga.notes« informieren über Amiga-spezifische Besonderheiten von Zoo. »Changes« sind eine Art Kurz-Information für jeden, der Zoo bereits aus früheren Versionen kennt. Alle Änderungen sind hier zusammengefaßt. »Readme« ist eine Liste aller Dateien, und »Poster« die

dos aus Tabelle 1 beginnen mit einem anderen Zeichen. Zoo erkennt die Eingabe des Minuszeichens mit einem Buchstaben dahinter bereits als Schlüsselwort. Mit diesen zehn Schlüsselwörtern sind Sie in der Lage, alle wichtigen Manipulationen in Ihren Archiven durchzuführen.

Wenn Sie die erweiterten Kommandos nutzen wollen, finden Sie eine ausführliche Dokumentation im File »zoo.man«. Allerdings sollten Sie vorher et-

was Erfahrung mit dem leichter zu handhabenden Standard-Befehlssatz sammeln.

Oft werden Sie auf Public Domain-Disketten Dateien finden, die nicht mit Zoo, sondern mit dem Programm »Arc« gepackt sind (beispielsweise »AnalytiCalc« von Seite 98). Dieses brauchen Sie natürlich zum Entpacken derselben Dateien. Sie finden Arc auf der Fish-Disk 58. In der Bedienung gibt es kaum Unterschiede zu Zoo. Allerdings bestehen bei Arc einige Einschränkungen, die Sie der Dokumentation der Fish 58 (dem enthaltenen »readme«-File) entnehmen können.

Auf der Programmservice-Diskette sind alle Files im Zoo-Format und können mit diesem Packer untersucht und entpackt werden. Wenn Sie häufig mit Zoo packen und entpacken, ist es sicher am besten, wenn Sie dieses Programm in den C-Ordner Ihrer Workbench kopieren.

Nach dem Lesen dieses Artikels können Sie Ihre Datenbestände durch effektiveres Speichern um 50 Prozent reduzieren. Lassen Sie Ihre Dateien nicht mehr zu diskettenfressenden Ungetümen heranwachsen, sondern packen Sie sie vorher in den Zoo. so

## Programmservice

Auf unserer Programmservice-Diskette sind alle enthaltenen Files mit »Zoo« gepackt. Bevor Sie diese Diskette entpacken, sollten Sie diesen Artikel aufmerksam lesen.

Natürlich wird dadurch die Arbeit mit der Diskette etwas unkomfortabler, aber wir hätten ohne diese Maßnahme höchstens die Hälfte der Programme unterbringen können.

Sollten Sie noch Probleme haben, vom CLI aus das Directory zu wechseln, eine Datei anzeigen zu lassen und Befehle einzugeben, dann finden Sie auf Seite 76 einen Grundlagenartikel zu diesem Thema.

alle im Directory »zoo« enthaltenen Dateien in Ihr Archiv. Sie werden feststellen, daß die gepackten Files um unterschiedliche Anteile komprimiert werden. Dies liegt an der unterschiedlichen File-Struktur. Textdateien können meist enger gepackt werden als Programmdateien. Außerdem kann bei längeren Dateien meist mehr gespart werden als bei kurzen.

Mailbox-»Anschrift« des Autors, Rob Tillotson.

Sehen Sie sich in Bild 1 nochmals die Befehlsliste an, sie enthält weit mehr Befehle als die drei bisher behandelten. In Tabelle 1 finden Sie eine Aufstellung der wichtigsten Kommandos und deren Wirkungsweise.

Sie können alle Kommandos auch abkürzen. Alle Komman-

Schlüsselwort	Funktion
-add	Fügt einem bestehenden Archiv ein neues File hinzu. Existiert das angegebene Archiv noch nicht, so wird es angelegt.
-comment	Fügt jedem archivierten File einen Kommentar an. Dieser kann bis zu 65535 Zeichen lang sein. Wenn bereits ein Kommentar bestand, wird dieser von Zoo vorgeschlagen. Wenn Sie den Kommentar nicht verändern wollen, geben Sie nur <RETURN> ein.
-delete	Löscht die angegebenen Einträge aus dem Archiv.
-extract	Restauriert ein archiviertes File aus dem angegebenen Archiv. Wird kein Filename angegeben, restauriert Zoo das ganze Archiv und schreibt das Ergebnis in das aktuelle Directory.
-freshen	Überschreibt einen älteren Eintrag mit einem neuen File gleichen Namens.
-list	Gibt Informationen über die archivierten Files aus (siehe Text).
-move	Fügt das angegebene File einem bestehenden Archiv hinzu. Danach wird das Original-File gelöscht.
-print	Restauriert ein archiviertes File (wie »-extract«), allerdings werden die restaurierten Daten direkt auf den Standard-Ausgabe-Kanal geschickt (meist der Bildschirm).
-test	Testet, ob die archivierten Daten restaurierbar sind oder ob beim Versuch sie zu restaurieren ein Fehler auftritt. Wenn ja, werden diese Fehler angezeigt.
-update	Testet, ob ein File gleichen Namens bereits archiviert ist. Wenn nicht, wird es angelegt. Im andern Fall wird es überschrieben, wenn es älteren Datums war.

Tabelle 1. Experimentieren Sie ausgiebig mit diesen Schlüsselwörtern, um einen Eindruck von »zoo« zu erhalten



**Transparenz ist nicht die**

**Stärke von Amiga-DOS.**

**Vollen Zugriff haben Sie nur**

**auf Textfiles, interne**

**Zusammenhänge bleiben**

**im verborgenen.**

**Mit »DiskX« behalten Sie die**

**Kontrolle über Ihre**

**Massenspeicher.**



Von Christian Wolf

**M**it »DiskX« haben Sie Zugriff auf alle internen Informationen Ihrer Disketten oder Festplatten. Was Sie schon immer über Amiga-DOS wissen wollten, bleibt nicht länger verborgen. Ob Sie wissen wollen, wie die Directory-Struktur aussieht oder wie einzelne »Hunks« verkettet sind, ob Sie Disketten mit Schreibfehlern retten wollen oder einem Virus auf der Spur sind – DiskX macht's möglich.

DiskX von der Fish 158 ist ein Disketten-Monitor, der den Vergleich mit kommerziellen Produkten nicht zu scheuen

# GLASNOST für Ihre Disketten

braucht. Sein Hauptanwendungsgebiet ist das sektororientierte Editieren von Disketten. Aber auch Files können indirekt verändert werden. Vielleicht sind Sie noch im Zweifel über Sinn oder Unsinn eines solchen Programms. Einige gut durchdachte Funktionen sorgen dafür, daß DiskX nicht zur Untätigkeit verbannt wird, wie

Sie gleich noch sehen werden. Bevor Sie sich diesen Disk-Monitor näher ansehen, beachten Sie folgendes: Die neueste Version benötigt die »arp.library« von der Fish-Disk 123 (oder RPD 122). Kopieren Sie diese in den »Libs«-Ordner Ihrer Start-Diskette. Diese Library stellt dem Programmierer viele interessante Funktionen zur Verfü-

gung. Fehlt sie bei Aufruf von DiskX, dann werden Sie durch eine Fehlermeldung darauf aufmerksam gemacht, und das Programm kann nicht gestartet werden. Falls Sie die arp.library aus irgendwelchen Gründen nicht installieren wollen, finden Sie auf Fish 71 eine Version von DiskX, die ohne diese Bibliothek auskommt.

Nach dem Start von DiskX wird ein eigener Screen geöffnet (Bild 1) und das aktuelle Verzeichnis auf »df0:« gesetzt. Damit sind wir gleich bei einer Besonderheit des Disk-Monitors: Es lassen sich nicht nur, wie bei anderen Programmen dieser Art üblich, Diskettenlaufwerke ansprechen, sondern jedes be-



liebige eingebundene Device, zum Beispiel die resetfeste RAM-Disk »RAD:« unter Kickstart 1.3 oder die Festplatte »dh0:«. Das gewünschte Device wählen Sie im Menü »Unit«, in dem sämtliche »gemounteten« Devices nicht nur durch den Device-Namen, sondern auch durch die entsprechende Treiber-Software (beispielsweise »ASDG-RAM« für die »vd0:«) repräsentiert sind. Es werden nur Geräte angezeigt, die in Spuren, Sektoren und Köpfe unterteilt sind – mit einem Disketten-Monitor die Devices »prt:«, »ser:« oder »par:« zu untersuchen, wäre nicht sehr sinnvoll.

## Daten werden roh gelesen

DiskX kennt zwei grundsätzlich verschiedene Betriebszustände, und zwar den Hex- und den Interpretier-Modus. Im Hex-Modus bekommt der Benutzer den tatsächlichen Inhalt eines Sektors zu sehen, das sind üblicherweise 512 Byte. Devices, die hiervon abweichen, können von DiskX in der aktuellen Version noch nicht korrekt gelesen werden. Da Sie mit der normalen Bildschirmauflösung nur die Hälfte der Daten von einem Block beziehungsweise Sektor auf einmal darstellen können, schalten Sie mit Hilfe des »Other Half«-Gadgets zwischen den beiden Hälften hin und her.

Der Cursor kann sowohl mit der Maus als auch mit Hilfe der Cursortasten an eine beliebige Stelle des aktuellen Datenblockes gesetzt werden. Haben Sie sich dazu entschieden, eine bestimmte Stelle zu verändern, können Sie nach Druck auf <RETURN> Ihre Änderungen im ASCII-Format eingeben – diese Eingabe wird ebenfalls mit <RETURN> wieder abgeschlossen. Wer nur einige Bytes ändern möchte, kann nach Betätigung des »Modify«-Gadgets diese auch in dezimaler, hexadezimaler oder sogar binärer Form eingeben.

Wie bereits kurz erwähnt, hat DiskX jedoch noch einen zweiten Darstellungsmodus. Das »Show Type«-Gadget versetzt Sie in diesen.

Hier versucht DiskX sich einen Sinn aus den Daten jedes Blocks zusammenzureimen. Dazu werden alle Daten in gleicher Weise interpretiert, wie es auch Amiga-DOS versuchen würde. Es wird also zwischen »File Header Blocks«, »User Directory Blocks«, »File List Blocks«, »Root Blocks« und

»Data Blocks« unterschieden, die in einer für den Benutzer verwendbaren Form dargestellt werden. So werden beispielsweise von einem Datenblock nur die 488 relevanten Datenbytes angezeigt, bei einem »File Header Block« dagegen erhalten Sie Aufschluß über Filenamen, Protection, Länge, Kommentar, Datum, erster Datenblock und ähnliches. Sie können DiskX vom CLI aus mit der Option »INTERLACE« aufrufen. In diesem Fall würden aufgrund der verdoppelten vertikalen Auflösung alle Daten eines

Sie sich einfach dieses Kommandos und gelangen so in den ersten, meist 488 Byte großen Datenblock.

■ **N** oder »Next Block«: Falls Sie nicht nur am ersten Datenblock eines Files interessiert sind, benutzen Sie diesen Befehl, um sich in der Liste der Datenblöcke jeweils zum nächsten zu hangeln und sich diese anzusehen. Das geht natürlich nur solange, bis DiskX den letzten Block gefunden hat, der je nach Länge des Files zwischen 0 und 488 Byte groß sein kann.

■ **P** oder »Parent Block«:

gets in der linken unteren Ecke von »DiskX's Screen« zurückgreifen. Mit diesen ist eine bequeme, wie von Intuition gewohnte Auswahl der Spuren, Sektoren oder Köpfe möglich. In beiden Betriebszuständen läßt sich mit Hilfe des »Write«-Gadgets der aktuelle Sektor zurückschreiben, so daß Sie Ihre Disketten nach eigenen Wünschen modifizieren können.

DiskX verfügt über komfortable Möglichkeiten zum Auffinden von Text-Strings. Hierzu finden Sie ein weiteres Gadget, das nach dem Anklicken ein

DiskX V2.2a Special Boot-Save Edition, by Steve Tibbett

000:	44	4F	53	00	C0	20	0F	19	00	00	03	70	43	FA	00	18	DOS..	.....	PC...
010:	4E	AE	FF	A0	4A	80	67	0A	20	40	20	68	00	16	70	00	N...	J.g.	@ h.p.
020:	4E	75	70	FF	60	FA	64	6F	73	2E	6C	69	62	72	61	72	Nup.	'.	dos.librar
030:	79	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	y.....		
040:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
050:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
070:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
080:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		
0F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....		

Control Panel:

- Block: Other Half, Find...
- Track: Modify, Write
- Sector: Show Type, Quit
- Head: (empty)

Offset: 000  
Value: 68 (\$44)  
Type: Unknown  
Checksum: BAD  
Hit RETURN to Edit...

Bild 1. Auf dem Screen von »DiskX« finden Sie viele vertraute Bedienungselemente

Blockes auf einmal zu sehen sein, so daß Sie sich die Benutzung des »Other Half«-Gadgets sparen können.

Um sich durch den »Datendschungel« eines Devices (beispielsweise Floppy, Hard-Disk oder RAM-Disk) zu wühlen und von einem Block zum nächsten zu gelangen, gibt es verschiedene Befehle. Diese sind sowohl per Tastendruck als auch per Menü-Item oder Gadget selektierbar:

■ **R** oder »This Disk's Root Block«: Mit diesem Kommando wird DiskX angewiesen, den RootBlock des Devices einzulesen. Dieser stellt sozusagen die Wurzel der Fileorganisation dar und wird vom Amiga-DOS als erster Anhaltspunkt zum Auffinden der Daten benutzt.

■ **F** oder »First Block«: Sicher findet man manchmal einen »File Header Block« im endlosen Gewirr der Daten, das bedeutet den Hinweis auf ein File. Um auf die Daten dieses Files Zugriff zu erhalten, bedienen

Möchten Sie dagegen von den Daten eines Files wieder in den Ursprungsblock, also in den »File Header Block«, zurückkehren, dann ist dieser Befehl die Lösung. Befindet sich DiskX schon einige Verzeichnisse »tiefer« in den hintersten Winkeln des Devices, dann gelangen Sie nach mehrmaliger Anwendung von »Parent Block« automatisch in den »Root Block«.

## Wer sucht, der findet

Die letzten vier Kommandos sind nur im Interpretier-Modus anwendbar und lassen recht komfortables Editieren eines Files zu. Wer allerdings auf die Datenorganisation von Amiga-DOS keinen Wert legt oder andere als Amiga-DOS-Disketten editieren will (beispielsweise von Software-Produkten, die ein eigenes Aufzählungsformat verwenden), kann auf die Gad-

Fenster öffnet und den Benutzer dazu auffordert, den zu suchenden Text einzugeben. Danach muß lediglich noch entschieden werden, ob ab der aktuellen Position (»From Here«) oder ab Block 0 mit der Suche begonnen werden soll. Hat man dem eingegebenen Text einen Stern (»\*)« vorangesetzt, wird bei der Suche Groß- und Kleinschreibung nicht unterschieden, DiskX ist dann also »case insensitive«. Wer sich noch nicht so ganz schlüssig ist, wonach er überhaupt suchen soll, der kann mit der Menü-Funktion »Search for any ASCII Text« dem Programm anweisen, nach einer beliebigen ASCII-Zeichenkette zu suchen und diese anzuzeigen.

In starkem Zusammenhang mit der Such-Funktion steht die Möglichkeit dieses Disk-Monitors, bereits gelöschte Dateien zu restaurieren. Um beispielsweise die Datei »dh0:Verloren« wiederzubekommen, führen Sie folgende Schritte durch:



- Selektieren Sie im Menü »Unit« das Item »dh0: device-name.device« und schalten Sie damit auf das entsprechende Device um.

- Klicken Sie das »Find...«-Gadget an, um in dem daraufhin auftauchenden Requester den Namen einer versehentlich gelöschten Datei einzugeben. Beachten Sie dabei, daß dem Namen ein Klammeraffe vorangehen muß, der DiskX mitteilt, daß nach dem »File Header« der angegebenen Datei gesucht werden soll. Eventuell ist es notwendig, vorher mit der »Show Type«-Funktion in den Interpretier-Modus umzuschalten. Die Wartezeit ist von der Größe des gewählten Speichermediums abhängig, die RAM-Disk zu durchsuchen dauert nicht so lange wie für eine 20-

MByte-Platte benötigt wird. Festplattenbesitzer sollten sich für diese Zeit eine andere Beschäftigung suchen, das Warten kann einige Zeit dauern.

- Wählen Sie die Menü-Funktion »Attempt to Recover File«, mit der DiskX angewiesen wird, die Daten zu dem gefundenen »File Header« aufzuspüren. Dazu müssen allerdings sämtliche »Data Blocks« des Devices durchsucht werden, von denen genau einer den Zeiger auf den »File Header« enthält, also den Zeiger auf seinen »Parent Block«. Der passende Block ist der erste Datenblock des Files.

- Nach erfolgreicher Suche erscheint ein File-Requester, der Sie nach einer neuen Datei fragt, in die die wiedergefundenen Daten geschrieben werden sollen. Beachten Sie dabei: Die

neue Datei darf auf keinen Fall in jenem Device erzeugt werden, das das gelöschte File enthält, weil dessen Daten sonst möglicherweise überschrieben werden, bevor sie vollständig ausgelesen sind (der Speicherbereich ist für Amiga-DOS während dieser Zeit als frei gekennzeichnet!). Der beste Platz ist in diesem Fall die RAM-Disk, was im File-Requester daher als Default-Device vorgeschlagen wird. - Mit der Betätigung über das »OK«-Gadget oder mit <RETURN> wird der »Recover«-Prozeß in die Wege geleitet und das File gerettet.

Wer etwas Englisch kann, sollte auf keinen Fall versäumen, einen Blick in die mitgelieferte Anleitung zu werfen. Der Autor dieses Programms - Steve Tibbett - erklärt darin viele

interessante Aspekte von DiskX. Diese Texte sind recht witzig und informativ gehalten, im Gegensatz zu den sonst eher spröden »readme«-Files.

Wenn Sie Ihren Amiga selbst programmieren, sollten Sie sich auf jeden Fall den Sourcecode von DiskX ansehen. Er gibt Ihnen eine Vorstellung von der Arbeitsweise eines sehr guten Programmierers. Auch wenn zum vollständigen Verstehen sehr gute Kenntnisse nötig sind, ist es doch für jeden lohnenswert. In jedem Fall werden Sie mit DiskX eine Menge Einblicke sammeln, die ohne dieses Utility verschlossen bleiben. so

»DiskX« finden Sie auf der Fish-Disk 158, die arp-Library ist auf Fish 123 oder RPD 122 erhältlich.

**M**artin Kopp und Stefan Stempel aus Mannheim sind die geistigen Väter eines PD-Programms, das in jede Sammlung gehört. »TurboBackup V1.00« kopiert komplette Disketten auf eine, zwei oder drei angeschlossene Laufwerke. Das Kopieren einer Quelldiskette auf ein Ziellaufwerk dauert zirka 105 Sekunden. Sind zwei oder drei Zieldisketten parallel zu beschreiben, erhöht sich die Zeit um zirka 40 beziehungsweise 70 Sekunden. Bei der Gestaltung wurde auf nutzlose Grafikspielereien, wie man sie von einigen Konkurrenten kennt, verzichtet. Vielmehr wurde auf Funktionalität und sinnvolle Zusätze Wert gelegt. Allerdings sind mindestens zwei Laufwerke für den Betrieb notwendig. Vielleicht ändert sich das in einer folgenden Version.

Die große Stärke des Backup-Programms liegt in der Datensicherheit. Das beginnt damit, daß zwar ein Schalter zum Desaktivieren des Verify vorhanden ist, dieses aber demonstrativ abgelehnt wird. Die Autoren drücken deutlich aus, daß der Zeitvorteil, der damit verbunden wäre, sich keinesfalls lohnt. Denn was nützen die gewonnenen Sekunden, wenn man sich später über eine fehlerhafte Kopie schwarz ärgert?

Aber dieses ist nicht der einzige Vorteil auf dem Sektor Sicherheit. Werden beim Kopieren Fehler entdeckt, zeigt TurboBackup diese an. Fehlerhafte Spuren führen zu einer Warnung. Das Programm versucht dann viermal automatisch, erneuert diese Spur zu beschrei-

## Kopieren? Aber sicher!

**Ein gutes Kopierprogramm gehört in jede Diskettensammlung. »TurboBackup« kopiert komplette Disketten in weniger als zwei Minuten — und das sicher.**

Von Ralf Sablowski

ben. Sind diese Versuche vergeblich, erscheint im großen Kasten unten die Frage, ob Sie erneut vier Versuche genehmigen. Das wiederholt sich, bis die Schreibversuche erfolgreich sind, oder der Kopierer mit »Abort« beschließt, es aufzugeben. Beim gleichzeitigen Kopieren auf mehrere Disketten kann mit »Remove« das Laufwerk aus dem Rennen genommen werden, auf dem der »Verify Error« auftrat.

Die Prüfroutinen von TurboBackup nehmen ihre Aufgabe sehr ernst. Fehler werden häufiger als bei anderen Backup-Programmen entdeckt. Sollte bei einer Quelldiskette regelmäßig ein Lesefehler auftreten, sollten Sie diese Diskette schnell kopieren. Sie erkennen den Fehler daran, daß im Raster der Quelldiskette ein »r« dick umrahmt ist. Bringt ein Backup keinen Erfolg, retten Sie Files auf der Diskette mit einem File-Kopierprogramm, beispielsweise Diskmaster. TurboBackup finden Sie auf der Fish-

Disk 139. Das Kopierprogramm ist auch auf der **Programmservice-Diskette** zu diesem Sonderheft zu finden. Im Directory »TurboBackup« existieren fünf Files:

```
TURBOBACKUPV1.00
TURBOBACKUP.doc
TURBOBACKUP.readme
ReadMe.First
Execute.Me
```

Die Programme auf der Service-Diskette sind mit »Zoo« gepackt. Eine genaue Anleitung zum Entpacken finden Sie auf Seite 135 in diesem Heft. Auf der Fish 139 sind die Files im Directory »TurboBackup« nicht gepackt. Daneben existiert ein Programm »TurboBackup V1.00.arc«. Dieses enthält die mit »arc« gepackten Files. Sinnvoll ist diese Version zum Weitergeben an Freunde mittels Datenfernübertragung. Da die gepackten Files erheblich kürzer sind, spart man beim Übertragen Zeit und Telefonkosten.

Sie können das Backup-Programm jetzt sofort starten (bei

der Programmservice-Diskette nach dem Entpacken, siehe oben). Die Workbench-Freunde klicken die Schublade »TurboBackup« und dann das Programm-Icon für TurboBackup an. Nach kurzer Ladezeit erscheint die Oberfläche des Kopierprogramms. Oben sehen Sie vier Raster, die vier zu verwendende Laufwerke symbolisieren. Darunter befinden sich die Schalter, mit denen Diskettenstationen aktiviert werden. Die Schalter unter nicht angeschlossenen Laufwerken sind in Schattenschrift dargestellt, Sie können diese Schalter natürlich nicht benutzen.

Klicken Sie mit der linken Maustaste den Schalter unter dem Raster für df0: an. Der Status des Schalters ändert sich in »Source« (Quelle). Klicken Sie den Schalter für df1: zweimal an, erscheint der Text »Destination« (Ziel) in diesem Feld. Klicken Sie noch einmal in den Schalter für df1:, erscheint wieder »Off«. Legen Sie so das Quell- und Ziellaufwerk fest.

Unten ist ein großer Kasten, in dem alle Meldungen des Programms erscheinen.

Legen Sie nun eine Quelldiskette in das Laufwerk df0: und eine leere Diskette in df1:. Die Schalter für df0: und df1: müssen nun auf Source (df0:) beziehungsweise Destination (df1:) stehen. Starten Sie den Kopiervorgang durch Anklicken des Schalters »Start«. Haben Sie versehentlich den Schieber an der Diskette in der falschen Stellung, und somit den Schreibschutz aktiviert? Dann mault TurboBackup energisch. Nehmen Sie die Diskette aus

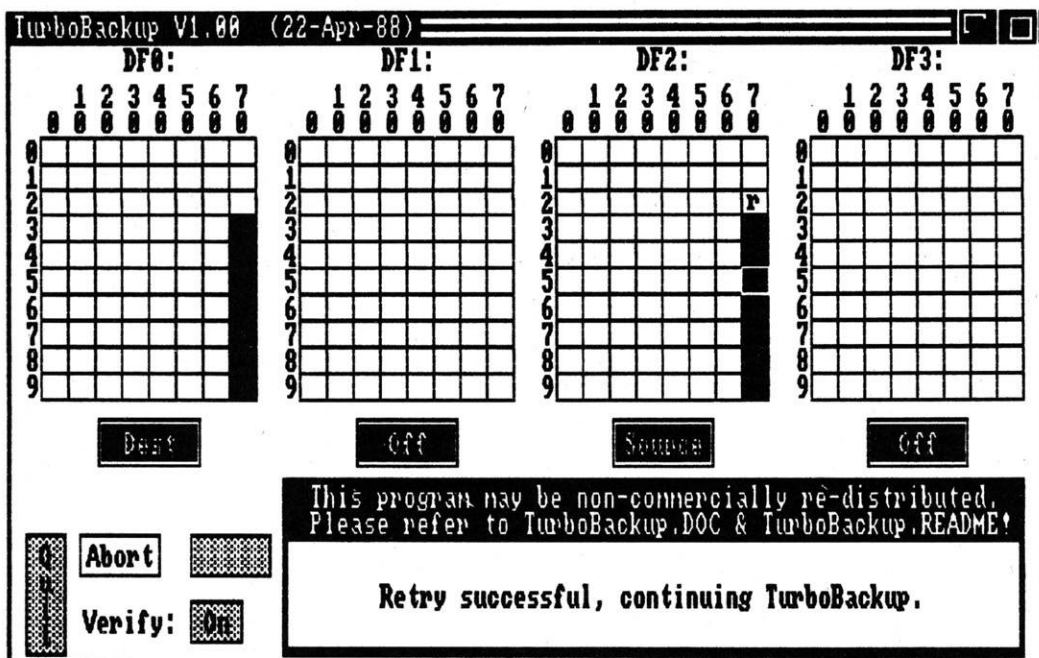


Bild 1. »TurboBackup« verzichtet auf grafischen Schnickschnack. Hohe Geschwindigkeit und Datensicherheit zeichnen das Kopierprogramm aus.

dem Laufwerk df1: und ändern Sie die Schalterstellung. Legen Sie die Diskette erneut ein, und starten Sie den Kopiervorgang.

In dem Raster für df0: erscheint nun für jede gelesene Spur ein »r«. Dies steht für »read«, also lesen der Spur. Rechts daneben erscheinen in den entsprechenden Feldern im Wechsel »w« und »v«. Diese sind die Abkürzungen für »write« (Schreiben) und »verify« (Überprüfen). Sollte einmal ein Lesevorgang mehr als einen Versuch erfordern, zeigt TurboBackup dies mit einem farblich hervorgehobenen Rand um das Kästchen für die betreffende Spur an. Treten mehrere Fehler auf, sollten Sie die Quellscheibe schnell für eigene Zwecke noch einmal kopieren. So schützen Sie sich mit Hilfe von TurboBackup vor »faulen« Daten auf der Diskette.

Kurz vor dem Ende des Kopiervorgangs erscheint ein tiefer Ton. Sind alle Kopien einwandfrei, ertönt schließlich ein hoher Ton (Beep). Klicken Sie nun auf »Quit« - Turboprint streikt mal wieder. Und das mit Recht. Die Warnung »There are still some identical disks in the drives!...« schützt vor heillosen Konfusion des Amiga-DOS. TurboBackup erzeugt zwei identische Disketten. Das kann das DOS nicht verkraften, die Folge wäre eine Reise ins Land der Gurus. Sparen Sie sich diese Strapaze, nehmen Sie die Disketten aus den Laufwerken, das Duplikat ist fertig.

Gehören Sie zu den Amiga-Anwendern, die mit Workbench

und Maus nicht »verheiratet« sind? Für sie hält TurboBackup einige Feinheiten bereit. Der Start über das CLI erfolgt mit dem Aufruf

```
turbobackupV1.00 df0:
df1: [df2: df3:]
```

Geben Sie den Aufruf bitte in einer Zeile ein und drücken abschließend <RETURN>. Das Programm startet, das Laufwerk df0: ist automatisch »Source«, df1: ist »Destination«. Die Laufwerke df2: und df3: können, so vorhanden, zusätzlich als »Destination« definiert werden. Vielleicht interessiert es Sie auch, daß entweder im Window- oder Screen-Modus gestartet werden kann. Geben Sie ein:

```
turbobackupV1.00 df1:
df0: -s
```

Das Backup-Programm startet nun im Screen-Modus. Quelle ist df1:, Ziellaufwerk ist df0:. Sie können den TurboBackup-Screen nun ganz an den unteren Rand ziehen, bis er fast

von der Bildfläche verschwunden ist - natürlich auch während des Kopierens.

Die Workbench-Benutzer können den Screen-Modus im »Info« zum Programm einstellen. Klicken Sie das Symbol (Icon) für TurboBackup einmal an und wählen Sie im »Workbench«-Menü den Punkt »Info«. Klicken Sie den »ADD«-Schalter in der Zeile »TOOL TYPES« an. Geben Sie ein:

```
DISPLAYMODE=screen
```

Drücken Sie anschließend <RETURN> und verlassen das Info-Feld mit »SAVE«. Die geänderten Eintragungen werden gespeichert. Der Neustart von TurboBackup erfolgt im Screen-Modus.

Für Mausmuffel bietet das Backup-Programm als Alternative Tastaturkommandos. Die Tabelle 1 zeigt alle Kürzel.

TurboBackup ist voll Multitasking-fähig. Allerdings sperrt das Programm alle Diskettenlaufwerke im System. Sie können also nicht auf irgendwelche

Taste	Aktion
0 bis 3	Schalter für dfx: verändern
s	Starten
r	Retry, wiederholen bei Fehlern
a	Abort, Abbruch bei Fehlern
q	Quit, Ende des Programms
SHIFT q	wie q
Sollten mehrere Ziellaufwerke eingestellt sein und beim Schreiben ein Fehler auftreten, können Sie mit der entsprechenden Zahlentaste (<0> bis <3>) »Remove« auslösen. Das Laufwerk wird aus dem Kopiervorgang eliminiert, die anderen Kopiervorgänge laufen weiter.	

Tabelle 1. Die Alternative zur Maussteuerung: Mit diesen Tasten steuern Sie TurboBackup

Daten auf Diskette zugreifen. Mit Festplatte und RAM-Floppies gibt es keine Probleme. Wenn Sie während des Kopierens beispielsweise mit einem Editor arbeiten wollen, starten Sie diesen vor dem Aufruf von TurboBackup. Daten können, falls notwendig, auf die RAM-Disk ausgelagert werden. Die Autoren weisen in der beiliegenden Dokumentation darauf hin, daß TurboBackup nur

## Multitasking mit TurboBackup

wenig Prozessor-Zeit in Anspruch nimmt. Das Multitasking ist also sehr sinnvoll, Kaffeepausen beim Kopieren sind nicht mehr angesagt!

Auf einige wichtige Details gehen die Autoren in der Beschreibung kurz ein. Eine Warnung zuerst: Das Backup-Programm prüft nicht, ob die Source-Diskette schreibgeschützt ist. Die Verantwortung für zerstörte Disketten liegt bei Ihnen. Wir meinen jedoch, genau wie Steffen Stempel und Martin Kopp, daß es ratsam ist, den Schreibschutz auf allen Disketten ständig zu aktivieren. Eine nützliche Nebenerscheinung ist dabei, daß Viren keine Nahrung finden - sie können sich auf schreibgeschützten Disketten nicht festsetzen (jedenfalls ist uns kein Virus bekannt, der den Schreibschutz per Software desaktiviert).

Die Autoren versichern außerdem, daß in TurboBackup kein Virus enthalten ist.

Das Programm kopiert alle Disketten im normalen Amiga-DOS-Format. Kopiergeschützte Originale sind nicht zu duplizieren. Das wird die Besitzer von geschützten Originalen zwar stören. Es war aber ganz und gar nicht im Sinne der Autoren, Software-Piraten das Leben zu erleichtern.

Sollten Sie in sehr seltenen Fällen Probleme mit der Spur 79 haben, hilft »Retry«, anschließend »Abort« und »Start« meistens weiter. In der Kombination TurboBackup/ClickToFront verschwindet die Titelzeile nach einem Doppelklick. Der Screen ist dann nicht mehr zu bewegen. Die Autoren verweisen auf Intuition, das dabei nicht mehr mitspielt.

Mit den Hinweisen aus diesem Artikel stehen Ihnen die vielen Vorteile von TurboBackup zur Verfügung. Nutzen Sie das Kopierprogramm für legale Zwecke, Sie werden Ihre Freude an den fehlerlosen Backups haben.



Den Amiga nur mit der Workbench zu steuern, artet schnell in umständliche Klickerei aus. Ganz zu schweigen vom ständigen Wechseln zwischen Tastatur und Maus. »MachII« erledigt dieses Problem. Dieses Programm kann allerdings weit mehr.

von Andreas Lietz

**M**achII ist ein Hilfsprogramm für den Amiga, das mit Hilfe einer geschickten Funktionstastenbelegung und vielen anderen Zusätzen die Arbeit mit dem Computer angenehmer macht. Dieses Programm hat so viele Funktionen, daß man sie in wenigen Sätzen unmöglich aufzählen kann. Am wichtigsten ist sicherlich die Fähigkeit, die Funktionstasten komfortabel belegen zu können – bis zu dreißig Funktionstasten kann man frei definieren. Das können natürlich viele andere Hilfsprogramme auch. Bei MachII gibt es aber einige Besonderheiten, die dieses Utility von anderen Programmen abheben: Man kann eine Funktionstaste nicht nur mit Zeichen, sondern auch mit Mausklicks belegen (und beides natürlich in einer Tastendefinition mischen). Es gibt aber nun keine umständliche Eingabe von Mauskoordinaten, sondern die Sache geht viel einfacher: Um eine Funktionstaste zu belegen, drückt man erst einmal <ALT ESCAPE> und dann die gewünschte Funktionstaste. Nun zeichnet MachII alle betätigten Tasten und Mausklicks auf, bis man wieder <ALT ESCAPE> betätigt. Auf diese Weise lassen sich beispielsweise oft verwendete Menüpunkte problemlos auf Funktionstasten legen. Natürlich läßt sich das eingegebene »Makro« (so heißen die Funktionstastenbelegungen bei MachII) auch »per Hand« verändern – wobei allerdings bei der deutschen Tastatur so mancher Ärger bevorsteht.

Neben den sehr vielseitigen Makro-Funktionen hat MachII noch eine Reihe anderer Dinge, die dem Computerbenutzer das Leben leichter machen, beispielsweise die automatische

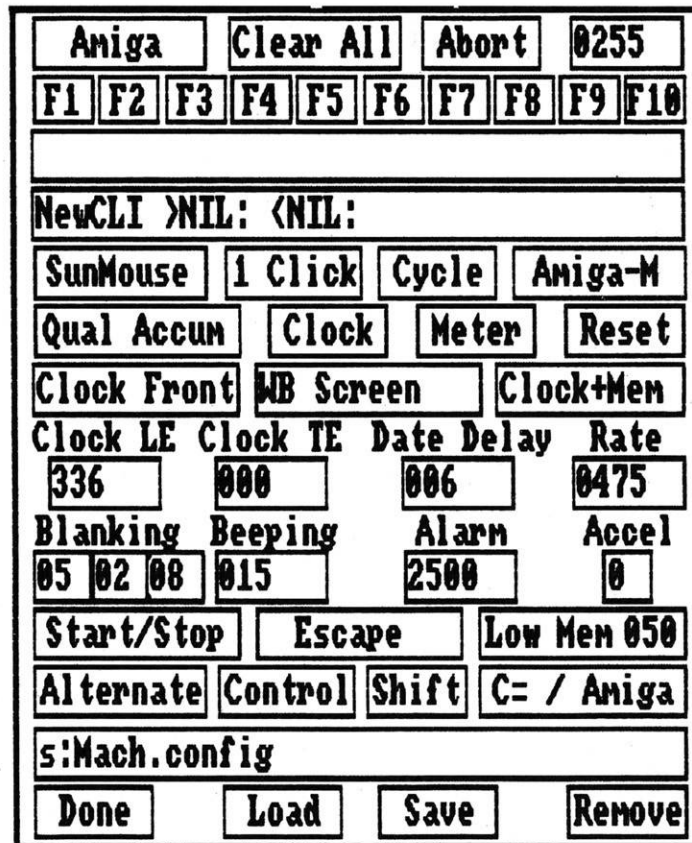


Bild 1. Das Steuerfeld von MachII eröffnet viele Funktionen

# Amiga " mit Überschall

Aktivierung des Fensters, über dem der Mauszeiger steht. Genaueres dazu finden Sie in Tabelle 1, in der alle Funktionen von MachII zusammengefaßt sind.

Wenn Sie dieses Programm verwenden möchten (der Autor dieses Berichtes möchte es nicht mehr missen), so sollten Sie sich die Fish-Disk Nummer 163 oder die Programmservice-Diskette zu diesem Heft beschaffen. Kopieren Sie das Programm MachII auf Ihre Workbench-Diskette. Da sich MachII nicht von der Workbench aus starten läßt, binden Sie es am besten gleich in die Startup-Se-

quence ein, um es beim Laden der Workbench automatisch starten zu lassen. Dazu fügen Sie bitte am Ende Ihrer Startup-Sequence (befindet sich im »s«-Verzeichnis Ihrer Diskette) mit einem Editor folgende Zeile an: `run MachII`

oder besser

`RunBack :MachII`

falls Sie das »RunBack«-Programm besitzen. Letzteres hat den Vorteil, daß sich das CLI-Fenster, in dem das Programm gestartet wurde, nun auch wieder schließen läßt, während das bei »Run« nicht der Fall ist! »RunBack« befindet sich auf

der Fish-Disk 142 (siehe auch Seite 137).

Haben Sie die Startup-Sequence nach Ihren Wünschen modifiziert? Dann lösen Sie einen Reset aus (mit der Tastenkombination <CTRL AMIGA AMIGA>) und warten ein wenig. Hat alles geklappt, so müßt jetzt

Can't Open Configuration File

auf Ihrem Bildschirm zu sehen sein. Nun klicken Sie auf »Cancel« – und sehen kurz danach ein kleines Fenster in der Titelleiste stehen, in dem abwechselnd Uhrzeit, Datum und verfügbarer Speicherplatz angezeigt werden. Bewegen Sie den Mauszeiger über dieses Fenster und drücken Sie die rechte Maustaste. Sie sehen das Steuerfeld von MachII (Bild 1). Hier werden alle Programmfunktionen kontrolliert. Das erste Anzeichen, daß MachII seine Arbeit aufgenommen hat, ist das Verschwinden des Mauszeigers, wenn Sie ihn etwa zehn Sekunden lang nicht bewegen.

Um das Programm auszuprobieren, drücken Sie <AMIGA ESC>. Nun müßte ein CLI-Fenster auf Ihrem Bildschirm auftauchen. Wenn Sie mit der Maus auf dieses Fenster fahren, wird es automatisch aktiv. Sie erkennen dies daran, daß die Titelleiste des Fensters von Geisterschrift in Normalschrift wechselt. Drücken Sie nun <ALT ESC>. Der Computer piepst, und in dem kleinen Fenster mit der Uhr steht »Select Hotkey...«. Drücken Sie <F1>. Nun tippen Sie irgendetwas, beispielsweise

echo "So einfach belegt man eine Funktionstaste"

und drücken nach dem Druck auf <RETURN> wieder <ALT ESC>. Es piepst nochmals, und die Uhr erscheint wieder. Wenn Sie nun auf <F1> drücken (achten Sie darauf, daß das CLI-Fenster noch aktiv ist!), erscheint das, was Sie vorher eingegeben haben. Das funktioniert auch mit Mausklicks hervorragend: Aktivieren Sie die Makro-Eingabe für die F2-Taste (wie oben für <F1> beschrieben) und vergrößern Sie das CLI-Fenster, so daß es den ganzen Bildschirm bedeckt. Wie üblich wird die Eingabe des Makros mit <ALT ESC> beendet. Nun holen Sie mit <AMIGA ESC> ein neues CLI auf den Schirm. Drücken Sie <F2>, um es auf die volle Bildschirmgröße zu bringen!

MachII kann die Funktionstasten auf drei Ebenen mit Ma-

kros belegen. Erstens: »normal«, wenn also nur die Funktionstaste selbst gedrückt wird. Zweitens: zusammen mit einer der Amiga-Tasten. Drittens: zusammen mit <CTRL> und einer der Amiga-Tasten. Aber drücken Sie in der Euphorie nicht gleich beide Amiga-Tasten und <CTRL> - sonst müssen Sie einen Moment warten, bis MachII nach dem dadurch ausgelösten Reset wieder geladen wurde.

Um ein eingebundenes Makro abzuspielden, haben Sie bis jetzt immer die dazugehörige Funktionstaste gedrückt. Aber es geht auch anders: Drücken Sie einmal <SHIFT ALT ESC> - es piepst. Nun tippen Sie »ech« und <RETURN>. Sofort wird der angefangene »echo«-Befehl von MachII vervollständigt. Nach dem Druck dieser Tastenkombination läßt sich also jedes Makro aufrufen, indem man seine ersten Buchstaben und <RETURN> tippt. Nur bei gespeicherten Mausclicks funktioniert das natürlich nicht.

Wenn Sie eine Funktionstaste drücken, die nicht mit einem Makro belegt ist, gibt MachII diese Information weiter an das nächste Programm, das diese Information verwerten kann. Die Ausführung eines Makros läßt sich mit <ESC> abbrechen.

Die ersten drei Zeilen des MachII-Steuerfeldes dienen der Arbeit mit den Makros. Man kann diese auch »per Hand« eingeben und editieren. Um sich Ihre beiden Makros anzusehen, aktivieren Sie wieder das Steuerfeld, indem Sie das kleine MachII-Fenster anklicken und auf die rechte Maustaste drücken (<CTRL AMIGA ESC> hat denselben

## Steuerung per »Gadget-Sammlung«

Effekt). In der ersten Zeile des Steuerfeldes klicken Sie zweimal auf das Wort »Amiga«, bis »None« erscheint. Nun können Sie unter »F1« beziehungsweise »F2« in der nächsten Zeile die von Ihnen programmierten Makros ansehen. Auf diese Weise lassen sich alle Makros anzeigen: Im ersten Feld der ersten Zeile stellt man die Funktionstasten-Ebene (»Amiga«, »Ctrl-Amiga« oder »None«) ein, im zweiten klickt man dann auf die entsprechende Taste.

In der ersten Zeile gibt es noch drei weitere Funktionen, »Clear All«, »Abort« und »0255«.

**Clear all** löscht den Inhalt aller Makros und ist deshalb mit Vorsicht zu genießen.

**Abort** bricht eine angefangene Makro-Definition ab und gibt dem Makro seinen alten Inhalt wieder (dies ist sehr praktisch, wenn man einmal beim Makro-Definieren die falsche Funktionstaste erwischt hat!). 255 ist die maximale Makro-Länge, die sich natürlich auch vergrößern läßt. Eine Veränderung dieser Zahl wird allerdings erst wirksam, nachdem Sie auf »Done« geklickt haben. Außerdem können Sie diese Zahl nicht ändern, solange Sie gerade ein Makro definieren. Diese Zahl verändert die Länge von programminternen Puffern, die bei der Definition eines Makros und bei der Anzeige des Steuerfeldes verwendet werden. Für die Speicherung der Makros selbst wird nur so viel Platz verwen-

## Vielseitige Makro-Befehle

det, wie sie auch wirklich brauchen.

Um neben den Eingaben per Tastatur auch Steuerzeichen und Mausclicks in Makros speichern zu können, verwendet MachII eine Art Mini-Sprache, deren sämtliche Befehle einen Buchstaben haben und mit »\« beginnen. Bei der Aufnahme von Mausclicks in ein Makro speichert MachII die Mausposition und welcher Knopf der Maus gedrückt oder losgelassen wird. Ein Beispiel: »\Mx155 \My075 \Mld« bedeutet, daß der Mauszeiger beim Abspielen des Makros an die X-Position 155 und die Y-Position 75 bewegt wird und daß der linke Mausknopf »gedrückt« wird. Das Drücken und das Loslassen der Mausknöpfe wird in zwei verschiedenen Befehlen gespeichert, um die Makros so flexibel wie möglich zu machen. Nach jedem zweiten Klick wird (falls nötig) ein »Pausezeichen« \P eingesetzt, um auch Pausen bei der Mauseingabe zu berücksichtigen. Sie können diese Pausen aus den Makros löschen, wenn Sie sie nicht brauchen. Sie sollten sich nicht darauf verlassen, daß ein Fenster beim Abspielen automatisch von MachII aktiviert wird. Leider funktioniert das Eingeben von Text in ein Fenster eines Programms, das gerade per »Makro-Mausclick« aktiviert wurde, nicht immer. Da hilft nur eines: ausprobieren!

Die »Makro-Sprache« von MachII hat noch viele andere Befehle, die in Tabelle 2 voll-

## Mach-Makros

Die Makro-Befehle beginnen mit dem Zeichen »\«. Alle Tasten des Amiga lassen sich in einem Makro ansprechen - natürlich auch die Sonderzeichen. Sie werden in Makros so dargestellt:

Befehl	Erklärung
\N	RETURN-Taste
\T	Tab
\E	Escape
\D	Delete
\B	Pfeil nach links
\H	Help-Taste
\u	Cursor nach oben
\d	Cursor nach unten
\l	Cursor nach links
\r	Cursor nach rechts
Die Zeichen auf der Zehnertastatur werden ebenfalls mit einem vorangestellten »\« gekennzeichnet, beispielsweise ist \0 die Null der Zehnertastatur.	
\F1	Funktionstaste 1
bis	
\F0	Funktionstaste 10. Beachten Sie, daß es nicht »\F10«, sondern »\F0« heißen muß!
Bei den Funktionstasten gibt es eine Besonderheit: Wenn Sie eine Funktionstaste aufrufen, die mit einem Makro belegt ist, so wird dieses Makro ausgeführt. Ist eine aufgerufene F-Taste unbelegt, so wird ihr Tastencode normal an das Betriebssystem weitergegeben. Auf diese Weise können Sie die Funktionstastenbelegung anderer Programme auch weiterhin nutzen.	
Um Tasten zusammen mit Control, Shift oder einer anderen sogenannten »Qualifier«-Taste in einem Makro zu »drücken«, müssen Sie folgende Befehle der eigentlichen Taste voranstellen:	
\C	für Control
\S	für eine der beiden Shift-Tasten
\A	für eine Alt-Taste. Schreiben Sie bitte immer »\A...« statt <ALT ... >.
\L	für die linke Amiga-Taste
\R	für die rechte Amiga-Taste
Beispiel: »\Cc« ergibt <CTRL c>; »\S\A\x« ergibt <SHIFT ALT x>.	

### Die eigentlichen Makro-Befehle, die in MachII einen Vorgang auslösen:

\Y	Datum in der Form < Wochentag Monat Tag, Jahr > schreiben.
\y	Datum in der Form < Monat Tag Jahr > schreiben. Leider gibt es beim Datum mit der deutschen Tastatur Probleme.
\b	sofortige Bildschirmabschaltung
\f < Name >	Eine MachII-Konfigurationsdatei mit dem Namen <Name > laden. Steht dieser Befehl nicht am Ende eines Makros, so muß nach dem Namen ein »\« folgen. Weitere Befehle lassen sich aber natürlich nur dann ausführen, wenn in der geladenen Konfiguration dasselbe Makro existiert!
\X < String >	Einen String ausführen (mit »run« <String >). Auch hier muß nach dem String ein »\« kommen, falls dieses Kommando nicht am Ende des Makros steht. Achten Sie unbedingt darauf, daß ein mit diesem Befehl gestartetes Programm keine Eingaben aus dem CLI erwartet! Im Text finden Sie nähere Hinweise zu diesem Befehl.
\Pnn	nn Sekunden Pause. Während einer Pause können keine Eingaben über die Funktionstasten erfolgen.
\Q	MachII verlassen
\G	einen Piepston erzeugen
\a	Das folgende Makro automatisch beim Start von MachII ausführen. Dieser Befehl muß am Anfang des Makros stehen. Auch mehrere Makros können ein »\a« am Anfang haben. Von diesem Befehl gibt es eine Variante: \ahhmm, wobei hhmm eine Zeit im 24-Stunden-Format ist. MachII führt das damit versehene Makro zur gewünschten Zeit aus. Beispiel: »\a1045 \G«: Es piepst um 10.45 Uhr.
\h	Einen Hilfs-Screen mit den ersten Buchstaben der Makro-Definitionen zeigen. Beim zweiten Aufruf verschwindet der Screen. Mit »\hHöhe« stellen Sie die Höhe des Screens in Pixeln ein.



Die Mausbefehle	
\M	ist der Anfang jedes Mausbefehls
\Mxnnn	X-Koordinate des Mauszeigers auf nnn stellen
\Mynnn	Y-Koordinate des Zeigers auf nnn stellen
\Mld	linken Mausknopf »drücken«
\Mlu	linken Mausknopf »loslassen«
\Mrd	rechten Knopf »drücken«
\Mru	rechten Knopf »loslassen«

**Tabelle 2. Alle Makro-Befehle auf einen Blick**

ständig aufgeführt sind. Natürlich brauchen Sie die meisten davon nicht, wenn Sie Makros einfach per <ALT ESC> eingeben und später wieder abspielen. Um das Programm voll auszunutzen, ist es aber oft nötig, ein Makro per Hand im Steuerfeld einzugeben oder dort zu ändern. Hier gibt es aber ein Problem: MachII funktioniert nicht zufriedenstellend mit der deutschen Tastaturbelegung! Offenbar hat der (amerikanische) Programmierer die Tastatur-Tabellen, die in anderen Ländern verwendet werden, nicht berücksichtigt. Das merkt man beispielsweise daran, daß ein beim Makro-Definieren getipptes »Y« als »Z« im Steuerfeld erscheint. Diese Tatsache ist dann wichtig, wenn Sie in einem Makro im Steuerfeld Buchstaben eingeben, die später in einem Amiga-Fenster (etwa im CLI) erscheinen sollen. Um einen Doppelpunkt auf den Bildschirm zu bringen, müssen Sie ein Größer-Zeichen (>) tippen, sonst haben Sie später ein »ö« auf dem Bildschirm stehen. Die meisten Sonderzeichen erreichen Sie

## Tastatur-Probleme

auch, wenn Sie zusammen mit dem Zeichen die ALT-Taste drücken. In Tabelle 3 finden Sie die Sonderzeichen zusammen mit den Tasten, die Sie dafür drücken müssen. Diese Maßnahme gilt übrigens nur, wenn Sie mit der deutschen Tastaturbelegung (Aufruf mit »setmap d«) arbeiten! Wenn Sie eine amerikanische Tastaturbelegung verwenden (»setmap usa0«), gibt es keine Probleme. Ein sehr wichtiger Makro-Befehl, den Sie nur im Steuerfeld eingeben können, ist »X<Name>«. Er dient dazu, Programme automatisch über ein Makro auszuführen. Wenn dieser Befehl nicht als letzter in einem Makro steht, muß er mit dem Backslash (»\«) abgeschlossen sein, beispielsweise so:

```
\Xnewcli 0/0/640/256/
MyCLI\echo Dies ist ein
Test\Nendcli\N
```

Damit sehen Sie auch gleich, wie man zum Start eigener Programme ein CLI auf den Bildschirm bringen kann. Alle Befehle, die in diesem CLI ausgeführt werden sollen, müssen dann mit »\N« (für RETURN) beendet werden. Grundsätzlich kann man natürlich mit \X nicht nur ein CLI-Fenster öffnen, sondern jedes beliebige Programm starten. Achten Sie aber unbedingt darauf, daß Ihr Programm keine Eingaben aus dem - nicht vorhandenen - CLI verlangt, sonst gibt es ein totales Chaos!

Bei dem eben genannten Makro-Befehl macht sich das Tastatur-Problem von MachII besonders bemerkbar. Wenn Sie nämlich ein CLI öffnen und darin Ihre Programme starten, so müssen Sie die Befehle für dieses CLI in der vorher angesprochenen Schreibweise (beispielsweise mit dem Größer-Zeichen als Doppelpunkt) eingeben. Befehle, die **direkt** hinter einem \X stehen, brauchen

ren Name in der Zeile darüber eingegeben wird.

**Load** lädt diese Datei unter demselben Namen.

**Remove** entfernt MachII aus dem Speicher - es läßt sich dann nur aus dem CLI wieder starten.

Bevor Sie Ihren Computer ausschalten, vergessen Sie nicht den Klick auf »Save«, falls Sie im Steuerfeld etwas geändert oder neue Makros definiert haben!

In der vierten Zeile des Steuerfeldes steht der Befehl, der beim Druck auf <ALT ESC> von MachII ausgeführt wird (im Moment NewCli). Er läßt sich

## Wer's mag: die Sonnenmaus

beliebig verändern. So können Sie - vorausgesetzt Sie haben die Workbench 1.3 - in dieses Feld schreiben:

```
NewShell NEWCON:0/0/640/
256/MyShell
```

Das von dieser Funktion geöffnete CLI hat dieselbe Stack-Größe wie MachII. Viele Programme brauchen aber einen größeren Stack; Sie sollten ihn also mit dem »stack«-Kommando entsprechend einstellen. Wenn Programme unerklärlicherweise abstürzen, kann es an der Stack-Größe liegen.

Wenn ein String-Gadget aktiv ist, arbeitet die SunMouse nicht. Vorsicht: Wenn Wordperfect einen Requester anzeigt, versucht es diesen aktiv zu halten. Das kann Probleme mit der SunMouse geben. Schalten Sie diese Funktion also lieber ab, wenn Sie gleichzeitig mit Wordperfect arbeiten.

**Click to Front:** Wenn Sie im Gadget neben »SunMouse« »1 Click« oder »2 Click« einstellen, können Sie jedes Fenster durch einen einfachen oder doppelten Klick in den sichtbaren Bereich dieses Fensters hinter anderen hervorholen. Sie brauchen dabei nicht das Close-Gadget zu treffen. Diese Funktion wird ausgeschaltet, wenn man »0 Click« einstellt.

Wenn **Cycle** aktiv ist, bedeutet das, daß Sie ein Fenster nach hinten schieben können, indem Sie den linken Mausknopf festhalten und dabei den rechten drücken. Hat der gerade verwendete Screen kein oder nur ein Fenster, so wird er selbst nach hinten gestellt.

Mit <AMIGA M> kann man normalerweise (ohne MachII) den Workbench-Screen nach hinten schieben. Ist diese Funktion aktiv, so dient <Linke AMIGA M> dazu, zwischen den verschiedenen Amiga-Screens hin- und herzuschalten, während <Linke AMIGA N> wie bisher den Workbench-Screen nach vorne bringt. Wenn die »SunMouse« eingeschaltet ist, wird das Fenster unter dem Mauszeiger auf dem neuen Screen automatisch aktiv.

**Qual Accum:** Viele Leute finden es sehr schwierig oder gar unmöglich, Tastenkombinationen wie <CTRL SHIFT F1> einzugeben. MachII hat die Fä-

## Eine Uhr für alle Fälle

higkeit, Steuertasten, die gedrückt werden, zu merken und sie zur ersten danach gedrückten »Nicht-Steuertaste« zu »addieren«. Auf diese Weise könnten Sie zum Beispiel zuerst Control, dann Alternate und dann F1 tippen, um die Tastenkombination <CTRL SHIFT F1> einzugeben. Diese Funktion wird allerdings von fingerfertigen Anwendern eher als Belästigung empfunden, Sie sollten sich daher die Aktivierung gut überlegen. Haben Sie einmal eine falsche Steuertaste gedrückt, so können Sie mit <CAPS LOCK> oder dem rechten Mausknopf noch einmal von vorne anfangen.

Tastaturprobleme	
Um das zu erhalten...	müssen Sie das tippen
/	&
—	/
)	(
=	)
—	?
:	<
:	>
ö	:
Ö	:
·	ü
~\$	#
&	,
ß	—
?	—

**Tabelle 3. Eine Umwandlungshilfe für die deutsche Tastatur**

diese Tasten-Übersetzung aber nicht! Man kann nur hoffen, daß der Autor von MachII sich in einer seiner nächsten Programmversionen der nicht-amerikanischen Nutzer erbarnt und dieses Problem aus der Welt schafft.

Das MachII-Steuerfeld hat noch viele weitere Funktionen. In der letzten Zeile stehen die »grundsätzlichen Dinge«:

**Done** schließt das Steuerfeld.

**Save** speichert die MachII-Konfiguration in einer Datei, de-

Mit diesem - immerhin stattlichen - Befehlsumfang ist MachII aber noch lange nicht am Ende. Im mittleren Bereich des Steuerfeldes finden Sie noch so manchen Leckerbissen:

**SunMouse** schaltet die bereits angesprochene Fähigkeit von MachII ein und aus, immer das Fenster zu aktivieren, über dem der Mauszeiger steht. Um Icons verschieben zu können, ist diese Funktion ausgeschaltet, solange Sie den linken Mausknopf gedrückt halten.

**Clock:** Die Uhr am oberen Bildschirmrand kennen Sie ja schon. Sie wird jede Sekunde weitergezählt und bringt sich, wenn sie von einem Fenster verdeckt wird, alle vier Sekunden selbständig nach vorne. Bei manchen Programmen verdeckt sie wichtige Informationen in der Titelseile, so daß sie sich auf Wunsch über das »Clock«-Gadget im Steuerfeld ganz ausschalten läßt. Wollen Sie die Uhr nur zeitweise »nach hinten klicken«, so schalten Sie »Clock Front« auf »Clock Depth« um – die Uhr bekommt dann zwei (fast) unsichtbare Depth-Gadgets, die wie bei einem normalen Fenster funktionieren. Der Abstand, in dem die aktuelle Zeit angezeigt wird, läßt sich im »Date delay«-Gadget einstellen – steht dieser Wert auf Null, so wird ständig die Zeit angezeigt, nicht aber das Datum.

**Clock + Mem:** Hier wird eingestellt, ob in dem kleinen MachII-Fenster die Uhrzeit und der verfügbare Speicher oder nur eines von den beiden angezeigt werden soll. Wenn Sie nur noch weniger als 50 KByte Speicher haben, erscheint die Speicherkapazität in der Uhr in Rot. Dieser Wert kann unter »Low Mem« geändert werden.

Sie können die Uhr mit der Maus beliebig positionieren oder die Werte bei »Clock LE« und »Clock TE« (X- und Y-Koordinate der linken oberen Ecke des Fensters) per Hand eingeben. Natürlich erscheinen hier beim Aktivieren des MachII-Steuerfeldes die aktuellen Werte.

## Jederzeit alarmbereit

Über dem »Clock TE«-Gadget finden Sie ein Gadget, in dem sich einstellen läßt, ob sich die Uhr auch automatisch auf andere Screens als den Workbench-Screen setzt. Hier ist zunächst einmal »WB Screen« eingestellt – die Uhr bleibt also immer auf dem Workbench-Screen. Wenn Sie statt dessen »Screen Depth« einstellen, bekommt die Uhr links ein unsichtbares Gadget, mit dem Sie sie auf andere Screens klicken können. Wird »Clock to Top« aktiviert, so bringt sich die Uhr immer an die »Oberfläche« der Anzeige, egal welcher Screen sich dort gerade befindet. »Screen Depth« und »Clock to Top« verändern den Close-Screen-Vektor, damit MachII sein Fenster wieder von einem Screen entfernen kann, falls

dieser geschlossen wird. Wenn ein anderes Programm ebenfalls diesen Vektor verändert hat, gibt es eine Fehlermeldung. Sie können dann entweder das Programm beenden, das diesen Vektor verstellt hat, oder auf »Cancel« klicken.

Unter **Alarm** läßt sich eine Alarmzeit im 24-Stunden-Format eingeben, bei der MachII einen Warnton auslöst. Da man diesen Ton aber leicht überhört, ist es wesentlich wirkungsvoller, Alarm per Makro mit der »\a«-Funktion auszulösen, beispielsweise so:

```
\a1630\Xsay "Alarm!
Alarm! Alarm!"
```

Damit gibt der Amiga um 16.30 Uhr per Sprachausgabe Alarm. Der Befehl »say« muß sich dazu im C-Ordner befinden, das »Narrator.device« im Ordner »devs« der Bootdiskette.

Wer schon immer zu wenig Platz auf dem Schreibtisch hatte, wird den »Maus-Speeder« von MachII mögen. Unter **Accel** können Sie den Beschleunigungs-

diesen Zweck zu schätzen wissen. Dazu gibt man unter **Rate** die Kosten der Verbindung pro Stunde ein und aktiviert den Zähler mit einem Klick auf **Meter**. Zurücksetzen läßt er sich mit **Reset**. Sie müssen allerdings zumindest »Clock+Mem« eingestellt haben, damit der Zähler angezeigt wird. Gezählt wird – wie könnte es anders sein? – in US-Dollar.

## MachII hält Sie wach

MachII hat eine automatische Bildschirm- und Mauszeigerabschaltung eingebaut, die mit den Zahlen im Feld **Blanking** kontrolliert wird. Die erste Zahl ist die Anzahl der Minuten, nach denen der Bildschirm abgeschaltet wird, wenn nichts getippt und der Mauszeiger nicht bewegt wurde. Ist dieser Wert Null, so wird der Bildschirm nicht abgeschaltet. Wie stark der Bildschirm bei der Abschaltung »gedimmt« werden

soll, läßt sich mit der zweiten Zahl einstellen: 00 macht den Bildschirm völlig schwarz, während 15 der normalen Bildschirmhelligkeit entspricht. Sinnvoll sind wohl Zahlen zwischen 0 und 6. 0 schaltet grundsätzlich den ganzen Bildschirm schwarz, während bei anderen Werten nur der vorderste Screen gedimmt wird (!). Mit »\b« kann man in einem Makro eine sofortige Bildschirmabschaltung auslösen.

Der dritte Wert im »Blanking«-Feld enthält schließlich die Anzahl der Sekunden, nach denen der Mauszeiger verschwindet, falls er nicht bewegt wird. Mit einer Null an dieser Stelle wird diese Funktion ausgeschaltet. Vorsicht: Sie sollten diese Funktion vor dem Arbeiten mit »Preferences« außer Betrieb setzen, da der Mauszeiger sonst möglicherweise auf Nimmerwiedersehen verschwindet.

Möglicherweise haben Sie nun schon ein wenig mit MachII gearbeitet und sind zu der Auffassung gekommen, daß das Programm zwar sehr gut ist, daß aber Kombinationen wie <SHIFT ALT ESC> doch sehr umständlich einzutippen sind. Stimmt genau. Deshalb gibt es die Gadgets **Start/Stop**, **Escape**, **Alternate**, **Control** und **Shift**. Damit lassen sich die Tastenkombinationen, die man zur Bedienung von MachII braucht, beliebig verändern. Die vier Tastenkombinationen von MachII sind in Tabelle 4 aufgeführt. Im »Start/Stop«-Gadget müssen Sie den Namen aus der Tabelle einstellen. Wenn Sie auf das »Escape«-Feld klicken, so rotieren dort alle möglichen Tasten, die von MachII abgefragt werden können. Klickt man auf die linke Seite des Feldes, geht das Programm weiter durch die Liste der Tasten; ein Klick auf die rechte Seite geht einen Schritt zurück. »Num Pad+Taste« sind die Tasten der Zehnertastatur;

## Kurzübersicht MachII

**Funktions-Tastenbelegung:** Die Funktionstasten lassen sich dreifach mit Makros belegen. Diese Makros sind durch eine kleine »Makro-Sprache« sehr vielseitig und leicht zu definieren. Makros lassen sich zu einer bestimmten Zeit automatisch starten.

**SunMouse** macht immer das Fenster aktiv, über dem der Mauszeiger steht.

**Maus-Beschleuniger** bewegt den Mauszeiger schneller.

**CLI** startet ein CLI oder ein anderes Programm per Tastendruck.

**Bildschirmabschaltung** verhindert ein Einbrennen des Bildschirms.

Auch der Mauszeiger schaltet sich auf Wunsch nach einiger Zeit ab.

**ClickToFront** bringt ein Fenster durch Klick auf eine beliebige Stelle nach vorne oder in den Hintergrund.

»< AMIGA M >« schiebt jetzt auch den Workbench-Screen nach hinten.

**Uhrzeit und/oder Speicher** lassen sich in einem Fenster in der Titelseile anzeigen.

**Alarm** kann nach beliebiger Zeit eingestellt werden (normalerweise alle 15 Minuten).

**Zähler** kontrolliert Ihre Online-Kosten bei der Datenfernübertragung.

Alle Funktionen werden über ein komfortables Steuerfeld eingestellt und sind abschaltbar. Das Programm verbraucht etwa 40 KByte Speicher.

Tabelle 1. Die Funktionen von MachII in Kurzfassung

gungsfaktor für die Maus einstellen, dabei ist 0 die normale Geschwindigkeit. Bei 9 zischt der Zeiger dafür so los, daß ein vernünftiges Arbeiten kaum noch möglich ist!

»Um Sie wachzuhalten«, wie der Programmierer von MachII schreibt, kann man unter **Beeping** in Minuten eingeben, wie oft das Programm einen Pieps-ton erzeugen soll. Wer's mag... Mit einer 0 schaltet man das Gepiepse ab.

Wer häufig Datenfernübertragung (DFU) betreibt, wird vielleicht die automatische Kostenrechnung von MachII für

## Mach-Steuerung

Normaleinstellung	Name	Erklärung
< ALT ESC >	Start/Stop	Beginnt und beendet eine Makro-Definition
< AMIGA ESC >	Execute	Führt den Befehl in der 4. Zeile des Steuerfeldes aus (normalerweise NewCli)
< CTRL AMIGA ESC >	Requester	Bringt das MachII-Steuerfeld auf den Bildschirm
< SHIFT ALT ESC >	Matcher	Sucht ein Makro nach seinen Anfangsbuchstaben und führt es aus

Tabelle 4. Mit diesen Shortcuts wird MachII bedient



»Enter« ist die RETURN-Taste dieser Tastatur! Die verschiedenen »Arrows« sind die Cursor-tasten. In der nächsten Zeile des Steuerfeldes läßt sich nun einstellen, welche Steuertasten zusammen mit der oben gewählten Taste gedrückt werden müssen, um die angegebene Funktion auszulösen. Bei »Alternate«, »Shift« und »C= / Amiga« reagiert das Programm jeweils auf die linke und die rechte dieser Tasten.

Sie sollten bei der Einstellung der Tastenkombinationen daran denken, daß diese Kombinationen dann von MachII »geschluckt« und von anderen Programmen nicht mehr verwendet werden können. Dies

## Ein Mauszeiger verschwindet

kann hinderlich sein, wenn Sie mit mehreren Programmen ar-

beiten, die dieselben Kombinationen abfragen.

Wenn Sie für MachII ein paar gute Makros entwickelt haben, schreiben Sie uns! Und wenn Sie dem Programmierer Hinweise geben wollen, wie er sein Programm verbessern kann, schreiben Sie an:

Polyglot Software  
10431 Ardyce Court  
Boise, Idaho 83704  
U.S.A

Der Autor weist ausdrücklich

darauf hin, daß er für konstruktive Kritik und Verbesserungsvorschläge dankbar ist. Dies richtet sich nicht nur an Programmierer, sondern an jeden, der eine gute Idee beitragen kann. In jedem Fall sind die 40 KByte Speicherplatz, die MachII belegt, ganz sicher nicht verschwendet. Sie erhalten ein Programm, das besonders durch seine Makros eine unvergleichliche Vielfalt an Funktionen aufweist. so

Von Peter Sack

**D**as Problem ist immer wieder das gleiche. Nachdem man das seitenlange Programm endlich eingetippt hat (und es für fehlerfrei hält), läßt man noch den Compiler laufen, der ebenfalls nicht meckert. Bevor man jetzt minutenlang das Laufwerk rotieren läßt, noch schnell ein Testlauf... und schon meldet sich dieser hämisch blinkende rote Kasten. Gleiches gilt für Texte oder Daten, die in mühseliger Arbeit eingetippt wurden. Kurz vor Schluß passiert es, der Amiga meldet sich unvermittelt ab und sendet noch einen letzten Gruß seines Guru.

Man könnte entweder die Wartezeit in Kauf nehmen und sicherheitshalber ständig auf Diskette zwischenspeichern (ich höre alle ständig unter Zeitnot leidenden Computerfreaks enttäuscht aufstöhnen) oder risikofreudig den zweiten (beschriebenen) Weg wählen. Oder gibt es doch eine dritte Möglichkeit?

Anstatt zur Datensicherung ein langsames Diskettenlaufwerk zu verwenden, könnte man auch auf eine schnelle RAM-Disk zurückgreifen. Das Problem der auf der Workbench-Diskette enthaltenen RAM-Disk ist jedoch, daß sie nach einem Systemabsturz überschrieben wird. Das läßt sich allerdings umgehen.

Die ASDG-RAM-Disk des amerikanischen Programmierers Perry Kivolowitz gehört zur Gattung der resettefesten Programme. Das bedeutet, daß ihr ein System-Reset (wenn er nicht gerade durch Stromausfall oder Ausschalten ausgelöst wurde) nichts anhaben kann. Sie enthält auch nach dem gefürchteten Systemabsturz die vorher gespeicherten Daten und macht sie nach erneutem Booten wieder zugänglich. Dabei belegt sie immer nur soviel

# Rettung für Ihre Daten

**Wie oft passiert es, daß der Amiga abstürzt und Sie das Ergebnis stundenlanger Arbeit nicht auf Diskette gespeichert haben? Viele Anwender beginnen dann von vorne mit der Arbeit. Wir stellen Ihnen eine bessere Lösung vor.**

Speicherplatz wie tatsächlich für die Daten benötigt wird. Damit ist sie flexibler als die auf der Workbench 1.3 mitgelieferte RAM-Disk namens »RAD:«. Dennoch kann Sie von jedem Programm, im CLI und auf der Workbench wie ein normales Diskettenlaufwerk angesprochen werden. Aber diese Super-RAM-Disk schützt nicht nur Ihre Daten vor Abstürzen, sondern auch Ihren Geldbeutel vor allzu großer Erleichterung. Als PD-Programm ist sie grundsätzlich kostenfrei, der Autor bittet lediglich im Rahmen des bei Shareware üblichen Beitrags um eine Unterstützung in Höhe von zehn Dollar. Sie finden ASDG-RAM auf Fred Fish 58, Auge 2 und Panorama 13d.

Zuerst sollten Sie sich eine der drei oben genannten Disketten besorgen, die Sie bei jedem Public-Domain-Versand bestellen können. Inzwischen sind PD-Disketten oft sogar in Computerläden erhältlich. Sollten Sie Probleme im Umgang mit dem CLI haben, so finden Sie in diesem Heft auf Seite 76 einige wertvolle Hinweise.

Nachdem diese resettefeste RAM-Disk keines der Standard-Laufwerke (wie »df0:« oder »df1:«) ist, wird sie beim Reset auch nicht vom System erkannt. Sie braucht daher eine besondere Behandlung, bevor sie unter dem Namen »vd0:« angesprochen werden kann. Im folgenden Abschnitt erfahren Sie, wie Ihr neues Laufwerk

beim System angemeldet wird und wie Sie es für Ihre Ansprüche optimal einrichten.

Fertigen Sie zuerst eine Kopie Ihrer Original-Workbench an. Da die ASDG-RAM-Disk ein wenig Platz benötigt, klicken Sie auf dieser Kopie die »Demo«-Schublade an und wählen den Punkt »Discard« aus dem Workbench-Menü. Außerdem sollten Sie den »Calculator« in der »Utilities«-Schublade durch einfaches Anklicken und anschließendes Wählen von »Discard« löschen. Nun ist auf der Workbench-Diskette Platz für unsere Zwecke. Keine Angst - die gelöschten Programme sind keineswegs verschwunden, solange Sie mit der Kopie arbeiten. Sie sind selbstverständlich auf der Original-Diskette weiterhin vorhanden.

Booten Sie jetzt mit der Kopie Ihrer Workbench-Diskette (indem Sie gleichzeitig die drei Tasten CTRL, rechte AMIGA und linke AMIGA drücken) und öffnen Sie das CLI. Haben Sie zwei Diskettenlaufwerke, so legen Sie die Kopie Ihrer Workbench-Diskette in das zweite Laufwerk »df1:«. Falls Sie kein zweites Diskettenlaufwerk haben, so geben Sie im CLI bitte die folgende Befehlssequenz ein:

```
makedir RAM:c
copy c:cd RAM:c
copy c:copy RAM:c
copy c:dir RAM:c
copy c:makedir RAM:c
path RAM:c
```

Darauf legen Sie die Diskette mit der ASDG-RAM-Disk in das interne Laufwerk und geben Sie ein:

```
cd df0:
dir
```

Nun erscheinen einige Directory-Namen. Einer enthält das Wort »ASDG«. Beispielsweise finden Sie auf der Fish-Disk den Namen »asdg.rdd (dir)«. Wechseln Sie mit »Cd« in dieses Unterverzeichnis. Wenn das Directory »asdg.rdd« heißt, so geben Sie ein:

```
cd asdg.rdd
```

Anschließend sehen Sie sich das Directory dieses Unterverzeichnisses mit »dir« an. Folgende Dateien müssen dort vorhanden sein:

**asdg.vdisk.device**  
**Mountlist**

Sind diese nicht enthalten, sollten Sie die Diskette umtauschen. Gut wäre es, wenn zusätzlich folgende Dateien enthalten sind:

**CleanRamDisk**  
**DeleteRamDisk**  
**Disclaimer**

Bringen Sie zuerst das CLI-Fenster auf die volle Größe und lesen Sie dann die Text-Datei »Disclaimer« mit:

```
Type Disclaimer
```

Hier finden Sie neben den Copyright-Vermerken einige Hinweise, wie Sie bei der Firma ASDG eingetragener Benutzer werden können. Die anderen beiden Directory-Einträge sind

Installiertes RAM	maximaler HighCyl-Wert	maximale Größe der RAM-Disk
512 KByte	15	128 KByte
1 MByte	31	256 KByte
1,5 MByte	95	768 KByte
2 MByte	143	1144 KByte
2,5 MByte	191	1,5 MByte

**Tabelle 1. Der optimale »HighCyl«-Wert bei verschiedenen RAM-Konfigurationen**

Utilities, die für den Betrieb der RAM-Disk hilfreich (aber nicht unbedingt nötig) sind.

Bevor Sie die RAM-Disk benutzen können, muß noch einiges an Vorarbeit geleistet werden. Am besten kopieren Sie das komplette ASDG-Unterverzeichnis in die normale RAM-Disk und von dort auf Ihre neue Workbench. Geben Sie dazu ein:

```
makedir RAM:asdg
copy df0:asdg.rrd
RAM:asdg all
```

Entriegeln Sie den Schreibschutz der Kopie Ihrer Workbench-Diskette und legen Sie diese in das interne Diskettenlaufwerk »df0:« ein. Um alle Dateien aus der RAM-Disk auf die Workbench zu kopieren, geben Sie ein:

```
Join RAM:asdg/Mountlist
devs:Mountlist AS devs:
MountL
Delete devs:MountList
Rename devs:MountL devs:
MountList
Copy RAM:asdg/asdg.
vdisk.device devs:
Copy RAM:asdg/CleanRam
Disk# ? df0:System
Copy RAM:asdg/DeleteRam
Disk# ? df0:System
```

Sicher wundern Sie sich über den merkwürdigen »join«-Befehl? Dieser verbindet zwei Dateien und legt sie unter dem Namen einer dritten ab. Wir wenden dieses Verfahren an, um eine eventuell vorhandene Mountlist nicht zu beschädigen. Im nächsten Schritt wird die alte Mountlist gelöscht, und die neu entstandene erhält den Namen »Mountlist«, damit sie später vom »mount«-Befehl erkannt wird.

Die resetfeste RAM-Disk braucht einige Grundeinstellungen, die in der Mountlist festgelegt sind. Diese sollten Sie mit einem Editor Ihrer speziellen Speicherkonfiguration anpassen:

```
Ed RAM:asdg/Mountlist
```

Sobald das Editor-Fenster erscheint, bewegen Sie den Cursor mit den Cursorstasten bis an das Ende des Textes. Dort finden Sie einige Zuweisungen,

die Ihnen vielleicht aus Basic bekannt sind. Diese Zuweisungen haben mit Basic nichts zu tun, sondern sind Anweisungen an AmigaDOS.

Überprüfen Sie zuerst, ob Ihr Amiga mehr als 512 KByte Speicher hat. Wenn Sie stolzer Besitzer eines Amiga 2000 sind, so ist dieser in jedem Fall mit mindestens 1 MByte ausgerüstet. Der Amiga 500 hingegen wird mit 512 KByte ausgeliefert. Nur mit einer Speichererweiterung erreicht er den gleichen Speicherausbau. Sie kön-

## CHIP oder doch nur FAST?

nen das auch in der Kopfzeile der Workbench erkennen, wenn Sie auf das Fenster der Workbench klicken. Dort sehen Sie eine Zahl mit mindestens fünf Stellen. Ist diese größer als 500000, so hat Ihr Amiga mehr als 512 KByte Speicher.

Man nennt die 512 KByte Speicher der Grundversion (die natürlich auch im Amiga 2000 enthalten sind) »CHIP«-Memory, weil die sogenannten Custom-Chips nur auf den Speicher in diesem Bereich zugreifen können. Dagegen hat der 68000er-Prozessor beim Zugriff auf den restlichen Speicher »freie Hand«, er wird nicht durch andere Chips gestört. Daher nennt man diesen Bereich »FAST«-Memory. Wenn Sie also nur 512 KByte besitzen, bestehen diese mit Sicherheit nur aus Chip-RAM. Sollten Sie dagegen auch Fast-RAM haben, so gehen Sie im Editor mit dem Cursor in die Zeile, die die Eintragung »BuffMemType = 3« enthält, und ändern Sie die 3 in eine 5. Damit legen Sie die RAM-Disk – soweit möglich – ins Fast-RAM.

Entsprechend der RAM-Größe Ihres Amiga entnehmen Sie nun bitte aus Tabelle 1 den richtigen Wert für die HighCyl-Zuweisung. Bitte beachten Sie, daß nur ungerade Werte gewählt werden können. Haben Sie sich für einen Wert entschieden, tragen Sie diesen hinter »HighCyl« im Editor ein.

Arbeiten Sie mit Kickstart und Workbench 1.3, so gehen

Sie anschließend mit dem Cursor auf das »#« in der letzten Zeile und drücken solange die Leertaste, bis Sie in derselben Spalte sind wie die übrigen Eintragungen. Nun tippen Sie:

```
Mount = 1
```

und drücken <RETURN>. Verlassen Sie den Editor mit einem Druck auf die ESC-Taste, »x« und abschließendem <RETURN>.

Ihre Mountlist ist jetzt so eingerichtet, daß die RAM-Disk mit Ihrer Konfiguration optimal zusammenarbeitet. Zuletzt müssen Sie die RAM-Disk noch so anmelden, daß Sie bei jedem Laden der Workbench automatisch eingerichtet wird und auf der Workbench erscheint. Dazu verändern Sie die »Startup-Sequence«:

```
Ed SYS:s/Startup-
Sequence
```

Dort bleiben Sie mit dem Cursor gleich in der ersten Zeile.

Tippen Sie:

```
Mount vd0:
Dir >NIL: vd0:
```

und schließen Sie jede der beiden Zeilen mit einem <RETURN> ab. Nun können Sie wieder die ESC-Taste, »x« und <RETURN> drücken. Nach dem nächsten Booten ist die RAM-Disk automatisch aktiv. Sie erkennen dies an dem neuen Diskettensymbol (»ASDG-RAM«). Wenn Sie ohne zu booten mit Ihrem neuen Laufwerk arbeiten wollen, geben Sie im CLI die beiden oben genannten Zeilen ein.

Die RAM-Disk ist jetzt resetfest eingerichtet. Vielleicht kommt Ihnen dieses Verfahren etwas umständlich vor. Aber bedenken Sie, daß eine resetfeste RAM-Disk nicht zu den »normalen« Laufwerken gehört, sondern – wie beispielsweise auch eine Festplatte – gezielt eingebunden werden muß. Bei jedem neuen Bootvorgang wird die RAM-Disk automatisch angemeldet (solange Sie die beiden Zeilen in der Startup-Sequence nicht entfernen), und Sie brauchen sich um nichts mehr zu kümmern. Einige Dinge sollten Sie aber beachten:

– Nur was auf der ASDG-RAM-Disk oder auf Diskette und Festplatte abgespeichert wurde, bleibt nach einem Reset erhalten. Die Daten auf Ihrer herkömmlichen RAM-Disk bleiben nach wie vor verschwunden.

– Nach einem Reset muß der Amiga als erstes wieder mit einer Diskette gebootet werden, auf der die ASDG-RAM-Disk installiert wird, damit die Dateien regeneriert werden können.

Sollten Sie mit einer anderen Workbench booten, sind Ihre Daten endgültig verloren.

– Die ASDG-RAM-Disk ist kein Garant für Datensicherheit. Sicherer sind Ihre Dateien auf Diskette oder auf Festplatte untergebracht. Besonders wenn Sie in Ihren Programmen direkt auf die Hardware zugreifen, kann es zu Abstürzen kommen, die sich weder um Gurus noch um Reset-Vektoren kümmern. Sie kennen sicher Abstürze, die nur mit Aus- und wieder Einschalten des Computers zu beheben sind.

– Die ASDG-RAM-Disk kann auf der Workbench angeklickt werden und in File-Requestern und im CLI unter dem Namen »vd0:« angesprochen werden. Sie benötigen – wenn auf ihr keine Datei gespeichert wurde – etwa 0,2 KByte des Speichers.

– Der CLI-Befehl »Addbuffers« sollte keinesfalls auf die ASDG-RAM-Disk angewendet werden, da er sie nur langsamer macht. Die Daten werden in diesem Fall unnötigerweise zwischen zwei Speicherbereichen hin- und hergeschaufelt.

Um Ihr neues Laufwerk zu testen, kopieren Sie einfach beliebige Files nach »vd0:«. Dies geschieht wie gewohnt mit »Copy« (vom CLI aus) oder indem Sie das gewünschte Icon mit der Maus in das Fenster von »ASDG-RAM« transportieren und dort loslassen. Drücken Sie darauf gleichzeitig beide AMIGA-Tasten und <CTRL>. Normalerweise sind nach einem so ausgelösten Reset alle Daten auf der herkömmlichen RAM-Disk verschwunden. Wenn Sie nach erneutem Booten (siehe oben) aber jetzt im CLI eingeben:

```
dir vd0:
```

dann sind alle vorher kopierten Dateien vorhanden. Der Reset konnte »vd0:« nichts anhaben.

## Wartung muß sein

»CleanRamDisk« und »DeleteRamDisk« sind zwei Utilities, die zusammen mit der ASDG-RAM-Disk geliefert werden. Sie dienen zur Wartung und Optimierung der Disk. Wenn Sie sämtliche Anweisungen in diesem Artikel eingegeben haben, so müßten diese Programme sich jetzt in der »System«-Schublade der Kopie Ihrer Workbench-Diskette befinden. Um dies zu überprüfen, geben Sie im CLI folgende Zeile ein:

```
dir SYS:system
```



Sie sehen die beiden neuen Einträge, aber wozu sollen sie gut sein? Wenn Sie die ASDG-RAM-Disk häufig benutzen (was zu erwarten ist) und Ihnen auffällt, daß Ihr freier RAM-Speicher knapp wird, so rufen Sie das Programm »CleanRamDisk« auf. Es optimiert die RAM-Disk und entfernt überflüssige Datenblöcke, so daß RAM-Speicher frei wird. Ihre Daten und Dateien bleiben jedoch vollständig erhalten, für sie besteht keine Gefahr. »Dele-

teRamDisk« konfiguriert die ASDG-RAM-Disk so, daß sie nach dem nächsten Reset und erneutem Installieren die auf ihr gespeicherten Dateien löscht und somit wieder ganz frei ist. In diesem Fall sollten Sie aber daran denken, alle noch benötigten Dateien auf ein »sicheres« Laufwerk auszulagern.

Die Geschwindigkeit der ASDG-RAM-Disk ist - verglichen mit Diskettenlaufwerken - enorm. Sie können dies problemlos testen, indem Sie ein

Programm, das normalerweise einige Ladezeit in Anspruch nimmt, auf »vd0:« kopieren und von dort starten. Nehmen Sie als Beispiel den »MicroEmacs« von der Extras-Diskette. Kopieren Sie ihn mit

```
copy ExtrasD:tools/
MicroEmacs vd0:
```

und starten Sie ihn mit  
vd0:MicroEmacs

Sie werden eine Veränderung feststellen: Statt endlos auf der Diskette zu suchen,

kommt das Programm jetzt sofort zur Sache. Dies ist sicherlich ein entscheidendes Argument für das »asdg.vdisk.device«. Ab jetzt warten Sie nicht mehr auf langsame Diskettenzugriffe, sondern speichern auf ein schnelles und trotzdem zuverlässiges Medium. Allerdings sollten Sie stets daran denken, daß vor dem Ausschalten alle Daten auf eine »normale« Diskette kopiert werden. Sonst stehen Sie wieder vor demselben Problem... so

## OLL Hintergrundiges im

Von Klaus Sonnenleiter

**D**er Amiga hat die angenehme Eigenschaft, multitaskingfähig zu sein. Sie können Programme starten, die von da an völlig unabhängig von der aufrufenden Umgebung laufen - als sogenannter »Hintergrundprozeß«. Das Problem ist jedoch oft, daß die meisten Programme diese Möglichkeit nicht nutzen. Kann man sie dazu zwingen?

Vielleicht kennen Sie das Problem des CLI-Befehls »run«, der für diesen Zweck vorgesehen ist. Run startet zwar einen Hintergrundprozeß, dieser bleibt mit der Standardin- und -ausgabe an die aufrufende Umgebung aber gebunden. Im Klartext heißt das, Ihr mit run gestartetes Programm läuft zwar unabhängig von anderen, aber versuchen Sie einmal, das CLI mit »endcli« zu schließen.

Solange noch mindestens ein mit run gestartetes Programm läuft, verweigert Ihnen endcli seinen Dienst. Zwar bricht das CLI seine Arbeit ab. Aber das Fenster bleibt geöffnet bis das letzte von dort gestartete Programm beendet ist.

Dabei handelt es sich nicht etwa um eine besonders perfide Hinterlistigkeit des Betriebssystems oder des »run«-Befehls. Vielmehr ist dieses Eigenart des CLI als Sicherheitsnetz gedacht für Programme, die zur Ein- und Ausgabe das CLI-Fenster benutzen. Da jedoch die meisten Programme (ausgenommen die CLI-Befehle) ei-

**Die enormen Fähigkeiten des Amiga werden im Zusammenhang mit Multitasking erst richtig deutlich. Leider gibt es dabei mit dem »run«-Befehl einige Probleme. »Runback« heißt die Lösung.**

gene Fenster benutzen, ist diese Vorkehrung unnötig.

Es muß lediglich dafür Sorge getragen werden, daß dieser Fall - sollte er dennoch auftreten - sauber abgefangen wird. »Runback« von Daniel Barrett ist eine Erweiterung der beiden Programme »Runbackground« von Robert Peck und »Which« von Carolyn Scheppner. Dieses Programm arbeitet im wesentlichen wie der Run-Befehl des CLI. Der einzige Unterschied ist die beschriebene Installation eines Hintergrundtasks.

### Wirklich hintergründig

Dieser wird vom CLI abgekoppelt, indem für die Ein- und Ausgabe das »NIL:«-Device verwendet wird. Dieses Device ist kein echtes (wie etwa »PRT:« oder »DF0:«), sondern ein gedachtes. Es ist das »schwarze Loch« von Amiga-DOS, in dem Sie alle unerwünschten Daten verschwinden lassen. Dieses Vorgehen verhindert, daß ein als Hintergrundprozeß installiertes Programm versucht, in das CLI-Fenster zu schreiben (oder aus diesem zu lesen), nachdem es geschlossen wurde. Dies würde zum ungeliebten Guru führen.

Runback verhindert diesen ungeliebten Kumpan durch die Umlenkung ins NIL. Starten Sie auf diese Weise jedoch niemals ein Programm, das für die Eingabe das CLI benötigt. Es würde nutzlos im Speicher stehen.

Für alle anderen Anwendungen ist Runback »die« Lösung. Beispielsweise können Sie damit Programme von der Startup-Sequence aus starten und das CLI-Fenster schließen. Diese laufen wie ein von der Workbench gestartetes Programm.

Runback finden Sie in der neuesten Version auf der Fish-Disk 152 oder auf der Programmservice-Diskette zu diesem Heft. Dieses beseitigt ein Problem, das bei älteren Versionen auftauchte. Bei diesen suchte Runback nicht automatisch den Suchpfad ab. Daher mußte beim Start der genaue Pfad des gewünschten Programms eingegeben werden, auch wenn dieses im C-Directory lag.

Bei der neuen Version geben Sie nur ein:

```
runback progName
```

Für »progName« setzen Sie den Namen eines im aktuellen Suchpfad enthaltenen Programms ein. Außerdem können Sie einen zusätzlichen Parameter übergeben, der Ihrem Task

für eine definierte Zeit den Diskettenzugriff sichert:

```
runback -6 progName
```

hält die Batchdatei, in der diese Zeile enthalten ist, für etwa sechs Sekunden an. Dies ist besonders hilfreich, wenn Sie Runback von der Startup-Sequence aus aufrufen. Sie vermeiden damit unnötiges Durcheinander bei den Diskettenzugriffen. Der Verzögerungswert kann zwischen 0 und 9 liegen. Bei 0 gibt es keine Pause.

Auch ein Programm wie Runback ist zwangsläufig mit Einschränkungen verbunden. So muß zum Gebrauch von Runback das File »Run« im C-Ordner vorhanden sein. Es ist nicht möglich, den Ein- oder Ausgabekanal des gestarteten Programms beim Start umzulenken, da Runback selbst eine Umlenkung ins NIL vornimmt. Die Gesamtlänge der übergebenen Parameter ist maximal 227 Zeichen. Diese Einschränkung dürfte kaum stören.

Die Beschränkungen, die von Runback auferlegt werden, sind von geringer Bedeutung, verglichen mit den Vorteilen. Bereichern Sie Ihre Startup-Sequence um »Autostart«-Programme. Einige Kandidaten sind »VirusX« oder »ScreenX« (siehe Seite 141 und 146). Eine andere Möglichkeit ist, Ihre Textverarbeitung oder ein beliebiges anderes Programm zu starten. Sie brauchen sich nicht mehr um nutzlos im Weg stehende CLI-Fenster zu kümmern, Runback entzieht diesem die Kontrolle über Ihre Programme.

**H**aben Sie schon einmal versehentlich eine Diskette gelöscht? Noch nie? Dann besitzen Sie entweder keinen Amiga oder haben bisher großes Glück gehabt. Irgendwann einmal passiert es jedem: Durch Eingabe von »format drive df0:« – statt »df1:« und vorschnelles <RETURN> ist eine Diskette ruiniert. Was Sie dabei gelöscht haben, sind bis zu 880 KByte mehr oder weniger wichtiger Daten.

Was speichern Sie eigentlich auf Ihrer Festplatte? Wichtige oder weniger wichtige Daten? Wird Ihnen langsam mulmig?

Wenn Sie wichtige Daten auf Ihrer Festplatte aufbewahren, die womöglich auch noch häufig verändert werden, sollten Sie mindestens einmal täglich ein Backup von der Platte machen. Bei Großrechnern ist dies gängige Praxis.

Aber auch wenn sich auf Ihrer Platte »nur« relativ unkompliziert zu beschaffende Programme und Daten befinden, ist es enorm mühsam, alles wieder neu auf die Platte zu packen, was vorher in vielleicht 80 MByte enthalten war. Wieviel einfacher wäre es, wenn irgendwo ein Duplikat aller Daten auf der Platte existierte. Nach einem »Betriebsunfall« verlieren Sie diese dann nicht auf einen Schlag.

Eine zweite Platte ist jedoch viel zu teuer für diesen Zweck. Daneben ist es nicht gerade effektiv, ständig eine identische Kopie parat zu halten. Ein Kompromiß ist das Speichern auf mehrere einzelne Disketten. Dies ist allerdings sehr zeitraubend und nervtötend, wenn Sie »per Hand« kopieren.

»MRBackup« entlastet Sie bei dieser Arbeit. Dieses Programm von Mark R. Rinfret finden Sie auf der Fish-Disk 129 oder auf unserer Programmerservice-Diskette. Es liegt inzwischen in der verbesserten Version 2.1 vor.

Um MRBackup zu starten, klicken Sie einfach auf das entsprechende Icon – wenn Sie die Version von der Programmerservice-Diskette besitzen, muß diese vorher entpackt werden (beachten Sie dazu die Hinweise auf Seite 125). Sie sehen ein neues Fenster, das sich auf einem eigenen Screen öffnet. Es enthält einige Schalter (Gadgets), das auffälligste ist ein rotes Stop-Zeichen. Drücken Sie die rechte Maustaste und sehen Sie sich die Menüs an. Unter »Operations« finden Sie die wichtigsten Steuerelemente. Die Einstellung der wichtigsten Parameter erfolgt unter »Flags«,

ein weiteres Menü (»windows«) zeigt das geschlossene Fenster an und reaktiviert es gegebenenfalls.

Auf der rechten Seite sehen Sie ein »Pathname Specifications Window«. Hier stellen Sie den Pfad ein, aus dem die Quelldateien geholt werden. Zusätzlich legen Sie fest, wohin diese ausgelagert werden. Starten Sie den Backup-Vorgang mit dem Menüpunkt »Backup«. Ein graues Gadget-Feld erscheint links oben. Geben Sie

steht der Name des Gerätes, von dem ein Backup gezogen werden soll. Voreingestellt ist hier »DH0:«. »Backup Path« gibt an, wohin kopiert werden soll. Beachten Sie, daß diese Pfade für das Zurückkopieren nicht verändert werden müssen. »DH0:« bleibt also auch beim Restaurieren der Dateien »Home«.

In »Exclude Path« können Sie ein File angeben, das eine Schablone für Dateien enthält, die vom Backup ausgeschlos-

```
home = DH0:
backup = DF1:
list = s:MRBackup.
      list2
exclude = s:MRBackup.
        xcld
compression = YES
listing = YES
speech = YES
format = YES
bigfiles = YES
archivebits = YES
```

Dabei bedeuten:

**home** und **backup** die oben beschriebenen Einstellungen des Kopierwegs.

**list** ist das File, welches das Protokoll aufnehmen soll. Geben Sie hier nichts an, so wird es auf den Drucker geschickt.

**exclude** ist das oben beschriebene Exclude-File. Diese vier Werte können auch nachträglich im »Pathnames«-Fenster verändert werden.

**compression** spart Platz auf der Backup-Diskette, wenn hier »YES« angegeben wird. Allerdings wird mehr Zeit benötigt, um die Files zu packen und dann zu speichern als sie unverändert auf Diskette zu schreiben.

**listing** schaltet das Protokoll an oder ab.

**speech** schaltet die Sprachausgabe an oder ab. Wenn Sie Ihren Lautstärkeregler nicht so wieso auf Null stehen haben, sollten Sie hier »NO« angeben – die synthetische Sprache des »Say«-Befehls klingt nicht gerade angenehm.

Stellen Sie bei **format** »YES« ein, wenn die Zieldisketten noch nicht formatiert sind. Jede wird in diesem Fall automatisch formatiert.

**bigfiles** erlaubt MRBackup, zu große Files (mehr als 1756 Blocks) über mehr als eine Diskette zu verteilen. Dazu muß das »format«-Flag ebenfalls gesetzt sein.

**archivebits** veranlaßt MRBackup, vor dem Kopieren das Archivbit jedes Files zu testen. Das Archivbit wird bei jeder Veränderung eines Files gesetzt. MRBackup löscht dieses Bit, wenn das File gesichert ist. Sie haben damit eine einfache und effektive Kontrolle, wenn Sie diese Fähigkeit konsequent anwenden.

Die letzten sechs Parameter können Sie auch über das »Flags«-Menü einstellen.

Wenn Sie MRBackup regelmäßig anwenden, wird Ihnen viel Ärger erspart bleiben. Ziehen Sie lieber einmal zu oft als zu selten ein Backup Ihrer Festplatte – ganz nach dem Motto »sicher ist sicher«. □

**Sicher ist sicher**  
 Von Klaus Sonnenleiter  
**Immer mehr Amiga-Besitzer kaufen sich eine Festplatte.**  
**Damit steigt die Nachfrage nach guten Backup-Programmen für diese schnellen Massenspeicher.**  
**»MRBackup« ist eine zuverlässige Kopierhilfe dieser Gattung.**

hier das Datum Ihres letzten Backups an. Auf diese Weise werden nur Files kopiert, die seitdem verändert wurden (Voraussetzung ist eine batteriegepufferte Uhr). Dies erspart Ihnen das Warten auf ein komplettes Backup der ganzen Festplatte. Wenn Sie das Datum eingestellt haben (beim ersten Mal geben Sie am besten ein Datum an, das älter ist als Ihre Platte), klicken Sie auf »OK«.

Ein Requester fordert Sie auf, eine Diskette zum Formatieren in »DF0:« einzulegen. Wenn dies abgeschlossen ist, beginnt der Backup-Vorgang. Dabei wird ein Protokoll auf den Drucker ausgegeben. Diese Ausgabe können Sie jedoch auch in ein beliebiges File umlenken. Normalerweise ist die Druckervariante jedoch durchaus die sinnvollste, denn damit können Sie das Protokoll zusammen mit Ihren Backup-Disketten aufbewahren. So wissen Sie jederzeit, welche Daten auf welcher Diskette sind und in welcher Version.

Werfen Sie jetzt einen Blick auf das Fenster an der rechten Seite. Dort finden Sie einige String-Gadgets. In »Home Path«

sein sollen. Dieses File schreiben Sie mit einem beliebigen Editor (als ASCII-File) und kopieren es am besten in den »S:«-Ordner. Sie könnten damit beispielsweise alle Dateien mit der Endung ».info« ausschließen, indem Sie eine Zeile in dieses File schreiben:

```
*.info
```

oder alle Files, deren Name an einer beliebigen Stelle die Zeichenkombination »den« enthält:

```
*den*
```

Das File kann auch Kommentarzeichen enthalten. Stellen Sie einer Zeile als erstes Zeichen ein Doppelkreuz (»#«) voran, deutete MRBackup dieses als Kommentar.

Das Backup-Programm sucht beim Start im »S:«-Ordner nach einem File mit dem Namen »MRBackup.init«. Dies enthält die Standardeinstellungen für alle Parameter. Ein Beispiel ist auf der Programm-Diskette enthalten. Die Parameter werden in folgender Form gesetzt:

```
Option = Wert
```

Sehen Sie sich das mitgelieferte Beispiel an. Es enthält folgende Zeilen:



**Machen Sie mit !**

# **AMIGA-SONDERHEFT**

**Diesen Beitrag im AMIGA-Sonderheft fand ich besonders gut:**

Ausgabe: \_\_\_\_\_ / \_\_\_\_\_ Seite: \_\_\_\_\_

Artikel: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Ich wünsche mir für eine der nächsten Ausgaben folgende Themen:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Ich möchte an der redaktionellen Gestaltung mitarbeiten.  
Meine Vorschläge:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Ich kann folgende(s) Programm(e) zur Veröffentlichung anbieten:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Dieses Problem habe ich:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



**Ich besitze einen**

**A500** \_\_\_ **A1000** \_\_\_ **A2000** \_\_\_ **A2500** \_\_\_

**mit 1** \_\_\_ **2** \_\_\_ **3** \_\_\_ **mehr** \_\_\_ **Laufwerken**

**mit einer Festplatte** \_\_\_

**Ich verwende einen Drucker** \_\_\_

**mit 9 Nadeln** \_\_\_ **24 Nadeln** \_\_\_

**Zusätzlich besitze ich einen**

**C 64** \_\_\_ **PC** \_\_\_ **Atari ST** \_\_\_ **andere** \_\_\_

.....  
**Diese Note (1 bis 6, 1 am besten) gebe ich dem**

**AMIGA-Sonderheft:** \_\_\_

**Das sollte am AMIGA-Sonderheft besser werden:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

.....  
Name: \_\_\_\_\_

Alter: \_\_\_\_\_ Jahre

Adresse: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

Telefon: \_\_\_\_\_

.....

Bitte schicken Sie die Mitmachkarte  
in einem Briefumschlag  
an folgende Adresse:

**Markt & Technik Verlag AG**  
**Redaktion Sonderhefte**  
**Stichwort: Mitmachkarte Amiga**  
**Hans-Pinsel-Straße 2**  
**8013 Haar b. München**

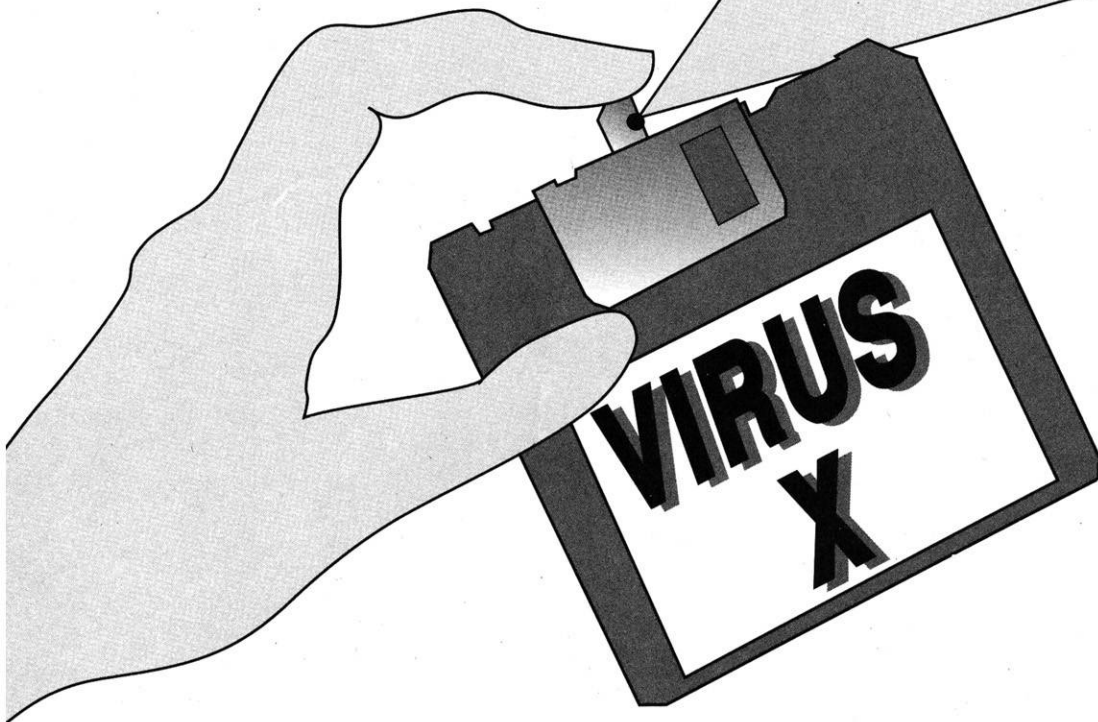


**Schreiben Sie uns!**



# Der stille Wächter

Viren sind auf dem Amiga fast schon etwas alltägliches geworden. Kaum ein Benutzer bleibt von diesen widerlichen Mini-programmen verschont. Mit einem Virus-Killer bleiben Sie trotzdem Herr über Ihre Disketten. »VirusX« lauert als Wächter im Hintergrund.



Von Christian Wolf

**O**hne Virus-Killer sind Sie schutzlos. Einige Programmierer betreiben ein dümmliches Hobby: Sie entwickeln und verbreiten Viren für den Amiga. Viren sind kurze Programme, die Effekte verschiedener Art bewirken. Gemeinsam ist allen Viren, daß sie Daten auf Disketten zerstören. Einige infizieren sogar Dateien auf der Festplatte.

Da einige Insider merkwürdigerweise immer neue Kreativität beim Entwickeln von Viren vergeuden, findet man inzwischen weltweit eine Vielzahl dieser Datenkiller. Die wichtigsten stellen wir im Artikel »Kleine Viruskunde« vor.

Der Amerikaner Steve Tibbett, auch bekannt unter dem Namen Mister X, hat eine gefährliche Gegenwaffe entwickelt. Gefährlich ist »VirusX« (Bild 1) für die bisher am häufig-

sten verbreiteten Datenkiller. Das Programm eliminiert alle Spezies, die wir in der kleinen Viruskunde auf Seite 142 vorstellen.

Steve Tibbett ruft im »docs«-File zum VirusX alle Anwender eindringlich auf, neue Viren an ihn zu senden. Nur dadurch kann er sein PD-Programm weiterentwickeln und neue Varianten bekämpfen. Hier seine Adresse:

Steve Tibbett  
USA-2710 Saratoga  
Pl. # 1108  
Gloucester, Ontario  
K1T 1Z2

VirusX finden Sie in der Version 3.1 auf der Fish-Disk 175. Das Programm ist ebenfalls auf der Programmservice-Diskette zu diesem Sonderheft vertreten. Zusätzlich finden Sie das

Programm »KV« (KillVirus) und vier »DOC«-Files. Die Informationen zu diesem Artikel sind eine überarbeitete und erweiterte Übersetzung dieser Text-Files.

## VirusX geht auf Posten

VirusX starten Sie von der Workbench wie gewohnt durch Anklicken des Programm-Symbols (Icon). Der Virus-Checker öffnet ein eigenes Fenster, von dem Sie nur die Titelzeile »VirusX 3.10 by Steve Tibbett« sehen. Klicken Sie dieses Fenster mit der linken Maustaste an, so daß die Geisterschrift verschwindet. Drücken Sie nun einmal auf die rechte Maustaste. Das Fenster klappt nach unten hin auf. Sie sehen den Statusbericht, in dem die Anzahl der geprüften Disketten, die der installierten und die Anzahl der verschiedenen lokalisierten Viren. Klicken Sie er-

neut mit der rechten Maustaste, während der Mauszeiger sich über dem VirusX-Fenster befindet. Dieses schrumpft wieder auf seine ursprüngliche Größe. Der Viruskiller arbeitet im Hintergrund rege weiter.

Legen Sie nun eine beliebige Diskette in das Laufwerk. Viren oder ein »Non-Standard-Bootblock« werden vom Programm erkannt. Die Aktivität von VirusX erkennen Sie, wenn Sie nach einem Diskettenwechsel das Fenster wieder auf volle Größe bringen. Die Anzeige für »Disks Checked« erhöht sich bei jeder Diskette.

Natürlich ist VirusX auch vom CLI aus zu starten. Mit der Zeile

```
run virusx <RETURN>
```

wird der Schutzpatron in die Betriebsamkeit des CLI eingeflochten. Wer eine individuelle Position des kleinen Fensters bevorzugt, der kann dieses beispielsweise mit

```
run virusx 320 100
```

bei der Bildschirmposition 320/100 öffnen. Von nun an kann mit Gelassenheit in die Zukunft geblickt werden - VirusX informiert Sie exakt über Infektionsgefahren.

Ist Ihr Amiga bereits von einem Virus befallen, wird dies beim Aufruf von VirusX sofort gemeldet. Der Virus kann in den meisten Fällen sogar identifiziert werden. Sie können dann den Störenfried wahlweise ignorieren oder von VirusX entfernen lassen. Jede einglegte Diskette wird auf »Infektion« untersucht, so daß bei ständiger Benutzung von VirusX die am häufigsten benutzten Disketten kuriert werden. Bei positivem Untersuchungsergebnis (das heißt: Virus wurde entdeckt) kann dieser auch von der Diskette entfernt werden.

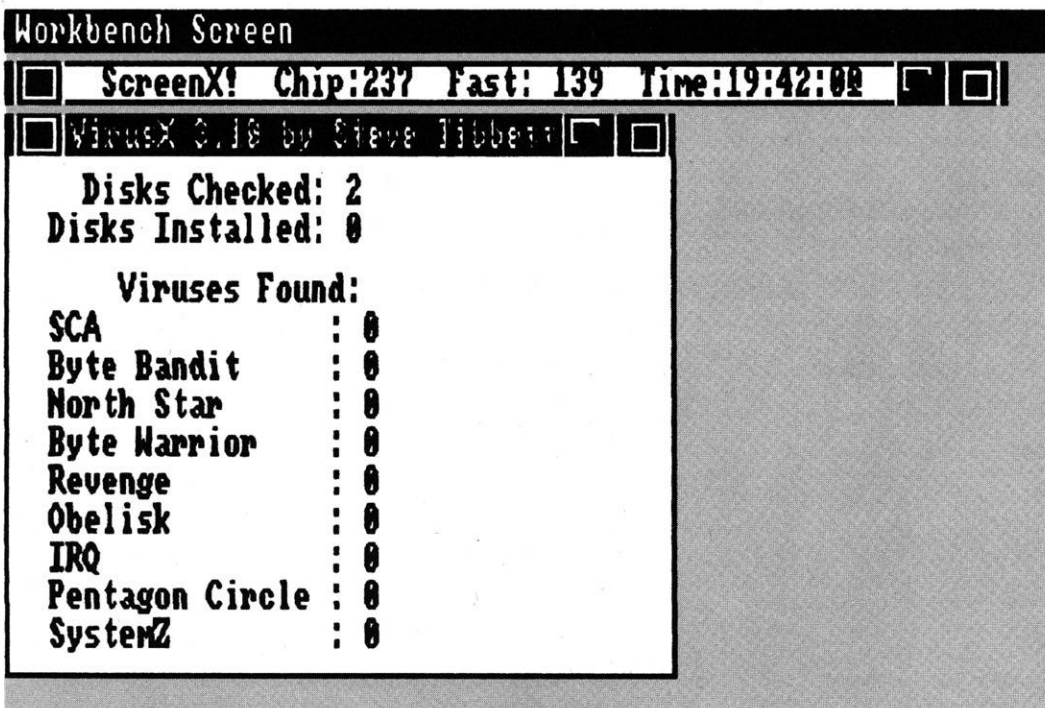


Bild 1. Der Statusreport von VirusX zeigt die entdeckten Virustypen

Da VirusX alle Boot-Blöcke meldet, die nicht das Standard-Format besitzen, sind kommerzielle Programme bedroht. Sollten Sie beispielsweise ein bootfähiges Spiel einlegen, warnt Sie der Virusdetektor. In diesem Fall ignorieren Sie unbedingt dessen Meldung. Sollten Sie den Bootblock versehentlich überschreiben, zerstören Sie damit das Spiel!

Manchmal findet VirusX auch andere Disketten mit einem »Nonstandard-Boot-Code«, also einem Programmcode, der nicht vom Kommando »Install« erzeugt wurde. Das ist dann möglicherweise ein Virus, der dem Programm nicht bekannt ist, also nicht identifiziert werden kann. Lassen Sie dann den vermeintlichen Virus erst einmal »am Leben« und klicken »Ignore it«. Drückt man nun bei aktiviertem VirusX-Fenster eine der Tasten <0> bis <3>, so wird der Boot-Code der Diskette im entsprechenden Laufwerk

**Viren sind in der gesamten Computerszene gefürchtet. Steve Tibbett beschreibt in seiner Anleitung zu VirusX die bekanntesten Amiga-Vertreter dieser ekelhaften Programmattung. Wir haben die wichtigsten Passagen seiner Aufzählung für Sie ins Deutsche übersetzt.**

# Kleine Viruskunde

**V**iren auf dem Amiga. Kleine böse Programme, die Schrecken unter allen Besitzern dieses Computers verbreiten. Gerüchte, Angst und Zorn. Die gesamte Szene ist durch die sinnlosen, in zahlreichen Mutationen auftretenden Schädlinge bedroht. Welche Taktiken setzen Viren ein, wie arbeiten sie? Steve Tibbett beschreibt im »docs«-File

zu VirusX die bekanntesten Datenkiller. Aus der englischen Anleitung wird jedoch nicht jeder schlau. Wir haben die wichtigsten Hinweise von Steve ins Deutsche übertragen und um einige Details ergänzt. Bis auf den IRQ V41 sind alles Bootblock-Viren, das heißt sie kopieren sich auf den Bootblock jeder installierten Diskette. Bevor Sie sich in die Beschrei-

bung der Arbeitsweise der verschiedenen Viren vergraben, ein wichtiger Tip: Sollten Sie auf einen Virus stoßen, den VirusX oder andere Virenkiller nicht eliminieren, schalten Sie den Amiga und jede Erweiterung, beispielsweise eine Festplatte, aus. Warten Sie zirka 15 Minuten und booten von einer Diskette, die nicht verseucht ist. Hoffentlich haben Sie noch eine!

Schicken Sie die vermutlich verseuchte Diskette an Steve Tibbett. Er hat das Zeug dazu, dem neuen Virus das Handwerk zu legen. Die Adresse von Steve finden Sie im Artikel zu VirusX. Legen Sie die infizierte

der, indem er den Bildschirm abschaltet und den Amiga in Lethargie versetzt. Sie beheben diesen Effekt mit gleichzeitigem Drücken der Tasten <ALT-rechts>, <Amiga-links>, <Space>, <Amiga-rechts> und <ALT-links>.

Ist der Byte Bandit-Virus im Speicher, holt er sich den »high memory pointer« und setzt sich direkt hinter den Zeiger auf den ersten Hunk des RAM. Der Virus befindet sich also immer an verschiedenen Speicherstellen. Er klemmt sich in in den Interrupt-Server und den Zeiger des Trackdisk.device. Der Byte Bandit ist resetfest.

Der Virus, der einer der ersten auf dem Amiga war, kontrolliert jede neu eingelegte Diskette und kopiert sich auf jede bootfähige. Der Byte Bandit verbreitet sich deshalb mit rasanter Geschwindigkeit. Er kopiert sich nach dem Installieren erneut auf die Startdiskette, muß also vorher unbedingt aus dem Speicher gelöscht werden.

Wenn VirusX diesen Virus findet, zeigt das Programm zusätzlich den »Copy-Counter«, also die Summe aller bisher von dem Zweig, in der Ihr Exemplar vegetiert, erzeugten Bootblock-Kopien an. Infizieren Sie eine weitere Diskette, wird der Copy Counter um eins erhöht. Der Zähler des Byte Bandit, den Steve gezähmt hat, steht auf 879...

Diskette auf keinen Fall mehr in irgendein Amiga-Laufwerk.

Doch jetzt zur Beschreibung der verschiedenen Viren. Wenn Sie den Amiga gut kennen, entwerfen auch Sie vielleicht bald eine Gegenwaffe für einen neuen Vertreter dieser lästigen Programme.

**Byte Bandit-Virus:** Der Byte Bandit meldet sich hin und wie-



(df0: bis df3:) als ASCII-Code angezeigt. Der Virus ist dann eventuell anhand eines im Bootblock versteckten Textes zu entlarven.

Die statistische Auswertung der Prüfungsergebnisse ist nach längerem Arbeiten mit verschiedenen Disketten interessant. Solange sich VirusX im Speicher befindet, protokolliert es alle Checks.

VirusX erschien schon in mehreren Varianten. Die neue Version (V3.1) kann sogar den IRQ-Virus aufspüren, der zur Fortpflanzung einen anderen Weg geht als die bisher bekannten Schädlinge. Die sogenannten Bootblock-Viren infizieren Disketten durch Beschreiben des Bootblocks mit deren eigenem Programmcode. Auf diese Weise haben Viren die Chance, sich auf allen installierten Disketten einzunisten. Das sind alle die Disketten, die nach der Aufforderung, die Workbench einzulegen,

vom Amiga klaglos akzeptiert werden. Der IRQ-Virus dagegen hängt sich wie ein Schmarotzer an verschiedene CLI-Befehle und ist somit wesentlich schwieriger zu lokalisieren. Die vom IRQ bevorzugt angegriffenen Dateien sind Files im c-Ordner, beispielsweise »dir« oder »copy«. Außerdem fällt der IRQ häufig das erste lauffähige Programm in einer Startup-Sequence an. Das kann zum Beispiel der DOS-Befehl »setmap« sein, oder auch Programme wie »MachII«.

Beim Aufruf eines auf diese Weise verlängerten Befehls gelangt der Virus dann in den Speicher. Der IRQ kann sich in diesem Fall auch auf andere Disketten verbreiten.

Den IRQ Virus V41 lokalisiert VirusX. Diesen Linkvirus beseitigt das Programm aus dem Speicher. VirusX sucht auf dem gleichen Weg wie der IRQ nach Dateien und prüft diese auf Virusbefall. Allerdings ist der

Linkvirus auf Disketten oder Festplatten vor VirusX sicher.

Den Generalangriff auf diesen Linkvirus starten Sie mit dem zusätzlichen Programm, das Sie ebenfalls im VirusX-Ordner finden: KV (KillVirus). Dieses eliminiert alle Dateien, die vom IRQ V41 befallen sind. Sie müssen beim Aufruf von KV

## KV jagt Viren auf der Festplatte

die Datei als zusätzlichen Parameter angeben, die geprüft werden soll. Dabei können auch Directories durchsucht werden. Geben Sie als Aufruf im CLI ein:

```
kv DATEINAME
```

Für DATEINAME setzen Sie den Namen des zu prüfenden Files ein. Directories durchsuchen Sie beispielsweise mit

```
kv dh0:c/#?
```

Nach diesem Aufruf durchsucht KV das c-Directory der

Festplatte mit dem Device-Namen »dh0:«.

VirusX gehört auf jede Start-Diskette. Binden Sie das Programm am besten in die Startup-Sequence ein. Ergänzen Sie diese mit der Zeile

```
virusX
```

Kopieren Sie das Programm vorher in den c-Ordner, kann nichts schiefgehen. Damit sind Sie vor bösen Überraschungen geschützt.

Besonders Amiga-Anfänger sollten auf VirusX keinesfalls verzichten. Die Immunität Ihrer Disketten sollte den geringen Aufwand für die Beschaffung und Installation des Programms wert sein. Die Freaks werden sich die Viren wahrscheinlich weiterhin zur Belustigung als kleine Haustiere auf den Disketten halten... Wir raten davon ab, denn allzusehr schnell wird eines der bösartigen Tierchen unabsichtlich verbreitet.

rs

**Revenge-Virus:** Verhält sich genauso wie der Byte Bandit. Zusätzlich wartet dieser Schädling nach einem Reboot zirka eine Minute. Danach verändert der Revenge-Virus den Mauszeiger in ein bestimmtes Teil der männlichen Anatomie.

Der Name Revenge-Virus stammt wahrscheinlich daher, daß dieser Störenfried nach dem SCA- und Byte Bandit-Virus sucht. Findet er einen seiner Kollegen, manipuliert er diesen auf Diskette, so daß der Amiga abstürzt, wenn der alternative Virus auftaucht. Wenn der SCA also auftritt, verändert der Revenge-Virus diesen. Taucht nun der Byte Bandit auf, stürzt der Amiga ab. Steve ist sich allerdings bei dieser Deutung nicht völlig sicher.

Der Revenge-Virus hält sich im RAM, indem er den Cool-Capture-Vektor so verbiegt, daß er auf dessen Code zeigt. Danach fängt er Aufrufe von DoIO() und wartet auf Aufrufe

Er aktiviert außerdem einen Interrupt, der die vertikale Austastlücke zählt, bis es Zeit zum Verändern des Mauszeigers ist.

Diesen Virus aus dem RAM zu entfernen, ist sehr leicht. Ein Joystick in Port 2 booten und dabei den Feuerknopf drücken – fertig. Der Bildschirm färbt sich dabei kurz rot.

VirusX schafft Ihnen den Virus von der Diskette, aus dem Speicher und damit vom Hals.

**Byte Warrior-Virus:** Wie der Byte Bandit. Der Byte Warrior ist allerdings nicht programmiert, irgend etwas Schädliches zu unternehmen. Wenn dieser Virus einen anderen lokalisiert, warnt er mit einem Alarmton. Er schreibt sich allerdings auch auf jede bootfähige Diskette. Im Virus ist eine versteckte Botschaft. Die Väter des Virus bitten darum, den Virus nicht zu löschen und ihn zu verbreiten. Ja, ja...

**North Star-Virus:** Ein Virus, der Sie vor anderen warnt. Das hält Steve wie auch wir für eine ziemlich merkwürdige Idee, weil dieser gleiche Schäden anrichten kann wie jeder andere.

**SCA-Virus:** Das »Original«, der Veteran unter den Amiga-Viren. Er meldet sich mit der frohen Botschaft »Something wonderful has happened. Your Computer is infected with a Virus«. Der SCA lauert im RAM auf neues Futter und kopiert sich auf jeden Bootblock, den er findet. Sie entfernen den Virus, indem Sie beim Booten den linken Mausknopf gedrückt halten.

**IRQ-Virus:** Arbeitet mit einer neuen Methode. Er wird auch als Link-Virus bezeichnet. Damit ist nicht gemeint, daß diese Spezies besonders link ist. Vielmehr bezeichnet man ihn des-

halb so, weil er sich an diverse Files »linkt«, also ein Programm um den IRQ-Code verlängert. Das geschieht allerdings nur mit lauffähigen (executable) Files. Der IRQ ist, anders als seine Kollegen, nicht im Boot-Block zu finden.

Der IRQ ist im Gegensatz zu den Bootblock-Viren recht harmlos. Es zerstört keine kommerziellen Spiele. Der Link-Virus ändert die Titelzeile des ersten gestarteten CLI-Fensters. Er versucht, sich auf jede eingelegte Diskette zu kopieren. Wenn Sie eine schreibgeschützte einlegen, erscheint ohne erkennbaren Grund der Requester »Volume X ist write-protected«. Sie wissen nun, warum Sie den Schreibschutz **nicht** entfernen sollten.

Das Hauptangriffsziel des IRQ sind Files im c-Ordner. Außerdem hängt er sich mit Vorliebe an Dateien, die in der Startup-Sequence aufgerufen werden. Das kann beispielsweise der Befehl »run« aus dem System-Ordner sein.

Diese Kette von Ereignissen kann zum Beispiel durch den IRQ ausgelöst werden:

- Sie erhalten ein neues Programm, beispielsweise per DFÜ, das infiziert ist.
- Sie starten dieses Programm.
- Der Link-Virus installiert sich im Speicher. Er manipuliert dazu den OldOpenLibrary()-Vektor und fügt einen KickTagPtr an (wer weiß warum?).

- Sie befinden sich auf df0: und starten ein Programm, das den OldOpenLibrary()-Vektor benutzt (schwer vorherzusagen, welches das nicht macht). Der Virus öffnet dann die Startup-Sequence und greift sich den ersten Filenamen heraus. Wenn das zugehörige Programm ausführbar ist, hängt sich der IRQ auf der Diskette an dieses. Ist das Programm nicht ausführbar, hängt sich der Virus häufig an den »dir«-Befehl im c-Ordner.

Mit VirusX entfernen Sie den IRQ V41 aus dem Speicher. Zum Eliminieren des Link-Virus ist zusätzlich zum VirusX das Programm »KV« (KillVirus) mitgeliefert. Mit diesem Utility verbannen Sie den IRQ von Ihren Disketten. Die genaue Anleitung finden Sie im Artikel zu VirusX in diesem Heft.

Die Geschichte der Viren ist wohl leider noch nicht abgeschlossen. Neue Arten werden auftauchen, neue Viruskiller gegen diese entwickelt werden. Irgendwie erinnert mich dieser Zweikampf an den Wettlauf, den die Kopierschützer mit den Crackern bestreiten. Mein Vorschlag an die Programmierer: Beenden Sie die Rennen. Nur dann können Sie mit aller Kreativität nützliche Programme entwickeln. Viren sind abschaulich, sie lähmen die gesamte Entwicklung des Amiga. Und daran hat wohl niemand Interesse.

rs

# Aus unserem Bücher-Regal

Der PD-Pool wird durch zahlreiche Neuerscheinungen von Tag zu Tag unübersichtlicher. Einige Fachbücher widmen sich dem Thema PD für den Amiga. Die wichtigsten stellen wir Ihnen hier vor.

## Das große Amiga Public Domain-Buch

Der Klassiker unter den PD-Büchern für den Amiga - so könnte man dieses Werk bezeichnen. Auf 350 Seiten beschreibt es eine Auswahl von Public Domain-Programmen verschiedener Serien. Neben einer ausführlichen Einleitung als Wegbegleiter im Reich der PD-Software bietet das Buch unter anderem folgende Kapitel: Anwendungsprogramme, Grafiksoftware, Arbeitshilfen, Programmierhilfen.

Jeder dieser Abschnitte befaßt sich mit verschiedenen Programmen aus dem jeweiligen Bereich. Der Anwender wird dabei nicht mit einer reinen Funktionsbeschreibung alleine gelassen. Zu jedem Programm



den Umgang mit den verschiedenen Utilities dar.

Bei der Auswahl wurden die Fish-, Faug- und die Panorama-Serie berücksichtigt.

Im Anhang befindet sich eine Übersicht über alle Programme der Fish-Disks 1 bis 127, der Faug-Reihe von 1 bis 39 und Panorama von 1 bis 17b. Neben ein paar Stichworten zur Funktion werden alle Programme nach ihrer Art bestimmten Kategorien zugeordnet.

Das Buch ist für PD-Interessierte sehr nützlich. Allerdings verliert man beim Lesen hin und wieder die Übersicht. Zwar erfolgt bei jedem Programm eine Unterteilung in Kapitel, wie Beschreibung, Herkunft etc., jedoch sind die Namen der Programme nicht klar hervorgehoben. So geht vor allem beim kurzen Nachschlagen schnell der Überblick verloren.

Als zusätzlichen Service bietet Technic-Support alle im Buch beschriebenen Programme, auf zehn Disketten zusammengefaßt, zum Kauf an.

*Martin Jobst/rs*

Das große Amiga Public Domain Handbuch, Ram/Hertwig (Hrsg.), Technic Support 1988, 350 Seiten, ISBN 3-926847-01-8, Preis 49 Mark.

## Amiga Public Domain-Dokumentation

Mit der Serie von Fred Fish setzt sich dieses erst kürzlich erschienene Buch in erster Linie auseinander. Alle Program-

me, die bis Fish 146 erschienen sind, werden meist in ein, zwei Absätzen kurz beschrieben. Die Angaben beziehen sich in erster Linie auf die Funktion und den Aufruf der einzelnen Programme. Teilweise wird auch auf deren Bedienung eingegangen. Somit ist das Buch als Arbeitshilfe nur bedingt geeignet. Es ist eher als Nachschlagehilfe nach bestimmten Programmen gedacht. Als solche eignet es sich dann auch wunderbar.

Vor allem durch die übersichtliche Gestaltung findet man schnell ein bestimmtes Programm und Informationen dazu, wie Name, Aufruf und vor allem Funktion. Eine alphabetische Aufstellung aller Programme der Fish-Disks am Ende des Buches hilft zudem beim Auffinden von Programmen.

In einem weiteren Kapitel wird auf die deutsche Serie RW von Rainer Wolf eingegangen. Auch hier werden alle Programme der Serie bis Disk 14 kurz beschrieben.

Weiterhin findet auch die Serie des Computer-Clubs International (CCI) kurz Erwähnung. Ein farbiger Bildteil in der Mitte des Buches zeigt eine Auswahl grafischer Leckerbissen der Fish-Disks.

Das Buch hilft jenen weiter, die sich über die Funktionen einzelner PD-Programme informieren wollen. Eine Arbeitsanleitung ist es jedoch nicht. Jeder muß selbst entscheiden, ob sich die Anschaffung eines derartigen Nachschlagewerkes lohnt.

Das dürfte nicht auf die Amiga-Besitzer zutreffen, die bereits Listen mit Kurzbeschreibungen, beispielsweise aus diesem Sonderheft, besitzen.

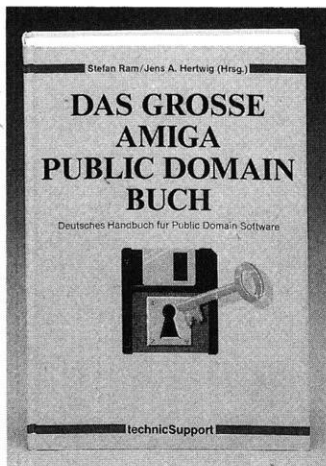
*Martin Jobst/rs*

Amiga Public Domain-Dokumentation, Laub/Wenzl, Markt & Technik 1989, 296 Seiten, ISBN 3-89090-675-3, 49 Mark.

## Das zweite Amiga Public Domain-Buch

Das bewährte Konzept vom ersten Band dieser Reihe (siehe oben) wurde konsequent beibehalten. Eine Auswahl von Programmen wird in aller Ausführlichkeit beschrieben und erklärt. Das Kapitel Arbeitshilfen nimmt auch in diesem Band wieder einen großen Teil ein. Auf rund 100 Seiten werden dort allerhand nützliche Utilities beschrieben, die die tägliche Arbeit mit dem Amiga erleichtern. Positiv schlägt zu Buche, daß die Auswahl durch das Kapitel Programmiersprachen erweitert wurde. So wird dort der Modula-Compiler »M2Amiga«, vor allem dessen Installation und Bedienung, erklärt.

Interessant ist auch, daß nun auch auf Programme der Serie RPD, die in Deutschland relativ neu ist, eingegangen wird. Neben der Beschreibung von Programmen der RPD-Serie findet man im Einleitungskapitel eine kurze Charakteristik dieser Reihe. Die Einleitung hilft auch jenen weiter, die im PD-Dschungel neu sind. Die nötigen Kenntnisse für den Umgang mit Dis-



gibt es eine ausführliche Bedienungsanleitung. Außerdem wird auf Probleme beim Arbeiten mit einzelnen Programmen eingegangen. Da kann sich die Abhandlung von einem Programm schon mal über zwanzig Seiten erstrecken. Darin ist auch der unmittelbare Wert des Buches zu sehen. Es wartet nicht mit einer Unmenge von verschiedenen Programmen auf und beschreibt kurz deren Funktion - dazu sind ja schließlich Übersichtslisten da -, sondern stellt eine Arbeitshilfe für



ketten sind hier genauso zu finden wie Grundlagen zur frei kopierbaren Software. Kurzberichte über die PD-Szene schließen sich an.

Die Listen im Anhang wurden erweitert. Fred Fish wurde fortgeführt von 128 bis 146. Die RPD-Disketten sind bis zur Diskette 123b gelistet. Wie auch schon der erste Band, wurde das gesamte Buch mit dem Satzprogramm TeX auf dem Amiga erstellt. Alle beschriebenen Programme sind gesammelt auf elf Disketten als Paket zum Preis von 95 Mark erhältlich.

Insgesamt wieder ein gelungenes Werk, das sich auf den Lorbeeren des Vorgängers nicht ausruht und einige Verbesserungen bringt. So die beschriebene Erweiterung der Kapitel und das leicht verbesserte Druckbild. Ein würdiger Nachfolger.

Martin Jobst/rs

Das zweite Amiga Public Domain Buch, Leithaus/Hertwig (Hrsg.), Technik Support 1988, 384 Seiten, ISBN 3-926847-05-0, 49 Mark.

## Das dritte Amiga Public Domain-Buch

Rund 90 weitere Programme beschreibt der Dritte im Bunde der PD-Bücher von Technik-Support. Die bewährte Auswahl an interessanter Software aus dem PD-Pool wurde um zwei Kapitel erweitert. Der Abschnitt »Diskettenmagazine« informiert über diese neue Art der Informationsverbreitung. Beschrieben werden die drei deutschen PD-Magazine »AMIGA Faszination«, »AmigaJUICE« und »Geti«. Dabei wird Einblick in Inhalt, Absichten und Nützlichkeit dieser Magazine gegeben. Äußerst hilfreich ist das Kapitel »Update-Informationen«. Es informiert über neue Versionen von in Band I und II beschriebenen Programmen, etwa »ConMan 1.3«, »VirusX 1.6« oder »Xicon 2.01«.

Das Einleitungskapitel gibt Arbeitshinweise für Neulinge

im Reich der PD-Software und beschreibt in der »anderen PD-Story« ein Anwendungsbeispiel, verpackt in eine ansprechende Geschichte.



Die Listen am Ende des Buches wurden ebenfalls fortgeführt. Auf rund 50 Seiten wird ein Überblick über gängige Serien gegeben. Gelistet werden

die Serien ACS von 1 bis 90, A.U.S.T.R.I.A. von 1 bis 6, FAUG von 40 bis 51, Fred Fish von 147 bis 172, Panorama von 17 bis 30c, RPD von 124 bis 150 und TBAG von 1 bis 19. Wie bei den Vorgängern werden alle Programme in einer systematischen, einer alphabetischen und in einer Liste nach Disketten aufgeführt. Die Übersichtlichkeit wurde durch klar abgehobene Bereichsüberschriften erheblich verbessert.

Mit seinen 415 Seiten bietet das Buch wieder eine Menge Stoff für alle Freunde der freikopierbaren Software. Gerade in der ausführlichen Beschreibung der einzelnen Programme und dem Schildern möglicher Probleme bei der Arbeit mit diesen, liegt der Wert dieses Buches. Im Herbst dieses Jahres soll der Nachfolger – als vorerst letztes Buch dieser Reihe – erscheinen.

Das dritte Amiga Public Domain Buch, Leithaus/Hertwig (Hrsg.), Technik Support 1989, 415 Seiten, ISBN 3-926847-06-9, 49 Mark.

# Programmservice

## preiswert wie nie

Der Programmservice wird beim Markt & Technik Verlag groß geschrieben. Die Programme, die in den AMIGA-Sonderheften veröffentlicht werden, sind auf den Disketten dieser Reihe zu finden. Vielleicht haben Sie schon eine Diskette aus dieser Reihe erworben. Der Preis für diese Produkte lag bisher, wie auch in Zukunft, bei 29,90 Mark. Sind Sie erstaunt, daß für das Sonderheft 5 eine Diskette im gleichen Format für nur 11,90 Mark angeboten wird? Das hat mehrere Gründe, auf die wir an dieser Stelle kurz eingehen müssen.

Sollten Sie schon eine Diskette zum AMIGA-Magazin oder AMIGA-Sonderheft erworben haben, kennen Sie das Leistungsangebot, das damit zusammenhängt: Bei auftretenden Problemen können Sie sich jederzeit an den Programm-Support wenden. Fragen zu den bisher veröffentlichten Programmen konnten wir in der Redaktion beantworten, fehlerhafte wurden soweit möglich korrigiert. Dieser Service muß bei der Diskette zum Sonderheft 5 entfallen.

Während in anderen Heften Leser-Listings veröffentlicht

**Die Service-Diskette zu diesem Sonderheft kostet nur 11,90 Mark. Dieses einmalige Angebot hat viele Gründe.**

werden, handelt es sich bei den Programmen auf der aktuellen Diskette um PD-Software aus aller Welt. Alle Autoren sind in kurzen »Read\_Me« oder »Doc«-Files auf der Diskette zu finden. Wenden Sie sich bitte bei allen Fragen an die Autoren. Nur diese haben das nötige Know-how, nur diese können bei Problemen weiterhelfen.

»Warum sind die Service-Disketten nicht immer so preiswert?« Diese Frage wird vielleicht bei einigen Lesern auftauchen. Auch diese möchten wir beantworten.

Wie schon erwähnt, sind die Programme in den AMIGA-Sonderheften normalerweise Leser-Listings. Diese werden uns zur Veröffentlichung angebo-

ten. Die besten Einsendungen wählen wir aus, davon profitieren dann alle Leser. Das Recht zur Veröffentlichung eines Programms wird dabei vom Markt & Technik Verlag dem Einsender »abgekauft«. Bei der Veröffentlichung erhält der Autor ein angemessenes Honorar. Diese Zahlungen entfallen bei einer Public Domain-Diskette selbstverständlich. Außerdem ist der niedrige Preis der Diskette nur dadurch zu realisieren, daß für den Programm-Support keine Kosten anfallen – die übliche Unterstützung entfällt.

Vielleicht sind Sie von PD-Anbietern Preise gewohnt, die unter 10 Mark liegen. In den Anzeigen wird mit Preisen »ab 3 Mark« oder »ab 4 Mark« gewor-

ben. Bedenken Sie dabei, daß auf der Service-Diskette die Perlen des PD-Marktes enthalten sind. Die niedrigen Preise einiger Anbieter sind nur dann zu realisieren, wenn eine ganze Reihe von verschiedenen Disketten abgenommen werden. Dann verringern sich sowohl der Arbeitsaufwand für den Versender, die Porto- und Verpackungskosten. Das trifft natürlich bei der Service-Diskette, die in dieser Form einzigartig ist, nicht zu.

Trotz des niedrigen Preises für die Diskette wünschen wir, daß keine Probleme mit der Diskette auftreten werden. Deshalb hier ein Wort zum Packer »ZOO«, mit dem die Daten auf der Service-Diskette komprimiert wurden. Der Packer erlaubt es, mehr als 880 KByte Daten auf der Diskette zu speichern. Der Umgang mit ZOO ist in diesem Sonderheft auf Seite 125 ausführlich beschrieben. Bitte lesen Sie dort nach, wie Sie die Daten auf der Service-Diskette entpacken und auf eine andere Diskette kopieren.

Wir wünschen Ihnen viel Vergnügen mit den sorgfältig ausgewählten Programmen, die die Arbeit mit dem Amiga für jeden erleichtern.

rs

## Flottes Auskunftsbüro

Informationen über den verbleibenden Speicherplatz und die aktuelle Zeit zeigt »ScreenX«. Das Hilfsprogramm von Steve Tibbett kann noch mehr.

Von Christian Wolf

Vielleicht kennen einige unter Ihnen das Utility »mclk«, das dem Benutzer genaue Auskunft sowohl über Zeit als auch noch freien Speicherplatz gibt. Die Anwender mit geringem Speicher werden dies schon aus dem Grunde schätzen, da man so schon frühzeitig und intuitiv gewissen Situationen aus dem Weg gehen konnte, die sonst leicht zu einem Systemabsturz führten. Das Amiga-Betriebssystem erweist sich nämlich bei Speicherknappheit als äußerst empfindlich. Aber auch der Programmierer, der sich mangels Zeitgefühl des öfteren mal bis spät in die Nacht seinen Problemen widmet, kommt so auf seine Kosten (oder auch nicht...). Denn allzu leicht verpaßt man so manche Verabredungen oder Termine, eine ständige und konsequente, vielleicht auch mahnende Uhrzeit schafft hier Abhilfe.

»ScreenX« geht über die eben geschilderten Fähigkeiten noch hinaus. Mit

run ScreenX

wird das Hilfsprogramm aufgerufen. Zunächst wird ein Screen geöffnet, der dann die volle Leistung offenbart. Wie der Name schon vermuten läßt, bekommt man nun einen Überblick über alle im System geöffneten Screens. Im Normalfall sind dies der »Workbench Screen« und »ScreenX's Screen«. Nachdem nun einer der gerade vorhandenen Screens angeklickt, also ausgewählt wurde, lassen sechs Gadgets verschiedene Aktionen zu:

### Pop to Front

Der gewählte Screen wird sozusagen über alle Screens hinweg nach vorn gezogen. Dies ist sinnvoll, da nicht alle Screens

durch das »nach-hinten-legen« anderer erreicht werden. Danach schließt sich »ScreenX's Screen« und man bekommt die übliche Zeit- und Speicheranzeige in einem kleinen Fenster zu sehen.

### Push to Back

Der gewünschte Screen wird nach hinten gelegt. Dessen Vorgänger gelangt somit nach vorn.

### Update Screen List

Hat sich in der Zwischenzeit in der Screen-Liste etwas geändert, wird mit dieser Funktion der Überblick aktualisiert.

### Cycle all Screens

Es werden der Reihe nach sämtliche Screens der Liste nach vorn gezogen, bis der Benutzer einschreitet.

### Close Screen

Wie Sie wohl erraten werden, kann hiermit der gewünschte Screen geschlossen werden. Dies wird mit Sicherheit zum Absturz des Systems führen, wenn der betroffene Screen noch von irgendwelchen Programmen benutzt wird.

### Save Screen to IFF:

Hiermit läßt sich der spezifizierte Screen als IFF-ILBM-File speichern, so daß sich die Grafik Daten eines Screens in beliebigen Malprogrammen weiterverwerten lassen. Dazu findet man noch ein weiteres Gadget, in dem sich der Name des Files eingeben läßt.

Üblicherweise möchte man »ScreenX« in seine Startup-Sequence einbauen und seine zusätzlichen Fähigkeiten nur auf Wunsch herbeirufen. Mit

run screenX -s

wird das Utility gestartet, so daß die Zeit- und Speicheranzeige in der rechten oberen Ecke installiert wird. Um nun »Screen« aus diesem »schlafenden Zustand« zu wecken, klickt man dessen Fenster an und drückt die rechte Maustaste — und schon ist es »wach«.

Ein erneuter Druck auf die Maustaste läßt es wieder einschlafen. Um ScreenX's Anzeige möglichst genau zu halten, erfolgt zirka 6mal in der Sekunde eine Auffrischung des Fensters. Mit der Option »-E« wird »ScreenX« angehalten, dies nur noch alle zwei Sekunden zu tun, so daß man wirklich nicht mehr um fehlende CPU-Zeit bangen muß. rs

## Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

**Chefredakteur:** Hans-Günther Beer

**stellv. Chefredakteur:** Gottfried Knechtel – verantwortlich für den redaktionellen Teil

**Chef vom Dienst:** Susanne Kirmaier

**Redaktion:** Ralf Sablowski, Klaus Sonnenleiter

**Redaktionsassistenten:** Brigitte Bobensteller, Sylvia Derenthal, Helga Weber (202)

**Mitarbeiter der Redaktion:** Martin Jobst, Andreas Lietz, Christian Wolf, Heiko Schlichting, Thorsten Froese, Jan Ippen, Peter Sack, Dietmar Inäbnit, Marco Vitolini-Naldini, Nikolaus Huber

Alle Artikel sind mit dem Kennzeichen des Redakteurs

(kn = Gottfried Knechtel, rs = Ralf Sablowski, so = Klaus Sonnenleiter) und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

**Art-director:** Friedemann Porscha

**Layout:** Erich Schulze (Chellayout), Marian Schwarz, Johanna Schneider

**Fotografie:** Sabine Tennstaedt, Ilona Wiewiorra, Janos Feltser

**Titelgestaltung:** Friedemann Porscha, Rolf Boyke

**Spritzgrafik:** Norbert Raab, Ewald Standke

**Computergrafik:** Werner Nienstedt

**Auslandsrepräsentation:**

Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 5656, Telex: 862329 mut ch

USA: M&T Publishing Inc., 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351

Österreich: Markt & Technik Ges. mbH, Hermann Raniger, Große Neugasse 28, A 1040-Wien, Tel. 0043-222-8579455, Telex: 047-132532

**Manuskripteinsendungen:** Manuskripte und Programm Listings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programm Listings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Produktionsleiter:** Klaus Buck (180), Wolfgang Meyer (Stellv.) (887)

**Anzeigenleitung:** Alicia Clees (313) – verantwortlich für Anzeigen

**Anzeigenformate:** 1/2 Seite ist 266 Millimeter hoch und 185 Millimeter breit (2 Spalten à 86 Millimeter oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter.

**Anzeigenverwaltung und Disposition:** Lisa Landthaler (233)

**Anzeigen-Auslandsvertretung:** England: F. A. Smyth & Associates Limited, 23a, Aylmer Parade, London, N2 0PQ. Telefon: 0044/1/3405058, Telefax: 0044/1/3419602  
Taiwan: Third Wave Publishing Corp., 1-4 Fl. 977 Min Shen E. Road, Taipei 10581, Taiwan, R.O.C., Tel. 00886/2/7630052, Telefax: 00886/2/7658767, Telex: 078529335

**Vertriebsleiter:** Helmut Grünfeldt (189)

**Verkaufsleiter Abonnement:** Benno Gaab (740)

**Verkaufsleiter Einzelhandel:** Robert Riesinger (364)

**Vertrieb Handelsauflage:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1,

**Bezugsmöglichkeiten:** Leser-Service: Telefon (089) 46 13-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

**Preis:** Das Einzelheft kostet DM 16,-

**Druck:** SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

**Urheberrecht:** Alle in diesem Heft erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen, gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Für Schaltungen, Bauanleitungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind.

**Sonderdruck-Dienst:** Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-776.

© 1989 Markt & Technik Verlag Aktiengesellschaft  
Redaktion »Sonderhefte«

**Redaktionsdirektor:** Michael M. Pauly

**Vorstand:** Otmar Weber (Vors.), Bernd Balzer

**Leiter Unternehmensbereich »Populäre Computerzeitschriften«:**  
Eduard Heilmayr, Werner Pest

**Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:** Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 46 13-0, Telex 5-22052

**Telefon-Durchwahl im Verlag:** Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

**Mitteilung gemäß Bayerischem Pressegesetz:** Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Carl-Franz von Quadt (Vorsitzender), Dr. Robert Dissmann (stellv. Vorsitzender), Eduard Heilmayr



Mit **ComicSetter** können Sie Ihre eigenen Cartoons schreiben, zeichnen und editieren – auch wenn Sie kein Zeichenprofi sind. Figuren und Hintergrundszenen werden fertig mitgeliefert, Sie müssen sie nur nach Ihren Wünschen zusammenstellen. Beim Entwerfen von Szenen stehen Ihnen eine einfach zu bedienende Benutzeroberfläche und eine Vielzahl von Mal- und Zeichenwerkzeugen zur Verfügung. Erfinden Sie die Helden Ihrer Geschichte. Plazieren Sie sie nach Belieben in den verschiedenen Szenen.

Und das alles natürlich in einer fast unbegrenzten Farbvielfalt.

Sie werden erstaunt sein, in welcher kurzer Zeit Sie Ihre Comics zu Papier bringen können.

**Bestell-Nr.: 54130**  
**Preis: DM 198,-\*** (sFr 178,-\*/öS 1980,-\*)  
**Update 54130U**  
**DM 49,-\*** (sFr 49,-\*/öS 490,-\*)

Zusatzdisketten zu **ComicSetter** mit einer Vielzahl von Figuren und Szenen aus den Bereichen Superhelden, Science Fiction und Funny Figures:

**ComicArt Super Heroes**  
 Bestell-Nr.: 54123  
**ComicArt Science Fiction**  
 Bestell-Nr.: 54124  
**ComicArt Funny Figures**  
 Bestell-Nr.: 54125

Preis je Produkt: DM 69,-\*  
 (sFr 62,-\*/öS 690,-\*)

\*Unverbindliche Preisempfehlung

# COMIC SETTER



**ENTWERFEN SIE  
 IHRE EIGENEN  
 COMICS**

Markt&Technik-Produkte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler.

**Markt&Technik**  
 Zeitschriften · Bücher  
 Software · Schulung

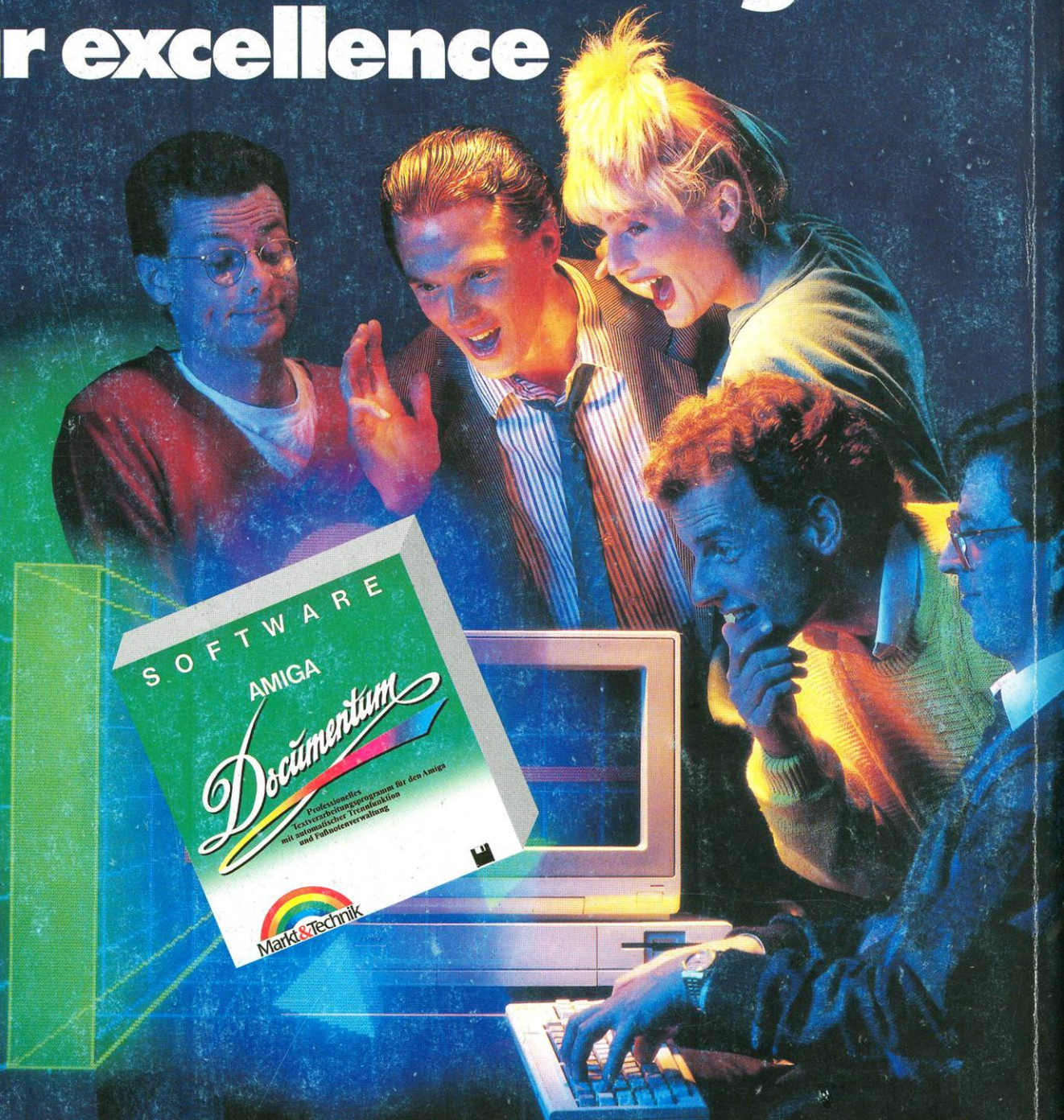
Fragen Sie Ihren Fachhändler nach unserem kostenlosen Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!

**Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0**

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 37, 6300 Zug, Telefon (042) 440550, Fax: (042) 41 57 70.  
 ÖSTERREICH: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 587 1393-0;  
 Ueberreuter Media Verlagsges.m.bH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 48 15 43-0.



# Textverarbeitung par excellence



Textverarbeitungsprogramme auf dem Amiga wurden bisher von Profis belächelt. Sie waren langsam wie Schreibmaschinen, bunt wie Kindergeburtstage, absturzsicher wie Starfighter oder teuer wie Schweizer Uhren. Bis Documentum kam... Rasend schnell • Automatische Trennhilfe • Fußnoten-

verwaltung • Editier-/WYSIWYG-Bildschirm • Kopf-/Fußzeilen • Seitennumerierung • Blockbearbeitung • Alle Amiga-Zeichensätze nutzbar • Tastatur- oder Mausbedienung • Buchdruckoption

• Suchen/Ersetzen • NLQ-Druck für alle Drucker • multitaskingfähig • und vieles, vieles mehr... Für Briefschreiber, Studierende, Autoren, Bürokraten - eigentlich für alle. Bestell-Nr. 54122

**DM 149,-\*** (sFr 135,-\*/öS 1490,-\*)  
\* Unverbindliche Preisempfehlung



Markt&Technik-Produkte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler.

**Markt&Technik**  
Zeitschriften · Bücher  
Software · Schulung

Fragen Sie Ihren Fachhändler nach unserem kostenlosen Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 37, 6300 Zug, Telefon (042) 440550, Fax: (042) 415770.  
ÖSTERREICH: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0;  
Überreuter Media Verlagsges.m.bH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.