

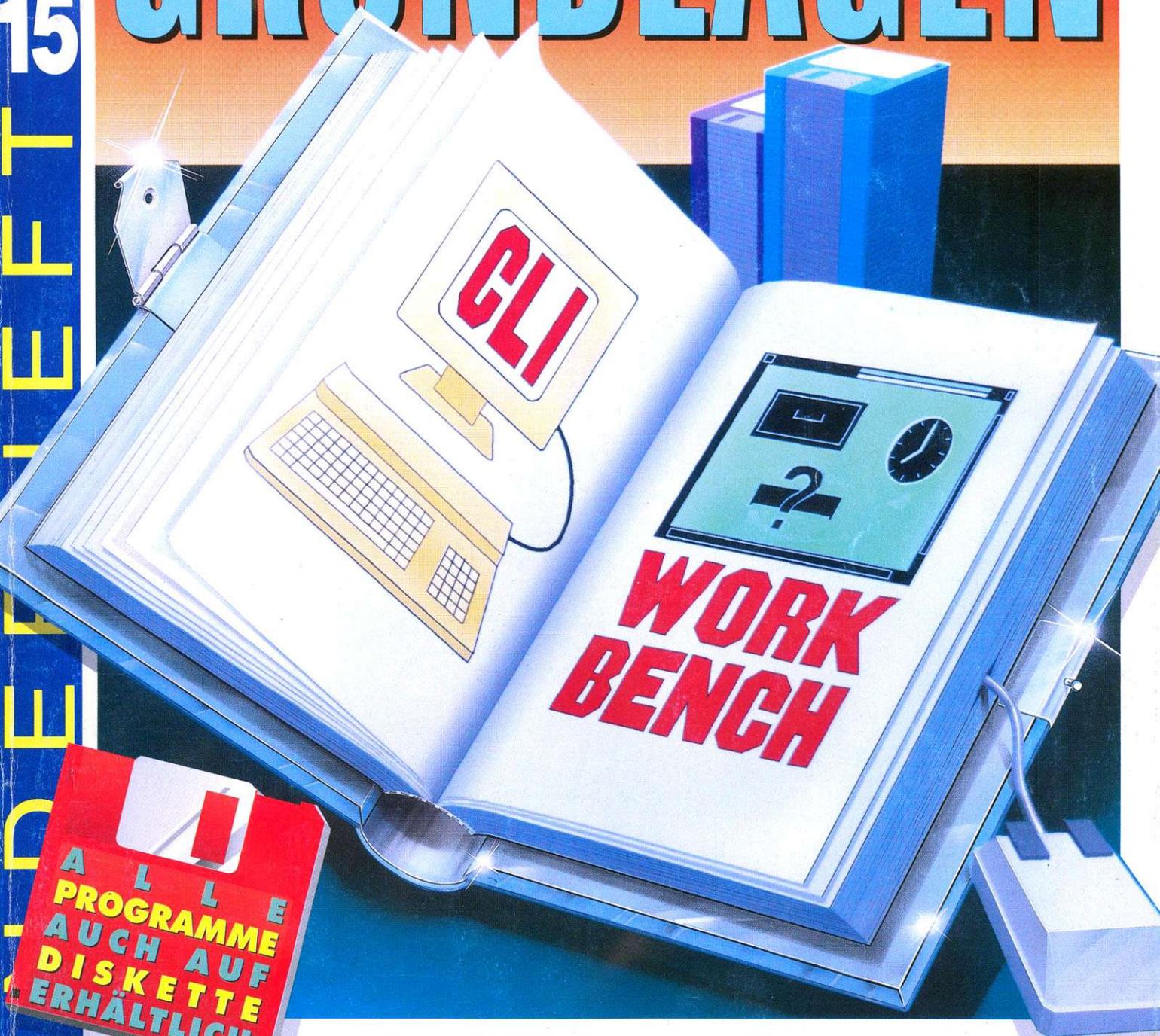
AMIGA
15
EFF
E
D
S

AMIGA

Markt&Technik

DM 16,-
ÖS 120,-/Sfr 16,-/Lit 16000
hfl 21,-/dkr 75,-/fmk. 43,-

GRUNDLAGEN



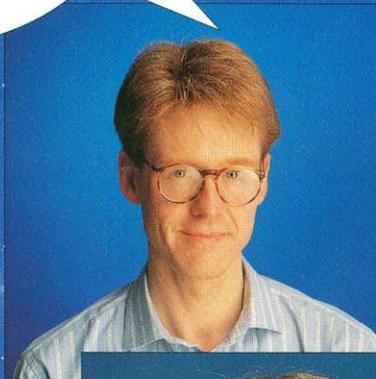
- **Leicht verständlich:** Einführung in das CLI mit vollständiger Befehlsübersicht
- **Durchleuchtet:** Workbench 1.3 und Batch-Dateien
- **Wertvoll:** Listings mit Eingabehilfe
- **Professionell:** Kompletter Kurs für die Benutzeroberfläche

Die neue POWER PLAY ist da!

Der Nachfolger von Populous ist da: Powermonger startet zum Großangriff auf alle Diskettenschächte.

Powermonger

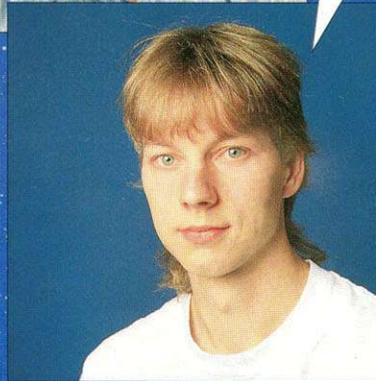
Electronic Arts' neues Strategiespiel bietet eine Weltsimulation mit einzigartigem Detailreichtum. Der neueste Streich der Bullfrog-Truppe ist strategisch wesentlich gehaltvoller als der Vorgänger.



Welches Videospiele-System ist das Beste? Wir vergleichen und bewerten die wichtigsten Konsolen.

Videospiele-Vergleich

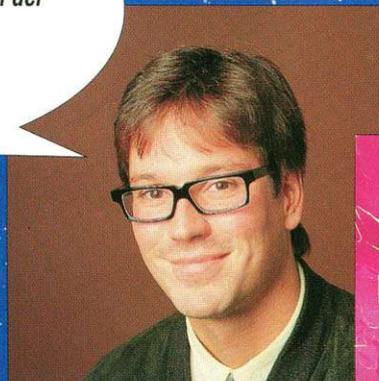
In unserem ausführlichen Vergleichstest der Videospiele bleibt keine Frage unbeantwortet. Wer bietet die beste Grafik, wo klingen die Bässe am kräftigsten, wofür gibt's die brilliantesten Spiele? Acht Seiten kompetente Kaufberatung werden geboten.



Weihnachten naht: Was für einen Computer kauft sich der ambitionierte Spielefreak?

Computer-Vergleich

Computer sind teuer: Um Euch vor einem Fehlgriff zu bewahren, nehmen wir die wichtigsten Computersysteme ausführlich unter die Lupe - ganz unter dem Aspekt des Spielers.



12/90 DAS GROSSE COMPUTER- UND VIDEO-SPIELE-MAGAZIN

Markt & Technik
DM 6,50

6S 50 - /str. 650
Lit. 6900/hlt. 8-
dtkr 35.-/rmx. 24.-

Holt Euch
POWER PLAY
jetzt bei
Eurem
Händler!

POWER PLAY



Nur zum Spielen?

● Viereckige Augen, verbissener Gesichtsausdruck, Joystick in den schweißnassen Händen. Dieses Bild schwebt vielen vor, wenn vom Amiga die Rede ist. Leider sind Heimcomputer immer noch – oder schon wieder? – mit dem Klischee nächtelanger Spielsucht behaftet. Daher wundert es nicht, wenn Eltern auf den Wunsch ihres Spröbblings nach einem Computer oft mit Ablehnung reagieren. Zugegeben: Spielen ist die schönste Nebensache der Welt. Aber es läßt sich weiß Gott auch noch anderes mit einem Computer anstellen.

● Die »kleinen« Computer verfügen über eine breite Palette an Möglichkeiten, die den Anwender unterstützen:

– Gesammelte EDV-Erfahrungen können im späteren Berufsleben genutzt werden.

– Die Schöpfung eigener Programme fördert das kreative Denken.

– Viele Anwendungsprogramme (z.B. Text- und Datenverarbeitungen) helfen, anfallende Arbeiten leichter zu bewältigen.

● Der Amiga mauserte sich zu einer vollwertigen Anwendungsmaschine. Mittlerweile gibt es auf dem großen Software-Markt interessante Beweise für diese Entwicklung.

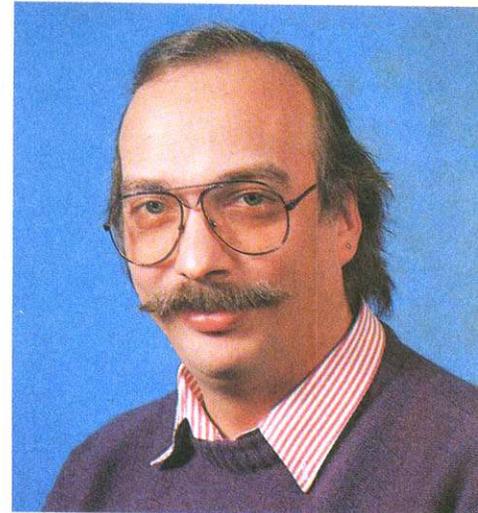
● Als solche sind die hervorragenden Textverarbeitungsprogramme, wie z.B. »Beckertext« oder »Wordperfect« zu nennen. Auch für die Verwaltung Ihrer Daten sind Programme erhältlich, die sich vor professioneller Software, wie sie für größere Rechner angeboten werden, nicht zu verstecken brauchen.

● Auch die Produktlinie des Amiga, die vom Amiga 500 bis zum High-Tech-Amiga 3000 reicht, unterstreicht die Vielfalt dieser Maschine. Für den

privaten Anwender ermöglicht gerade der Amiga 500 einen preiswerten Einstieg in die Computerwelt. Eines haben aber alle gemeinsam: die Workbench und das CLI.

● Daher finden Sie in diesem Sonderheft alles Wissenswerte und Interessante zu diesen beiden Themen. Steigen Sie ein zu einer Rundreise in Ihrem Computer.

● Mit dem Heimcomputer lassen sich eine Menge nützlicher Dinge anstellen. Warum denn immer nur spielen?



Ihr

Andreas Greil
(Leitender Redakteur)

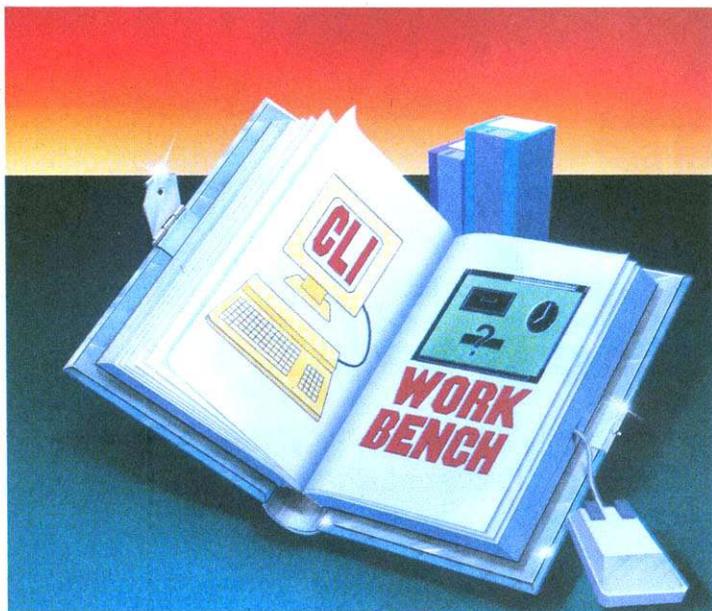


Bild 1. Damit das CLI kein Buch mit sieben Siegeln ist, lesen Sie den Kurs ab

SEITE 34

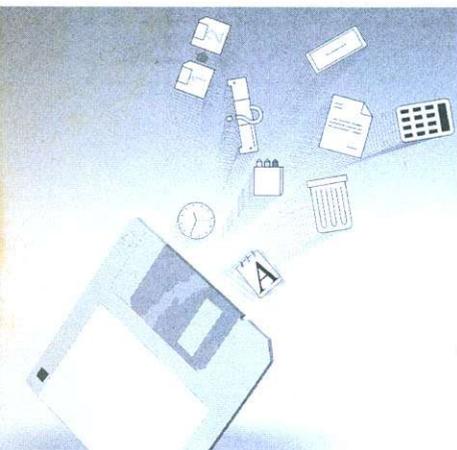


Bild 2. Unzählige Icons stellen eine Vielzahl von Programmen auf der Workbench dar. Was es mit diesen Utilities und Tools auf sich hat, erfahren Sie ab

SEITE 66

Bild 3. Die Handelsimulation »Reeder« führt Sie in die Welt der Kaufleute und Seemänner

SEITE 90



Workshops

6 Ein Buch ohne Siegel

Für alle, die Ihr Handbuch schon längst in die Ecke gefeuert haben. Ein Workshop zur Workbench mit Tips aus der Praxis für die Praxis.

34 CLI - der Hausherr im Amiga

Nicht alles was der Amiga kann, läßt sich über die Workbench erledigen. Den vollen Zugriff erhalten Sie nur über das CLI. In diesem umfangreichen Kurs erfahren Sie alles Wissenswerte über diese Benutzeroberfläche.

Grundlagen

66 Entdeckungsfahrt durch die Workbench

Die Einsatzgebiete der Workbench sind vielfältig. In diesem Beitrag zeigen wir Ihnen Schritt für Schritt den Inhalt der mitgelieferten »Workbench 1.3D«- und »Extras 1.3D«-Diskette. Gleichzeitig gehen wir darauf ein, welche Files für den eigenen persönlichen Gebrauch wichtig sind.

77 Ein Bummel durch Ihren Computer

Aus welchen Teilen besteht ein Computer? Wie arbeiten diese zusammen? Welche Peripherie ist nötig? Wenn Sie diese Seiten gelesen haben, dürften kaum noch Fragen offen bleiben.

82 Hoch- und andere Stapeleien

Batch-Processing ist die einfachste Art, mit dem Amiga umfangreiche Befehlsfolgen abzuarbeiten. Wie das geht, verdeutlicht Ihnen dieser Artikel.

Listings

87 Checksummer

Das Abtippen von Listings ist eine fehlerträchtige Angelegenheit. Schonen Sie sich Ihre Nerven und benutzen Sie unseren Checksummer.

90 Herr der Meere

Es ist ein harter Weg zum erfolgreichen Reeder. Die Umstände machen es nicht leicht. Aber vielleicht ernennt man Sie sogar zum Bürgermeister. Lassen Sie sich von der Handelssimulation »Reeder« faszinieren.

103 Das Supermenü

Sie kennen sicherlich die ansprechenden Menüleisten in professionellen Programmen. Mit »SuperMenusII« können Sie jetzt solche Menüs in Ihr eigenes Basic-Programm einbauen. Dazu stehen noch einige zusätzliche Grafik-Features zur Verfügung.

Sonstiges

3 Editorial

106 Impressum

NEU!

Software-Paket mit
ausführlichen
Anleitungen im Heft

AMIGA POWER DISC

Markt & Technik
DM 19.80

ISSN 150-
08 19 80 / Lfd. 17000
Nr. 20 - Okt. 89

**Das Heft mit
der Diskette ist da!**

Das Super-Softwarepaket für den Amiga-User.

"Earl-Datenbank" - Dateien jetzt mit Grafik. Diese universelle Datenbank verarbeitet Text, Daten und Grafik beinahe wie von selbst.

"Project-T" - Deutschland im Klötzchen-Fieber. Suchen Sie ein Geschicklichkeitsspiel, das Sie voll und ganz fordert? Dann ist unser "Project-T" genau das Richtige für Sie.

"Sternenkampf" - Weltraumspaß für zwei Akteure. Zwei Piloten fliegen durch das Weltall und versuchen, sich gegenseitig abzuschießen. Diesem Action-Weltraumspiel kann niemand widerstehen.

"Troff" - hitziger Hindernislauf. Das Spiel, nach dem "Tron"-Prinzip aus dem Walt Disney-Film, steht für Action, Spannung und höchste Anforderungen an die Geschicklichkeit.

**Amiga Power-Disk gibt es
ab 07.11.90 an Ihrem Kiosk**

Die Workbench-Oberfläche

Ein Buch ohne Siegel

Sie haben die Handbücher zu Ihrem Amiga längst in die Ecke gefeuert? Sie suchen verzweifelt nach Literatur, die Ihnen die Workbench näherbringt? Mit Tips aus der Praxis für die Praxis? Können Sie haben.

von Thomas Hauser

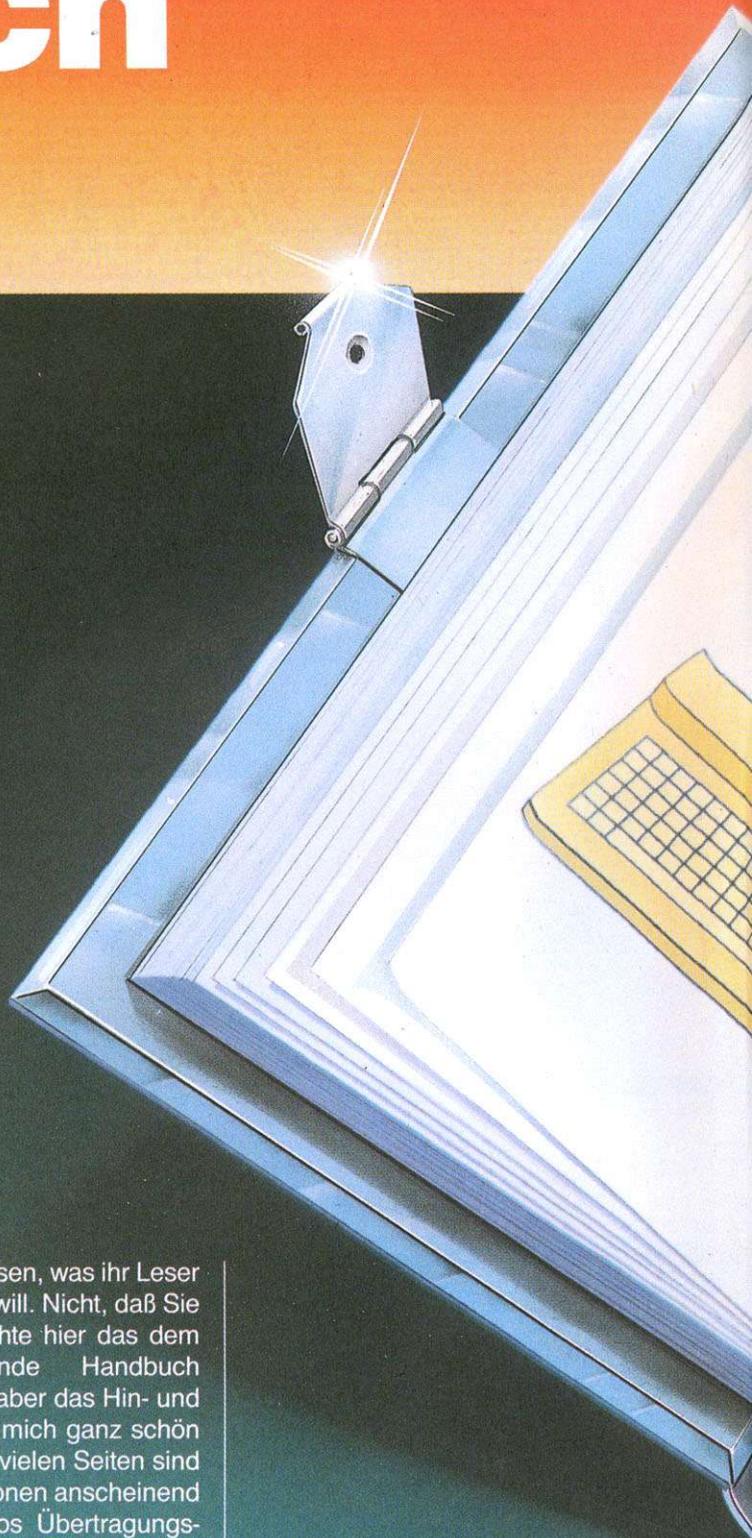
Handbücher im allgemeinen und Computerhandbücher im besonderen sind immer Glückssache. Entweder setzt spätestens nach Seite 274 das große Gähnen ein, oder man landet nach viermal Umblättern im Index. Sicher – nichts ist schwerer, als es jedem recht zu machen. Wo der eine mehr erklärt haben möchte, sieht der andere nur Platzverschwendung. Manche dieser Handbücher lesen sich, als ob einer, der wußte, wovon er sprach, versucht hätte, sein Wissen einem anderen mitzuteilen, der dann über etwas schrieb, von dem er eigentlich keine Ahnung hatte: Kommunikationsinfarkt!

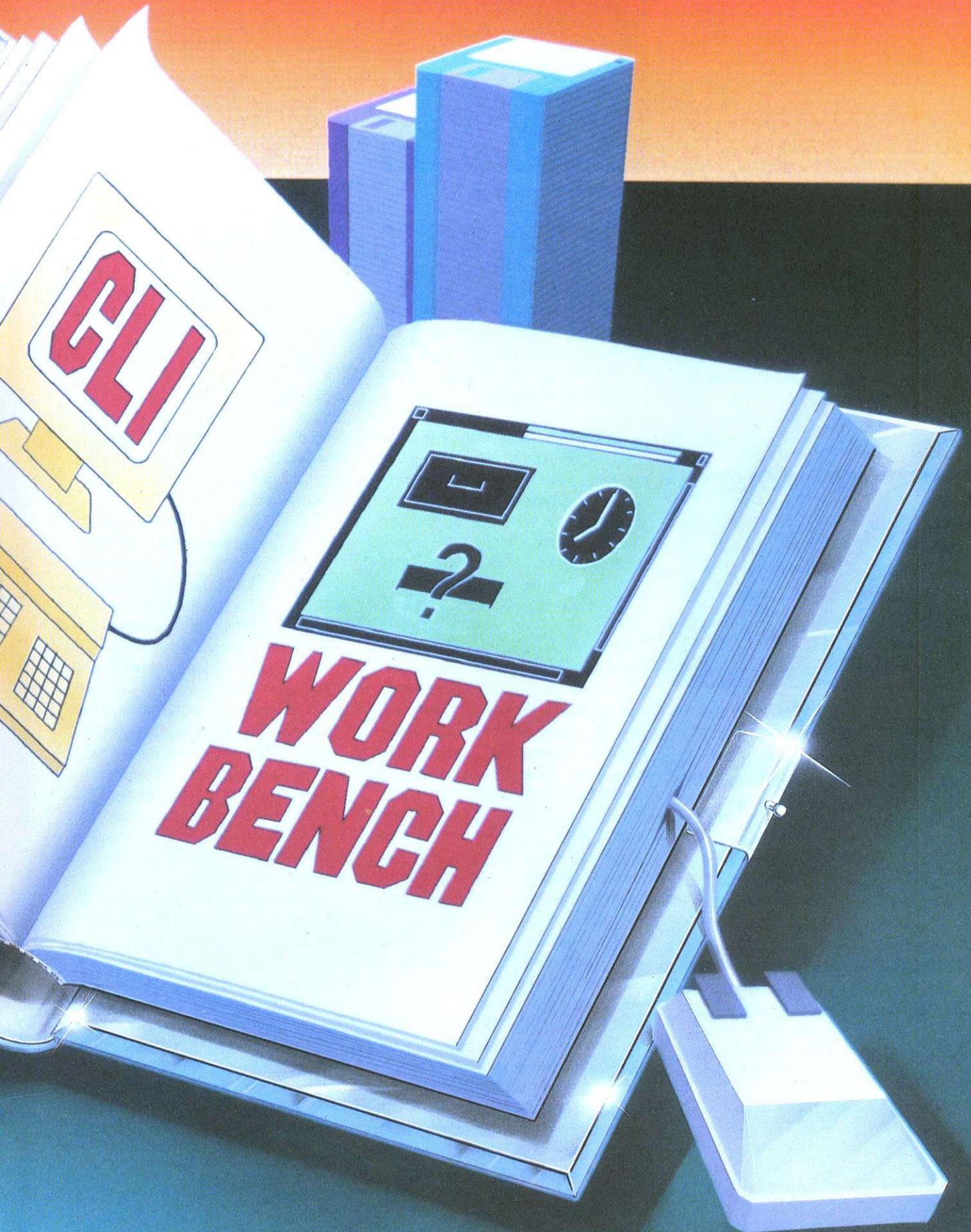
Meist endet so ein Handbuch dann in der Rundablage (= Papierkorb), und der geplagte Anwender holt sich seine Informationen von anderen, bereits praxisgeschädigten Gleichgesinnten. Nicht umsonst erfreuen sich Bücher mit sinnigen Titeln wie »Erste Schritte mit ...«, »Der Einstieg in ...« und »... ohne Ballast« ungebrochener Beliebtheit. Werden sie doch (meist) von Leuten ge-

schrieben, die wissen, was ihr Leser leidet und was er will. Nicht, daß Sie glauben, ich möchte hier das dem Amiga beiliegende Handbuch schlechtmachen, aber das Hin- und Hergeblättere hat mich ganz schön genervt. Trotz der vielen Seiten sind manche Informationen anscheinend sang- und klanglos Übertragungsverlusten zum Opfer gefallen. Oder haben Sie brauchbare Informationen zu den »TOOL TYPES«-Einträgen gefunden? Noch dazu entschloß sich Commodore bei Erscheinen von Kickstart und Workbench 1.3, dem Handbuch der 1.2-Version nur einen Zusatz für 1.3-Spezifikationen beizulegen. Anstatt eine revidierte Neuauflage des Handbuchs auszugeben.

Jetzt blättert der Amiga-Benutzer schon in zwei Handbüchern.

Ich nehme an, Sie haben die Verkabelung Ihres Amigas und deren Peripherie zumindest soweit erfolgreich abgeschlossen, daß auf Ihrem Monitor die berühmte **Workbench-Hand** zu sehen ist. Für den Fall, daß bei Ihnen jetzt anstelle von »Work-





bench« sowas ähnliches wie »Kickstart« zu lesen ist, sitzen Sie vor dem Urvater, dem Amiga 1000.

Legen Sie, wie vom Bildschirm gefordert, die Kickstart-Diskette ein und warten Sie, bis auch auf Ihrem Bildschirm die Aufforderung zum Einlegen der Workbench-Diskette erscheint. Von jetzt an gilt das Gesagte für alle Amiga-Besitzer.

Die Workbench-Hand auf dem Bildschirm gibt Ihnen zu Beginn eine wichtige Information: die Betriebssystemversion. Je höher diese Zahl, desto moderner und leistungsfähiger das Betriebssystem. Diese Zahl hat aber nichts mit der Versionsnummer der Software zu tun.

Wahrscheinlich steht dort »V1.2«. Die Mehrzahl der in Gebrauch stehenden Amigas wurde mit dieser Betriebssystemversion ausgeliefert. Deren Nachfolger wiederum, die Version 1.3, sei Ihnen gewünscht –

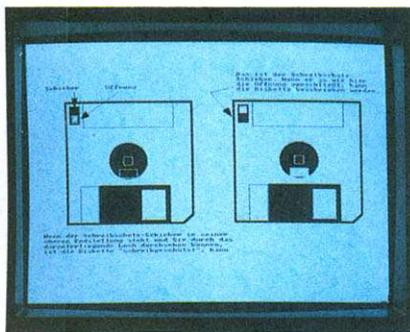


Bild 1. Bei geschlossenen Fenstern ist die Diskette beschreibbar

mit ihr sind Sie auf dem aktuellen Stand der Technik. Was dieses »Betriebssystem« eigentlich ist? In kurzen Worten skizziert, ist das Betriebssystem des Amigas in zwei Hauptgruppen aufgeteilt. Ein Teil wird als »Kickstart« bezeichnet und ist nichts anderes als ein Programm, das in Ihrem Amiga eingebaut ist (wie, soll uns hier nicht weiter interessieren). Dieses Programm versetzt diese herrliche Ansammlung von High-Tech erst in die Lage, auf das Einlegen einer **Boot-Diskette** (eine Diskette, mit der Sie den Amiga star-

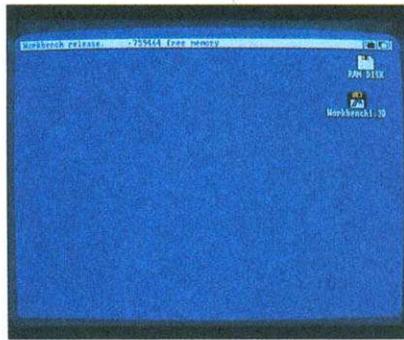


Bild 2. Die Workbench

ten können) zu reagieren und das gesamte Computersystem »hochzufahren«. Die zweite Hauptgruppe ist »Amiga-DOS« (»DOS« – »Disk-Operating-System« – Diskettenbetriebssystem), neuerdings auch – und, wie ich meine, treffender – als »Amiga-OS« (»OS« für »Operating System) bezeichnet. Amiga-DOS ist aber kein eigenes Programm mehr, sondern eine Ansammlung von Programmen, Tabellen usw., die auf jener Diskette gespeichert sein müssen, mit der Sie das ganze Computersystem hochfahren – eben jene besagte »Boot«-Diskette. Nein, die Boot-Diskette hat nichts mit einer Ruderpartie zu tun. »Boot« kommt aus dem Englischen und wird »Buuut« ausgesprochen.

Bei diesem Booten arbeitet der Computer eine »Startup-Sequence« ab, mit der alle vom System benötigten Ressourcen bereitgestellt werden. Was diese »Startup-Sequence« bewirkt, können Sie ausführlich in dem Artikel über die Batch-Dateien auf Seite 82 nachlesen. Die Unterschiede zwischen den Kickstart-Versionen 1.2 und 1.3 sind nicht allzu groß und für den reinen Workbench-Betrieb unwesentlich (mehr darüber lesen Sie im Artikel über das »CLI« auf Seite 34 in diesem Heft).

Die Auftrennung zwischen »Kickstart« und »Amiga-DOS« bringt es auch mit sich, daß sowohl vom einen wie auch vom anderen Versionen existieren, die aber nicht zwangsläufig miteinander verkoppelt sind. »Kickstart 1.2« – egal, ob im Computer eingebaut oder von externer »Kickstart«-Diskette geladen – verlangt zwar nach der »Workbench 1.2« (womit eigentlich die Workbench-Diskette gemeint ist, auf der das Betriebssystem »Amiga-DOS 1.2« gespeichert ist), akzeptiert aber auch

die Workbench V1.3. Umgekehrt ist es auch zulässig, mit »Kickstart 1.3« die »Workbench 1.2« zu verwenden (von einigen kleinen Einschränkungen einmal abgesehen).

Unser Amiga wartet noch immer auf die Workbench-Diskette – also rein mit der Floppy ins interne Laufwerk. Die Laufwerk-Betriebsanzeige leuchtet auf (beim Amiga 500 ist das die grüne Leuchtdiode, beim 2000er die rote), und der Computer erwacht zum Leben. Am besten, Sie rühren jetzt nichts an, bis die Laufwerks-LED wieder erloschen ist. Ihr Amiga weiß schon, was er tut. Bei dieser Gelegenheit fällt mir schon wieder so eine Richtlinie ein, die permanent propagiert wird, aber nur zur Hälfte stimmt. Vom Handbuch bis zu allen mir bekannten Fachzeitschriften, wird davor gewarnt, die Diskette aus dem Laufwerk zu entfernen, während die Laufwerksanzeige leuchtet, da sonst die Diskette zerstört werden kann. Nirgends steht, daß dies nur für Schreibzugriffe gilt.

Wird von der Diskette nur gelesen (z.B. gerade jetzt, wo Sie Ihre Workbench-Disk mit gesetztem Schreibschutz im Laufwerk haben), kann gar nichts passieren, sorgsame Handhabung einmal vorausgesetzt. Dafür steht aber auch nirgends, daß nach einem Schreibzugriff die Laufwerks-LED zwar erlischt, daß aber kurz danach noch einmal ein Schreibzugriff auf die Diskette erfolgt. Warten Sie daher nach dem Erlöschen der Laufwerks-LED noch ein paar Sekunden, bevor Sie die Diskette aus dem Laufwerk nehmen. Falls Sie noch nicht wissen, was dieser »Schreibschutz« ist, sehen Sie sich bitte das Bild 1 an.

Auf Ihrem Monitor ist jetzt also die Workbench-Hand verschwunden und hat einem weißen Rahmen auf blauem Untergrund Platz gemacht. Innerhalb dieses Rahmens lesen Sie verschiedene Meldungen. Auf diese wird im CLI-Artikel näher eingegangen. Mit dem echten Workbench-Betrieb hängen sie nur mittelbar zusammen. Für uns wird es wieder interessant, wenn der Bildschirm so aussieht wie in Bild 2. Das ist »die

AMIGA

Markt&Technik

SONDERHEFTE

1 Rund um den Amiga

Ausführliche Kurse, Bauanleitung, Tips & Tricks



2 Listings der Spitzenklasse

Tolle Anwendungen, Tips, Tricks und Tools



3 C und Assembler

Zwei Super-Kurse und viele Tips und Tricks



4 Basic und Spiele

10 Spitzenspiele zum Abtippen in AMIGA-Basic!



5 Public Domain

Software zum Nulltarif: Das Beste aus aller Welt



6 Anwendungen

Workshops, Top-Programme fast umsonst



7 Spiel-Programme

Knowhow, Projekte, Spiele



8 Starthilfe

Know-how für den Einstieg

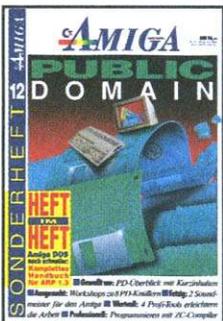
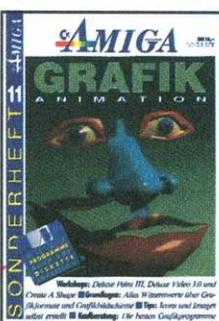
9 Die besten Listings des Jahres

20 Spitzenprogramme zum Sparpreis



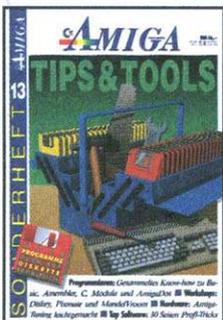
11 Grafik-Animation

Workshops, Tips und Kaufberatung



12 Public Domain

Workshops, Grundlagen und Programmieren



13 Tips & Tools

Programmieren, Hardware und Top-Software

Die AMIGA Sonderhefte bieten umfassende Informationen zu speziellen Themen rund um den AMIGA. Diese Ausgaben erhalten Sie umgehend von Computer Service Ernst Jost, Markt&Technik Leserservice, Postfach 140 220, 8000 München 5. Am besten füllen Sie gleich den Coupon aus und schicken ihn los.

AMIGA BESTELLCOUPON

Ich bestelle: _____ Ausgaben AMIGA SH Nr. _____
 _____ " " " Nr. _____

Insgesamt: _____ Ausgaben für 16,-DM pro Exemplar zzgl. Versandkosten

Name, Vorname _____
 Straße, Hausnummer _____
 PLZ, Wohnort _____
 Die Bezahlung erfolgt nach Erhalt der Rechnung. Schicken Sie bitte den ausgefüllten Coupon an: Computer Service Ernst Jost, Markt&Technik Leserservice, Postfach 140220, 8000 München 5

Workbench« schlechthin, jene Arbeitsoberfläche, die Kommunikation zwischen Mensch und Computer so herrlich einfach macht.

In dem weiß unterlegten Streifen am oberen Bildschirmrand wird die Anzahl der freien Speicherplätze angezeigt. Je nach Größe des Arbeitsspeichers Ihres Amigas steht dort eine mehr oder weniger große Zahl. Erwarteten Sie aber bitte nicht, dort jene Zahl zu finden, von der der Verkäufer (oder auch die Verkäuferin) sprach, als Sie Ihren Amiga erwarben! Durch das Betriebssystem wird bereits eine erhebliche Anzahl Speicherplätze belegt – da kann der Verkäufer wirklich nichts dafür. Wenn Sie den Rat, nichts anzurühren, beherzigt haben, werden Sie in der linken oberen Ecke des Bildschirms einen kleinen roten Pfeil entdecken. Haben Sie an der Maus herumgefummelt, während der Amiga die »Workbench« aktivierte? Dann suchen Sie mal schön nach dem Pfeilchen. Jetzt dürfen Sie das Mäuschen verschieben.

Wenn Sie die Maus bewegen, bewegt sich der Pfeil mit. Maus nach rechts, Pfeil nach rechts, Maus nach hinten (von Ihnen weg), Pfeil nach oben. Geht doch super, oder? Dieser Pfeil ist der **Mauszeiger** oder auch Maus-Pointer, und er ist die wichtigste Verbindung zwischen Ihnen und dem Computer. Ohne diesen Zeiger ist Ihre Workbench nicht mehr als ein nettes Bildchen auf dem Monitor. Führen Sie den Pfeil jetzt auf das kleine Bildchen am rechten Bildschirmrand, das wie eine Diskette aussieht und mit »Workbench1.3D« untern ist. So ein Bildchen nennt man auch **Piktogramm**. Im Amiga-Sprachgebrauch hat sich für diese Symbolbildchen die Bezeichnung **Icon** (»Eik'n« gesprochen) durchgesetzt. Icons sind Symbole und ste-

hen für Disketten, Programme, Verzeichnisse usw. Die »Workbench« ist eine symbolorientierte Schnittstelle zwischen Computer und Benutzer. Den Unterschied zu einer kommandoorientierten Schnittstelle werden Sie spätestens dann bemerken, wenn Sie sich mit dem »CLI« beschäftigen (machen wir später und an anderer Stelle in diesem Heft).

Zurück zu unserem Maus-Zeiger. Die Spitze des Pfeils muß ungefähr in der Mitte des diskettenähnlichen Symbols stehen, das die Workbench-Diskette repräsentiert. Nicht der ganze Zeiger ist für die Steuerung des Computers relevant, sondern nur ein kleiner Punkt in der Pfeilspitze. Drücken Sie einmal kurz auf die linke der beiden Tasten, die in die Maus eingebaut sind. Wenn Sie genau aufgepaßt haben, wird Ihnen nicht entgangen sein, daß sich das kleine Diskettenbildchen, das »Icon«, etwas verändert hat. Diese Veränderung im Aussehen ist die Bestätigung des Computers dafür, daß er Ihren Wunsch, sich mit diesem Icon und der davon vertretenen Workbench-Diskette zu befassen, verstanden hat – Sie haben das Symbol »selektiert«. Bloß weiß der arme Siliziumhaufen damit noch nicht, was Sie eigentlich wollen.

Der erste Klick ein Treffer

Irgendwas hat es mit der Diskette »Workbench1.3D« zu tun, das ist ihm schon klar. Bloß was? Machen wir doch gleich eine Kopie der Workbench-Diskette. Grundsätzlich sollten Sie nie mit Originaldisketten arbeiten, sondern immer nur mit Arbeitskopien. Amiga-DOS gehört nicht unbedingt zu den betriebssichersten Operating-Systemen. Es ist zwar nicht ganz so schlimm, wie es sich vielleicht anhört, aber in Ihrem eigenen Interesse sollten Sie diesen Rat beherzigen. Die erzwungene Ausnahme von dieser Regel stellen nur kopiergeschützte Programme und Disketten dar. Bei diesen bleibt Ihnen gar nichts anderes übrig, als mit dem Original zu arbeiten.

Also kopieren wir die Workbench-Diskette. Ich bin davon überzeugt, daß jener Händler, der Ihnen den Amiga, vor dem Sie sitzen, verkauft

hat, Sie nicht aus dem Geschäft ließ, ohne Ihnen auch mindestens eine Packung Leerdisketten zu verkaufen. Für Ihren Amiga sind nur solche Disketten geeignet, die die Aufschrift »DS-DD« tragen. »DD« bedeutet »Double Density«, übersetzt also »doppelte Dichte«, und ist ein Kennzeichen dafür, wie viele Spuren (»Tracks«) aufgezeichnet werden können. »DS« bedeutet »Double Sided« – doppelseitig verwendbar. Dies heißt nicht, daß Sie die Diskette umdrehen können wie eine Schallplatte! Es bedeutet, daß beide Seiten der in der Diskette steckenden Magnetscheibe beschrieben werden können, was das Laufwerk Ihres Amigas aber ganz von selbst tut.

Die richtige Diskette

Eine andere Kennzeichnung für Amiga-taugliche Disketten ist der Aufdruck »Double Sided – 135 tpi«. Alles andere, also z. B. »SS-DD«, sollten Sie nicht einmal in die Nähe Ihres Amigas lassen. Es besteht durchaus die Möglichkeit, daß solche »falschen« Disketten anfangs funktionieren, nach einiger Zeit aber Teile der gespeicherten Daten verlieren. Auch zuviel des Guten kann fatale Folgen haben. Die neuen »HD«-Disketten (»HD« – »High Density« für »hohe Schreibdichte«), die besonders im MS-DOS-Bereich zunehmend Beliebtheit erlangen, sind nicht für den Standard-Amiga geeignet. Es werden zwar schon Zusatzlaufwerke für unsere Freundin angeboten, die mit solchen Disketten arbeiten können, aber das ist wohl doch eher als Ausnahme anzusehen.

Nehmen Sie jetzt bitte eine dieser »leeren« Disketten, entfernen Sie die Workbench-Diskette aus dem internen Laufwerk und legen Sie statt dessen die Leer-Disk ein. Die Laufwerk-LED leuchtet ca. 3 Sekunden auf, und Sie hören, daß das Laufwerk arbeitet. In dieser Zeit prüft der Amiga, was Sie ihm da eigentlich unterjubeln wollen. Sie wissen, daß die Diskette leer ist, der Amiga weiß das nicht. Er erkennt lediglich, daß er mit dieser Diskette nichts, aber auch

schon gar nichts, anfangen kann. Ignorieren kann und darf er diese Diskette aber nicht, und deshalb wird sie als Icon mit der Untertitelung »DF0:BAD« auf der Workbench angezeigt. Mit »BAD« ist das englische »bad« für »schlecht«, »unbrauchbar« gemeint. Als »DF0:« wird das interne, das »Boot«-Laufwerk bezeichnet. Im Gegensatz zu MS-DOS, das seine Laufwerke mit Buchstaben bezeichnet, numeriert Amiga-DOS die Laufwerke von 0 bis 3. Aber wir wollten ja die Workbench-Disk kopieren.

Steht Ihr Mauszeiger noch über dem Symbol für die Workbench-Diskette? Okay, dann drücken Sie jetzt die linke Maustaste und halten sie in dieser gedrückten Stellung. Wenn Sie jetzt die Maus bewegen, löst sich eine Kopie des Diskettensymbols vom Vorbild und folgt dem Mauszeiger (die linke Taste noch immer drücken!). Verschieben Sie diese Kopie des Workbench-Icons so, daß es das Icon der leeren Diskette überdeckt. Jetzt können Sie die linke Maustaste loslassen (daß dabei der »Schatten« der Diskette verschwindet, soll Sie nicht irritieren). Aus der Tatsache, daß Sie das Icon einer »guten« Diskette über das Icon einer anderen Diskette geschoben haben, schließt der Amiga messerscharf, daß Sie den Inhalt der »obenliegen-

Aus eins mach zwei

den« Diskette auf die »darunterliegende« kopieren wollen. Um diesem Ihrem Wunsch nachkommen zu können, braucht der Amiga das Diskettenkopierprogramm **DiskCopy**, das sich auf der Workbench-Diskette befindet. Die haben Sie aber durch die neue Diskette ersetzt. Also verlangt der Amiga nach »seiner« Workbench-Disk. Der Amiga hat eine viel feinere Möglichkeit, Ihnen seine Bedürfnisse mitzuteilen – er öffnet einen **System-Requester** (Bild 3). In solchen »Nachfragekästchen«, die der Amiga auf dem Bildschirm bei Bedarf einblendet, steht immer eine Nachricht für Sie. Den Erhalt einer



Bild 3. Ein System-Requester

solchen Nachricht müssen Sie quittieren, und zwar durch das Anklicken eines der Gadgets. Was ein **Gadget** ist, wollen Sie wissen? Gadgets (singular »Gadget«) sind Eingabefelder, dienen also ebenso wie die Icons der Kommunikation Mensch-Maschine. Der Amiga kennt verschiedene Arten solcher Gadgets. Solche, die nur darauf warten, angeklickt zu werden, und andere, die wir später kennenlernen werden (»Proportional-Gadgets«, »String-Gadgets« usw.). Der Begriff »Gadget« zeugt übrigens vom feinen Humor der Amiga-Entwickler bedeutet er doch frei übersetzt soviel wie »Dingsda«. In unserem Requester sehen Sie zwei solcher Gadgets, denen Sie im weiteren Verlauf dieses Beitrags noch öfters begegnen werden: »Retry« und »Cancel«. »Retry« bedeutet »versuch's nochmal«, »Cancel« soviel wie »vergiß es«. Der Amiga erwartet also von Ihnen, daß Sie die Diskette mit dem Namen »Workbench1.3D« wieder in irgendein Laufwerk einlegen. Als intelligentes System zwingt Sie der Amiga nämlich nicht, die angeforderte Diskette wieder in jenes Laufwerk einzulegen, aus dem Sie sie entnommen haben. Wenn Sie nur über ein Laufwerk verfügen, ist das zwar ein netter Zug des Amigas, aber sonst hat es keinen größeren Nutzen für Sie. Den werden Sie erst kennen und schätzen lernen, wenn Sie Ihr Computersystem um ein Zusatzlaufwerk erweitert haben. Kommen wir dem Wunsch des Amigas nach der Workbench-Disk probeweise einmal nicht nach. Positionieren Sie den Mauszeiger über dem »Retry«-Gadget, und klicken Sie es mit einem kurzen Druck auf die linke Maustaste an. Der Amiga »probiert nochmal«, ob er die angeforderte Diskette endlich findet, was Sie am kurzzeitigen Ver-

schwinden des System-Requesters erkennen. Der ist aber gleich wieder da, denn überlisten läßt sich ein so cleveres System nicht so leicht. Zum Vergleich klicken Sie jetzt »Cancel« an – der Requester verschwindet, und das Icon der Workbench-Diskette sieht wieder so aus wie ganz zu Beginn unserer Kopier-Session.

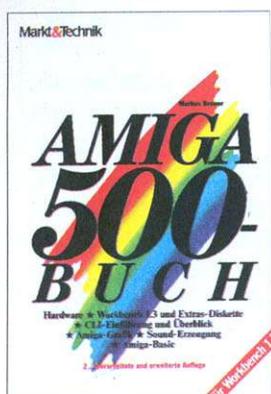
Aber wir wollten ja eine Kopie anlegen. Also bitte nochmals Mauszeiger über dem Icon der Workbench-Disk plazieren, linke Maustaste drücken und gedrückt halten und den »Schatten« des Workbench-Disketten-Icons über jenem der neuen Diskette plazieren. Was'n nu los? Nicht genug, daß der Bildschirm kurz »aufgeblitzt« hat (das ist ein optisches Signal des Betriebssystems, daß etwas schiefgelaufen ist), in der weiß unterlegten Zeile auf dem oberen Bildschirmrand steht jetzt auch nicht mehr »Workbench release. xxxxxx free memory«, sondern »Disks are of incompatible type and cannot be copied«, frei übersetzt: Die beiden Disketten haben sowenig gemeinsam, daß ein Kopieren nicht möglich ist. Meiner Meinung nach handelt es sich dabei um einen Bug (einen Fehler) im Betriebssystem, denn diese Aussage ist zwar grundsätzlich richtig, bei einer unformatierten Diskette aber auch unumgänglich.

Wie bitte? Ach so, ja – »unformatiert«. Eine soeben gekaufte Diskette ist nicht nur leer, sie ist sogar »total leer«. Damit der Amiga Daten auf einer Diskette ablegen kann, muß diese durch den **Formatiervorgang** dafür vorbereitet werden. Dabei wird durch ein Programm die Magnetschicht in Bereiche eingeteilt. Diese Bereiche nennt man »Spuren« (»Tracks«), »Sektoren« und »Zylinder«. Die »Spuren« werden in Form konzentrischer Ringe auf beiden Seiten der Magnetscheibe aufgetragen. Dieses »Auftragen« ist nichts anderes als ein definiertes Magnetisieren der kleinen Partikel, aus denen die Magnetschicht besteht.

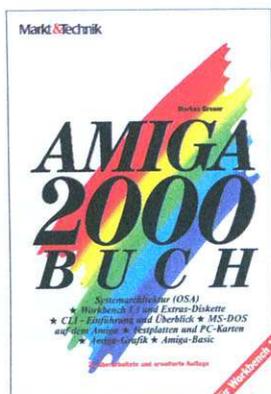
Zwischen den Ringen wird immer ein kleiner Abstand freigelassen, da-

Spitzenbücher zum Amiga

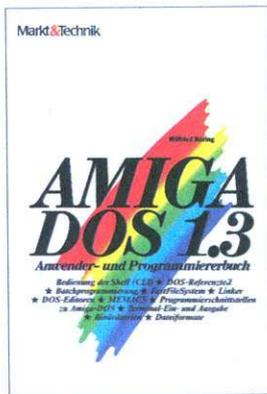
Hol Dir, was Dir



M. Breuer
Amiga-500-Buch
 Eines der erfolgreichsten Commodore-Bücher in aktueller Überarbeitung. Alles über Hardware, Software, Zubehör und eine ausführliche Beschreibung der Workbench 1.3. Durch viele Abbildungen und Beispiele werden Sie mit der Bedienung des Amiga 500 schnell vertraut. In einem übersichtlichen Nachschlageteil werden die Shell-Befehle erläutert. 2., überarb. Auflage 1989, 541 Seiten
 ISBN 3-89090-300-2
DM 49,-



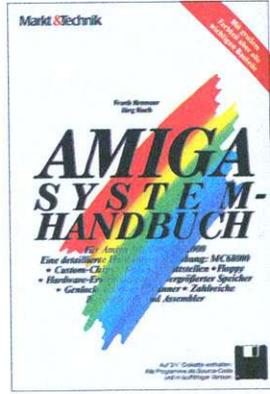
■ **NEU**
 M. Breuer
Amiga-2000-Buch
 Vollständig überarbeitete und aktualisierte Neuauflage dieser fundierten Einführung in die Bedienung des Amiga 2000 und 2500. Mit leichtverständlicher Beschreibung der Hardware und der verschiedenen Ausbaumöglichkeiten. Dazu ausführliche Darstellung von Workbench 1.3, CLI und der neuen Shell. 2., überarb. Auflage 1990, 672 Seiten
 ISBN 3-89090-287-1
DM 59,-



W. Häring
Amiga-DOS 1.3
 Im ersten Teil werden dem Anwender die Shell- und die Amiga-DOS-Befehle, Startup-Sequenzen und die Arbeit mit dem Editor erklärt. In einem speziellen Anhang finden Sie viele nützliche Tipps zur Konfiguration Ihres Druckers. Der zweite Teil wendet sich an künftige Programmierer und macht sie mit dem Betriebssystem und der Hardware vertraut. 1989, 392 Seiten,
 ISBN 3-89090-802-0
DM 69,-



■ **NEU**
 P. Lukowitz/O. Pfeiffer
Amiga Datenstruktur-Lexikon
 Alle Systemdatenstrukturen werden unter Angabe des Offsets aufgelistet und ausführlich beschrieben. Zu den vier wichtigsten Programmiersprachen – C und MODULA-2, Basic und Assembler – finden Sie detaillierte Benutzerhinweise und Anwendungsbeispiele. Referenzlisten der Systemroutinen machen die Vorgänge endgültig transparent. 1990, ca. 250 Seiten,
 ISBN 3-89090-250-2
DM 69,-



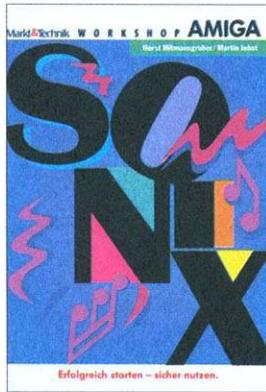
J. Kremser/F. Koch
Amiga-Systemhandbuch
 Aufbau, Vergleich und Erläuterung des Amiga 500, 1000 und 2000, die Custom-Chips Denis, Agnus und Paula, Ansprechen der Cooper-, Blitter-, Playfield-, Audio- und CIA-Hardware in C und Assembler, Aufbau externer Hardware wie Floppy und SideCar, PAL-Modulator, RAM-Erweiterung, Genlock-Interface, Amiga 500 in der MS-DOS-Welt. 1988, 421 Seiten, inkl. Diskette
 ISBN 3-89090-550-1
DM 79,-

h besser macht!



NEU

H. Mitmansgruber/M. Jobst
**M&T-Workshop
 Amiga Sonix**
 Wenn Sie noch keine Erfahrung in Sachen Musik und Amiga besitzen, dann erfüllt dieses Buch alle Voraussetzungen für einen erfolgreichen und sicheren Start. Es bietet: Tutorium mit einer Einführung in die Musiklehre, Komponieren (extra Notenheft liegt bei), ausführliche Referenz. Für Sonix-2.0-Anwender. 1990, 272 Seiten
 ISBN 3-89090-897-7
DM 39,-



NEU

F. Belzner
**Kreative Grafik
 auf dem Amiga**
 Anhand zahlreicher Programm- und Bildbeispiele erhalten Sie eine Fülle von Tips und Tricks, mit denen Sie die Möglichkeiten der Programmiersprache GFA-Basic 3.0 voll ausschöpfen können. Dazu zwei große Grafikprogramme mit ganz neuartigen Optionen, die Sie anwenden können, ohne GFA-Basic 3.0 zu besitzen. 1990, 360 Seiten, inkl. Diskette
 ISBN 3-89090-227-8
DM 79,-



W. Häring
**Schnellübersicht
 Amiga-DOS 1.3**

Alles schnell im Griff: Grundlagen und Peripherie, Massenspeicher, Verzeichnisse und Laufwerke, Dateien und ihre Sicherung, Ein- und Ausgabe – sowie Antworten auf alle Fragen, die bei der täglichen Arbeit auftreten. 1989, 292 Seiten,
 ISBN 3-89090-730-X
DM 39,-

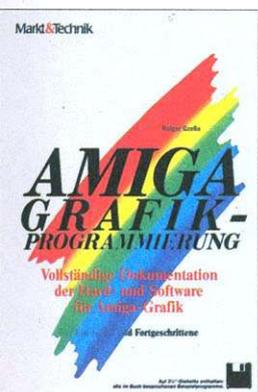


P. Wollschläger
**Schnellübersicht
 Amiga-Basic**

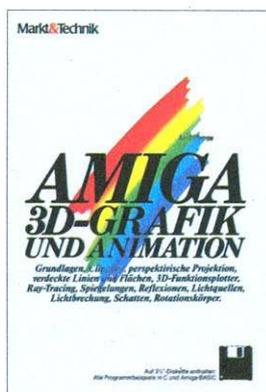
Die Beschreibungen sind problemorientiert aufgebaut, und die Informationen werden so vermittelt, wie sie bei der täglichen Arbeit auftreten. Eine ausklappbare Themenübersicht und Querverweise erleichtern die Arbeit zusätzlich. 1989, 336 Seiten,
 ISBN 3-89090-736-9
DM 39,-



* unverbindliche Preisempfehlung



NEU
 H. Gzella
**Amiga-
 Grafikprogrammierung**
 Der Einsteiger lernt zunächst den Umgang mit den einfachen Grafikbefehlen und ist bald in der Lage, farbenprächtige Grafiken darzustellen. Der Fortgeschrittene findet Erläuterungen zu Themen wie Sprites und Animationen, Copper, Blitter, Fraktal- und Vektorgrafik, Laufschriften, Scrolling und vieles mehr. 1990, 552 Seiten, inkl. 2 Disketten
 ISBN 3-89090-339-8
DM 89,-



A. Plenge
**Amiga - 3-D-Grafik
 und Animation**
 Angefangen bei einfachsten Problemstellungen, lernen Sie, professionelle 3-D-Grafiken auf Ihrem Commodore Amiga zu planen, zu programmieren und darzustellen. Auch scheinbar komplizierte Grafiken werden einfach und nachvollziehbar erklärt. 1988, 376 Seiten, inkl. Diskette
 ISBN 3-89090-526-9
DM 69,-



NEU
 B. Lugert
**Freie Malerei
 auf dem Amiga**
 Eine Einführung in die Computermalerei und ihre Disziplinen; besonders amigaspezifische Maltechniken mit Programmen wie »DPaint« und »PhotoPaint«. Der Autor möchte zu Impressionen anregen und Computeranwender aus der kühlen Welt der Bits in den Himmel der schönen Künste entführen. Mit vielen Bildern und literarischen Texten. 1990, 94 Seiten
 ISBN 3-89090-233-2
DM 98,-

Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften und in den Fachabteilungen der Warenhäuser.



mit sich die Magnetisierungen nicht gegenseitig stören. Auf jeder Seite der Magnetscheibe werden 80 solcher Ringe magnetisiert, insgesamt also 160. Ringe, die einander gegenüberliegen (auf »Oberseite« und »Unterseite« der Magnetscheibe) bilden zusammen einen »Zylinder«. Zusätzlich wird noch jeder »Ring« in »Sektoren« unterteilt. Warum das alles so ist, würde hier aber wirklich den Rahmen sprengen – lassen wir's dabei bewenden. Halt, eine Bemerkung muß ich noch anbringen. Es hat sich eingebürgert (und ich werde mich auch daran halten), von »80 Spuren« zu sprechen, wenn eigentlich 80 Zylinder gemeint sind. Bei Bedarf wird dann noch zwischen Ober- und Unterseite unterschieden, meist durch den Zusatz »Head 0« oder »Head 1«, womit die Schreib-/Leseköpfe auf Ober- und Unterseite gemeint sind.

Zurück zu unserem »Bug«. Schmeißen Sie die leere Diskette raus und legen Sie die Workbench-Disk wieder ein, dann tickt der Amiga wieder richtig. Um endlich eine Kopie der »Workbench«-Disk machen zu können, müssen Sie nun wieder die leere Diskette einlegen. Die Anzeige »df0:BAD« kennen Sie ja schon, und wie Sie einen Kopiervorgang einleiten, wissen Sie auch

Das Formatieren

schon (»Schatten« über das Icon der leeren Diskette ziehen). Nach dem Loslassen der linken Maustaste erscheint wieder der bekannte Requester. Diesmal wollen wir aber der Anforderung des Amigas entsprechen. Tauschen Sie die leere Diskette gegen die Boot-Diskette aus, und klicken Sie das »Retry«-Gadget an. Wenn Sie schnell genug waren mit dem Anklicken von »Retry«, wird der Amiga das Anklicken des »Retry«-Gadgets zwar akzeptieren (zu bemerken am kurzzeitigen Verschwinden des Requesters), aber gleich wieder das Einlegen der Workbench-Diskette monieren, obwohl Sie diese ja schon eingelegt haben.

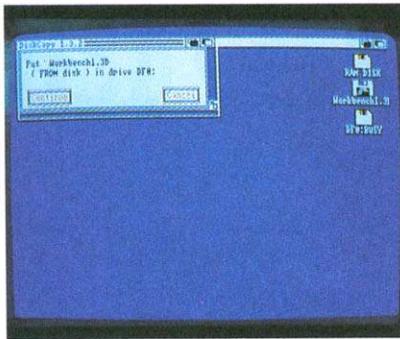


Bild 4. Ein Programm-Requester

Die Ursache für dieses scheinbare Fehlverhalten ist darin zu suchen, daß das DOS (Disk-Operating System – Diskettenbetriebssystem) eine gewisse Zeit (meist etwa 3 Sekunden) braucht, um von einer neu eingelegten Diskette Daten einzulesen und diese Diskette dann dem übrigen Betriebssystem als vorhanden zu melden. Erst nach dieser »Anmeldung« wird das Anklicken von »Retry« mit einem dauerhaften Verschwinden des Requesters belohnt – bloß keine falsche Hektik. Der Amiga holt sich nun das Diskettenkopierprogramm »DiskCopy« in den Speicher und startet es. Danach werden Sie wieder mittels Requester aufgefordert, die Quelldiskette (also jene, die kopiert werden soll) in das interne Laufwerk namens »df0:« einzulegen. Sie sehen, jetzt ist es nicht mehr gleichgültig, wo die Disketten eingelegt werden, sondern es wird ein Laufwerk explizit verlangt. Beim Aufruf von »DiskCopy« stellt das Programm zuallererst fest, welche Disketten und welche Laufwerke vom Kopiervorgang betroffen sind, und auf diese beziehen sich alle folgenden Operationen, wobei das Laufwerk (oder auch die Laufwerke, wenn Sie mehrere besitzen), das von »DiskCopy« okkupiert wurde, auf der Workbench mit »BUSY«, »beschäftigt«, als nicht mehr frei verfügbar, gekennzeichnet wird. Daß die Aufforderung zum Diskettenwechsel jetzt nicht vom Betriebssystem, sondern von einem Programm ausgeht, erkennen Sie an der Beschriftung des Requesters (Bild 4). Stand da vorher noch »System-Request«, sehen Sie jetzt in der ersten Zeile »DiskCopy 1.3.2«, womit der Zusammenhang zwischen Requester und Programm klar wird. Übrigens, sollte bei Ihnen nur »Disk Copy« zu lesen sein, sind

Sie nicht auf dem letzten Software-Stand der Workbench-Technik. Sie mögen vielleicht die Version 1.3 haben, aber dank Commodore ist »1.3« nicht gleich »1.3«. Der Unterschied ist in der »Versionsnummer« (im Original als »Release« bezeichnet) erkennbar, auf die wir im weiteren Verlauf noch zu sprechen kommen. Für unser Kopiervorhaben hat dieser kleine Unterschied eine große Bedeutung, wie Sie gleich feststellen werden. Mit dem letzten Requester wurden Sie aufgefordert, die Quelldiskette in »df0:« einzulegen. Daß diese im gegenständlichen Fall schon eingelegt ist, stört »DiskCopy« nicht die Bohne, was Sie aber nicht daran hindern soll, das »Continue«-Gadget anzuklicken, um das Programm zum »Weitermachen« zu animieren. »DiskCopy« liest jetzt in Abhängigkeit vom freien Speicherplatz entweder die Spuren 0 bis 26 (bei einem 512-KByte-Amiga) oder die Spuren 0 bis 40 (bei 1 MByte Gesamtspeicher) in das Memory ein. Den Fortschritt des Kopiervorgangs sehen Sie in einem Kontrollfenster, das »DiskCopy« öffnet. Das Beispiel in unserer Abbildung sagt aus, daß soeben die Spur 22 gelesen wird und daß noch 57 Spuren abzuarbeiten sind. Genaugenommen müßte es »Zylinder« heißen, denn bei 80 Spuren ist Schluß, doch an solche Feinheiten müssen Sie sich als Amiga-Fan gewöhnen. Ist der verfügbare Speicher aufgebraucht, erscheint wieder ein Requester.

Diesmal werden Sie aufgefordert, die Zieldiskette, also jene, auf die die Daten geschrieben werden sollen, in »df0:« einzulegen. Tun Sie, wie Ihnen vom Programm geheißen und klicken Sie danach das »Continue«-Gadget an. Ich glaube, es ist klar, daß man durch Anklicken von »Cancel« das Programm abbricht – oder? Nach dem Klick auf »Continue« schreibt »DiskCopy« die im Speicher stehenden Daten Spur für Spur auf die Zieldiskette. Und jetzt kommt der große Unterschied zwischen »Disk-Copy 1.3.2« und der älteren Version

So macht Lernen Spaß!



**Erdkunde I
Bundesrepublik und DDR**
Bestell-Nr. 38774

**Erdkunde II – Vereinigte
Staaten von Amerika**
Bestell-Nr. 38776



Mathematik I – Geometrie
Bestell-Nr. 38777

Mathematik II – Algebra
Bestell-Nr. 38778

**Mathematik III
(Bruchrechnen)**
Bestell-Nr. 38786

**Physik I – Mechanik,
Wärmelehre, Optik**
Bestell-Nr. 38779

Englisch I
Bestell-Nr. 38775

Deutsch I (Grammatik)
Bestell-Nr. 38787

Die nächste Prüfung ist gerettet: Die neuen Lernprogramme »Spielend lernen« für Ihren Amiga ersetzen stupide Paukerei durch Abenteuer, Spaß und Quiz. Trockenes Schulwissen wird spannend aufbereitet und einprägsam vermittelt. Jedes Programm hat einen Lernteil und ein Quiz oder Spiel zur Überprüfung des neu erworbenen Wissens.

Jedes Programm nur
DM 49,-*

(sFr 45,-*/öS 490,-*)

*Unverbindliche Preisempfehlung

Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser.


Markt&Technik

Zeitschriften · Bücher
Software · Schulung

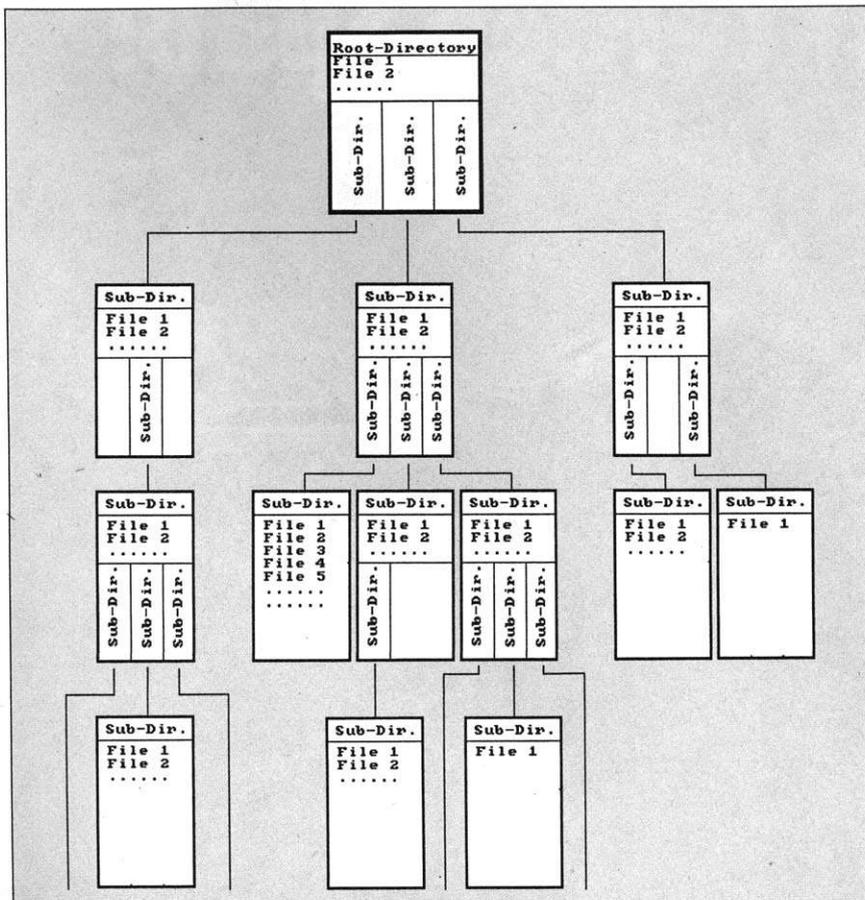


Bild 5. Die Verzeichnishierarchie einer Amiga-DOS-Diskette

»Disk Copy«. Während die alte Version die Daten nur auf die Diskette schreibt und ansonsten auf Gott und die Qualität der Diskette vertraut, überprüft »DiskCopy 1.3.2« nach dem Schreiben der Spuren 0 bis 26 (oder 0 bis 40) in einem zweiten Durchlauf, ob die Daten auch wirklich richtig auf die Diskette geschrieben wurden. Erkennlich ist das durch die Anzeige von »ver'ing Track«, also »überprüfe Track«, im Informationsfenster von »DiskCopy«. Allein schon aus diesem Grund sollten Sie trachten, die neueste Workbench-Diskette zu besitzen. Nach dem Ende dieses Überprüfungsvorgangs (oder nach dem Ende des Schreibvorgangs bei der Vorgängerversion) erscheint wieder der bekannte Requester und fordert Sie auf, die Quelldiskette in »df0:« einzulegen. So geht es weiter, bis alle 80 Spuren (Spur 0 bis 79) gelesen, geschrieben und eventuell überprüft

wurden, was je nach Speicherausbau bis zu drei Diskettenwechsel erfordert. »DiskCopy« verabschiedet sich von Ihnen mit der Meldung »DiskCopy finished«. und schließt sein Ausgabefenster. Anstelle des traurigen Icons mit der Untertitelung »df0:BAD« sehen Sie jetzt auf Ihrer Workbench ein Icon gleichen Aussehens wie jenes der Workbench-Diskette, untertitelt mit »copy of Workbench1.3D«.

Sie besitzen jetzt eine Arbeitskopie der Workbench-Diskette, jener Startdiskette also, die Commodore Ihrem Amiga beigelegt hat. Es ist nun an der Zeit, dieses Original an einem sicheren Ort, fernab von Netzspannung führenden Leitungen, Transformatoren, Dauermagneten und stärkeren Wärmequellen, aufzubewahren. Tun Sie das mal, ich hole mir inzwischen eine Tasse Kaffee.

Original sicher verstaut? Okay. Belassen Sie die Kopie namens »copy of Workbench1.3D« in »df0:«. Schließlich ist es ja eine exakte Kopie der Originaldiskette, und wir sollten mit ihr genauso arbeiten können wie mit dem Original, nicht wahr? Na, ich will

Sie lieber nicht aufs Glatteis führen. So, wie Ihr System jetzt ist, können Sie mit der Kopie die Originaldiskette nicht völlig ersetzen. Obwohl auf der Kopie dieselben Programme enthalten sind wie auf der Originaldiskette, bestehen der Amiga und sein Betriebssystem in manchen Fällen darauf, Programme und Kommandos von jener Diskette laden zu können, mit der das System »gestartet« wurde. Da führt kein Weg dran vorbei. Sie glauben mir nicht so recht? Na gut, lassen wir's darauf ankommen.

Beschäftigen wir uns mit dem Inhalt unserer Kopie. Dazu klicken Sie das Disketten-Icon mit der Untertitelung »copy of Workbench1.3D« **zweimal** an. Wenn Sie das mit einer gemüthlichen Pause zwischen den beiden Betätigungen der linken Maustaste tun, wird der erzielte Effekt nahe Null sein. Das Disketten-Icon wird zwar sein Aussehen geringfügig ändern, um Ihnen anzuzeigen, daß es sich angesprochen fühlt. Das ist aber auch schon alles.

Sie müssen die beiden Tastenbetätigungen (für die sich der Ausdruck »Mausklick« eingebürgert hat) innerhalb einer bestimmten Zeit durchführen, damit sie als »Doppelklick« akzeptiert werden. Als Belohnung für den Streß öffnet der Amiga nach dem erfolgreichen Doppelklick ein neues Fenster. Während der Inhalt dieses Fensters eingelesen wird, verwandelt sich der Mauszeiger in die berühmte »Schnarchblase« mit dem »Doppel-Z«. Damit wird Ihnen angezeigt, daß der Amiga momentan beschäftigt ist und für weitere Wünsche (sprich Eingaben) kein offenes Ohr hat. Um den Inhalt dieses Fensters verstehen zu können, machen wir nun einen kleinen Ausflug in die Verzeichnisstruktur und die File-Symbolik einer Amiga-DOS-Diskette.

Sehen Sie sich bitte Bild 5 an. Als modernes Diskettenbetriebssystem verwaltet Amiga-DOS Verzeichnisse und Unterverzeichnisse (**Directories** und **Subdirectories**). Unterverzeichnisse können Files und weitere Unterverzeichnisse enthalten usw. In

AMIGA
SONDERHEFT

PROGRAMM- SERVICE

Direkt bestellen statt abtippen!

Die aktuelle Diskette zum Sonderheft 15:

Workshops

Alle Listings die in den Workshops zur Workbench und dem CLI abgedruckt sind, finden Sie in vollem Umfang auf der Programmservice-Diskette. Die Workshops zur Workbench und dem CLI finden Sie auf Seite 6 bzw. auf Seite 34.

Reeder

Tauchen Sie ein in die Welt der Windjammer. Werden Sie vom gerissenen Händler zum Bürgermeister. Aber um dieses Ziel zu erreichen, müssen Sie in der komplexen Handelssimulation »Reeder« viele Abenteuer überstehen und ge-

schicktaktieren. Denn Ihre Mitspieler lassen nichts unversucht, ihre Handelsgesellschaft in den Bankrott zu treiben. Wie Sie das verhindern können, lesen Sie ab Seite 90.

SuperMenu II

Die Basicerweiterung für anspruchsvolle Programmierer. Geben Sie Ihren Programmen eine professionelle Note. Mit »SuperMenu II« werden Menüleisten mit eigenen Logos zum Kinderspiel. Wie Sie »SuperMenu II« in Ihre Programme einbauen und den komfortablen Logo-Editor bedienen, lesen Sie ab Seite 103.

Checkie 42

Alle Listings in den Amiga-Sonderheften sowie dem Amiga-Magazin sind mit dieser Eingabehilfe versehen. Damit Ihnen zukünftig keine ärgerlichen Tippfehler mehr in Ihren Listings unterlaufen, finden Sie den Checksummer »Checkie 42« ab Seite 87.

Weiterhin finden Sie auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderheftes 15 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 45015

DM 24,90*

sFr 22,90*/öS 240,-*

* Unverbindliche Preisempfehlung



Markt & Technik
Zeitschriften · Bücher
Software · Schulung

**Weitere Angebote
auf der Rückseite!**

AMIGA PROGRAMMSERVICE

Bit für Bit nur Hits... ...das Beste aus zwei Jahren Amiga-Magazin!

Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

AmigaDat – Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung.

Manager – Das komfortable Haushaltsbuch.

Disketi – Drucken von Diskettenlabels.

MasterCruncher – Leistungsfähige Daten- und Programmkomprimierung.

Recover – Retten von gelöschten Daten.

Resi – Macht Programme resistent.

MouseCreator und PointerMaker – Generieren Sie Ihre eigenen Mauszeiger.

DiskSpy – Problemloses Ändern von Daten direkt auf Diskette.

AmigaSort – Bringt Ordnung in Ihre Diskettensammlung.

Fade – Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.

VirusControl – Der komfortable Virenschutz.

TrapHandle – Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action

Troof – Das spannende Motorradrennen der Zukunft.

Quadron – Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen

Poker – Wann bekommen Sie den Royal-Flush.

Billard – Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.

Domino – Verblüffende Umsetzung des bekannten Spiels.

Kniffel – Eine grafisch verblüffende Würfelspielumsetzung für bis zu vier Spieler.

3D-Tic-Tac-Toe – Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest

Eliza – Der Amiga als Psychotherapeut.

Arriba – Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readme-Datei auf den jeweiligen Disketten.

Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:

Bestell-Nr. 47901 **DM 39,90*** sFr 35,90*/öS 390,-*

Amiga-Sonderheft 13

Disk-Utility: Ein nützliches Utility für alle, die mit CLI-Befehlen auf Kriegsfuß stehen. Strukturieren, Formatieren und Kopieren ganzer Disketten und einzelner Files wird zum Kinderspiel. Selbst das Starten eines Programms von dieser Oberfläche ist möglich. **BiPF:** Der Grafikräuber, mit dem Sie alle Bilder aus Programmen stehlen können. »BiPF« findet die Bitplanes eines Programms und speichert die Bitmap als IFF-Bild. **PassWd:** Es ist manchmal etwas schwer, wichtige Disketten oder die Festplatte und damit die darauf befindlichen Daten vor unbefugtem Zugriff zu schützen. Mit »PassWd« erhalten Sie ein Werkzeug, mit dem Sie spielend Ihre Geheimnisse behalten. **DPaint III – DVideo:** Begleitend zu unseren Tips und Tricks zum Datentransfer zwischen DPaint III und DVideo, finden Sie auf der Programm-Service-Diskette alle Bilder, die in diesem Beitrag erarbeitet werden. Zusätzlich geben wir Ihnen ein Animations-Beispiel, das ebenfalls in dem Artikel erstellt wird. Zwei Disketten!

Bestell-Nr. 45013 **DM 29,90*** sFr 26,90*/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.

3,5"-Diskette für Amiga
Bestell-Nr. 45802 **DM 29,90*** sFr 26,90*/öS 299,-*

Amiga-Sonderheft 11

IconMaster: Ein nützliches Utility zum Erstellen eigener Piktogramme für die Workbench. Vorhandene Icons lassen sich verändern und IFF-Grafiken als Bildvorlagen nutzen. Ihrer Kreativität sind keine Grenzen gesetzt. **Maus:** Der Einheits-Mauspfeil gehört der Vergangenheit an. Mit diesem Programm installieren Sie einen Pointer nach Ihren Wünschen. Dabei stehen Ihnen 15 Farben zur Verfügung. Damit noch etwas Bewegung ins Spiel kommt, läßt sich der neue Mauszeiger sogar noch animieren.

DPaint-III-Workshop: Begleitend zu unserem großen DPaint-III-Workshop finden Sie auf der Programm-Service-Diskette alle Bilder, die in diesem Kurs erarbeitet werden. Zusätzlich geben wir Ihnen ein Animations-Beispiel, das ebenfalls in dem Artikel erstellt wird. **Deluxe Video 3.0:** Eine Ergänzung zu unserem Workshop über DPaint III stellt dieser Kurs dar. Auch zu diesem Artikel finden Sie alle erarbeiteten Beispiele auf der Programm-Service-Diskette. **Create-a-Shape:** Im Heft stellen wir Ihnen diesen Animationskünstler vor. Dabei wird Schritt für Schritt eine kleine Animation erstellt. Damit Sie auch gleich die einzelnen Schritte nachvollziehen können, finden Sie auf der Programm-Service-Diskette eine Demo-Version dieses hervorragenden Animations-Programms. Sie können alles damit anstellen, nur nicht abspeichern. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 11 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 45011 **DM 29,90*** sFr 26,90*/öS 299,-*

Drei Disketten!

Amiga-Sonderheft 14

»Silver« & »Reflections«

Alle Objekte, Szenen und Frames, die Sie für die beiden Workshops von Seite 6 und Seite 22 (Amiga SH 14) benötigen, finden Sie auf dieser Diskette. Ebenso die komplette Animation des »Turbo Silver«-Workshops.

»3-D-Digitizer«

3-D-Objekte einlesen kinderleicht, mit dem »3-D-Digitizer«. Lassen Sie sich von der komfortablen und leistungsfähigen Steuersoftware des 3-D-Digitizers mit integriertem Animator begeistern. Bauanleitung und Beschreibung finden Sie ab Seite 72 (Amiga SH 14).

»Time Manager«

Wer dieses Animations-Tool kennt, will es nicht mehr missen. Der »Time Manager« regelt den Zeitablauf Ihrer Animation. Wenn Bilder oder Logos nach dem Einblenden stillstehen sollten, mußte man bis jetzt dasselbe Bild x-mal hintereinander anzeigen lassen. Das kostet Speicherplatz und läßt die Ladezeit der Animation ins Unendliche wachsen. Anders beim »Time Manager«, mit ihm können Sie Animationen an x-beliebiger Stelle stoppen und warten lassen. Ab Seite 102 (Amiga SH 14) zieht Sie dieses Super-Tool in seinen Bann.

»Ray-Tracing in Basic«

Lernen Sie anhand eines Basic-Programms, wie Ray-Tracing funktioniert. Sehen Sie, wie mit einfachen Algorithmen fantastische Bilder entstehen. Die Beschreibung dieses lehrreichen Basic-Programms finden Sie ab Seite 111 (Amiga SH 14). Weiterhin finden Sie auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 14 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 45014 **DM 29,90*** sFr 26,90*/öS 299,-*

Zwei Disketten! * Unverbindliche Preisempfehlung

Wichtig: Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,- können Sie Software-Disketten Ihrer Wahl aus dem Programmservice-Angebot im Wert von DM 180,- bestellen – egal, ob diese DM 19,90, DM 29,90 oder DM 89,- kosten. Sie sparen DM 30,-!

Das Super-Software-Angebot finden Sie in den Zeitschriften **Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST-Magazin, PC Magazin, Computer live.**

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken! Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingehafteten Zahlkarte direkt beim Verlag bestellen.
Kennwort: »Super-Software-Scheckheft«, Bestell-Nr. W156

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.

Bestellungen bitte nur gegen Vorauskasse an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. 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) 5871393-0; Microcomput-ique, E. Schiller, Gögglstraße 17, A-3500 Krems, Telefon (02732) 741 93; MES-Versand, Postfach 15, A-3485 Haitzendorf; 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 kein Bargeld einschicken!

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.

diesen Verzeichnissen und Unterverzeichnissen sind dann die einzelnen Files abgelegt. Diese Files können entweder lediglich Daten enthalten, ausführbare Programme oder Text- bzw. Scriptdateien sein. Daß da auch noch andere Files (wie z.B. jene, in denen das Aussehen der Icons gespeichert ist) abgelegt sein können, sei hier nur der Vollständigkeit halber erwähnt. Solche Verzeichnisstrukturen nennt man »Baumstruktur«, weil wie bei einem Baum alle Äste und Ästchen letztendlich aus dem Stamm entspringen.

Die oberste Ebene der Verzeichnishierarchie stellt das Root-Directory dar (»root« - »Wurzel«). Es mag etwas unlogisch klingen, die oberste Ebene als Wurzel zu bezeichnen und eine Analogie zu einem Baum zu ziehen, es hat sich auf jeden Fall so eingebürgert, und wir wollen auch dabei bleiben. Um von der iconorientierten Workbench-Arbeitsoberfläche aus von einem Verzeichnis zum nächsten wechseln oder auch lauffähige Programme starten zu können, muß ein entsprechendes Icon vorhanden sein, das Sie anklicken können. Ohne Icon gibt es keine Zugriffsmöglichkeit auf Workbench-Ebene - Ende der Fahnenstange.

Nach diesem kleinen Ausflug in die Theorie kehren wir zu unserem Fenster auf der Workbench zurück, das sich nach dem Doppelklick auf das Disketten-Icon von »copy of Workbench1.3D« geöffnet hatte. Wie auch in Bild 6 sehen Sie auf Ihrem Monitor innerhalb des Fensterrahmens den von entsprechenden Icons repräsentierten Inhalt des Root-Directory, auf den Sie von der Workbench aus zugreifen können. Sie bemerken die Einschränkung? In den Verzeichnissen der Workbench- und auch anderer Disketten stehen viel mehr Files, als Sie Icons auf der Workbench sehen.

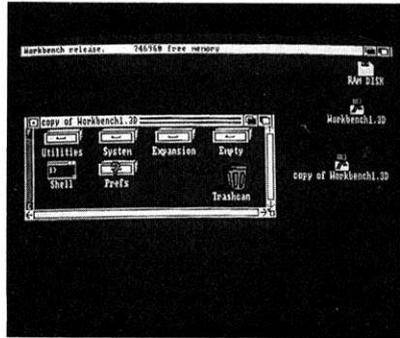


Bild 6. Das erste sog. »Root«-Verzeichnisfenster ist geöffnet

Aber wie bereits zu Beginn erwähnt, ist von der Workbench aus ein Zugriff auf ein File oder ein Directory nur dann möglich, wenn dafür ein Icon existiert. Jene Icons, die wie kleine **Schubladen** aussehen, repräsentieren Unterverzeichnisse, das gleiche gilt für den niedlichen Mülleimer namens **Trashcan**, der eine Sonderrolle spielt, wie Sie gleich sehen (und lesen) werden. Im Gegensatz zu den erwähnten Subdirectory-Icons ist das Icon mit dem Namen »Shell« ein »Project«-Icon. Auch darüber später mehr. Bevor wir uns jedoch in die Tiefen der Verzeichnisse stürzen und uns mit den darin enthaltenen Programmen befassen, möchte ich Ihnen die für nahezu alle Windows gültigen Bedienungselemente und Anzeigebereiche kurz erläutern (»Windows« nennt man diese eingerahmten Bildschirmbereiche auf der Workbench). Zur besseren Über-

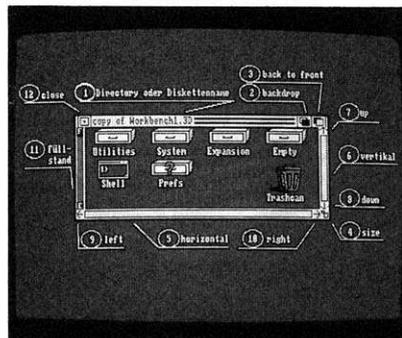


Bild 7. Die Bedienungselemente eines Workbench-Verzeichnisses

sicht habe ich Bild 7 mit Erläuterungen und Bezugszahlen versehen. Innerhalb des Rahmens sehen Sie die Bezeichnung des zu diesem Fenster gehörenden Directory (1). Im Falle des Hauptverzeichnisses steht hier der Name der Diskette. Die beiden Quadrate in der rechten oberen Ecke sind das **backdrop**-Gadget (2) und

das **back to front**-Gadget (3). Um die Wirkung dieser beiden Gadgets zu demonstrieren, ist es notwendig, ein zweites Window zu öffnen. Führen Sie den Mauszeiger über das Icon »Prefs«, und klicken Sie zweimal mit der linken Maustaste. Es geht uns jetzt nur um die Wirkung der Gadgets (2) und (3), also lassen wir den Inhalt des neuen Windows einmal unbesprochen. Worauf es uns ankommt ist, daß das neue Window mit der Bezeichnung »Prefs« jetzt einen Teil des Windows »copy of Workbench1.3D« überdeckt (Bild 8). An das »Trashcan«-Icon kämen wir jetzt gar nicht mehr ran. Mit einem Klick auf das Gadget »back to front« (3) im Fenster mit der Bezeichnung »copy of Workbench1.3D« legen wir das

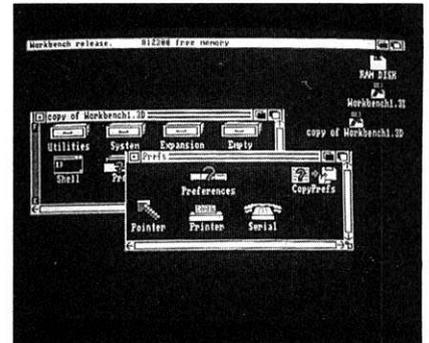


Bild 8. Windows sind wie Notizzettel stapelbar

überdeckte Window nach oben, so daß nunmehr das »Prefs«-Fenster nur teilweise sichtbar ist. Genau in die andere Richtung wirkt das »backdrop«-Gadget (2). Klicken Sie nun dieses an, und das »copy of Workbench1.3D«-Fenster liegt wieder hinter dem »Prefs«-Window. Nach dieser Demonstration stört uns das »Prefs«-Window mehr, als es uns nützen kann. Um Windows vom Bildschirm entfernen zu können, haben die Amiga-Entwickler das **close**-Gadget (4) vorgesehen. Sie finden es in der linken oberen Ecke jedes

Fensters, also auch im Window mit dem Titelleisteneintrag »Prefs«. Mauszeiger auf das »close«-Gadget des »Prefs«-Fensters stellen, einmal linke Maustaste gedrückt, weg ist es. Natürlich hat auch unser »copy of Workbench1.3D«-Window dieses Schließ-Gadget, aber wir sind ja noch nicht fertig. In der rechten unteren Ecke unseres Workbenchwindows ist das **size**-Gadget (4) angebracht (size = Größe).

Wie der Name schon andeutet, wird mit diesem Gadget die Größe des Windows beeinflusst. Wie alles auf der Workbench geht auch das ganz einfach. Positionieren Sie den Mauszeiger mit seiner Spitze auf diesem Gadget. Wenn Sie jetzt die linke Maustaste drücken und gedrückt halten, folgt die rechte untere Ecke des Fensters den Bewegungen der Maus. Bewegen Sie die Maus nach rechts und gleichzeitig zu sich hin,

Kontrollpult

wird das Fenster sowohl in der Höhe als auch in der Länge größer. Umgekehrt führt eine Bewegung der Maus (Maustaste gedrückt halten nicht vergessen!) nach links und von Ihnen weg zu einer Verkleinerung des Fensters. Verschieben des ganzen Fensters ist auch kein Problem. Dazu zeigen Sie mit dem Maus-Pointer auf die Titelzeile des Fensters und drücken die linke Maustaste. Halten Sie sie gedrückt und verfahren mit der Maus, folgt das Fenster der Bewegung des Zeigers, ohne seine Größe zu verändern. Auf diese Art hätten wir uns auch den Zugriff auf das »Trashcan«-Icon im Beispiel mit dem Vordergrund-Gadget verschaffen können. Allerdings kann kein Fenster den Workbench-Bildschirm »verlassen«. Wenn Sie beim Verschieben eines Windows mit einem Rand des Fensters an den Rand des Workbench-Bildschirms stoßen, ist es aus mit der Herumfahrelei. Gleiches gilt für Icons. Übrigens läßt sich

der ganze Bildschirm, genauer gesagt sein Inhalt, rauf und runter schieben. Ähnlich einem Fenster. Dazu packen Sie mit dem Mauszeiger nicht die Titelleiste eines Windows, sondern die Titelzeile mit der Aufschrift »Workbench.release«. Jetzt können Sie mit der Maus den ganzen Bildschirminhalt rauf und runter schieben.

Panoramafenster

Na, genug mit dem Größen-Gadget und seinen Verwandten gespielt? Okay, dann verkleinern Sie jetzt bitte das Window auf etwa halbe Höhe und halbe Länge. Sie sehen, daß nicht mehr alle Icons in den Rahmen passen. Und wie kommen wir jetzt an die unsichtbaren Icons heran? Natürlich ohne das Fenster wieder zu vergrößern. Dafür gibt es die beiden weißen **Schiebebalken** (5) und (6) sowie die vier **Scroll-Gadgets** (7), (8), (9) und (10). Die beiden Schiebebalken haben zwei Funktionen. Einerseits geben Sie durch ihre Größe eine Indikation dafür, wie groß das Fenster in der jeweiligen Richtung sein müßte, um alle Icons darstellen zu können, andererseits lassen sich diese Balken mit der Maus anpacken (Zeigerspitze drauf und linke Maustaste gedrückt halten) und ziehen - nach Loslassen der Maustaste bleibt dann der Balken an der aktuellen Position, und der Fensterinhalt wird dementsprechend verschoben. Sind also der horizontale Balken (5) halb so lang und der vertikale Balken (6) ein Viertel so hoch wie das Fenster, bedeutet dies, daß in horizontaler Richtung das Fenster doppelt so lang und in vertikaler Richtung viermal so hoch sein müßte, um alle Icons gleichzeitig angezeigt zu bekommen. Probieren Sie das doch gleich aus. Ähnlich, nur feiner abgestuft und mit direkter Ergebniskontrolle, funktionieren die Scroll-Gadgets. Stellen Sie doch bitte die beiden Balken (5) und (6) in die Mitte ihrer Schiebebereiche. Zeigen Sie mit der Spitze des Mauszeigers jetzt auf das **up**-Gadget (7), und drücken Sie die linke Maustaste. Der Inhalt des Fensters scrollt von oben nach unten durch den vom Fensterrahmen gebildeten Bildschirmausschnitt. Gleichzeitig wandert der Bal-

ken (5) mit nach oben. Machen Sie dasselbe mit dem **down**-Gadget (8) - jetzt scrollt der Fensterinhalt von unten nach oben. Es ist sicher leicht zu erraten, welche Funktionen und Auswirkungen die Scroll-Gadgets **left** (9) und **right** (10) haben. Fast haben wir's. Was noch zu erwähnen bleibt, ist der **Füllstandsanzeiger** (11). »E« am unteren Ende bedeutet »empty«, also leer, »F« steht für »full«, voll. Diese »Tankanzeige« versinnbildlicht die belegte und freie Speicherkapazität der zugehörigen Diskette. Sie finden diese Anzeige nur in Fenstern, die das Root-Directory einer Diskette repräsentieren. Ist das Feld zwischen »E« und »F« völlig ausgefüllt, bedeutet dies, daß die Diskette voll ist und keine weiteren Daten darauf gespeichert werden können. Bei einer frisch formatierten Diskette hingegen ist dieser Bereich frei.

Die Ablage

Vergrößern Sie nun bitte wieder das Fenster mit der Aufschrift »copy of Workbench1.3D« so weit, daß alle Icons sichtbar sind, die beiden Balken (5) und (6) ihre jeweiligen Bereiche voll ausfüllen und das Window ungefähr zwei Drittel des Bildschirms einnimmt. Sie haben jetzt unterhalb der Icons und rechts derselben viel freien Raum. Den werden Sie auch brauchen, denn wir wollen uns jetzt mit diesen Icons und deren Position innerhalb des Fensters beschäftigen. Icons lassen sich beliebig verschieben und plazieren. »Packen« Sie doch jetzt das Icon »Utilities« mit dem Mauszeiger an (linke Maustaste...) und ziehen Sie das Icon nach rechts unten in die Ecke des Windows. Sie haben sicher bemerkt, daß sich das bewegte Icon verändert hat - die Schublade ist jetzt geöffnet. Was das auf sich hat, erfahren Sie, wenn wir zu den Workbench-Menüpunkten kommen. Hier und jetzt interessiert uns nur, daß wir die Icons auf diese herrlich bequeme Art und Weise verschieben und neu plazieren können. Übrigens sind Sie

nicht an den Bereich innerhalb des Fensterrahmens gebunden. Packen Sie doch gleich mal die Schublade »Expansion« mit dem Mauszeiger an. Diesmal bewegen Sie die Maus aber so, daß der Mauszeiger mit dem daran klebenden »Expansion«-Icon aus dem Rahmen heraus auf die Workbench-Oberfläche gleitet. Wenn Sie jetzt die linke Maustaste loslassen, bleibt das Icon auf der Workbench. Selbst dann, wenn Sie das Fenster mit einem Klick auf das »close«-Gadget schließen! Auf diese Art legen Sie benötigte Schubladen oder auch Programme auf der Workbench zur weiteren Verwendung ab wie Werkzeug auf eine Werkbank, das Sie aus dem Werkzeugkasten nehmen. Wenn Sie den Werkzeugkasten dann wieder verschließen, haben Sie noch immer die benötigten Werkzeuge zur Hand. Merken Sie sich genau, wo momentan die Icons »Utilities« und »Expansion« plaziert sind. Schließen Sie das Fenster durch Anklicken des »close«-Gadgets. Das »Expansion«-Icon liegt noch immer an seiner durch Sie festgelegten Position auf der Workbench-Oberfläche. Und wie steht's mit dem »Utilities«-Icon? Öffnen Sie wieder das »Root«-Window durch zweimaliges Anklicken des Diskettensymbols »copy of Workbench 1.3D«. Das »Utilities«-Icon steht wieder an seiner ursprünglichen Position links oben und nicht mehr rechts unten, wo es beim Schließen des Fensters abgelegt war. Warum? Wenn Sie ein Fenster schließen, werden alle Verschiebungen von Icons innerhalb des Fensters wieder rückgängig gemacht. Die Position der einzelnen Icons ist in besonderen Files, den »info«-Dateien gespeichert. Wollen Sie eine solche Verschiebung auch beim nächsten Öffnen des Fensters vorfinden, müssen Sie dies dem Betriebssystem mitteilen, das daraufhin die neue Position auf der Diskette abspeichert. Dafür gibt es einen eigenen Befehl in der Workbench-Menüleiste, mit der wir uns aber erst später befassen.

Icons sind, wie bereits erwähnt, Symbole für Programme, Verzeichnisse, Bilder usw. Die Workbench ist eine symbolorientierte Benutzeroberfläche und stellt – wie der Name »Workbench« schon andeutet – eine Art elektronische Werkbank, einen Schreibtisch, dar. Nun gibt es zwei grundverschiedene Arten von Schreibtischen – aufgeräumte und chaotische. Commodore, der Amiga und die Workbench stellen es Ihnen frei, ob Ihr Amiga-Schreibtisch aufgeräumt oder penibel geordnet sein soll. Zu diesem Zweck gibt es auf der Workbench Schubladen (ein Vergleich mit Ringordnern ist ebenso treffend) und einen Papierkorb.

Ordnung muß sein

Wenn Sie zu den künstlerisch-intuitiv-chaotischen Typen zählen, werden Sie Schubladen kaum verwenden und alle von Icons repräsentierten Dateien im Hauptverzeichnis (»Root«-Directory – Sie erinnern sich) einer Diskette eintragen. Lassen Sie sich aber warnen: Je mehr Icons in einer Schublade eingetragen sind (und das Root-Directory ist eigentlich auch nichts anderes als eine Schublade), desto länger dauert es, bis nach dem Öffnen eines Fensters auch das letzte Icon eingelesen ist. Da sind die ordnungsliebenden User schon besser dran. Natürlich kann man alles übertreiben, und wer für jede Datei ein eigenes Verzeichnis als Unterverzeichnis eines Unterverzeichnisses erstellt, darf sich nicht wundern, wenn die linke Maustaste rotglühend in den Streik tritt. Wie immer liegt die Wahrheit in der Mitte, und ein Beispiel für diese Mitte ist die Workbench-Disk selbst. Fünf Ordner, ein Papierkorb und eine direkte Kommandoleitung sorgen für Überblick und raschen Zugriff. Was wo abgelegt ist, müssen Sie sich zwar selbst merken, aber dafür können Sie die Ordner ja auch aussagekräftig beschriften (im Falle der Workbench-Diskette hat das Commodore schon für Sie erledigt).

Haben Sie schon einmal das Arbeitsgebiet und damit den Schreibtisch eines Kollegen übernommen? Dann war sicher eine der ersten Aktivitäten, den Schreibtisch und die Ordner des Vorgängers nach den eigenen Vorstellungen zu ordnen,

Rechnungen nicht mehr nach Datum, sondern nach Lieferant geordnet abzulegen, die Ordner umzubeschriften und dergleichen mehr. Auf solche Aktivitäten ist natürlich auch die Amiga-Workbench eingerichtet, bloß geht alles viel leichter, schneller und bequemer als in einem Papierbergwerk. Dankenswerterweise hat uns Commodore eine Schublade zum Üben mit auf den Weg gegeben – »Empty« ist die sinnige Bezeichnung dieses Ordners. Da Sie das Fenster mit dem Titel »copy of Workbench1.3D« noch geöffnet haben, wird es Ihnen nicht schwerfallen, nun die Schublade »Empty« mittels Doppelklick zu öffnen. Siehe da, nomen est omen – die Schublade »Empty« ist wirklich leer. Jetzt geht's los. Verschieben wir doch mal probeweise einen Teil des Inhalts des Ordners »System« in den Ordner »Empty«. Da könnte ja jeder kommen und uns vorschreiben, was in welchem Ordner abgelegt zu sein hat.

Schreibtischtäter

Um dieses Vorhaben verwirklichen zu können, müssen wir natürlich auch den Ordner »System« aufschlagen, denn durch den Deckel kommen wir nicht an die Papiere. Also bitte klicken Sie auch das »System«-Icon an. So, jetzt herrscht schon ganz schöner Betrieb auf unserer Workbench. Machen wir doch etwas Platz! Die Einhakhefter »System« und »Empty« haben wir aus dem Ringordner »copy of Workbench1.3D« herausgenommen. Wozu liegt der Ringordner noch auf dem Schreibtisch? Weg damit. »Close«-Gadget anklicken und weg ist er. Daß das »System«-Fenster jenes von »Empty« teilweise überdeckt, könnte unser Vorhaben zwar nicht behindern, doch ist es sicher angenehmer, beide Fenster frei zugänglich zu haben. Verschieben Sie das »Empty«-Window soweit, daß es frei zugänglich ist. Sie wissen doch noch, wie das geht, oder? Na eben. Ich schla-

ge vor, wir holen das Programm »NoFastMem« aus dem »System«-Ordner heraus und legen es im Ordner »Empty« ab. Das funktioniert genauso wie in unserem Beispiel, als wir den Ordner »Expansion« auf die Werkbank legten, denn die Vorgangsweise beim Verschieben von Ordnern und Programmen ist grundsätzlich gleich: Icon »NoFastMem« mit dem Mauszeiger anfahren, durch Gedrückthalten der linken Maustaste »packen« und dann den Mauszeiger samt angeklebtem Icon so verfahren, daß das Icon in den Bereich des Fensters »Empty« kommt. Lassen Sie jetzt die linke Maustaste los, und das Laufwerk Ihres Amigas erwacht zum Leben. Anders als beim reinen Verschieben von Icons innerhalb eines Fensters oder auch beim Ablegen eines Icons auf der Workbench interpretiert der Amiga die soeben getätigte Aktion als den Befehl, das Programm »NoFastMem« dauerhaft aus der Schublade »System« herauszunehmen und dafür im Ordner »Empty« abzulegen. Dies bedarf einer Änderung des Inhaltsverzeichnisses der Workbench-Diskette, und weil der Amiga eine sehr ordnungsliebende Maschine ist, wird diese Änderung sofort durchgeführt und nicht erst nächsten Montag. Machen Sie dasselbe jetzt mit dem Programm »InitPrinter«.

Die Icons werden übrigens genau an jener Position abgelegt, an der Sie die linke Maustaste loslassen, und diese Position wird im Gegensatz zu unserer Verschiebung mit dem »Utilities«-Icon automatisch abgespeichert. Nun wäre es etwas mühsam, wollte man viele Icons nacheinander auf die beschriebene Art von einem Ordner in den anderen transferieren. So wie Sie aus einem Ordner nicht nur Einzelblätter, sondern auch einen ganzen Stapel herausnehmen und diesen mit etwas Geschick in einem anderen Ordner ablegen können, so ist dies auch auf

der Workbench vorgesehen. Allerdings bedarf dies Ihrer zweiten Hand. Sie haben bemerkt, daß immer nur ein Icon als »aktiv« gekennzeichnet wird. Jeder Klick auf ein weiteres Icon innerhalb desselben Ordners macht das zuvor aktive wieder inaktiv. Wie also mehrere gleichzeitig aktivieren? Man nehme die Shift-Taste. Wie bitte?

Sieben auf einen Klick

Mann, die mit dem Pfeil nach oben! Zwei gibt es davon, eine links, eine rechts! Nicht auf der Maus, auf der Tastatur. Eine unter der <Ctrl>-Taste und eine unter der <Return>-Taste. Na endlich! So, wir nehmen jetzt die Programme »DiskCopy«, »Format« und »SetMap« aus dem »System«-Ordner heraus und legen sie im »Empty«-Ordner ab. Der Trick ist der: Halten Sie eine der <Shift>-Tasten gedrückt, während Sie das erste Icon, meinetwegen »SetMap«, anklicken. »SetMap« wird »aktiviert« - Image-Änderung. <Shift>-Taste weiter gedrückt halten und nacheinander »DiskCopy« und »Format« anklicken. Sie sehen jetzt etwas, was Sie noch nie zuvor sahen: drei gleichzeitig als »aktiv« deklarierte Icons im selben Window. Die <Shift>-Taste ist noch immer gedrückt. Eines der drei aktivierten Icons fangen Sie jetzt auf die bekannte Art und Weise (nach dem »Fangklick« können Sie übrigens die <Shift>-Taste wieder loslassen) und ziehen es in das »Empty«-Window - und siehe da, die beiden anderen kommen folgsam mit. Aber nicht nur die beiden anderen folgen der Bewegung des ersten. Auch das »InitPrinter«-Icon wandert. Was soll das? Tja, von unserer letzten Transaktion ist im Ordner »Empty« noch immer das Icon »InitPrinter« aktiviert. Obwohl das Fenster selbst inaktiv ist (erkennlich an der gerasterten, kontrastarmen Schrift in der Titelzeile des Windows). Achten Sie bitte unbedingt darauf, daß auch wirklich nur jene Icons aktiviert sind, die Sie bearbeiten wollen. Die beste Methode, mit Sicherheit alle Icons (auch solche, die durch andere Fenster verdeckt sind) zu deaktivieren, ist ein Klick außerhalb eines Fensters auf

die Workbench. Sozusagen ein Schlag auf den Schreibtisch. Und was nun? Ich muß Ihnen leider sagen, ich habe bis jetzt nur einen seriösen Ausweg aus dieser Situation gefunden (ausgenommen vielleicht so brutale Dinge wie »Diskette aus dem Laufwerk nehmen« oder »<CTRL> <Amiga-links> <Amiga-rechts>«): Halten Sie die linke Maustaste gedrückt, drücken Sie die rechte Maustaste dazu, lösen Sie die linke Maustaste und erst danach wieder die rechte. Andernfalls müssen Sie wohl oder übel irgendwann die linke Maustaste wieder loslassen, und dann läuft die Transaktion. Im konkreten Fall würde das bedeuten, daß die aktivierten Icons und damit natürlich auch die zugehörigen Programme in jenen Ordner umkopiert würden, über dem zum Zeitpunkt des Loslassens der linken Maustaste der Mauszeiger gerade steht. Also bitte Vorsicht. Ich hoffe, jetzt steht einem erfolgreichen Verschieben mehrerer Files gleichzeitig nichts mehr im Wege. Muß ich noch erwähnen, daß mit derselben Prozedur auch Ordner in andere Ordner gesteckt werden können? Wohl kaum, also lasse ich's auch.

Halt, eine Bemerkung will ich doch noch anbringen. Der »Trashcan«-Ordner ist das Workbench-Äquiva-

Alles im Eimer

lent zum Papierkorb neben dem Schreibtisch. Files (Papiere) und Schubladen (Ordner), die nicht mehr gebraucht werden, werfen Sie mit der oben beschriebenen Methode einfach in den Mülleimer. Und solange die Fußbodenkosmetikerin den Papierkorb nicht ausgeleert hat, steht einem Zurückholen versehentlich weggeworfener Dinge (Daten) nichts im Wege. Glücklicherweise kommt auf der Workbench die Putzfrau nur auf Anforderung - dazu gleich mehr, wenn wir uns mit der Menüleiste befassen. Es muß Ihnen aber klar sein, daß durch die bloße Transferierung von Files oder ganzen Ordnern in den »Trashcan«-Ordner kein Bit auf der Diskette frei

wird! Es ist wie in einem Büro – Sie werfen etwas in den Papierkorb und gewinnen dadurch Platz auf Ihrem Schreibtisch, aber das Büro selbst wird dadurch nicht leerer. Ich möchte Sie nun bitten, die momentan im »Empty«-Ordner abgelegten Files wieder in den »System«-Ordner zurückzulegen. Nicht nur deshalb, damit wir für den weiteren Ablauf wieder eine originalgetreue Kopie der Workbench-Diskette zur Verfügung haben, sondern auch, weil einige der bearbeiteten Programme von anderen Programmen nur im »System«-Ordner gesucht werden, und wenn sie dort nicht vorgefunden werden, gibt das Ärger im System. Außerdem wird etwas Übung nicht schaden.

Kommen wir zur bereits mehrfach erwähnten Menüleiste. Um in diesen Artikel eine Struktur hineinzubringen, möchte ich die Menüleiste von links nach rechts abhandeln – ich hoffe, Sie haben nichts dagegen. Wo denn diese **Menüleiste** sei, fragen Sie? Die liegt »unter« der weiß unterlegten Titelzeile am oberen Rand des Bildschirms. Ja, richtig, es ist jene Zeile, in der der freie Speicherplatz angezeigt wird. Und wie Sie sehen, hat diese Zeile dieselben »backdrop«- und »back to front«-Gadgets, die wir auch bei den Fenstern vorfinden. Wie war das noch gleich – durch Anklicken des »backdrop«-Gadgets wird das aktive Fenster ganz nach hinten gelegt, und demzufolge werden jene Fenster, die bis dahin vom aktiven Fenster verdeckt wurden, sichtbar. Klicken Sie doch mal das »backdrop«-Gadget an ... Hören Sie auf, hören Sie auf! Und wenn Sie hundertmal klicken, davon wird die Menüleiste auch nicht sichtbar. Bevor Sie mich jetzt würgen (verstehen Sie denn keinen Spaß?) – lassen Sie die linke Maustaste in Ruhe, und drücken Sie die rechte. Der Inhalt der Titelleiste ändert sich drastisch und zeigt nun die **Hauptmenüpunkte**.

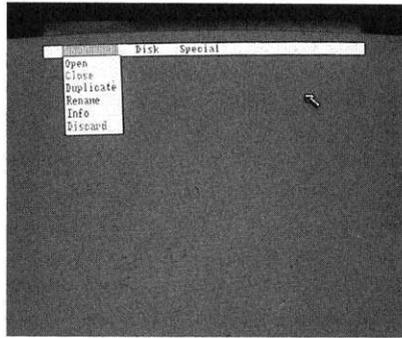


Bild 9. Bei Bedarf klappen die Submenüs nach unten

Sollten Sie schon einmal den Begriff **Menü-Taste** im Zusammenhang mit der Amiga-Maus gehört haben – jetzt wissen Sie, warum die rechte Amiga-Maustaste so bezeichnet wird. Eine Menüleiste hat mit den Charakteristika eines Fensters überhaupt nichts gemein, deshalb kann ein Klick auf das »backdrop«- oder das »back to front«-Gadget auf das Sichtbarwerden einer Menüleiste keinerlei Einfluß haben. Es spielt auch überhaupt keine Rolle, wo auf dem Workbench-Bildschirm sich der Mauszeiger zum Zeitpunkt des Drückens der Menütaste befindet. Ein Druck auf die Menütaste bringt immer die Menükopfleiste zum Vorschein. Bloß haben wir von den Hauptmenüpunkten allein recht noch wenig.

Intuition à la carte

Die Workbench-Power steckt in den Untermenüs, doch die bekommen wir erst zu Gesicht, wenn der Mauszeiger die Titelleiste, eigentlich die Menükopfleiste, berührt. Dies gilt nicht nur für die Workbench, sondern für alle Programme, die mit Menüzeilen versehen und den Commodore-Richtlinien entsprechend programmiert wurden. Drücken Sie die Menütaste, führen Sie den Mauszeiger mit gedrückter Menütaste in die Kopfleiste und bewegen Sie dann den Mauszeiger horizontal nach rechts, ohne die Menüleiste zu verlassen. So sehen Sie ein Submenü nach dem anderen, ohne eine echte Aktion auszulösen. Auch wenn die Bedeutung einzelner **Submenüpunkte** aus der Bezeichnung alleine heraus nicht klar ist, eines fällt Ihnen sicher auf: Manche Wörter sind voll und kontrastreich geschrieben, andere wieder nur angedeutet, gera-

stert wie ein Zeitungsbild. Sie vermuten richtig – »normal« geschriebene Menüpunkte sind solche, die bei den gegebenen, aktuellen Randbedingungen angewählt und aktiviert werden können, die »gerasterten« dagegen nicht. Um aus einem »gerasterten« einen »normalen« Menüpunkt zu machen, müssen Sie bestimmte Aktionen setzen, abhängig davon, welchen Menüpunkt Sie aktivieren wollen. Zum »WIE« und »WANN« kommen wir gleich. Schließen Sie jetzt bitte alle eventuell geöffneten Fenster durch einen Klick auf deren »close«-Gadgets, und klicken Sie einmal in die leere Fläche der Workbench, damit mit Sicherheit kein Disketten-Icon aktiviert ist und damit wir einen gemeinsamen Ausgangspunkt auf unserer Werkbank haben.

Beschäftigen wir uns zunächst wie angekündigt mit dem Submenü ganz links, das den schönen Titel **Workbench** trägt. Darin enthalten sind die Kommandos »Open«, »Close«, »Duplicate«, »Rename«, »Info« und »Discard« (Bild 9). Um die Wirkung von »Open« zu demonstrieren ist es notwendig, ein Disketten-Icon durch einmaliges Anklicken zu aktivieren. Danach drücken Sie die Menütaste (die rechte Maustaste) und führen den Mauszeiger über die Submenübezeichnung »Workbench«. In dem nun »heruntergeklappten« Submenü erkennen Sie, daß im Gegensatz zum ersten Mal vier der sechs Submenüpunkte, nämlich »Open«, »Duplicate«, »Rename« und »Info« voll geschrieben sind und nur noch die zwei verbleibenden Kommandos gerastert erscheinen. Ist ja auch irgendwo klar: Nachdem Sie dem Amiga durch das Anklicken eines Icons (in diesem Fall war es das Disketten-Icon der Workbench-Diskette) mitgeteilt haben, womit Sie sich befassen wollen, stehen Ihnen die vier genannten Punkte zur Verfügung. Und auch wenn Sie wirklich noch nicht wissen, was mit »Open« und »Close« gemeint

sein könnte, eines ist klar – man kann nur etwas schließen («close»), wenn zuvor etwas geöffnet («open») wurde. Aktivieren Sie bitte den »Open«-Befehl (Menütaste drücken – Mauszeiger in Menüleiste führen, bis der Schriftzug »Workbench« farbig unterlegt ist und das Submenü »herunterklappt« – Mauszeiger bei gedrückter Menütaste innerhalb des Submenükastens nach unten führen, bis »Open« farbig unterlegt ist – Menü-Taste loslassen). Das Ergebnis kennen Sie schon – es öffnet sich ein Fenster, in dessen Titelzeile »copy of Workbench1.3D« steht. Der »Open«-Befehl aus dem Workbench-Menü entspricht also dem Doppelklick der linken Maustaste auf ein Icon. Sehen Sie sich nochmal den Submenüblock von vorhin an. Jetzt ist auch »Close« voll ausgeschrieben und nicht nur gerastert. Wählen Sie »Close« an, wie Sie es zuvor mit »Open« getan haben – das Fenster »copy of Workbench1.3D« wird geschlossen, verschwindet. Was folgern Sie daraus, Doktor Watson?

Nur mit Original

Richtig – **Close** entspricht einem einfachen Klick mit der linken Maustaste auf das Schließ- oder »close«-Gadget im Fensterrahmen. Auch die Wirkung von **Duplicate** haben Sie bereits kennengelernt. Das Icon der Workbench-Diskette (bzw. deren Kopie) ist noch immer aktiv, wie Sie an der geänderten Farbgebung leicht erkennen. Aktivieren Sie bitte »Duplicate«. Mit dieser Reaktion des Amigas hatten Sie nicht gerechnet, wie? Verlangt dieses Stück Blech doch tatsächlich nach der **Original-Diskette**. Erinnern Sie sich noch, was ich Ihnen einige Seiten zuvor prophezeit hatte?

Es gibt Situationen, in denen der Amiga Kommandos von jener Diskette nachlädt, mit der er gestartet wurde. Er akzeptiert auch nur diese eine Diskette, und es ist dem Betriebssystem vollkommen egal, ob

der gesuchte Befehl oder das gesuchte Programm nun auf der aktuell eingelegten Diskette drauf ist oder nicht – die Boot-Diskette muß es sein. Da ist unser Freund/unsere Freundin stur wie ein Esel. Es ist Ihnen im Verlauf unseres gemeinsamen Rundgangs auf der Workbench vielleicht schon aufgefallen, daß das Icon der Boot-Diskette immer sichtbar ist, das

Noch mal von vorne

Icon der Diskette »copy of Workbench1.3D« aber immer gelöscht wurde, wenn die Diskette aus dem Laufwerk genommen wurde und kein zu dieser Diskette gehörendes Fenster geöffnet war. Auch das ist ein Hinweis des Amigas an Sie, daß die Boot-Diskette fest ins System eingebunden wurde. Und wenn Sie bis hierher alles so nachvollzogen haben, wie ich es beschrieben habe, dann muß Ihr Amiga jetzt nach der Original-Workbench-Diskette verlangen. Quod erat demonstrandum.

Damit der Amiga sich in Zukunft mit unserer Workbench-Kopie zufriedengibt, gibt es nur eine Möglichkeit: **Warmstart** mit <CTRL> <Amiga-links> <Amiga-rechts>. Warmstart nennt man jenes »Hochfahren« des Systems, bei dem nur bestimmte Tabellen und Speicherplätze im Memory neu initiiert werden, ohne daß deswegen der ganze Speicher gelöscht würde. Im Unterschied dazu bleibt beim Kaltstart, bei dem ja zuvor die Spannungsversorgung abgeschaltet wird, kein Stein mehr auf dem anderen. Ein Kaltstart ist im normalen Workbench-Betrieb kaum notwendig. Anders sieht das aus, wenn Sie kommerzielle Spiele in Betrieb nehmen. Der Großteil dieser Spiele reißt sich der Amiga derart rigoros unter den Nagel, daß nur noch der Griff zum Netzschalter die durcheinander gewürfelten Vektoren wieder zurechtbiegt (Vektoren sind eine Art Zeiger, die von Programmen abgefragt werden und die z.B. einem Programm mitteilen, wo im Speicher eine bestimmte Routine zu finden ist). Machen Sie jetzt bitte den berühmterbüchtigten Drei-Finger-Spinnen-griff und drücken Sie gleichzeitig die <CTRL>-Taste und beide Amiga-Tasten. Lassen Sie nach ca. 2 Sekunden los und warten Sie, bis auf Ih-

rem Bildschirm wieder die Workbench zu sehen ist.

Gesagt, getan. Ab nun gibt sich der Amiga auch mit der Kopie der Workbench-Diskette voll und ganz zufrieden. Aktivieren Sie bitte wieder das Icon der Diskette »copy of Workbench1.3D« durch einen einzelnen Klick mit der linken Maustaste (Voraussetzung ist natürlich, daß sich der Mauszeiger über dem genannten Diskettensymbol befindet). Wir wurden durch die Sturheit unseres Computers von unserem Vorhaben abgehalten, den »Duplicate«-Befehl auszuprobieren. Wählen Sie bitte wieder »Duplicate« aus dem Workbench-Menü. Was sich daraufhin tut, erinnert frappant an jene Vorgänge, als wir eine Kopie der Original-Workbench-Diskette »zogen«, denn »DiskCopy« wird aktiv.

Der Unterschied ist, daß »DiskCopy« bei »Duplicate« nicht die angeklickte (aktive) Diskette auf eine andere kopiert, sondern daß »Duplicate« die aktive Diskette als Zieldiskette sieht. Da es nicht in unserem Sinne sein kann, auf die mühsam erworbene Kopie der Workbench-Diskette zu kopieren, beenden Sie vorerst das »Duplicate«-Programm durch Anklicken des »Cancel«-Gadgets im Requester links oben. Machen wir

Der »DiskCopy« Trick

uns den Spaß und duplizieren wir gemeinsam die Boot-Diskette. Dazu legen Sie bitte eine neue (leere, unformatierte) Diskette in das interne Laufwerk namens »df0:«. Lassen Sie sich durch die Anzeige »df0:BAD« auf der Workbench nicht irritieren. Klicken Sie das »BAD«-Icon einmal an, um es zu selektieren, und wählen Sie danach den Menüpunkt »Duplicate«. Folgen Sie ab nun den Anweisungen der System- und DiskCopy-Requester, wie wir es bereits besprochen haben. Nach Abschluß des Kopiervorgangs haben Sie eine weitere Kopie der Workbench-Diskette, und unter dem Icon steht »copy 2 of Workbench1.3D«. Womit bewiesen wäre, daß Amiga-DOS tatsächlich Kopien durchnummeriert. Das funktio-

niert natürlich nur dann, wenn Sie eine Diskette mit Namen »copy of ...« duplizieren. Und wenn Sie die Diskette »copy 2 of Workbench1.3D« zweimal auf zwei verschiedene Disketten kopieren, haben Sie am Ende zwei Disketten, die beide den gleichen Namen tragen: »copy 3 of Workbench1.3D«.

Diese Durchnummerierung ist nicht nur ein netter Gag. Sie erspart Ihnen bei Anwendung des »Duplicate«-Kommandos die Mühe, die Kopie umzubenennen. Amiga-DOS vermag Disketten gleichen Namens und gleichen Icons zu unterscheiden – Sie auch? Seien Sie bitte nicht so vermessen, diese Frage mit »Ja« zu beantworten. Es geht vielleicht fünfmal gut, meinetwegen auch zehnmal. Spätestens dann, wenn Sie nicht wiedergutzumachenden Schaden anrichten können, werden Sie Disketten verwechseln. Auch die Arbeit am und mit dem Amiga gehorcht den Gesetzen von Edsel Murphy (»If anything can go wrong it will«, auf deutsch »wenn etwas schiefgehen kann, geht's auch schief«). Machen Sie lieber keine Experimente. »Duplicate« kopiert aber nicht nur Disketten. Ebenso gut vervielfältigen Sie damit einzelne Files oder auch ganze Directories. Der Unterschied in der Anwendung besteht lediglich darin, daß im Gegensatz zum zuvor praktizierten Diskettenduplizieren nun das entsprechende Original-File oder die Originalschublade anzuklicken ist. Versuchen Sie das mal mit dem »Shell«-Icon im Hauptverzeichnis. Nach kurzer Zeit erscheint unterhalb des Original-Icons ein zweites Icon gleichen Aussehens, aber mit der Bezeichnung »copy of Shell«. Versuchen Sie das auf der Workbench-Diskette aber bitte nur mit »Shell«, denn dieses File ist so klein, daß auch auf der an und für sich randvollen Workbench-Diskette noch genug Platz für die Kopie ist.

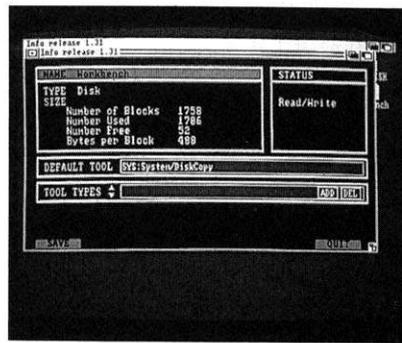


Bild 10. Das »Info«-Fenster zu einer kompletten Diskette

»Duplicate« kann die Kopie nur in jenem Verzeichnis ablegen, in dem auch das Original steht. Um Kopien von Files oder Schublade auf Diskette »A« auch auf Diskette »B« zu haben, müssen Sie das zu kopierende Icon mit der Maus »fangen« und in den Fensterrahmen des Zielverzeichnisses ziehen. In diesem Fall wird allerdings keine Umbenennung des Duplikats in »copy ...« vorgenommen. Hier sei eine Randbemerkung gestattet. Es gibt leider erst ab »Workbench 2.0«, (jener Version, die mit dem Amiga 3000 ausgeliefert wird) die Möglichkeit, auf Workbench-Ebene neue Directories (Schublade) anzulegen, ohne dazu einen bestehenden Ordner kopieren zu müssen. Beim Kopieren von Ordnern ist aber zu beachten, daß auch deren Inhalt mitkopiert wird. Wenn also Sie und ich ein neues, leeres Directory eröffnen wollen, bleibt Ihnen und mir nichts anderes übrig als den »Empty«-Ordner auf die bekannte Art und Weise zu kopieren und anschließend umzubenennen.

Womit wir auch schon beim nächsten Kommando im Menü »Workbench« wären: **Rename**. »Rename« dient zum Umbenennen von Files, Ordnern und Disketten. Wenn Sie den Menüpunkt »Rename« auswählen und (zuvor) das umzubenennende Icon aktiviert haben, öffnet sich auf dem Workbench-Screen ein »String«-Gadget, in dem der aktuelle Name des Delinquenten eingetragen ist. Dieser kann jetzt nahezu nach Belieben modifiziert werden. Schließen Sie die Eingabe durch Drücken von <RETURN> ab, und schon ist der neue Name eingetragen. Versuchen Sie das doch gleich mal an unserer Boot-Diskette. Klicken Sie das Disketten-Icon der

Boot-Diskette an, um es zu aktivieren, und wählen Sie danach »Rename« aus dem Menü »Workbench«. Ändern Sie den Namen der Diskette auf »Workbench«, und drücken Sie <Return>. Schon hat die Boot-Diskette einen neuen Namen. Das hat übrigens überhaupt keinen Einfluß auf den Amiga. Sie erinnern sich an unseren Warmstart, weil die Boot-Diskette nicht mehr greifbar war bzw. Sie sie in Sicherheit gebracht hatten. Das Betriebssystem merkt sich die Namensänderung und greift bei seiner File-Suche auch weiterhin auf die Boot-Diskette zu.

Informationsdienst

Um von der Workbench aus an verschiedene Informationen über File-Größe, Icon-Typ usw. herankommen zu können, wurde der nächste Menübefehl implementiert: **Info**. Je nach Icon-Typ gibt »Info« unterschiedlich strukturierte Informationen aus. Der Amiga unterscheidet grundsätzlich zwischen fünf verschiedenen Icon-Grundtypen: »Disk«, »Tool«, »Project«, »Drawer« und »Garbage«. »Disk« ist klar – damit werden Disketten symbolisiert. »Tool« steht für jedes direkt ausführbare Programm. »Project« ruft ein »Tool« auf und übergibt bei Bedarf an dieses aufgerufene Programm zusätzliche Informationen, stellt also sowas wie eine Fernsteuerung für ein »Tool« dar. Die übergebenen Informationen sind entweder in einem Text-File abgelegt oder im Infofenster eingetragen.

Mehr darüber gibt's bei der Besprechung des »Project«-Infos. »Drawer« ist die schon bekannte Schublade, also der Eingang zu einem Unterverzeichnis. »Garbage« ist die Typbezeichnung für den Mülleimer der Workbench, genannt »Trashcan«. Dies ist übrigens das einzige Icon und damit auch das einzige Verzeichnis, das Sie von der Workbenchoberfläche aus weder duplizieren noch löschen können!

In Bild 10 sehen Sie ein Beispiel für die Information, die Ihnen der

Amiga gibt, wenn Sie ein **Disk-Icon** anklicken und danach »Info« aus dem »Workbench«-Menü anwählen. Ganz oben steht der Name der Diskette. In der Rubrik »Status« steht entweder »Read/Write« wie in unserem Beispiel oder »Read Only«. Sie erfahren auf diesem Weg den gegenwärtigen Stand des Schreibschutzschiebers, ohne die Diskette aus dem Laufwerk nehmen zu müssen. Daß neben »Type« »Disk« steht, wird Sie wohl nicht weiter überraschen. Der Block unter »Size« gibt Auskunft über die Gesamtanzahl der auf dieser Diskette formatierten Blocks (das sind die »Sektoren« aus unserer kurzen Abstecher in die Diskettenstruktur), die Anzahl der von Dateien belegten Blöcke und darunter als Differenz der beiden Angaben die Anzahl der noch freien Blöcke. Die Eintragung »Bytes per Block« ist bei normalen Disketten immer gleich und informiert Sie dar-

»Info« Einträge

über, daß je Block 488 Byte Daten eingetragen werden. Ein eigener Block im Infenster ist dem »Default-Tool« gewidmet. Den Eintrag »SYS:System/DiskCopy« könnten Sie beliebig ändern oder auch löschen. Dafür klicken Sie mit dem Mauszeiger einfach in das Feld, in dem die genannte Eintragung steht. Sie könnten dort z.B. eintragen »Juppidu« – kein Problem, aber auch sinnlos. Der Eintrag »SYS:System/DiskCopy« steht nicht grundlos dort. Er sorgt dafür, daß das Programm »DiskCopy« aus dem Ordner »System« der Boot-Diskette aufgerufen wird, wenn Sie dieses Disketten-Icon über ein anderes ziehen. Ach ja, das habe ich Ihnen ja noch nicht ge-

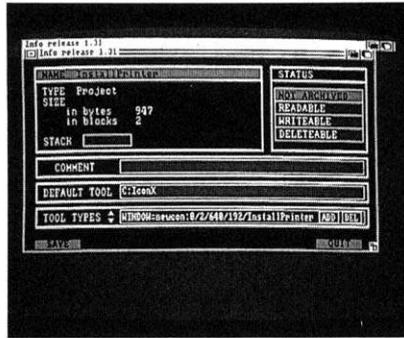


Bild 11. Beispiel für ein »Project«-Infenster – Install Printer

sagt: Der Amiga kann die Boot-Diskette unter zwei Namen ansprechen. Einerseits unter der Bezeichnung, die Sie auch auf der Workbench sehen (also z.B. »Workbench« oder auch »copy of Workbench1.3D«), und andererseits unter der Bezeichnung »SYS:«. Wann immer Sie sich auf die Boot-Diskette beziehen wollen, sollten Sie der Bezeichnung »SYS:« den Vorzug geben, denn dann sind Sie unabhängig von etwaigen »Rename«-Orgien. Das Eintragungsfeld in unserem »Disk«-Infenster namens »TOOL TYPES« ist leer und sollte es auch bleiben. Ich komme gleich auf diesen Bereich zu sprechen – noch etwas Geduld bitte. Unten links sehen Sie ein Gadget namens »SAVE«. Wenn Sie etwas im Infenster verändert haben (sei es ein »Default Tool«-Eintrag oder auch eine Stack-Angabe, über die wir uns gleich unterhalten), so müssen Sie das Fenster mit einem Klick auf »SAVE«-Gadget verlassen, sonst werden die Änderungen nicht abgespeichert. Ein Klick auf »QUIT« hingegen schließt das Infenster, ohne etwaige Änderungen abzuspeichern – eine Art Notausgang, wenn Sie geänderte Einträge lieber nicht abspeichern wollen und die schnellste Art, »Info« zu beenden, wenn Sie nur einen Blick auf die File-Informationen werfen wollten, ohne etwas zu ändern.

Das Ergebnis einer Info-Anforderung über ein **Project-Icon** und damit über eine »Project«-Datei sehen Sie in Bild 11. Als erstes fällt auf, daß das Feld »Status« plötzlich in vier Bereiche unterteilt ist, von denen in unserem Beispiel das oberste blau unterlegt ist. Die Bedeutung dieser vier Felder ist folgende. Vielleicht haben Sie schon etwas von den sog.

»Schutz-Bits« oder »Protection-Bits« gehört. Wenn nicht, im CLI-Artikel gehen wir näher darauf ein. Für uns und den Workbench-Betrieb ist nur deren Auswirkung wesentlich und ebenso natürlich auch die Art und Weise, wie wir auf diese Bits Einfluß nehmen können. Klicken Sie mal mit dem Mauszeiger irgendwo in das blau unterlegte Feld, in dem »not archived« steht. Haben Sie? Dann werden Sie bemerkt haben, daß sich nach dem ersten Mausklick die Eintragung auf »archieved« geändert hatte. Nach einem weiteren Klick kommt wieder »not archived« usw. Im normalen Workbench-Betrieb ist diese Eintragung von keinerlei Bedeutung. Wenn Sie aber Backup-Programme benutzen, die das Archiv-Flag abfragen, können Sie damit viel Zeit sparen. Sie wollen trotzdem wissen, was das ganze soll? Na gut, machen wir einen kurzen Abstecher in die Backup-Technik. Ein »Backup« ist eine Sicherheitskopie, wie wir sie zu Beginn von unserer Original-Workbench-Diskette angefertigt haben. Es gibt grundsätzlich zwei Arten von Backups: »Image-Backups« und »File-Backups«. Bei »Image-

Das Backup

Backups« wird ein Datenträger (z.B. eine Diskette) komplett und soweit möglich originalgetreu auf einen zweiten Datenträger kopiert. Bei »File-Backups« wird File um File, Datei um Datei, auf einen anderen Datenträger kopiert. Und hier kommen wir zur Anwendung des »Archive«-Flags. Gute File-Backup-Programme kopieren nicht alle Files auf den zweiten Datenträger, sondern nur jene, die seit der letzten Datensicherung (seit dem letzten Backup) verändert wurden.

Wie weiß nun das Backup-Programm, welche Dateien verändert wurden und welche nicht? Dazu wird das »Archive«-Bit herangezogen.

Nach einem »File-Backup« setzen solche Backup-Programme das »Archive«-Bit, tragen also auf dem Original ein, daß dieses File »archiviert« wurde. Wenn Sie dieses archivierte File irgendwann verändern und erneut abspeichern, löscht das Amiga-Betriebssystem dieses Bit wieder. Beim nächsten Backup-Durchlauf überprüft das Backup-Programm dann den Zustand der »Archive«-Bits aller Files und archiviert nur jene, die seit dem letzten Archivierungsvorgang verändert wurden, bei denen also das »Archive«-Bit nicht gesetzt ist. So einfach ist das. Und es spart eine Menge Zeit. Damit Sie aber auf Workbench-Ebene dieses Bit bewußt setzen oder löschen können (aus welchem Grund auch immer), haben die Workbench-Programmierer die Eintragung von »archived« oder »not archived« im Infofenster zugelassen. Steht dort »archived«, wird das Programm von einem Backup-Programm übergangen. Lautet die Eintragung »not archived«, wird ein Backup durchgeführt, weil die Datei in diesem Zustand »noch nicht archiviert« wurde. Soweit unser Abstecher in die Backup-Technik – zurück zu unserer »Project-Info« und zum Block »Status«. Unter »archived« sehen Sie »readable«, »lesbar«. Klicken Sie in dieses Feld, wechselt die Anzeige zwischen »readable« und »not readable«. Mit diesem Eintrag legen Sie fest, ob die Datei gelesen bzw. geladen werden kann (»readable«) oder nicht (»not readable«). Es wird wohl auf einige wenige Anwendungsfälle beschränkt bleiben, einer Datei das Prädikat »nicht lesbar« zuzuordnen, aber Sie wissen jetzt, wie's geht. Das nächste Feld trägt den Eintrag »writeable«, also »beschreibbar«. Da wird die Sache schon interessanter, denn hier haben Sie das auf Einzel-Files übertragene Software-Äquivalent des Schreibschutzschiebers der Diskette vor sich. Steht hier »writeable«, kann das File überschrieben werden. Ein Klick mit der Maus wandelt

den Eintrag in »not writeable« – eine Erklärung dazu spare ich mir. Daß ein File überschreibbar ist, muß noch lange nicht bedeuten, daß es auch gelöscht werden kann oder darf. Dafür ist der letzte Eintrag im Status-Block zuständig. »deleteable« steht da, also »entfernbar«, »löschar«. Das ist sozusagen die zweite Hälfte des Software-Schreibschutzschiebers. Lautet der Eintrag »deleteable«, kann das File gelöscht werden. Ein Klick in das Feld ändert den Eintrag

Archivdienst

auf »not deleteable« und schützt das File davor, gelöscht zu werden. Seien Sie aber auf der Hut! Für Programme wie »DiskCopy« oder »Initialize« stellt das kein Hindernis dar. Unter der Eintragung »TYPE Project« steht »Size«, und hier erfahren Sie etwas über die Größe der Datei. Einmal erfolgt die Angabe in Bytes, ein andermal wird angezeigt, wie viele Blöcke auf der Diskette von diesem File belegt werden. Normalerweise ist in den genannten Feldern die Länge der Textdatei eingetragen, die von diesem »Project«-Icon aufgerufen wird. Es gibt aber einen Fall, bei dem die Dateilänge null ist. Dann nämlich, wenn mit einem »Project«-Icon nur das »Default Tool« aktiviert wird, ohne daß eine Textdatei ins Spiel kommt (ein Beispiel für diesen Fall ist das »Shell«-Icon). Dann kommt was Neues: »Stack«. Der Stack (= Stapel) ist eine Art Zwischenspeicher, in dem Daten und Adressen während des Programmablaufs abgelegt und bei Bedarf wieder abgerufen werden. Wie viele Bytes für Stapelzwecke einem Programm zugeordnet werden, wird meist vom Programmierer festgelegt. Das ist übrigens eine heikle Sache, denn im Falle, daß ein Programm mehr Stack-Bytes braucht, als dafür zur Verfügung gestellt wurden, kommt das einem Ticket der Air India gleich. Bei einigen Programmen werden Sie im Infofenster im Feld »Stack« eine Eintragung finden, bei den meisten jedoch keine. Im Feld neben »Stack«, das in unserem Beispiel frei ist, kann eine beliebige Anzahl Bytes eingetragen werden. Einfach mit dem Mauszeiger hineinklicken und dann eine Zahl eintragen – probieren Sie's doch. Lassen Sie es aber bei diesem einen

Versuch bewenden, denn zu Beginn Ihrer Amiga-Karriere sollten Sie sich darum nicht weiter kümmern. Nur noch eins: Hüten Sie sich davor, einen vorgegebenen Eintrag im »Stack«-Feld zu verkleinern. Der Programmierer, der den Eintrag auf eine bestimmte Zahl setzte, wußte schon, was sein Programm braucht.

Das Feld unter »Stack« mit der Bezeichnung »Comment« wurde dafür vorgesehen, Kommentare eintragen zu können, die dem jeweiligen File zugeordnet sein und bleiben sollen. Maximal 80 Zeichen lang darf der Kommentar sein, den Sie nach dem Anklicken des Feldes rechts von »Comment« eintragen können. Soweit zum Kommentar, auch unter der Bezeichnung »Filenote« bekannt. Wie schon beim Infofenster zu »Disk« sehen Sie auch bei »Project« einen Eintrag namens »Default Tool«. Doch erst bei »Project«-Icons bekommt dieser Eintrag so richtig Sinn, bedeutet ein Eintrag in dieser Rubrik doch nicht mehr und nicht weniger, als daß durch Anklicken dieses Icons ein von diesem Icon völlig unabhängiges Programm gestartet wird, an das nach dessen Start bestimmte Informationen übergeben werden. Ein gutes Beispiel ist das »Project«-Icon einer »Notepad«-Datei. Wenn Sie ein Notepad-File abspeichern, wird für dieses File automatisch ein »Project«-Icon angelegt, in dessen »Default Tool«-Feld nicht nur der Programmname »Notepad« steht, sondern auch der komplette Pfad zu diesem Programm. Im »TOOL TYPES«-Feld dieses »Notepad«-Datei-Icons sind dann in mehreren Zeilen weitere wichtige Informationen für das »Notepad«-Programm eingetragen, wie z.B. Schriftart, Fenstergröße etc.

Informationen en gros

Wenn in »Default Tool« der Eintrag »C:IconX« oder auch »XIcon« steht, können Sie Gift darauf nehmen, daß eine Textdatei existiert, in der eine Reihe von Kommandos stehen. Die Abarbeitung dieser Kommandofolge

muß in einem CLI-Window erfolgen, und den Start eines neuen CLI-Prozesses und die Übergabe der im Text-File stehenden Daten übernimmt das Programm »IconX« bzw. sein Vorgänger »XIcon«. Auf diese Art werden Programme für den Workbench-User zugänglich, die eigentlich nicht von der Workbench aus aufrufbar sind (sowas gibt's). Wie man das macht, also wie man ein nicht workbenchtaugliches Programm von der Workbench aus aufrufbar macht, erkläre ich Ihnen gegen Ende dieses Artikels. Auch wie Sie ein Programm mit einem Icon versehen, werden wir an einem kleinen Beispiel durchexerzieren. Hier und jetzt sei noch angemerkt, daß selbstverständlich auch der Eintrag neben »Default Tool« von Ihnen selbst beeinflußt werden kann. Einfach in das Feld klicken, ... den Rest wissen Sie schon. Kommen wir nun zur letzten Zeile, die mit »TOOL TYPES« bezeichnet ist. Lang hat's gedauert, bis ich dahinterkam, wie ich mit dieser Zeile umzugehen habe.

Das Handbuch zur Workbench 1.2 schweigt sich ja ausführlich darüber aus, die Ergänzungsbroschüre zur Workbench 1.3 erwähnt die »TOOL TYPES« nur kurz, und auch in anderen Büchern ist Information darüber Mangelware. Damit Sie nicht auch raten müssen, hier in Kürze das Wesentliche. Einträge in »TOOL TYPES« haben nur dann einen Sinn, wenn das aufgerufene Programm nach solchen Einträgen sucht. Das ist aber alles andere als selbstverständlich. »Wordperfect« z.B. sucht nach dem Programmstart nach einem Eintrag in »TOOL TYPES«, mit dem die Backup Time festgelegt wird. »Superbase« holt sich Steueranweisungen aus den »TOOL TYPES«-Einträgen, und »XIcon« kann (muß aber nicht) über die »TOOL TYPES«-Einträge veranlaßt werden, zum Öffnen eines neuen CLI-Prozesses einen bestimmten Handler zu verwenden und eine vorgewählte Fenstergröße einzustellen. Als Beispiel dafür

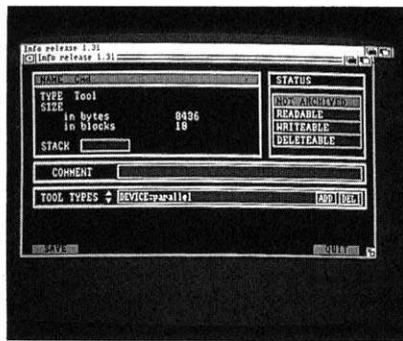


Bild 12. Informationen über ein Programm – die »Tool«-Info

mag »InstallPrinter« aus der »Utilities«-Schublade der Workbench-Disk herhalten oder auch das im Handbuch zu WB 1.3 erwähnte File »CMD« mit seinen Optionen. Für den Fall, daß mehrere solcher Einträge möglich oder gar notwendig sind (wie etwa bei »CMD«), wurden zwei kleine Scroll-Pfeile vorgesehen (gleich neben dem Schriftzug »TOOL TYPES« zu sehen). Sie blättern durch Anklicken eines der beiden Pfeile die Einträge einfach durch. Um einen Eintrag zu modifizieren, klicken Sie bitte in das Feld, in dem der Eintrag steht. Um einen Eintrag zu löschen, klicken Sie das »DEL«-Gadget ganz rechts an. Wollen Sie jedoch einen Eintrag zufügen, muß das »ADD«-Gadget angeklickt werden. Dann wird der gerade sichtbare Eintrag nach oben gescrollt, und zwischen diesem und einem etwaig folgenden Eintrag setzen Sie nun den gewünschten Zusatz. Um herauszufinden, welche Schlüsselwörter eventuell von einem Programm abgefragt werden, habe ich bis jetzt nur zwei Methoden gefunden. Leider fallen beide in die Kategorie »try and error«. Die eine ist, Übergabeparameter, die bei einem Aufruf des gewünschten Programms aus einem CLI-Fenster heraus übergeben werden können, in das »TOOL TYPES«-Feld einzutragen. Die andere, wenn auch mühsame Methode ist folgende: Sehen Sie sich das entsprechende File als ASCII-Hexdump an und probieren Sie Wörter, die Schlüsselwörter sein könnten, aus. Aber ich glaube, für den Workbench-Einsteiger ist das noch etwas zu früh. Lassen wir's dabei bewenden.

»Project« war etwas umfangreich, bedingt durch das erstmalige Auftreten von »TOOL TYPES« und anderen

Besonderheiten. Umso schneller geht's jetzt weiter. Als nächstes steht **TOOL** auf dem Programm. Ein Beispiel für ein Infofenster, das zu einem ausführbaren Programm gehört, sehen Sie in Bild 12. Abgesehen davon, daß es keinen Eintrag »Default Tool« gibt, gleicht das Bild dem von »Project« aufs Haar. Alles dort Erwähnte gilt auch für »Tool« und mehr gibt's dazu nicht zu sagen.

Nächster auf der Liste ist **Garbage**, der Mülleimer. Dessen Infofenster gleicht dem einer Schublade aufs Haar, also können wir uns gleich dem Themenkreis **Drawer** zuwenden. »Drawer« bedeutet bekanntlich »Schublade«, und das zu »Drawer« passende Infofenster holen Sie jetzt bitte auf den Bildschirm. Wie auch der Papierkorb bietet »Drawer« keine Information darüber, wie viele Files sich in der Schublade befinden oder wieviele Bytes der Inhalt der Schublade belegt. Eigentlich schade, denn das wären Informationen, an denen jeder Benutzer interessiert ist.

Der »Discard«-Befehl

Tja, und damit wären wir mit dem Menüpunkt »Info« zu Ende – hat ohnehin lange genug gedauert.

Der letzte Menüpunkt in der Rubrik »Workbench« ist **Discard**. Eine sehr freie, dafür aber zutreffende Übersetzung dieses Begriffs ist »wegwerfen«. Sie erinnern sich: Wenn Sie ein File oder Directory in den Mülleimer (»Trashcan«) werfen, ist es noch nicht endgültig für Sie verloren. Anders sieht es aus, wenn Sie ein Icon anklicken und danach »Discard« anwählen. »Discard« löscht die angeklickten Icons und die zugeordneten Files oder Directories unwiederbringlich! Das ist auch der Grund, daß die Workbench-Programmierer diesem Menüpunkt eine Sicherheitsabfrage spendiert haben. Nach dem Anwählen von »Discard« erscheint ein Programm-Requester namens »Workbench«, der Sie darauf aufmerksam macht, daß Sie im Begriff stehen, den »point of no return« zu überschreiten. Ein Klick auf »OK to

discard« verschafft Ihnen unweigerlich freien Platz auf der Diskette, ein Klick auf »forget it!« beweist, daß Vorsicht nichts mit Unentschlossenheit zu tun hat – besser ein Rückzug als ein Pyrrhus-Sieg. »Discard« läßt sich nur auf Files und Directories anwenden, nicht auf Disketten. In diesem Zusammenhang zeigt sich der Unterschied zwischen einem normalen Directory in Form einer Schublade und dem »Trashcan«-Verzeichnis. Ich habe bereits erwähnt, daß der Mülleimer eine besondere Art von Unterverzeichnis darstellt. Diese Besonderheit wird bei der Anwendung von »Discard« manifest: Bei angewähltem »Trashcan« ist »Discard« nicht aktivierbar (es bleibt bei der gerasterten Schrift im Menü).

Es ist fast nicht zu glauben – wir haben den Menüpunkt »Workbench« durchgeackert. Aber bloß keine Müdigkeit vorschützen. Gleich weiter mit **Disk**, dem zweiten Hauptmenüpunkt auf der Benutzeroberfläche. Gleich der erste Untermenüpunkt schafft Platz im Büro: **Empty Trash** leert den Papierkorb. Daß dieser Menüpunkt nur dann aktivierbar ist, wenn zuvor das »Trashcan«-Icon angeklickt wurde, ist logisch.

Zweiter Submenüpunkt innerhalb »Disk« ist **Initialize**. Mit diesem Kommando machen Sie eine leere Diskette Amiga-DOS-verträglich bzw. löschen Sie alle Daten von einer beschriebenen Diskette. »Initialize« formatiert eine Diskette, schreibt also die für Amiga-DOS wichtigen Spur- und Sektorinformationen auf die Magnetschichten. Machen wir doch gleich mal. Legen Sie eine jener neuen Disketten, die Ihnen der Amiga-Händler noch schnell unter den linken Arm schob, in das Laufwerk ein. Das Icon mit der Untertitelung »df0:BAD« kennen Sie ja schon. Aktivieren Sie dieses Icon mit einem Mausklick und wählen Sie danach aus dem »Disk«-Menü den Untermenüpunkt »Initialize«. Daß allmählich der Wunsch nach einem zweiten Laufwerk in Ihnen wach wird, kann ich gut verstehen. Es hilft aber nichts, Sie müssen der Aufforderung »Plea-

se insert disk Workbench in any drive« Folge leisten. »Initialize« braucht nämlich das Programm »Format« aus dem »System«-Drawer der Workbench-Diskette. Es steht Ihnen zwar frei, den »Initialize«-Vorgang durch Anklicken des »Cancel«-Gadgets abubrechen, aber das wäre nicht in unserem Sinne (zumindest jetzt nicht). Nachdem das »Format«-Programm geladen wurde, werden Sie aufgefordert, die zu initialisierende Diskette wieder in jenes Laufwerk einzulegen, in der sie zum Zeitpunkt des Klicks auf das Icon war. In unserem Falle ist das das interne Laufwerk, und deshalb steht im Requester »df0:«. Weil »Initialize« ebenso endgültig eventuell auf der Zieldiskette vorhandene Daten über

Das Formatieren

den Jordan schickt wie »Discard«, blendet die Workbench auch hier einen Requester ein, der Sie auf die Konsequenzen Ihres Vorhabens hinweist. Mit einem Klick auf »Continue« übernehmen Sie die volle Verantwortung für Ihr Treiben und geben dem Formatierprogramm grünes Licht. Während der Formatierungsprozeß läuft, werden Sie in einem Ausgabe-fenster über den aktuellen Stand der Dinge informiert. Dieses Ausgabe-fenster ähnelt in Inhalt und Aussehen jenem von »DiskCopy«, mit dem Unterschied, daß bei »Initialize« der Lesevorgang entfällt und daß auch bei älteren Versionen ein Überprüfungszyklus (»verify«) durchgeführt wird. Nach Abschluß des »Initialize«-Prozesses erhält die Diskette einen Namen, denn namenlose Disketten gibt es bei Amiga-DOS nicht. Na, welchen Namen wird die Diskette wohl verliehen bekommen? Ganz klar, »Empty« wird sie vom System getauft. Sinnig, wie? Sollten Sie das Spielchen gleich noch an einer zweiten Diskette ausprobieren wollen, steht dem natürlich nichts im Wege. Sie haben dann allerdings zwei Disketten mit Namen »Empty«, denn im Gegensatz zu »Duplicate« oder »DiskCopy« wird bei »Initialize« auf eine Durchnummerierung verzichtet. Machen Sie sich darüber keine Sorgen – Amiga-DOS kommt mit zwei Disketten gleichen Namens zurecht. Sie tricksen das Betriebssystem nur dann aus, wenn Sie mit einem Ko-

pierprogramm à la »Select-Copy« Disketten duplizieren, also identische Kopien herstellen. Sind dann Original und Kopie gleichzeitig auf der Workbench präsent, gibt's Probleme, und mit ein bißchen Pech sind nach einem Schreibzugriff Original und Kopie im Eimer.

Auf zum letzten Gefecht – **Special** steht auf dem Programm. Dieser Menüpunkt in der Workbench-Menüleiste birgt einige praktische Hilfsmittel für den Umgang mit der Benutzeroberfläche und den zugehörigen Icons und Fenstern. Klappen Sie das Menü einmal heraus (rechte Maustaste, Mauszeiger über »Special« in der Menüleiste positionieren). Gleich der erste Unterpunkt namens »Clean Up« ist gerastert dargestellt. Wenn Sie daraus messerscharf schließen, daß zuerst ein Icon aktiviert werden muß, ehe mit diesem Punkt etwas anzufangen ist, liegen Sie beinahe richtig. »Clean Up« ist ein Befehl, der nur auf den Inhalt von Windows anwendbar ist. Öffnen Sie doch bitte das Fenster mit dem Hauptverzeichnis der »Workbench«-Diskette durch einen Doppelklick auf deren Diskettensymbol. Wie Sie sehen, stehen alle Icons schön und sauber in Reihe und Glied, und die Größe des Fensters erlaubt, alle Icons zu sehen. Schnappen Sie sich das Größen-Gadget, und ändern Sie das Fensterformat damit von einem liegenden Rechteck in ein stehendes.

Bitte aufräumen

Ziehen Sie den unteren Rand des Fensters ruhig bis zum unteren Rand des Bildschirms – weiter geht's ohnehin nicht. Jetzt sollten nur noch die Icons der Directories »Utilities«, »System« und »Prefs« sowie das Project-Icon »Shell« zu sehen sein. Daß das aber nicht alles ist, wissen nicht nur Sie, sondern auch die Workbench. Mit »Clean Up« werden die Icons eines Fensters so umgruppiert, daß sie erstens geordnet dargestellt werden (also schön unter- und nebeneinander) und daß zweitens möglichst vie-

le Icons im zur Verfügung stehenden Fensterrahmen gezeigt werden. »Clean Up« erspart Ihnen also die Mühe, nach einer Größenänderung eines Windows alle Icons einzeln neu plazieren zu müssen. Bei der Neuordnung achtet »Clean Up« allerdings nur auf die Icon-Abmessungen und nicht auf die Namenslänge. So kann es durchaus vorkommen, daß nach Aktivieren dieser Funktion zwar alle Icons wie die Zinnsoldaten ausgerichtet sind, sich die File-Namen unterhalb der Icons aber überlagern. Und noch etwas: Wundern Sie sich nicht, wenn »Clean Up« manchmal nur gerastert dargestellt wird, obwohl genug Fenster geöffnet sind, in denen sich »Clean Up« austoben könnte. Der Befehl steht nur dann zur Verfügung, wenn das Schubladen-Icon aktiviert ist, zu dem das Fenster gehört, das Sie aufräumen wollen. Falls es sich um das Fenster mit dem Namen der Diskette in der Titelleiste handelt, muß das Disketten-Icon aktiviert sein. Schließen Sie jetzt bitte das Window durch einen Klick auf das »Close«-Gadget oder durch Anwahl von »Close« im Workbench-Menü, und öffnen Sie es gleich wieder. Wie Sie sehen, ist zwar die Fenstergröße dieselbe geblieben, die Icons stehen jetzt aber wieder so, wie sie vor »Clean Up« standen. Die neue Position der Icons wurde also nicht gespeichert! Dazu bedarf es nämlich mehr als des Befehls »Clean Up«.

Der nächste Befehl in der Kette der Spezialisten ist **Last Error**. Damit lassen Sie sich den zuletzt aufgetretenen Fehler in der Titelzeile des Workbench-Bildschirms anzeigen. Machen wir gemeinsam einen Versuch. Setzen Sie den Schreibschuttschieber Ihrer Workbench-Diskette auf schreibgeschützt, und versuchen Sie, den »Trashcan«-Ordner umzubenennen (falls Sie es vergessen haben – dazu verwendet man »Rename« aus dem »Workbench«-Menü). Den Requester »Disk is write protected« quittieren Sie mit »Cancel«. Hat Sie das Aufblitzen des Bild-

schirms erschreckt? Sollte es aber nicht. Das war nur ein Hinweis des Betriebssystems, daß Sie einen Fehler provoziert haben. Welcher Fehler das war, wissen Sie im konkreten Fall zwar ganz genau, er steht in leicht kryptischer Form nun aber auch in der Titelleiste des Workbench-Bildschirms: »Error while moving Trashcan: 214«. Diese Fehlermeldung wird bei der nächsten Betätigung einer Maustaste durch die Standardtitelleiste überschrieben, und Leute mit Kurzzeitgedächtnis haben dann Probleme, sich an den letzten Fehler und dessen Ursache, bzw. die Fehlernummer, zu erinnern (der Fehlercode »214« bedeutet, daß ein Schreibzugriff auf eine Diskette verlangt wurde, die schreibgeschützt war). Die Anwahl von »Last Error« holt die jeweils letzte Fehlermeldung neuerlich in die Titelleiste zurück – so oft Sie wollen. Es wird aber immer nur die letzte Fehlermeldung angezeigt. Zurückblättern ist nicht.

Ein Zeichenbefehl

Wenden wir uns kurz dem nächsten Menüpunkt zu. Es ist dies **Redraw** und meiner Meinung nach der Gott sei Dank unnötigste aller Workbench-Menüpunkte. Es scheint, als wäre er ein Überbleibsel aus jenen schönen Amiga-Pioniertagen, in denen Fenstermanipulationen immer wieder dazu führten, daß der Inhalt eines Fensters nicht automatisch aufgefrischt wurde. Es ist mir jedenfalls noch nie gelungen, mit standardmäßigen Workbench-Operationen den Bildschirminhalt dermaßen durcheinanderzubringen, daß ich ihn mit »Redraw« hätte auffrischen müssen. Das ist nämlich der Sinn von »Redraw«. Hoffentlich komme ich nicht doch noch in die Verlegenheit, diesen Befehl zu brauchen.

Der nächste Menüpunkt ist da ungleich wichtiger für den Benutzer: **Snapshot**. Es ist dies jenes Kommando, das Änderungen von Fenstergrößen und -positionen ebenso wie Icon-Verschiebungen dauerhaft speichert. Sie erinnern sich sicher an die Sache mit »Clean Up«. Sie meinen, daß mir hier ein Irrtum unterlaufen sei? Weil doch zumindest Fenstergröße und -position sehr wohl nach dem Schließen und nochmaligem Öffnen des Workbench Win-

dows dem geänderten Stand entsprachen? Pustekuchen! Das sieht nur so aus. Spätestens nach dem nächsten »Reboot«, also dem nächsten Hochfahren des Systems, sieht das Window wieder genauso aus wie auf der Originaldiskette. Um solche Änderungen dauerhaft zu speichern, bedarf es der Anwendung des »Snapshot«-Befehls.

Schnappschüsse

Dieser wirkt wie gesagt auf die Position von Icons aller Art ebenso wie auf die Position und die Größe von Fenstern. Allerdings sind dabei einige Bedingungen zu beachten. Wollen Sie, daß in Zukunft das Diskettensymbol der Workbench-Disk in der Mitte des Bildschirms angezeigt wird, müssen Sie das Icon an die gewünschte Stelle ziehen und danach den »Snapshot«-Befehl starten. Am Aufleuchten der Laufwerks-LED (und an mehr oder weniger starker Geräuschentwicklung des Laufwerks) erkennen Sie, daß die neuen Positionsdaten auf der Diskette eingetragen werden. Wichtig dabei ist, daß zum Zeitpunkt der »Snapshot«-Aktivierung das Diskettensymbol aktiviert ist. Das gilt auch für die Verewigung von Größe und Position des Hauptverzeichnis-Windows. Ganz allgemein formuliert: Um eine geänderte Dimension und/oder Position eines Windows abspeichern zu können, muß das dem Fenster zugehörige Schubladen-Icon aktiviert sein. Was das Hauptverzeichnis betrifft, so stellt das Disketten-Icon die Schublade dar. Daß ein Window aktiv ist (erkennbar an der voll gezeichneten Schrift) genügt nicht! Für das Abspeichern von Icon-Positionen innerhalb eines Fensters gilt, daß nur jene Icons von »Snapshot« berücksichtigt werden, die zum Zeitpunkt der Anwahl von »Snapshot« aktiviert sind. Eines können Sie allerdings nicht: die Position eines File- oder Directory-Icons außerhalb eines Windows, also direkt auf der Workbench, fixieren. Das funktioniert erst unter Workbench 2.0, wie sie mit dem Amiga 3000 ausgeliefert wird. An

dieser Stelle möchte ich Ihnen einen Tip geben. Sie erinnern sich, daß man Files und Directories einfach dadurch von einem Ordner in den anderen transferieren kann, indem man die gewünschten Icons bei gedrückter <SHIFT>-Taste anklickt (Mehrfachselektion) und dann in das Ziel-Window verschiebt. Ähnlich funktioniert das auch bei »Snapshot«. Haben Sie mehrere Icons verschoben und wollen deren neue Position abspeichern, geht das auf einen Schlag: mittels Mehrfachselektion die gewünschten Icons anklicken und dann »Snapshot« auswählen. Dabei ist allerdings Vorsicht geboten! Ich weiß nicht, ob es sich dabei um eine Meise im System handelt oder ob folgendes unvermeidlich ist. Nehmen wir an, Sie haben zwei Fenster geöffnet (ich spreche bewußt von einer Annahme - bitte versuchen Sie nicht, das Gesagte nachzuvollziehen). In beiden Fen-

Der Trick Mehrfachselektieren

stern stehen Icons, deren Position Sie geändert haben, und Sie wollen diese neuen Positionen abspeichern. Als erfahrener Workbench-User benutzen Sie die Möglichkeit der Mehrfachselektion. Zuerst werden die Icons des ersten Fensters angeklickt und dann jene des zweiten. Da passiert es: Sie bewegen die Maus, während die linke Maustaste gedrückt ist. Alle aktivierten Icons folgen als Duplikat den Bewegungen des Mauszeigers! Nicht so schlimm, denken Sie, und führen die Icons brav wieder an ihre Ursprungsposition. Wenn Sie jetzt die linke Maustaste loslassen (und irgendwann werden Sie das wohl müssen), werden die Files des ersten Fensters unbarmherzig in das zweite Fenster verschoben - Sie sind im File-Transfer-Modus gelandet! Besonders delikate wird dieses Fehlverhalten dadurch, daß die Position der Files im zweiten Window absolut betrachtet jener auf dem Workbench-Screen entspricht, d.h., daß sie im

zweiten Fenster außerhalb des sichtbaren Bereichs eingetragen werden. Die Scroll-Balken des zweiten Windows ändern zwar die Position, aber das ist nicht unbedingt auffällig. So manches unschöne Wort schlüpfte mir da schon durch die Lippen...

Bleibe noch zu guter Letzt die Workbench-Funktion **Version**. Nach Anwahl dieses Menüpunktes werden in der Titelleiste des Workbench-Bildschirms die Kickstart- und die Workbench-Versionsnummer angezeigt. Lassen Sie sich von den unerwartet großen Zahlen und den vielen Kommastellen nicht beeindrucken. Commodore als professionelle Software-Schmiede dokumentiert und registriert natürlich jede freigegebene Änderung der Software, selbst wenn diese Änderung noch so minimal ist. Betrachten Sie Zahlen wie »V1.2« und »V1.3.2« als grobe Indikation für den Aktualitätsgrad der Betriebssystem-Software. Detaillierte Unterschiede erkennen Sie nur an den Versionsnummern. Versionsnummer 34.28 gehört zu V1.3.2. Und weil Kickstart und Workbench voneinander relativ unabhängig sind, haben beide separate Versionsnummern bekommen.

Soweit zu den Workbench-Menüs. Prinzipiell haben wir gemeinsam alle Fähigkeiten der Workbench durchgespielt. Immer benutzten wir vorhandene Icons, um die gewünschten Aktionen auszulösen. Aber was ist mit Programmen, die noch kein Icon haben. Wie aktiviert man Programme von der Workbench aus, die dafür nicht programmiert wurden? Wie kann man das besonders schöne Icon von Programm A auch für Programm B verwenden? Die Beschäftigung damit soll den Abschluß dieses Artikels darstellen.

Beginnen wir mit dem einfachsten Fall. Auf der Diskette »Werner« befindet sich im Unterverzeichnis »Georg« das File »Karl«, dessen Icon Ihnen viel besser gefällt als jenes, das dem File »Gabi« auf der Diskette »Susi« zugeordnet ist. Zunächst überprüfen Sie, ob beide Icons vom gleichen Typ sind, also »Tool«, »Project« oder »Drawer«. Das geht am besten mit dem »Info«-Befehl. Wenn das File »Karl« bei »TYPE« den Eintrag »Tool« hat und der »Info«-Befehl beim File »Gabi« ein »Project« anzeigt, können

Sie sich schon mal nach einem anderen Icon umsehen. Sind die beiden Einträge gleich, also steht bei »Karl« wie bei »Gabi« der Eintrag »Tool« (oder »Project« oder...), haben Sie die erste Hürde schon genommen (ich hoffe, Sie entschuldigen, daß ich in diesem fortgeschrittenen Stadium nicht mehr jeden Diskettenwechsel oder Requester erwähne). Jetzt müssen Sie nur noch das häßliche Icon durch das gewünschte ersetzen. Zwei Wege bieten sich Ihnen für dieses Vorhaben an. Der erste Weg führt über das Programm »IconEd«, das Sie im Ordner »Tools« auf der »Extras1.3D«-Diskette finden.

Icons nach Maß

»IconEd« ist ein Editor für Icons, ein Werkzeug also, mit dem Sie Icons nach Ihrem eigenen Geschmack anfertigen. Eine Art Malprogramm, spartanisch ausgestattet und extrem unkomfortabel. Um aber auf Workbench-Ebene, also ohne den CLI in Anspruch zu nehmen, einem File ein anderes Icon zu verpassen, ist »IconEd« die beste Wahl. Starten Sie »IconEd« durch einen Doppelklick. Das nach dem Programmstart eingeblendete Informationsfeld schließen Sie mit einem Klick auf das »OK«-Gadget. Drücken Sie die rechte Maustaste, um das Menü von »IconEd« zu aktivieren. Selektieren Sie das Menüfeld »Disk« und wählen Sie den Unterpunkt »Load Data«. Nach Loslassen der rechten Maustaste erscheint in der rechten Hälfte des »Icon Editor«-Fensters ein blau unterlegtes Feld. Das schwarz unterlegte Feld, in dem »Name« steht, klicken Sie mit dem Mauszeiger an. Mit der -Taste löschen Sie »Name« und tragen dann ein:

Werner:Georg/Karl

Schließen Sie die Eingabe mit der <RETURN>-Taste ab und klicken Sie das Feld mit der Inschrift »Load Icon Image« an. Die Daten, die das Icon der Datei »Karl« ausmachen, werden in den Speicher geholt und das Icon selbst im Window des »IconEd« zweimal angezeigt. Aktivie-

ren Sie jetzt den Menüpunkt »Disk«/»Save Data«. Die rechte Hälfte des »IconEd«-Windows bekommt starke Ähnlichkeit mit jener, die für den Ladevorgang zuständig war. Tragen Sie in das Namensfeld Pfad und Name der Zieldatei ein, in unserem Beispiel also:

```
SUSI:Gabi
```

Abschluß wieder mit <RETURN> und der Einfachheit halber Anklicken von »Save Full Image«. Das Icon wird jetzt dem File »Gabi« auf der Diskette »Susi« zugewiesen. Das Original-Icon wird vom neuen Icon überschrieben. Wenn Sie das Icon der Diskette »Susi« zweimal anklicken, werden Sie im Root-Window das neue Icon vorfinden, untertitelt mit dem Programmnamen »Gabi«. War aber das Root-Fenster die ganze Zeit während Ihrer Icon-Manipulationen geöffnet, werden Sie zunächst keinen Unterschied feststellen. Erst nach dem nächsten Öffnen dieses Fensters sehen Sie das neue Icon auf dem Bildschirm. Ich muß Sie aber darauf hinweisen, daß »IconEd« nicht mit jedem Icon umzugehen in der Lage ist, sondern nur mit normalen Icons, die dem Workbench-Standard entsprechen. Sog. »Giant Icons«, übergroße Piktogramme also, werden von »IconEd« nur unvollständig verarbeitet und ebenso unvollständig abgespeichert. Die »Riesen« sind aber meist die schönsten, und genau so ein Icon haben Sie sich für das File »Gabi« in den Kopf gesetzt. Was ist zu tun? Wenn »IconEd« an den Grenzen seiner Leistungsfähigkeit angerannt ist, bleibt nur noch der Weg über den »CLI«. Tut mir wahnsinnig leid, aber das kann ich Ihnen in diesem Fall nicht ersparen. Im Hauptverzeichnis der Workbench-Diskette, dem »Root«-Window mit der Titelzeile »Workbench1.3D«, finden Sie ein Icon namens »Shell«. Klicken Sie dieses Icon zweimal an, um das zugehörige Programm zu starten. Es öffnet sich ein neues, längliches Window mit Namen »AmigaShell«. Dies ist die Kommunikationsschnittstelle namens »CLI«. Ich möchte hier und

jetzt gar nicht näher auf diese Schnittstelle eingehen. Wie gesagt, auf Seite 34 in diesem Heft gibt es einen sehr ausführlichen Artikel zum Thema »CLI«. Tippen Sie einfach folgende Zeile ein :

```
copy Werner:Georg/Karl.info  
to SUSI:Gabi.info
```

Schließen Sie auch diese Zeile wieder mit einem Druck auf die <RETURN>-Taste ab. Danach kopiert der Amiga die Datei »Karl.info« auf die Diskette »SUSI« und gibt dort der kopierten Datei den Namen »Gabi.info«. Mit dieser Namensgebung wird das Icon, das im Original zur Datei »Karl« gehörte, zuständig für das File »Gabi«. Durch die Änderung des Namens der ».info«-Datei hat sich aber absolut nichts am Aussehen des Icons oder gar an dessen Typ geändert. Es genügt also, das gewünschte ».info«-File zu kopieren und umzubenennen, schon hat man für ein Workbench-Programm ein neues Icon. Das alte Icon bzw. die alte Datei namens »Gabi.info« wird durch den Kopiervorgang von Amiga-DOS automatisch gelöscht.

Kein Close-Gadget

Um das Fenster mit dem Namen »AmigaShell« wieder wegzukriegen, tippen Sie bitte noch ein:

```
ENDCLI
```

und drücken wieder <RETURN>. Damit wird das Fenster geschlossen (ist Ihnen aufgefallen, daß es kein »close«-Gadget hatte?). War das nicht viel einfacher als der Weg über »IconEd«? Noch dazu, wo der CLI jedes File kopieren kann, ganz gleich, welche Daten darin eingetragen sind. Wenn Sie einer Diskette ein neues Icon geben wollen, gehen Sie ganz ähnlich vor. Der Unterschied ist lediglich, daß die Daten des Disketten-Icons in der Datei »disk.info« abgelegt sind. Eines sollten Sie aber nicht vergessen: Im ».info«-File sind nicht nur die Bild-, sondern auch die Positionsdaten des Icons gespeichert, was bedeutet, daß Sie höchstwahrscheinlich um eine Neupositionierung und die Anwendung des »Snapshot«-Kommandos nicht herumkommen.

Soweit zum Ersatz eines vorhandenen Icons durch ein anderes. Was aber tun, wenn ein Programm kein

Icon hat, Sie diesem File aber unbedingt ein solches geben möchten? Wie versehen wir eine iconlose Diskette, ein Unterverzeichnis mit einem Piktogramm? Auch keine Hexerei – mit ein bißchen Sorgfalt »Null Problem«. Ich weiß schon, im Normalfall werden Sie als Workbench-Benutzer das Vorhandensein eines Files ohne Icon gar nicht bemerken. Doch spätestens, wenn Sie einen CLI-Ersatz à la »CLImate« oder »DiskMaster« verwenden und dabei entdecken, wie viele bislang unbekannte Programme und Unterverzeichnisse sich auf Ihren Disketten befinden, wird der Wunsch in Ihnen wach werden, das eine oder andere Programm von der Workbench zu starten. Beginnen wir wieder mit einem einfachen Beispiel. Ein Unterverzeichnis ohne Icon verwehrt Ihnen auf Workbench-Ebene jeden Zugriff auf Dateien innerhalb dieses Unterverzeichnisses, selbst wenn diese Dateien mit Icons versehen sind – Sie können die Schublade nicht öffnen. Alles was Sie brauchen, um ein solches »verstecktes« Unterverzeichnis mit dem Doppelklick öffnen zu können, ist ein Schublade-Icon beliebigen Namens (das »Empty«-Icon tut's auch) und die Kenntnis des exakten Namens des Unterverzeichnisses. Mit diesen Kenntnissen bewaffnet verfahren Sie wie im Beispiel »Icon-Ersatz« beschrieben. Etwas diffiziler wird es, wenn es um ein »Executable«, ein ausführbares Programm, geht. Hier bewegen Sie sich insofern auf etwas unsicherem Boden, als nicht jedes Programm workbenchtauglich programmiert wurde, dies aber nicht »von außen« erkennbar ist. Da hilft nur die bewährte Mal-schauen-was-passiert-Methode. Wichtig ist, daß Sie ein »Tool«-Icon verwenden, also bitte mit dem »info«-Befehl aus der Workbench-Menüleiste überprüfen. Der Rest funktioniert wieder wie im Beispiel »Icon-Ersatz« beschrieben. Wenn nach dieser Aktion das Programm sich immer noch nicht starten läßt, müssen Sie den Weg des »Project«-Icons

wählen. Ein solches Icon wäre z.B. das »Shell«-Icon von der Workbench-Diskette. Nur geht das nicht so einfach wie im Fall des »Shell«-Aufrufs. Sie brauchen zum Start eines für einen Workbench-Start ungeeigneten Programms ein Script-File und das Programm »IconX«, das sich im »C«-Verzeichnis der Workbench-Diskette befindet. Was halten Sie vom Versuch, den CLI-Befehl »dir« (»Directory« - »Inhaltsverzeichnis«) von der Workbench aus zu aktivieren. Verwenden wir die RAM-Disk für dieses Unterfangen, dann bedarf es keines Schreibzugriffs auf die Workbench-Diskette. Öffnen Sie bitte mittels Doppelklick auf das »Shell«-Icon im Hauptverzeichnis der Workbench-Diskette ein CLI-Fenster (an dieser Stelle wollen Sie mir bitte nähere Erklärungen zum CLI und seinen Befehlen ersparen). Dann tippen Sie bitte ein:

```
copy shell.info to ram:
Versuch.info
```

Daß solche Eingaben immer mit einem <RETURN> abzuschließen sind, erwähne ich hier das letzte Mal. Dann geben Sie ein:

```
copy c:dir to ram:
```

Nun müssen Sie ein Script-File, also eine Textdatei mit einer Kommandoliste, schreiben. Verwenden wir den Commodore-Texteditor »Ed« - das »Notepad« ist für solche Dinge ungeeignet.

```
ed ram:Versuch
```

Nachdem sich das Window von »Ed« geöffnet hat, geben Sie bitte ein:

```
ram:dir SYS: all
```

Um diese Zeile abzuspeichern (und zwar im File »Versuch« auf der RAM-Disk), drücken Sie bitte zuerst die <ESCAPE>-Taste, dann <x> und dann <RETURN>. Blicke noch das Schließen des aktuellen CLI-Fensters mit

```
endCLI
```

Da Sie beim Aufruf des Texteditors »Ed« den Zusatz »ram:Versuch« angaben, befindet sich jetzt ein File namens »Versuch«, das nur die Zeile

»dir SYS: all« enthält, in der RAM-Disk. Was jetzt noch fehlt, ist der Eintrag »IconX« im »Default Tool«-Feld des Icons »Versuch«. Icon »Versuch« einmal anklicken, aus dem Workbench-Menü »Info« anwählen, im Infofenster den »Default Tool«-Eintrag »SYS:Utilities/DiskCopy« löschen und durch »SYS:c/IconX« ersetzen. »Save« anklicken. Wenn Sie jetzt das Icon »Versuch« anklicken, wird das Hilfsprogramm »IconX« geladen und gestartet, das in der Schublade »C« auf der Workbench-Diskette abgelegt ist. Dieses Programm öffnet ein CLI-Fenster und führt dann den nächsten (in unserem Fall einzigen) Befehl der Befehlsdatei mit Namen »Versuch« aus. Mit dem Kommando

Geheimtip »Iconx«

»ram:dir SYS: all« veranlassen Sie die Auflistung aller auf der Workbench-Diskette eingetragenen Files und Directories in jenem Fenster, das von »IconX« geöffnet wurde. Nach dem letzten File wird das CLI-Fenster dann automatisch geschlossen, und die Workbench ist wieder grafisch orientiert. Daß der »dir«-Befehl zuvor in die RAM-Disk kopiert wurde, hat nur Beispiel-Charakter. Ebensogut hätten Sie den »Dir«-Befehl direkt aus dem »C«-Verzeichnis der Boot-Diskette verwenden können. Das wäre also der Weg, mittels »Project«-Icon, Textdatei und »IconX« ein für den Workbench-Betrieb ungeeignetes Programm doch noch workbenchfähig zu machen.

Zu »IconEd« möchte ich Ihnen noch einen kleinen Trick verraten. Mit diesem Programm können Sie nicht nur Icons anfertigen oder deren Aussehen verändern, Sie können auch deren Typ ändern! Möchten Sie das Mülleimer-Piktogramm als Symbol für einen Ordner verwenden? Kein Problem. Laden Sie zuerst das Icon einer Schublade in Feld 1. Dann den Mülleimer in Feld 2. Nun wählen Sie in der Menüleiste von »IconEd« den Punkt »Copy/Exchange with Frame/1«. Die beiden Bilder haben jetzt ihre Plätze getauscht - aber auch ihre Typzuweisungen! Wenn Sie die Icons jetzt abspeichern, haben Sie ein Icon mit dem Aussehen des Mülleimers, aber den Attributen eines Directory-Icons und natürlich umgekehrt auch ein Icon mit dem

Aussehen einer Schublade, aber mit den Attributen von »Trashcan« (mit allen Konsequenzen).

Noch nicht besprochen habe ich das Icon rechts oben - die **RAM DISK**. Es ist dies eine Art »elektronische Diskette«, die vom Betriebssystem simuliert wird, indem ein bestimmter Speicherbereich vom RAM Ihres Amigas abgezwickelt und für die RAM-Disk »zweckentfremdet« wird. Der Vorteil einer solchen RAM-Disk gegenüber einer herkömmlichen Diskette ist der wesentlich schnellere Datenaustausch. Sie bedienen die RAM-Disk wie eine normale Diskette. Der Haken ist nur der, daß nach einem Reset des Computers (nach einer GURU-Meldung oder auch nach <CTRL>-<Amiga-links>-<Amiga-rechts>) alle in der RAM-Disk gespeicherten Informationen verloren sind. Natürlich erst recht nach einem Kaltstart, also wenn Sie die Finger nicht vom Netzschalter lassen konnten. Sollte Ihnen das dürrtige Standard-Icon für die RAM-Disk nicht gefallen, möchte ich Sie darauf hinweisen, daß wir Ihnen im Artikel über die »Startup-Sequence« einen Weg zeigen, wie Sie nach jedem Start Ihr eigenes RAM-Disk-Icon präsentiert bekommen können. Und noch was. Wundern Sie sich nicht, daß der Füllstandsanzeiger der RAM-Disk immer »voll« anzeigt. Im Gegensatz zur »resetfesten« RAM-Disk nimmt sich die RAM-Disk immer nur soviel Speicher, wie sie braucht. Eine gewöhnliche Floppy hat eine feste Speicherkapazität - wird davon nichts in Anspruch genommen, wird also nichts auf diese Diskette gespeichert, so ist die Diskette eben leer, was sich in einem entsprechenden »Pegelstand« auswirkt. Die Kapazität der RAM-Disk wird nur durch den zur Verfügung stehenden freien RAM-Speicher begrenzt.

Das war ein Ausflug in die Welt der Icons und Menüleisten. Ich hoffe, es hat Ihnen Spaß gemacht. Viel Freude mit der Amiga-Workbench! *pe*

Was bedeutet »CLI«? Zunächst die beiden geläufigen Schreibweisen: »Command Line Interpreter« ist die eine, »Command Line Interface« die andere Art, den Begriff »CLI« zu umschreiben. Doch ganz gleich, ob »Interpreter« oder »Interface«, die Betonung liegt auf »Command Line«, »Kommandozeile«.

Als der Amiga dem staunenden Computer-Volk präsentiert wurde, gab es zwei Dinge, die ihn deutlich von anderen Computern seiner Klasse unterschieden. Zum einen waren es die erstaunlichen Fähigkeiten seiner Hardware, zum anderen die benutzerfreundliche Arbeitsoberfläche namens »Intuition«, auch »Workbench« genannt. Für den Computerneuling war gerade diese ein wahrer Segen – anstatt mühsam Kommandos und deren Syntax erlernen zu müssen, genügte plötzlich ein »Klick-Klick«, und ab ging die Post. Doch läßt sich eines nicht verheimlichen: Nicht alles, was der Amiga kann, läßt sich von der Workbench aus erledigen. Zwar wird die Workbench immer komfortabler und funktionsreicher, den vollen Zugriff aber erhalten Sie eben nur über das CLI.

Wie in der guten alten Zeit der Heimcomputer kommunizieren Sie im CLI mit dem Computer per Tastatur. Ehe Sie sich jetzt mit Grausen abwenden, sollten Sie bedenken, daß selbst bei topaktuellen Betriebssystemen wie etwa »Unix« oberhalb eines bestimmten User-Levels die Befehlseingabe über die Tastatur allem anderen vorgezogen wird und kein professioneller Programmierer oder System-Operator auf diese Art der Kommunikation mit seinem Computer verzichten möchte. Die Zeiten haben sich eben gewandelt. War die Befehlseingabe per Tastatur früher der Normalfall und sozusagen das Los aller Computerbenutzer, so ist diese Art der Mensch-Computer-Kommunikation heute eher die Domäne der »Insider«, während komfor-

CLI - der Hausherr im Amiga

Die Workbench ist die bequemste Art, mit dem Amiga zu arbeiten. Auch kommt Sie dem Einsteiger mit der leicht bedienbaren Oberfläche sehr entgegen. Jedoch die wahre Power aber steckt im CLI.

table Benutzeroberflächen à la »Intuition« für die »Nur-User« konzipiert sind. Der Kenner spricht die Sprache seines Computers wie seine eigene. Als logische Konsequenz ergibt sich, daß ein Befehl aus tausend anderen viel schneller im Gehirn gefunden wird als auf einer symbolorientierten Benutzeroberfläche.

Wie funktioniert diese »Freak«-Kommunikation? Im Prinzip wie beim Militär: Einer befiehlt, der Rest folgt. Sie sind der Kompaniechef, das CLI ist der Feldwebel, der Amiga die Truppe. Ihre Befehle geben Sie über den Feldwebel an die Truppe weiter.

Sprachprobleme

Also lassen Sie den Feldwebel zum Befehlsempfang antreten. Auf dem Amiga geschieht dies dadurch, daß Sie auf der Workbench das »CLI«-Icon im »System«-Ordner mit einem Doppelklick aktivieren. Absichtlich wird hier das »CLI«-Icon angesprochen und nicht die »Shell« – erst die Arbeit, dann das Vergnügen.

Wenn nicht gerade alles schiefgegangen ist, müßte sich jetzt auf Ihrem Monitor ein neues Window geöffnet haben. In der Titelleiste dieses Windows finden Sie den Eintrag »New CLI Window«. Innerhalb des Fensters steht »1 >«, und neben dem »>« sehen Sie einen Cursor. Dieses neue Window umrahmt jenen Bereich, in dem die kommandozeilenorientierte Kommunikation zwischen

Ihnen und dem Amiga erfolgt. Der Cursor markiert den Beginn der Eingabezeile, der »Command Line«. Wie Sie aus den beiden Namensgebungen für »CLI« erkennen konnten, wird bei der Arbeit mit dem CLI der Inhalt einer Kommandozeile von eben diesem CLI interpretiert und dann ausgeführt. Trennen Sie sich für die nächsten Minuten von Ihrer heißgeliebten Maus und tippen Sie irgendetwas auf der Tastatur, vielleicht:

```
zeige mir alle Dateien
```

Drücken Sie danach die <RETURN>-Taste. Das Ergebnis ist ernüchternd:

```
Unknown command zeige
```

So geht's also nicht. Umgelegt auf unser Beispiel mit der Kompanie ist das so, als wollten Sie einem bayerischen Feldwebel einen Befehl in Finnisch erteilen – er versteht ihn nicht. Es ist also wichtig, eine gemeinsame Sprache zu sprechen. Im Computerjargon heißt das »die richtige Syntax einhalten« und Wörter verwenden, die beiden Seiten, dem User wie auch dem Computer, bekannt sind.

Ein Computer versteht nur das, was ihm beigebracht wurde, und so freie Wortschöpfungen wie oben sind im Amiga nicht vorgesehen. Sie werden noch sehen, daß auch das keine Hexerei ist – mit dem CLI können Sie Ihren Amiga solche Dinge lehren.

Welche Wörter versteht der Amiga? Aus welchen Kommandos setzt sich sein »Sprachschatz« zusammen? Ihr Amiga akzeptiert jedes Wort als gültigen Befehl, das er in Form eines Programms innerhalb des vorgegebenen Suchpfades findet. Wenn Sie jetzt nur noch »Bahnhof« verstehen, wundert mich das gar nicht. Bleiben wir doch bei dem Beispiel mit der Sprache. Ein Berliner, der sehr gut Englisch spricht, beherrscht im Prinzip drei Sprachen: Berliner Dialekt, Hochdeutsch und Englisch. Versuchen Sie sich mal den Ablauf des Verstehens eines gesprochenen (gehörten) Wortes folgendermaßen und vollkommen unwissenschaftlich vorzustellen: Zuerst wird der aktuell gesprochene Wortschatz, vielleicht der Dialekt, durchforstet. Findet sich dort das gehörte Wort nicht, schaltet der Denkprozeß um auf »Hochdeutsch« und sucht dort weiter. Wird auch da nichts gefunden, kommt die Speicherbank »englische Wörter« zum Einsatz. Und ist auch dort Ebbe, ist die zu erwartende Reaktion ein fragendes »Hä??«?. Ersetzen Sie in der genannten Reihenfolge »Berliner Dialekt« mit dem aktuellen Directory, »Hochdeutsch« mit dem Directory »C« und »Englisch« mit einem beliebigen Directory, das mittels »PATH« in die Liste der zu durchsuchenden Directories aufgenommen wurde. Wenn Sie in der Kommandozeile ein Wort eintragen und mit <RETURN> den Amiga beauftragen, dieses Wort gefälligst als Kommando aufzufassen, sucht der Amiga, genauer gesagt das »CLI«, zunächst im aktuellen Directory nach einem Programm-File dieses Namens. Wird er dort nicht fündig, durchstöbert er das Verzeichnis »C« jener Diskette, mit der Sie das System »hochgefahren«, gebootet, haben. Gibt's auch dort kein Erfolgserlebnis, wird das nächste (im Beispiel gleichzeitig das einzi-

ge) im Suchpfad eingetragene Directory abgeklappert. Entweder der Amiga findet dort ein ausführbares Programm dieses Namens, dann wird es geladen und gestartet. Andernfalls kommt »Unknown command xxxx« auf den Bildschirm. Damit dürfte auch klar sein, daß der Amiga keinen definierten, begrenzten Befehlssatz hat – jedes ausführbare File innerhalb des Suchpfades kommt einem vom Amiga akzeptierbaren Kommando gleich. Wenn vom Befehlssatz des CLIs die Rede ist, meint man im allgemeinen jene Sammlung von Programmen, die sich im »C«-Directory der Boot-Diskette befinden. Mit diesen wichtigen Programmen auf der Workbench-Diskette wollen wir uns in diesem Artikel befassen.

Nur auf Befehl

Lassen Sie mich kurz die wichtigsten Regeln der Befehlseingabe in der Kommandozeile zusammenfassen. Die Eingabezeile darf maximal 254 Zeichen lang sein. WER HAT HIER »MESCHUGGE« GESAGT? Sie werden noch staunen, wie lang eine einzige Kommandozeile werden kann, wenn Sie die CLI-Befehle voll ausnutzen. Wenn Sie beim Schreiben an den rechten Fenster- rand stoßen, lassen Sie sich davon nicht beirren und schreiben Sie ungeniert weiter. Drücken Sie auf keinen Fall <RETURN>, um in die nächste Zeile zu kommen. <RETURN> bedeutet immer, daß die Zeile fertig ist und der CLI den Inhalt der Zeile abarbeiten soll. Kommandos, Optionen (=Zusatzangaben zu einem Befehl), Filenamen etc. sind immer durch ein Leerzeichen voneinander zu trennen. Ausnahme von der Regel ist die Anwendung des Zeichens »|«, über dessen Bedeutung wir uns bei den »Jokerzeichen« unterhalten. Filenamen dürfen auch Leerzeichen enthalten, z.B., wenn Sie eine Datei »Mein erstes Programm« getauft haben. Dann müssen Sie den ganzen Filenamen in Anführungszeichen einbetten.

Damit Sie auch Programme aufrufen können, die außerhalb des aktuellen, des »C«- und eines in »PATH« deklarierten Directory abgelegt sind, ohne in jenes Directory wechseln zu

müssen, in dem das gesuchte File steht, kann dem Filenamen noch der komplette Pfad zu dem gewünschten File vorangestellt werden. Ein Beispiel: Sie befinden sich im Directory »Utilities« der Boot-Diskette und wollen das Programm »Disky« starten, das sich im Directory »System« der Diskette in Laufwerk df1: befindet. Die Eingabe von

Disky

natürlich gefolgt von <RETURN>, bringt wie erwartet das Ergebnis

Unknown command Disky

Hingegen weisen Sie mit folgender Eingabe dem Amiga den rechten Weg:

df1:System/Disky

Würde sich das Programm »DiskCopy« im Ordner »System« auf der Diskette »Test« im Laufwerk df2: befinden, wäre sowohl

Test:System/DiskCopy

als auch

df2:System/DiskCopy

von Erfolg gekrönt. Sie müssen also dem Amiga in irgendeiner Art und Weise mitteilen, wo genau sich jenes Programm befindet, dessen Ausführung Sie verlangen, sofern es sich nicht im aktuellen Verzeichnis befindet. »Wo genau« heißt konkret, in welchem Verzeichnis und bei Bedarf noch zusätzlich auf welchem Device (»Device« = Gerät). Zum Ausgleich für diese Mühe ist der Amiga nicht so pingelig, was die Schreibweise betrifft – Groß- und Kleinbuchstaben werden gleichwertig behandelt, so daß auch die Eingabe von

TEST:System/diskcopy

zum Erfolg geführt hätte. Sollten Sie sich jedoch vertippt haben, lernen Sie die Sturheit eines Computersystems kennen. Und was der Gipfel ist – Sie dürfen alles nochmal tippen. Das »CLI« bietet Ihnen nur eine Art der Kommandozeileneditierung: Solange Sie die Zeile nicht mit <RETURN> freigegeben haben, können Sie Tippfehler dadurch korrieren, daß Sie alles, was zwischen

dem Cursor und der fehlerhaften Stelle liegt, mit der <BACKSPACE>-Taste löschen und neu schreiben. Falls Sie den Fehler zu spät bemerken und den Druck auf die <RETURN>-Taste nicht mehr abfangen können, können Sie von vorne tippen. Bei langen Kommandozeilen soll das schon zu Tobsuchtsanfällen geführt haben. Unter anderem darum hat Commodore dem »CLI« ab der Workbench-Version 1.3 einen Nachfolger geschenkt – die »SHELL«. Die wollen wir jetzt benutzen, denn wozu mehr quälen als nötig? Verkleinern Sie das CLI-Fenster mit der Maus auf Minimalgröße (um Platz zu schaffen) und klicken im Hauptverzeichnis der Workbench das Icon »Shell« an.

Zauberwort »Shell«

Das nun geöffnete Window hat frapante Ähnlichkeit mit dem CLI-Fenster namens »New CLI Window«, das Sie soeben verkleinert haben. Lediglich der Window-Name »AmigaShell« und der geänderte Prompt zeugen im ersten Moment von der Anwesenheit der »Shell«. Die verbesserten Editierfunktionen der »Shell« wurden an anderer Stelle bereits erwähnt, nicht erwähnt wurde jedoch die »ALIAS«-Funktion. Mit Hilfe der »Shell« ist es Ihnen möglich, bestehende Kommandos (= Programme) unter einem anderen Namen anzusprechen. Machen wir gemeinsam ein Beispiel. Wie Sie wissen, wird mit dem Befehl »dir« das Inhaltsverzeichnis einer Diskette oder eines Unterverzeichnisses auf dem Bildschirm ausgegeben. Geben Sie das Kommando »dir« im neuen Fenster ein! Wurde das Verzeichnis ausgegeben? Okay. Angenommen, Sie möchten die Ausgabe nicht mit »dir«, sondern mit »Inhalt« veranlassen, aus welchem Grund auch immer. Ohne »Shell« müßten Sie den Befehl »dir« in »Inhalt« umbenennen. Soweit wäre das ja kein Problem. Was aber, wenn ein Freund an Ihrem Computer

arbeitet, der von dieser Namensänderung nichts weiß? »dir« gibt es dann nicht mehr und daß »Inhalt« der neue Name für »dir« ist, weiß der andere ja nicht. Noch schlimmer wird es, wenn Programme den »dir«-Befehl aufrufen und ins Leere greifen! Dem allen baut der »ALIAS«-Befehl vor. Mit

```
alias Inhalt dir
```

kann das »dir«-Programm sowohl mit »dir« als auch mit dem Befehl »Inhalt« aufgerufen werden. Aber nicht nur das! Ganze Befehlssequenzen lassen sich mit »alias« unter einem einzigen Namen ansprechen. Machen wir noch einen Versuch und einen Vorgriff auf den »list«-Befehl und dessen Zusatzkommandos. Geben Sie bitte folgende Kommandozeile im »Shell«-Window ein:

```
alias Liste list [ ]nodates block files
```

Die eckigen Klammern erhalten Sie durch gleichzeitiges Drücken von <ALT><Ü> bzw. <ALT><*>. Nach <RETURN> übernimmt der Amiga bzw. die »Shell« den Alias-Namen »Liste« in das Verzeichnis ausführbarer Befehle. Die beiden eckigen Klammern bedeuten, daß hier bei der Befehlsübergabe Zusatzangaben aus der aufrufenden Kommandozeile einzusetzen sind. Wenn Sie dieses Beispiel wirklich korrekt eingegeben haben und jetzt den Befehl eintippen:

```
Liste devs:
```

erhalten Sie eine Aufzählung der im Verzeichnis »devs« eingetragenen Files mit Angabe der Filegröße in Blocks ohne Datumsangabe und ohne Angabe der im selben Verzeichnis eingetragenen Subdirectories. Mit »alias« lassen sich also nicht nur Dateien mit einem Alternativnamen versehen, sondern es können auch Programmzusatzparameter voreingestellt werden. Damit ersparen Sie sich eine Unmenge an Tipparbeit, wenn Sie öfters ein Programm mit einem ganzen Rattenschwanz von Zusatzangaben aufrufen müssen. Falls Sie über ein schlechtes Gedächtnis verfügen, hilft Ihnen die »Shell« auf die Sprünge. Die Eingabe von

```
alias
```

ohne weitere Angaben bringt eine Liste aller Alternativbezeichnungen

und deren Bedeutung auf den Bildschirm. Um das Beispiel »Liste« wieder zu eliminieren, reicht die kurze Eingabe von

```
alias Liste
```

Damit ist der Begriff »Liste« für den Amiga wieder ein »Unknown command«. Andererseits steht es Ihnen frei, den Namen »Liste« für ein anderes »alias« zu verwenden. Bei jeder neuen Zuweisung eines vorhandenen »alias«-Namens wird dessen alte Bedeutung überschrieben. Sie werden übrigens »alias« vergeblich auf der Diskette suchen. »alias« ist fester Bestandteil der »Shell«. Die Zuweisungen selbst sind aber an den jeweiligen »Shell«-Task gebunden. Wenn Sie jetzt z.B. »endcli« eintippen, beenden Sie den laufenden Task, und das »Shell«-Window verschwindet. Klicken Sie jetzt wieder auf das »Shell«-Icon und geben im danach geöffneten Window »Inhalt« ein, erhalten Sie »Unknown command« als Antwort. Der Begriff »Inhalt« ist wieder unbekannt.

Ganz praktisch, dieses »Alias«, werden Sie jetzt denken, aber jedesmal alles einzutippen ist doch langweilig. Das fanden auch die Entwickler der »Shell« und schufen einen Ausweg. Im »S«-Verzeichnis der Boot-Diskette finden Sie einen Eintrag namens »Shell-Startup« (schauen Sie ruhig nach). In diesem File, es ist eine Textdatei, sind schon einige »Alias«-Zuweisungen eingetragen, und mit einem Texteditor steht es Ihnen frei, weitere Zuweisungen vorzunehmen. Sie könnten also die Definition von »Test«, wie wir sie zuvor im Direktmodus ausprobiert haben, in diese Datei eintragen, und bei jedem Start eines »Shell«-Tasks wäre diese Definition automatisch eingebunden. Welche Möglichkeiten dieses File außerdem bietet, wird im Artikel zu den Batch-Dateien ab Seite 82 in diesem Heft näher beschrieben.

Eine praktische Sache, der wir in diesem Artikel etwas mehr Raum geben wollen, sind die »Jokerzeichen« oder »Wildcards«. Amiga-DOS kennt sieben verschiedene Jokerzeichen: das Doppelkreuz »#«, das Fragezeichen »?«, das Apostroph »'«, das Prozentzeichen »%« und das »verbindende Trennzeichen« »|«. Mit dem Fragezeichen »?« ersetzen Sie jedes beliebige (zulässige) Zeichen in einem Dateinamen. Der Befehl

```
dir SYS:c/?nd????
```

beispielsweise zeigt alle Files im Verzeichnis »C« der Boot-Diskette, die an zweiter Stelle im Dateinamen ein »n« und an dritter Stelle ein »d« enthalten und überdies sechs Zeichen lang sind (nicht länger und nicht kürzer!). Das Ergebnis dieses Kommandos wäre also die Ausgabe von »EndCLI«, weil nur dieser Filenamen den Anforderungen entspricht. Hingegen brächte die Eingabe

```
dir SYS:c/?nd????
```

das Ergebnis »EndSkip« - jetzt waren nach der Zeichenfolge »nd« noch vier weitere Zeichen gefordert. Komplizierter wird es da schon bei der Verwendung von »#«. Mit diesem Zeichen verlangen Sie, daß das Zeichen, das Sie nach dem Doppelkreuz eingeben, im Filenamen gar nicht oder auch mehrfach vorkommen muß (klingt schizophren, wie?). Da helfen nur Beispiele:

```
dir SYS:c/#n?????
```

bringt folgendes Ergebnis:

```
Avail      Break
EndIf      Fault
IconX      Mount
NewCLI     Stack
Which
```

Ist Ihnen aufgefallen, daß alle Dateinamen fünf Zeichen lang waren, nur »NewCLI« hatte sechs Buchstaben? Die Eingabe verlangt nach allen Dateien, die entweder mit einem oder mehreren »N« beginnen und danach noch fünf Zeichen folgen oder - wenn kein »N« an erster Stelle steht - der ganze Dateiname nur fünf Zeichen lang ist (fünf Fragezeichen!). Lassen Sie aber das Doppelkreuz weg und geben nur ein

```
dir SYS:c/n?????
```

(wie immer <Return> nicht vergessen), erhalten Sie die Ausgabe

```
NewCLI
```

denn in diesem Fall verlangen Sie explizit, daß der Dateiname sechs Buchstaben lang sein und mit einem »N« beginnen muß. Um dem ganzen Verwirrungsspiel noch die Krone aufzusetzen, versuchen Sie doch mal folgende Eingabe:

```
dir SYS:c/#?n?????
```

Überrascht Sie das Ergebnis?

```
EndSkip    Install
```

```
NewCLI
```

Sicher nicht: Die genannte Zeichenkombination verlangt, wie Sie erkannt haben, daß vor dem ersten Auftreten des Buchstabens »N« eine beliebige Anzahl (»#« bedeutet null - bis mehrfach) beliebiger Zeichen (vertreten durch das Fragezeichen) stehen kann, aber nicht muß und daß die auszugebenden Filenamen nach dem »N« noch weitere fünf Zeichen (fünfmal das Fragezeichen) enthalten müssen.

Dagegen bringt die Eingabe von

```
dir SYS:c/#nd????
```

nur die Anzeige von

```
Date
```

zuwege, weil nur für diesen Filenamen unter allen im »C«-Verzeichnis eingetragenen Dateien zutrifft, daß an erster Stelle ein »D« steht und der ganze Filenamen vier Zeichen lang ist. Ein Filenamen, der mit »nd« oder »nnnd« beginnt, steht nicht im »C«-

Nimm den Joker

Directory. Gäbe es einen solchen und kämen nach dem »d« noch drei weitere Zeichen, wäre auch dieser angezeigt worden.

Das letztgenannte Beispiel soll uns auch für eine Demonstration der Patternfunktion mit mehr als einem zu filternden Zeichen dienen. Tippen Sie nach dem Prompt

```
dir SYS:c/#(nd)???
```

Daß Sie die Eingabe mit <RETURN> zur Bearbeitung freigeben müssen, ist Ihnen natürlich klar. Der einzige Unterschied zum vorherigen Beispiel besteht in der Einklammerung von »nd«. Durch sie wird nicht mehr nach Filenamen gesucht, die nach einem oder mehreren oder auch keinem »n« am Anfang des Na-

mens unbedingt mit »d« fortsetzen und nach dem »d« noch drei Zeichen haben. Sondern jetzt sind Filenamen im Fadenkreuz, die die Zeichenfolge »nd« einmal, keinmal oder beliebig oft zu Beginn stehen haben, danach aber noch drei Zeichen kommen. Das Ergebnis bei einer Original-Workbench-Diskette sieht so aus:

```
Ask Dir
Lab Run
Why
```

Es existiert also kein Filenamen, der mit der Zeichenkombination »nd« beginnt. Daher werden nur jene Filenamen ausgegeben, die aus drei Zeichen bestehen. Hätten Sie im »C«-Directory ein File namens »ndndndwut« und eines mit dem Namen »ndRUM« eingetragen, hätte der Computer diese auch ausgegeben.

Damit Sie unabhängig von der Namenslänge alle Dateien angezeigt erhalten, die mit einer gesuchten Zeichenfolge beginnen, müssen Sie zwei Jokerzeichen verknüpfen: das Doppelkreuz und das Fragezeichen. Rufen Sie sich die Definition der Doppelkreuz-Jokerfunktion nochmal ins Gedächtnis: Das dem Doppelkreuz nachgestellte Zeichen (bzw. die Zeichenfolge, wenn mehrere Zeichen zusammengeklammert sind) muß einmal, mehrfach oder auch gar nicht auftreten. Was passiert nun, wenn anstelle eines normalen Zeichens nach dem Doppelkreuz ein weiteres Jokerzeichen, nämlich das Fragezeichen, steht? Sie erinnern sich: Mit einem Fragezeichen ersetzen Sie ein beliebiges anderes Zeichen. In Kombination mit dem Doppelkreuz ersetzt ein Fragezeichen aber beliebig viele andere Zeichen, denn mit dem Doppelkreuz ist jede Anzahl von Wiederholungen des Fragezeichens und damit von durch das Fragezeichen ersetzten Zeichen definiert. Die Eingabe von

```
dir SYS:c/e#?
```

bringt demzufolge alle mit »e« beginnenden Filenamen auf den Schirm, unabhängig von der Gesamtlänge der Dateinamen (ein Zeichen lang muß der Dateiname aber mindestens sein, denn das »e« wird explizit angefordert):

```
Echo      Ed
Edit      Else
EndCLI    EndIf
EndSkip   Eval
Execute
```

Umgekehrt bringt die Eingabe von

```
dir SYS:c/#?e
```

alle Filenamen im »c«-Verzeichnis zur Anzeige, die mit einem »e« enden. Auf diese Weise können Sie natürlich auch alle Namen ausfiltern, die mit dem Buchstaben »R« beginnen und mit »e« enden:

```
dir SYS:c/r#?e
```

Na, was haben Sie für eine Ausgabe bekommen? Hoffentlich nur »Rename«, sonst wird's bedenklich.

Erfreulicherweise dürfen Sie zu suchende Zeichenfolgen (»Muster«=»Pattern«) auch kombinieren. Als Trennzeichen zwischen den Suchmustern muß »|« herhalten. Die Eingabe von

```
dir SYS:c/new???lend???
```

holt alle Dateinamen auf das aktuelle Ausgabegerät, die mit »new« oder »end« beginnen und insgesamt sechs Zeichen lang sind (drei Zeichen für »new« oder »end« und drei weitere Zeichen). Auch die Angabe von mehreren Suchmustern ist zulässig:

```
dir SYS:c/new#?|end????|?|???
?????
```

zeigt Ihnen alle Files im »c«-Ordner der Boot-Diskette, die

- entweder mit »new« beginnen und danach beliebig viele Zeichen tragen oder
- mit »end« beginnen und insgesamt sieben Zeichen lang sind oder
- zwei Zeichen lang sind oder
- acht Zeichen lang sind.

```
CD          Ed
EndSkip     FF
FileNote    If
NewCLI      NewShell
Resident    SetClock
SetPatch
```

Bleibe noch die Anwendung des Jokerzeichens »%«. Dieses benutzen Sie, wenn Sie eine Zeichenkette mit der Länge Null (auf gut deutsch keine Zeichenkette) definieren wollen. Auch hier ein Beispiel: Angenommen, Sie wollen alle Files im »C«-Ordner angezeigt bekommen, die mit »ed« beginnen und entweder mit »it« enden (»Edit«) oder keine weiteren Zeichen enthalten (»Ed«):

```
dir SYS:c/ed#(it)%
```

Ich muß zugeben, daß diese Jokerfunktion zumindest bei meiner täglichen Arbeit eher ein Schatten-dasein führt.

Apropos Schatten – so nützlich die Jokerzeichen sind, die Inkonsequenz, mit der sie von den einzelnen Programmen und Kommandos akzeptiert werden, kann einem schon den Spaß an der Freude verderben. Während z.B. ein

```
dir c/#?cli
```

erwartungsgemäß zur Ausgabe

```
EndCLI      NewCLI
```

führt, überrascht die an und für sich richtige Eingabe von

```
rename c/#?cli to c/#?SHELL
```

mit der Aussage

```
Can't rename c/#?cli as c/#
?SHELL
```

Leider sind die Abneigungen einzelner Kommandos gegenüber Jokerzeichen und Jokerkombinationen nicht gleich, so daß kein allgemeingültiges »Kochrezept« gegeben wer-

Die Pipeline

den kann. Sie müssen sich schon selbst ablagen. Wollen wir hoffen, daß diese Ungereimtheiten mit dem nächsten Release beseitigt sind.

Eine wichtige Neuerung der Workbench 1.3 und damit der »Shell« ist »PIPE«. Bitte versuchen Sie nicht, »PIPE« mit »PFEIFE« zu übersetzen. »PIPE« kommt von »PIPELINE« und stellt eine Datenverbindungsleitung von Prozeß zu Prozeß dar. Stellen Sie sich vor, Ihr Amiga wäre Teil einer ganzen Computeranlage mit mehreren vernetzten Computern. Sie wissen ja, wie es in so einem Netzwerk zugeht. Wenn Computer A eine Datei auf der zentralen Hard-Disk ablegt, kann Computer Y die Daten dieser Datei sofort weiterverarbeiten. Verbunden sind die Computer in ei-

nem solchen Netzwerk durch Datenleitungen und Netzwerk-Steuerprogramme (Ethernet, Tokenring, und wie sie alle heißen). Was im Großen mit viel Hard- und Software-Aufwand funktioniert, spielt sich im Kleinen, sprich in Ihrem Amiga, mit Hilfe der »PIPE« ab. Die »PIPE« ist innerhalb Ihres Amigas das, was das Netzwerk für den Computerverbund ist: eine Möglichkeit, Daten zu transferieren. Um Ihnen einen Begriff von den Vorteilen der »PIPE« zu vermitteln, gehen wir einmal einen Schritt zurück in die Geschichte der Amiga-Betriebssysteme. Nicht weit, bloß bis Workbench-Version 1.2. Sollte das folgende zu unklar sein, überspringen Sie es bitte, bis Sie mit den angeführten Kommandos vertraut sind. Um unter V1.2 den Inhalt der Datei »Startup-Sequence« alphabetisch sortiert und mit Zeilennummern versehen auf dem Bildschirm ausgeben zu können, müssen Sie folgende Eingaben durchführen:

```
type s/startup-
sequence to Hilfsfile_a
sort hilfsfile_a to
hilfsfile_b
type hilfsfile_b number
delete hilfsfile_a
delete hilfsfile_b
```

Abgesehen vom Tippaufwand und der Tatsache, daß Sie die Diskette nach Abschluß der Ausgabe aufräumen müssen, wird es lustig, wenn Sie größere bis große Dateien auf diese Art bearbeiten müssen. Angenommen, Sie müssen anstelle der relativ kurzen »Startup-Sequence« eine umfangreiche Adressendatei auf diese Art handhaben, kann es vorkommen, daß die benötigten Hilfsfiles die freie Kapazität der Diskette überschreiten. Dann bleibt Ihnen nur noch mehrmaliges Diskettenwechseln oder resignierend auf die geplante Aktion zu verzichten. Von der Zeit, die für die Schreibvorgänge auf der Diskette draufgeht, wollen wir gar nicht reden.

Ganz anders mit »PIPE«. Das zuvor genannte Beispiel sieht mit Hilfe der »PIPE« folgendermaßen aus:

```
type s/startup-
sequence to PIPE:a
sort from PIPE:a to PIPE:b
type from PIPE:b number
```

Der Clou dabei ist, daß alle drei Vorgänge parallel ablaufen können! Es liegt ganz allein bei Ihnen, wann Sie die Weiterverarbeitung starten – es geht sofort weiter. »PIPE« schreibt die Daten in einen Puffer, und diese Pufferdaten stehen sofort für andere Prozesse zur Verfügung. Es ist also nicht notwendig, zu warten, bis »type s/startup-sequence to PIPE:a« beendet ist, bevor der »sort«-Befehl gestartet werden kann. Und ganz egal, wie groß die zu bearbeitenden Datenmengen sind, für jeden »PIPE«-Aufruf werden nur 4 KByte Speicher benötigt. Ist die zu bearbeitende Datenmenge kleiner als die Puffergröße, werden die Daten in diesen Puffer geschrieben und stehen dann zur freien Verfügung. Ist die Datenmenge größer als der Pufferspeicher, wird das Auffüllen des Puffers gestoppt und das schreibende Programm wartet mit dem Nachfüllen, bis ein anderes Programm den Puffer etwas entleert hat. Machen wir doch einen Versuch. Öffnen Sie drei »SHELL«-Windows, indem Sie das »SHELL«-Icon dreimal mit einem Doppelklick ansprechen. Ordnen Sie die drei Windows untereinander an (benutzen Sie die Größen-Gadgets, um die Fenster entsprechend zu trimmen). Aktivieren Sie das oberste Fenster mit einem Mausklick. Tippen Sie in diesem Fenster

```
type s/startup-
sequence to PIPE:Test_a
```

aber drücken Sie noch nicht auf die <RETURN>-Taste. Danach aktivieren Sie, wieder mit einem Mausklick, das mittlere der drei »SHELL«-Fenster. Dort tippen Sie folgendes Kommando ein:

```
sort from PIPE:Test_a to
PIPE:Test_b
```

Drücken Sie diesmal ruhig wie gewohnt die <RETURN>-Taste. Sie werden sehen, daß die Kommandozeile zwar akzeptiert wird, aber es tut sich nichts. Als letztes kommt das dritte, das unterste Fenster dran. In dieses schreiben Sie

```
type from PIPE:Test_b number
```

Auch diese Zeile übergeben Sie wie gewohnt mit <RETURN> an den Kommandoprocessor. Aber auch hier tut sich (noch) nichts. Klicken Sie jetzt wieder in das erste Fenster, um es zu aktivieren, und drücken Sie dann auf <RETURN>. Als Folge dieser Aktion erhalten Sie im dritten Fenster den Inhalt der Datei »Startup-Sequence« ausgegeben, die Zeilen alphabetisch sortiert und fortlaufend nummeriert. Das ist »PIPING« in Reinkultur. Ich bin sicher, Ihnen fallen noch hundert andere Anwendungen ein. Auf einige kleine Macken bzw. Anforderungen der »PIPE« möchte ich Sie aber noch hinweisen, um Ihnen unliebsame Überraschungen nach Möglichkeit

Die Macken der »PIPE«

zu ersparen. Zuerst einmal: Beachten Sie bitte die Schreibweise der Kommandos in unserem Beispiel. In die »PIPE« schreiben Sie mit »TO«, aus der »PIPE« heraus in ein anderes, empfangendes Programm hinein schreiben Sie mit »FROM«! Das geht bei den genannten Kommandos nicht anders! Andererseits trägt das »AVAIL«-Kommando kein »TO«! Da müssen Sie mit dem Umleitungspfeil »>« arbeiten. Zudem müssen Sie jeder der Pseudo-Zwischendateien einen Namen geben, für den die normalen Filenamen-Konventionen des Amiga-DOS gelten. Drittens ist ein empfangender Prozeß solange blockiert, bis er aus der »PIPE« seine Daten bekommt. Erinnern Sie sich bitte, daß wir den mittleren und den untersten der drei »SHELL«-Tasks mit der <RETURN>-Taste gestartet hatten, ohne daß sich vorerst etwas tat. Erst als wir die »PIPEs« mit dem obersten Prozeß füllten, kam auch Leben in Prozeß 2 und 3. Stellen Sie sich die »PIPE« ruhig wie ein Rohr vor, das zuerst von oben gefüllt werden muß, ehe man es »von unten« ablassen kann. Die Rohranalogie macht auch

deutlich, daß Daten, die einmal aus dem Rohr abgelassen wurden, nicht mehr verfügbar sind – die sind weg. Sozusagen ausgetrunken vom empfangenden Prozeß. Viertens, um bei der Rohranalogie zu bleiben: Sie können ein Rohr erst dann wieder befüllen, wenn Sie es zuvor geleert haben. Sie haben zwar nahezu beliebig viele Rohre zur Verfügung, aber Rohr Nummer 1 kann erst dann wieder neu gefüllt werden, wenn es völlig leer ist, wenn also ein anderer Prozeß den Rohrinhalt übernommen hat. Das hat nichts mit der Datenmenge zu tun – auch wenn Sie nur zwei Byte in das Rohr Nummer 1 einfüllen und danach der Füllvorgang beendet wurde, müssen Sie zuerst die beiden Bytes abholen, ehe Sie wieder Daten einfüllen dürfen. Halten Sie sich nicht an diese Regel, folgt eine Fehlermeldung auf den Fuß. Hinweis Nummer 5 hängt mit der Datenmenge zusammen. Wenn die zu verarbeitenden Daten in den 4-KByte-PIPE-Puffer passen, kann die »PIPE« aus dem gleichen CLI-Task ausgelesen werden, aus dem sie befüllt wurde. Ist die Datenmenge aber größer als der Puffer, wird der Füllvorgang gestoppt. Damit ist der Füll-Task auch blockiert! Bitte achten Sie auf diese Dinge, dann werden Sie sicher ein PIPE-Fan.

Eine andere Art der Datenweitergabe ist die Anwendung der Umleitungspfeile »>« und »<«. Damit dirigieren Sie die Ausgabekanäle. Einige Beispiele sollen die Wirkung von »>« und »<« verdeutlichen. Das »dir«-Kommando haben Sie ja schon kennengelernt – das Inhaltsverzeichnis einer Diskette wird auf dem Bildschirm ausgegeben. Damit Sie diese Ausgabe auch auf andere Geräte (Drucker, Modem etc.) umleiten können, stehen die Umleitungspfeile zur Verfügung:

```
dir >prt: df2:c all
```

schickt die Auflistung des Inhaltsverzeichnis des »C«-Directory jener Diskette, die im Laufwerk »df2:« eingelegt ist, nicht wie gewohnt an das aufrufende CLI- oder Shell-Fenster, sondern an den Drucker. Beachten Sie bitte die Reihenfolge der Kommandos – zuerst müssen Sie dem »dir«-Kommando das Ziel bekanntgeben, ehe das zu listende Directory und eventuelle Zusatzanweisungen angegeben werden. Ähnlich funktioniert

```
type > ram:Testfile
df0:s/startupII
```

Doch werden hier die Daten in der Datei »Testfile« in der RAM-Disk abgelegt. Umgekehrt lassen sich Daten an Programme schicken, die normalerweise Eingaben z.B. von der Tastatur erwarten. Angenommen, Sie haben eine Datei »Testfile« in der RAM-Disk abgelegt, das folgenden Inhalt hat:

```
"*e[32mrequested Commands are
resident*e[0m"
```

Diesen File-Inhalt können Sie nun mit dem »echo«-Befehl folgendermaßen ausgeben:

```
echo < ram:testfile ?
```

Das Ergebnis sieht dann so aus:
,NOLINE/S,FIRST/K,LEN/K: requested Commands are resident

wobei die Worte attribuiert gemäß den im Filetext angegebenen Steuerzeichen ausgegeben werden. Es läßt sich leider nicht vermeiden, daß die Syntax-Help-Zeile mit ausgegeben wird. Sie müssen den »echo«-Befehl (und nicht nur diesen, sondern alle CLI-Kommandos, die Sie auf diese Weise fremdspeisen) mit dem Fragezeichen aufrufen, um die Eingabe von der Tastatur auf einen anderen Eingabekanal umleiten zu können. Mit einem Trick kann dieser Schönheitsfehler allerdings ausge- merzt werden. Ändern Sie die Zeile im File folgendermaßen:

```
"*n*eM*e[32mrequested Command-
s are resident*e[0m*eJ"
```

Der Unterschied liegt nur in den ersten fünf Zeichen! Damit führen Sie vor der Ausgabe des gewünschten Strings einen Wagenrücklauf und einen Zeilenvorschub aus, womit die ausgegebene Syntaxhilfe vom folgenden String überschrieben wird. Nicht unbedingt elegant, aber dafür wirkungsvoll. Bei diesen Zeichenfolgen handelt es sich um »ESCAPE-Sequenzen« – mehr davon und zu diesem Thema finden Sie im zugehörigen Textkasten.

Der Gipfel im Umleitungszirkus wird zweifellos mit der Umleitung der Ausgabe eines Kommandos bei gleichzeitiger Fremdsteuerung desselben erreicht:

```
echo > ram:test2
< ram:testfile ?
```

schreibt die Ausgabe des »ECHO«-Befehls in das File »test2«, wobei die das Echokommando steuernde Eingabe aus dem erstellten File »testfile« kommt.

Workbench 1.3

Noch eine Neuerung der Workbench-Version 1.3: Die Umgebungsvariablen, auf gut deutsch »Environment Variables« sind Daten, die mit dem Kommando »SETENV« im Arbeitsspeicher abgelegt werden und dort für Abfragen durch andere Programme zur Verfügung stehen. Eines der wenigen Programme, auf die diese Umgebungsvariablen Einfluß nehmen können, ist das Hilfsprogramm »more« aus der »Utilities«-Schublade der Workbench-Diskette. Im Normalfall verfügt dieses Programm über keine Verbindung zu einem Texteditor. Auch die Hilfsfunktion dieses Programms läßt in der Standardeinstellung eine solche Verbindungsmöglichkeit nicht erkennen. Wenn Sie aber mit

```
SETENV editor
Workbench1.3D:c/Ed
```

im System die Umgebungsvariable »editor« anlegen und ihr den Wert »Workbench1.3D:c/Ed« zuweisen, erkennt dies das Hilfsprogramm »more« und stellt Ihnen die ansonsten nicht vorhandene Funktion »E« zur Verfügung, mit der Sie das von »more« angezeigte Textfile vom in »EDITOR« definierten Texteditor verarbeiten lassen. Das funktioniert allerdings nur dann, wenn Sie »more«

aus dem »CLI« bzw. der »SHELL« heraus starten und nicht mittels Doppelklick. »More« ist das einzige mir bekannte Programm auf den beiden mitgelieferten Disketten, das eine Umgebungsvariable abfragt. Es steht Ihnen aber frei, beliebige andere Umgebungsvariablen anzulegen und z.B. aus Batch-Dateien heraus abzufragen und zu verarbeiten. »Batch-Dateien« sind Textdateien, in denen Kommandos eingetragen sind, die bei Aufruf einer solchen Datei mittels »Execute« der Reihe nach abgearbeitet werden (Näheres lesen Sie ab Seite 82).

Bevor wir uns jetzt den einzelnen Befehlen (= Programmen) zuwenden, die den Befehlssatz des CLIs und der Shell ausmachen, noch einige allgemeine Punkte: In diesem Artikel werden nur jene Kommandos besprochen, die sich im »C«-Directory Ihrer Workbench-Diskette befinden. Alle anderen Files, wie etwa »DiskCopy«, werden an anderer Stelle in diesem Heft behandelt. Alle Kommandos sind sowohl aus einem »Shell«-unterstützten Window als auch aus einem reinen »CLI«-Window heraus aufrufbar. Dabei darf aber nicht übersehen werden, daß ein »CLI«-Window mit einem »Alias«-Kommando nichts (!) anzufangen weiß – »alias« versteht einzig und allein die »Shell«. Weiters werden wir in diesem Artikel auf einige Programme nicht eingehen, die sich zwar im »C«-Verzeichnis befinden, die aber mit Kommandos nur sehr wenig gemein haben und meiner Meinung nach besser in einer »TOOLS«- oder »Utilities«-Schublade aufgehoben wären. Es sind dies die beiden Editoren »Ed« und »Edit« sowie »DiskDoctor«. Nun denn: frisch, fromm, fröhlich, frei – ans Werk! *pe*

Alle CLI-Befehle im »C«-Directory

*AddBuffers	*Ask	*Assign	*Avail	*Binddrivers
*Break	*CD	*ChangeTaskPri	*Copy	*Date
*Delete	*Dir	*DiskChange	*DiskDoctor	*Echo
-Ed	-Edit	*Else	*EndCLI	*EndIf
*EndSkip	*Eval	*Execute	*Failat	*Fault
*FF	*FileNote	*GetEnv	*IconX	*If
*Info	*Install	*Join	*Lab	*List
*LoadWB	*Lock	*Makedir	*Mount	*newcli
*NewShell	*Path	*Prompt	*Protect	*Quit
*Relabel	*RemRAD	*Rename	*Resident	*Run
*Search	*SetClock	*SetDate	*SetEnv	*SetPatch
*Skip	*Sort	*Stack	*Status	*Type
*Version	*Wait	*Which	*Why	.

Die mit »-« bezeichneten Dateien werden in diesem Artikel nicht besprochen.

CLI-Befehle kurz erklärt

ADDBUFFERS vergrößert den Laufwerk-Pufferspeicher

Schablone

Laufwerk/a,Buffergröße/a

Syntax

AddBuffers <Laufwerk:> <Anzahl Buffer-Blöcke >

»AddBuffers« vergrößert den jedem Laufwerk vom System zugeteilten Pufferspeicher. In diesem Pufferspeicher werden die vom Laufwerk kommenden Daten vor der Weiterverarbeitung abgelegt. Amiga-DOS stellt für jedes beim Systemstart aktive Laufwerk fünf Blocks à 512 Byte als Datenpuffer zur Verfügung. Eine Vergrößerung des Pufferspeichers erhöht die Datenflußgeschwindigkeit zwischen Laufwerk und Computer, kostet aber logischerweise wertvollen Speicher, noch dazu CHIP-Memory! Dieser Speicher wird erst wieder beim nächsten Warm- oder Kaltstart freigegeben.

Beispiel

```
AddBuffers df3: 20
```

setzt die Buffergröße für Laufwerk 3 auf den Wert 20 (entspricht etwa 10 KByte).

ASK stoppt den Ablauf einer Batch-Datei und gibt dem Benutzer Gelegenheit, Eingaben zu tätigen und damit den weiteren Ablauf entscheidend zu beeinflussen.

Schablone

Eingabeaufforderungs-Zeichenkette/a

Syntax

Ask [Zeichenkette]

»ASK« ist ein Programm, das einfache Eingaben zuläßt und anhand dieser Eingaben Flags setzt, die in Scriptdateien weiterverarbeitet und als Verzweigungsbedingungen verwendet werden. »ASK« erlaubt die Definition einer Zeichenkette, die zu Beginn ausgegeben wird und z.B. die Aufforderung enthält, eine Eingabe zu tätigen. Dieser Aufforderung ist durch die Eingabe von <Y> <RETURN> oder <N> <RETURN> zu entsprechen. Wird nur <RETURN> gedrückt, entspricht dies der Eingabe von <N> <RETURN>.

Beispiel

```
ask "Hast Du einen ATARI ?"
if not warn
echo "Servus Freund!"
else
echo "Du tust mir leid"
endif
```

ASSIGN ordnet einem frei wählbaren Device-Namen einem existierendes Directory oder File zu

Schablone

Name, Directory, LIST/s, EXISTS/s, REMOVE/s

Syntax

Assign

```
<frei_wählbarer_Name>:
[zugewiesene_Pfadangabe/File |
][... ..][LIST][EXISTS][REMOVE]
```

»ASSIGN« macht aus »frei_wählbarer_Name« ein Synonym für ein Directory oder auch eine komplette Pfadangabe. »frei_wählbarer_Name« wird so zu einem »logischen Device«. Nach einer »ASSIGN«-Zuweisung wird das logische Device vom Betriebssystem behandelt wie ein

physikalisches Gerät (z.B. ein Disketten-Laufwerk). Die erneute Zuweisung eines bereits vorhandenen Assignments überschreibt das alte.

Wahlschalter

LIST - gibt eine Tabelle aller Assignments (= »Zuweisungen«) auf dem aktuellen Ausgabegerät aus. Dieses Schlüsselwort ist eigentlich unnötig, da die Eingabe von »ASSIGN« alleine die gleiche Wirkung hat. Lediglich in Verbindung mit einem vorhandenen Assignment wird »LIST« interessant (siehe Beispiele).

EXISTS - prüft, ob der angegebene Name in der Liste der logischen oder physikalischen Devices erscheint. Wird der gesuchte Eintrag gefunden, übergibt »ASSIGN« den Return-Code »NULL«, andernfalls den Wert für den Fehlerlevel »WARN«. Die Auswertung kann mit einer »IF WARN«-Funktion erfolgen..

REMOVE - löscht das angegebene Device aus der Device-Liste des Betriebssystems. Dieser Zusatz hat die gleiche Wirkung wie die Eingabe des Befehls »ASSIGN zugewiesener_Name«. Wenn mit »ASSIGN PRINT: WP:PRINT« das logische Device »PRINT« der Diskette »WP:« zugeordnet wurde, wird diese Zuordnung mit »ASSIGN PRINT:« ebenso rückgängig gemacht wie durch die Eingabe von »ASSIGN PRINT: REMOVE«.

Beispiel

```
Assign WB: Workbench1.3D:
```

Die Diskette »Workbench1.3D« kann jetzt sowohl unter ihrem »richtigen« Namen »Workbench 1.3D« als auch unter der Bezeichnung »WB« angesprochen werden.

AVAIL gibt Auskunft über die RAM-Auslastung

Schablone

CHIP/s, FAST/s, TOTAL/s

Syntax

Avail [CHIPIFAST] [TOTAL]

»Avail« wertet die Speicherverwaltungslisten des Betriebssystems aus

und informiert über momentan belegten (»In-Use«) und freien (»Available«) Speicher in CHIP- und FAST-Memory. Ebenfalls informiert es über den größten, zusammenhängenden und noch nicht belegten Speicherbereich in CHIP- und FAST-RAM (»Largest«).

Wahlschalter

CHIP - ermittelt die Größe des freien CHIP-Memory.

FAST - gibt die Summe aller im FAST-RAM noch nicht belegten Speicherstellen aus

TOTAL - summiert freies CHIP- und FAST-RAM und gibt das Ergebnis auf dem Monitor aus.

BINDDRIVERS bindet externe Gerätetreiber ins System ein

Schablone

keine

Syntax

Binddrivers

Wenn im Verzeichnis »Expansion« der Boot-Diskette spezielle Gerätetreiber abgelegt sind, werden diese mit »BindDrivers« in das System eingebunden, sinnvollerweise in der »Startup-Sequence«. Solche Treiber werden im allgemeinen gemeinsam mit der zugehörigen Hardware-Erweiterung mitgeliefert.

BREAK schickt Control-Codes an andere Tasks

Schablone

Prozeß/a, A=ALL/s, C/s, D/s, E/s, F/s

Syntax

Break <Task-Nummer> [C D E F oder ALL]

»Break« schickt an den angegebenen Task ein Unterbrechungssignal. Viele, aber nicht alle Programme reagieren auf den Empfang von Control-Codes zwischen CTRL-C und CTRL-F, wobei die einzelnen Control-Codes verschiedenen Reaktionen des empfangenden Programms zugeordnet sein können. Meist bedeutet »CTRL-C« für das empfangende Programm die Aufforderung, seine Aktivitäten zu beenden.

»Break« ohne dezidierte Angabe eines Control-Codes schickt ein »CTRL-C« an den angegebenen Task.

Beispiel

Break 2 ALL

schickt an den Task Nummer 2 nacheinander die Codes »CTRL-C«, »CTRL-D«, »CTRL-E« und »CTRL-F«.

CD wechselt ins angegebene Verzeichnis oder gibt das aktuelle aus

Schablone

Directory, //s

Syntax

CD [[Pfad/]neues_Directory]] [...]

»CD« leitet sich ab von »Change Directory«, also »wechse Verzeichnis«. Wenn an »CD« der Name eines gültigen Directory (evtl. inkl. Pfadangabe) übergeben wird, macht »CD« dieses zum aktuellen Directory. Alle Files, die dann eingegeben werden, sucht Amiga-DOS dann zuerst in diesem Verzeichnis. Die Eingabe »CD« ohne Zusatz gibt das aktuelle Directory auf dem aktuellen Ausgabegerät aus. Um aus den Tiefen eines Sub-Directory eine Stufe »nach oben« in Richtung Hauptverzeichnis zu gelangen, verwendet man das Zeichen »/« anstelle der Eingabe Verzeichnisnamens.

Beispiel

Ganz gleich, welches Directory auf welcher Diskette gerade das aktuelle ist, mit

CD libs:

machen Sie das Directory »libs« auf der Boot-Diskette zum aktuellen Directory des Amigas

Beispiel

CD df1:Korrespondenz/Briefe/Anwalt

macht das Sub-Directory »Anwalt« auf der Disk in Laufwerk DF1: zum aktuellen Directory.

CHANGETASKPRI ändert die Priorität eines angegebenen Tasks

Schablone

PRIORITY/a, PROCESS/k

Syntax

ChangeTaskPri [PRI #] [PROCESS #]

Die Änderung der Priorität eines Tasks mittels »ChangeTaskPri« bedeutet eine Änderung der dem jeweiligen Task zur Verfügung stehenden CPU-Zeit. Das Multitasking-Betriebssystem des Amigas kennt Prio-

ritätswerte zwischen +127 und -128. Je höher der Wert (unter Berücksichtigung des Vorzeichens), desto mehr CPU-Zeit steht dem Task zur Verfügung. Prioritätswerte über +25 sind zu vermeiden, um nicht systeminterne Tasks wie z.B. das Beschreiben einer Diskette außer Tritt zu bringen. Zur Sicherheit empfiehlt Commodore als Obergrenze den Wert »+5«. Welche Prozeßnummer einem bestimmten Task zugeordnet ist, erfahren Sie mit »status«.

Beispiel

ChangeTaskPri -5 process 3

COPY kopiert Files und Directories

Schablone

FROM, To/a, ALL/s, QUIET/s, BUF=BUFFER/k, CLONE/s, DATES/s, NOPRO/s, COM/s

Syntax

Copy [FROM | TO] <Filename >

[| <Filename2 > | <Filename3 > | ...]

[TO IFROM]

<Filename|DIR > [ALL] [QUIET]

[BUF nn] [CLONE IDATES INOPRO

ICOM]

»Copy« dupliziert den Inhalt eines oder mehrerer Files (mehrere, wenn als Quelle ein Directory oder ein Jokermuster angegeben wurde), die unmittelbar nach »FROM« angegeben wurden, in jenes File (oder in jene Files...), das unmittelbar nach »TO« angegeben wurde. »FROM« und »TO« können weggelassen werden, wenn nach »Copy« zuerst das zu duplizierende File steht und daran anschließend das zu beschreibende File. Sollte das Zielfile bereits existieren, wird es ohne Warnung einfach überschrieben, es sei denn, seine Schutzbits verhindern ein Überschreiben.

Das »FROM«-Argument kann ein Filename, ein Directory oder ein beliebiges Jokermuster sein. Ein Führungszeichenpaar (»" "«) steht für das aktuelle Directory und kann als Joker für das Ziel-Directory (»TO«) eingesetzt werden. Ist das »FROM«-Argument ein Directory, so wird dessen Inhalt in das gleichnamige Directory des Zielgerätes kopiert. Existiert auf dem »TO«-Laufwerk das genannte Directory noch nicht, so wird es automatisch angelegt.

Wahlschalter

ALL - kopiert den kompletten Inhalt des Quell-Directory inklusive aller Sub-Directories und deren Inhalt in das Ziel-Directory. Wird dieser Schalter nicht gesetzt, werden nur die Files des Quell-Directory kopiert. Sub-Directories bleiben dabei unberücksichtigt.

QUIET - Ist dieser Schalter nicht gesetzt, gibt »Copy« Quell- und Zielpfad auf dem Monitor aus und informiert so den User über den Fortgang des Kopiervorgangs. Ist der Schalter »QUIET« gesetzt, werden diese Informationen unterdrückt. Kommt es während des Kopiervorgangs allerdings zu Fehlermeldungen, werden diese in jedem Fall angezeigt.

BUF=BUFFER - Mit dieser Option kann der während des ganzen Kopiervorgangs zur Verfügung stehende Disk-Cache beliebig vergrößert werden. Wie auch bei »Addbuffers« entspricht eine Buffereinheit 512 Byte, so daß mit der Angabe »BUF 300« der Cache-Speicher auf rund 150 KByte gesetzt wird. Voreingestellt ist die Bufferzahl 200, was bedeutet, daß dem »COPY«-Befehl im Normalfall 100 KByte als Kopierpuffer zugewiesen werden.

CLONE - Ein Kopiervorgang ohne diesen Schalter kopiert den Fileinhalt und die Schutzbits »rwd«, nicht jedoch zugehörige Filenotes und den Datum-/Zeiteintrag. Ein gesetzter »CLONE«-Schalter kopiert alles, was zu dem gewünschten File gehört, ausgenommen wird nur das Archive-Bit.

DATES - veranlaßt »COPY«, zusätzlich zu File-Inhalt und Schutzbits auch noch den Datum/Zeit-Eintrag zu kopieren.

NOPRO - beschränkt das Kopieren der Schutzbits auf jene, die von den jeweiligen Betriebssystem vorgegeben werden (z.B. unter WB_1.2 nur die Bits »rwd«).

COM - Im Normalfall (»COM« nicht gesetzt) werden Filenotes nicht mitkopiert. Wird die Option »COM« beim »Copy«-Aufruf ausgewählt, werden auch diese Kommentare mitkopiert.

Beispiel

```
copy * to prt:
```

macht aus Ihrem Amiga und Ihrem Drucker eine Speicherschreibmaschine. Diesen »Typewriter-Mode« beenden Sie mit <CTRL> <\>.

Beispiel

```
copy df2:s to ram:q CLONE ALL
```

kopiert alle Files und Unterverzeichnisse mit deren Inhalt, die im Verzeichnis »s« auf der Diskette in Laufwerk df2: eingetragen sind, in das Verzeichnis »q« des Device RAM: Ist in RAM: das Verzeichnis »q« noch nicht angelegt, wird dies durch den »Copy«-Befehl nachgeholt. Die Duplikate in RAM:q sind, abgesehen von einem eventuell vorhandenen Archive-Bit, exakte Kopien der einzelnen Quellfiles.

Beispiel

```
copy File_1|File_5|File_28 to prt:
```

kopiert nacheinander die Files »File_1«, »File_5« und »File_28« auf den Drucker (auch das geht). Ist in einem der Filenamen ein Leerzeichen enthalten, müssen die Filenamen in Anführungszeichen gebettet werden:

```
copy "File 1|File_5|File 28" to prt:
```

DATE Systemdatum setzen oder anzeigen

Schablone

Datum, Zeit, TO=VER/k

Syntax

Date [Datum] [Zeit] [TO/VER file]

»Date« setzt oder liest Systemzeit und/oder -datum. Um die aktuelle Systemzeit und das Systemdatum, also den Inhalt der Amiga-internen Zeitählung angezeigt zu bekommen, genügt die Eingabe von »DATE« <RETURN>. Soll besagter Inhalt nicht auf dem Bildschirm angezeigt, sondern z.B. in ein File eingetragen werden, ist eine der Optionen »TO« oder »VER« zu aktivieren.

Soll mit »Date« die Uhrzeit gesetzt werden, so ist das Format »HH:MM:SS« einzuhalten, also »Stunden:Minuten:Sekunden«. Ab Version 1.3 ist es nicht unbedingt erforderlich, führende Nullen mit einzugeben. Anstelle eines Datums kann auch der Wochentag eingegeben werden. Aus diesem wird dann das einzusetzende Datum ermittelt.

Es ist auf diese Art jedoch nicht möglich, ein Datum aus der Vergangenheit einzugeben. Jeder Wochentagsname bezieht sich für den »DATE«-Befehl immer auf die Zukunft.

Beispiel

```
Date 1-JAN-90
```

setzt nur das Systemdatum, beeinflusst nicht die Systemzeit.

Beispiel

```
date ver TXT_1:filename
```

leitet die Ausgabe des Befehls »DATE« nicht an den aktuellen Bildschirm weiter, sondern schreibt diese Ausgabe in das File »filename«. Ist das Zielfile noch nicht vorhanden, wird es automatisch angelegt, andernfalls gnadenlos überschrieben.

Beispiel

```
date 18:22
```

setzt die Systemzeit auf 18 Uhr, 22 Minuten, 0 Sekunden.

DELETE löscht eine beliebige Anzahl von Files oder Directories

Schablone

Files/..., ALL/s, Q=QUIET/s

Syntax

Delete <wildcards> [...] [ALL] [QUIET]

»Delete« löscht Files und/oder Directories. Maximal zehn Files im aktuellen Directory können mit einer einzigen Befehlszeile selektiv gelöscht werden. Beachten Sie bitte, daß Sie nur solche Files oder Directories löschen können, auf die kein Zugriff angemeldet ist. Ein solcher Zugriff kann auch ein aktuelles Assignment (siehe »ASSIGN«) sein.

Wahlschalter

ALL - Sind Sie sicher, daß sämtliche Einträge eines Directory inklusive Sub-Directories, Subsub-Directories, gelöscht werden sollen, so steht Ihnen der Schalter »ALL« zur Verfügung, um diese Absicht dem »Delete«-Kommando nahezubringen. Es ist allerdings nicht zulässig, ein virtuelles Device wie etwa »RAM:« als zu löschendes Ziel mit der »ALL«-Option anzusprechen!

Q=QUIET - Bei Gebrauch der »ALL«-Option und bei Mehrfachlösungen werden die Filenamen nach erfolgter Löschung als »...deleted« auf dem Schirm gemeldet. Wenn kein Bedarf an dieser Information besteht, kann sie durch Aktivieren des Schalters »Q«, der identisch ist mit dem Schalter »QUIET«, unterbunden werden. Fehlermeldungen (»...not found« etc.) werden natürlich trotzdem ausgegeben.

Beispiel

```
Delete #?
```

löscht alle Files und alle leeren Sub-Directories im aktuellen Directory, nicht aber Sub-Directories, in denen eventuell noch weitere Unterverzeichnisse existieren.

Beispiel

```
Delete File1 File3 File28 quiet
```

löscht die angegebenen Files nacheinander ohne Ausführungsmeldung

DIR alphabetisch sortierte Ausgabe des Inhaltsverzeichnisses von Directories, Devices und Volumes

Schablone

```
Dir, OPT/k, INTER/s, FILES/s, DIRS/s, ALL/s
```

Syntax

```
Dir [directory] [OPT ADFI] [ALL] [DIRS] [INTER] [FILES]
```

Einer der mächtigsten Befehle ist zweifellos »Dir«. Ohne jeden Zusatzbefehl werden damit Filenamen und Sub-Directory-Header des aktuellen Directory angezeigt. Files werden zweispaltig ausgegeben, Sub-Directories einspaltig mit dem Zusatz »(dir)«. Mit entsprechenden Pfadangaben kann der Inhalt jedes Directory aus jedem Directory heraus aufgelistet werden. Werden Muster vorgegeben, so erstreckt sich die Vergleichsfunktion auf Files und Directories.

Der Gebrauch des Schlüsselworts »OPT« mit anschließender Angabe einer den Optionen entsprechenden Buchstabenkombination ist ab Workbench-Version 1.3 ebenso zulässig wie der Gebrauch der vollständigen Schreibweise der einzelnen Wahlschalter.

Wahlschalter

FILES oder OPT F - listet nur die Namen von Dateien aus, nicht aber die Directory-Header. Die Directory-Struktur bleibt jedoch weiterhin erkennbar (Einrückungen).

DIRS oder OPT D - ist das genaue Gegenstück zur Option »FILES«. Filenamen werden unterdrückt, Directory-Header mit der Zusatzinformation »(dir)« ausgegeben.

ALL - zeigt alle Files und alle Directories und Sub-Directories - einfach jeden Eintrag, gleich, in welcher Tiefe der Directory-Struktur er angesiedelt ist. Durch die Einrückungen wird die Baumstruktur übersichtlich angezeigt.

INTER oder OPT I - ist der interaktive Modus von »Dir«. In Verbindung mit beliebigen anderen »Dir«-Schlüsselwörtern ist »INTER« eine der aktivsten Funktionen unter den »Dir«-Optionen. Nach jedem ausgegebenen File- oder Directory-Namen fordert das Programm durch Ausgabe eines Fragezeichens den User auf, den weiteren Ablauf der Geschichte zu bestimmen. Folgende Kommandos stehen zur Auswahl:

<RETURN> - Betätigen Sie nach dem Fragezeichen die Return-Taste, wird mit dem nächsten Eintrag (File oder Directory) fortgefahren.

B=BACK - führt aus einem Unterverzeichnis zurück in die nächsthöhere Etage der Directory-Struktur. In der obersten Ebene (im »Root«-Directory) wird damit der interaktive Modus abgebrochen.

DEL=DELETE - löscht das File, dessen Name soeben auf dem Monitor angezeigt wurde.

E=ENTER - Wird anstelle eines Filenamens der Header eines (Sub-) Directory angezeigt, steigen Sie durch die Eingabe von <E> <RETURN>, »ENTER« <RETURN> oder auch nur <RETURN> in dieses Directory ein. Leider wurde verabsäumt, bei zusätzlich aktiviertem »ALL«-Schalter eine

»Skip«-Funktion einzubauen. Sie können aber nach Anzeige des ersten Files mit das Sub-Directory wieder verlassen.

Q=QUIT - führt zu einem Abbruch der »Dir«-Session.

T=TYPE - listet das anstehende File auf dem aktuellen Ausgabegerät aus. Sie sollten allerdings darauf achten, nur ASCII-Dateien auflisten zu lassen, es gibt keine automatische Erkennung oder Umschaltung auf einen Hexdump-Mode.

C=COM - Wenn Sie nach dem Fragezeichen-Prompt »COM« oder auch nur »C« eingeben, wird in einer neuen Zeile der Prompt »Command ?« angezeigt. Sie befinden sich damit im Execute-Modus der »INTER«-Option, von dem aus Sie jedes beliebige Programm starten können. Wenn Sie wissen, was Sie wollen, können Sie unmittelbar nach dem »C«, also noch bevor Sie <RETURN> drücken, das aufzurufende Programm eingeben und dann erst <RETURN> drücken. Allerdings müssen Sie dieses Programm und den eventuell mit angegebenen Pfad in Anführungszeichen setzen, z.B. »com "SYS:c/execute s/startupll"«. Nach Abarbeitung des aufgerufenen Programms wird der »Dir«-Befehl von dem Punkt aus fortgesetzt, bei dem Sie in den Execute-Modus wechselten.

Beispiel

```
Dir df1: all dirs
```

zeigt alle Directory- und Sub-Directory-Header in Laufwerk df1: an. Die gleiche Wirkung hat die Eingabe von

```
dir df1: opt ad
```

Beispiel

```
Dir ?/#?b all inter
```

oder

```
Dir ?/#?b opt ai
```

durchsucht alle dem aktuellen Verzeichnis direkt entspringenden Verzeichnisse, deren Directory-Header nur ein Zeichen lang ist, nach Dateien, deren Name mit »b« endet und wechselt in den interaktiven Mode.

DISKCHANGE meldet dem Betriebssystem den Wechsel einer Diskette im Disketten-Laufwerk

Schablone

Dev/a:

Syntax

DiskChange <Drive: >

Amiga-DOS muß von jedem Wechsel eines Datenträgers informiert werden, um die jeweiligen Verwaltungstabellen zu aktualisieren. Geschieht dies, egal aus welchem Grund, vor einem Schreibzugriff einmal nicht, können Sie gleich den Befehl »Format« nachschlagen

Im Normalfall erfolgt die Information »Diskette wurde gewechselt« über die Betätigung eines im Laufwerk eingebauten Schalters und die »Diskchange«-Leitung, also hardwaremäßig. Viele 5¹/₄-Zoll-Laufwerke verfügen nicht über einen solchen Schalter, und das »FastFilesystem« der WB_1.3, das an sich nur für Festplatten ausgelegt wurde, fragt diese Leitung gar nicht ab. Für solche Fälle wurde der Befehl »DiskChange« implementiert.

Beispiel

Diskchange df2:

veranlaßt das Betriebssystem, die Diskette in Laufwerk df2: wieder neu zu aktualisieren.

ECHO gibt eine Zeichenkette auf dem aktuellen Ausgabegerät aus

Schablone

, NOLINE/s, FIRST/k, LEN/k

Syntax

Echo [" string "] [NOLINE] [FIRST n] [LEN n]

»Echo« dient der Ausgabe kurzer Zeichenketten, wobei diese Ausgabe auf jedem Output-Device inklusive Files erfolgen kann. Die meiste Anwendung findet »Echo« in Batch-Files, doch auch für das Senden von Escape-Sequenzen an den Drucker (Schrifteinstellung usw.) ist »Echo« die richtige Wahl. Die auszugebende Zeichenkette muß in Anführungszeichen eingebettet sein.

Wahlschalter

NOLINE - Mit diesem Schalter wird die im Standardfall durchgeführte Zeilenschaltung am Ende des auszugebenden Strings unterdrückt.

FIRST - definiert, ab welcher Zeichenposition innerhalb des Strings die Ausgabe erfolgen soll, also welche Zeichen ab String-Beginn unterdrückt und nicht ausgegeben werden sollen.

LEN - bestimmt, wie viele Zeichen des Strings ausgegeben werden sollen, wobei die Zählung am Ende des Strings beginnt und in Richtung String-Anfang verläuft, solange keine »FIRST«-Definition übergeben wird! Dann nämlich beginnt die Zählung mit dem ersten Zeichen, das von »FIRST« für die Ausgabe freigegeben wird.

Beispiel

Echo "Hello World!"

gibt den String »Hello World!« auf dem aktuellen Ausgabegerät aus.

ELSE leitet in Stapeldateien (=Batch-Dateien) jenen Befehlsblock ein, der abzuarbeiten ist, wenn eine vorausgegangene »IF«-Bedingung den Wert »falsch« ergibt

Schablone

else

Syntax

ELSE

»ELSE« wird nur in Zusammenhang mit einer »IF«-Abfrage verwendet und leitet jenen Befehlsblock ein, der abzuarbeiten ist, wenn die Bedingung, auf die mit »IF« geprüft wird, nicht zutrifft, also den Wert »FALSCH« oder »FALSE« ergibt. Im anderen Fall, also wenn die »IF«-Bedingung erfüllt ist, wird der zwischen »ELSE« und »ENDIF« liegende Befehlsblock ganz einfach übersprungen (siehe »IF«)

ENDCLI beendet einen CLI- oder Shell-Task

Schablone

EndCLI

Syntax

EndCLI

»EndCLI« beendet jenen CLI-Task, in dem es eingegeben bzw. aufgerufen wird. Ob nach dem Ende dieses Tasks auch das zugehörige Bildschirmfenster geschlossen werden kann oder nicht, hängt davon ab, ob noch andere Prozesse laufen, die aus diesem Task heraus gestartet wurden und die auf dieses Fenster Zugriffsrechte beim Betriebssystem angemeldet haben.

ENDIF symbolisiert das Ende eines »IF«-Bereichs in Batch-Files

Schablone

EndIF

Syntax

EndIF

»EndIF« markiert das Ende eines Blocks von Befehlen, der »unter Aufsicht« eines »IF«-Kommandos abzuarbeiten ist und auch mit einem »IF« beginnt (siehe »IF«).

ENDSKIP bricht einen »SKIP <Label >«-Suchlauf ab

Schablone

EndSKIP

Syntax

EndSKIP

»EndSKIP« bricht einen »SKIP Label«-Suchlauf ab und setzt den Fehlerlevel auf »WARN«. Da die Richtung, in der ein »SKIP Label«-Suchlauf stattfindet, vom User vorgegeben wird, läßt sich mit Hilfe des »EndSKIP«-Kommandos und des Fehlerlevels z.B. feststellen, von welchem Bereich einer Scriptdatei aus der »SKIP«-Suchlauf gestartet wurde. Nach Erreichen eines »EndSKIP« wird die Abarbeitung der Stapeldatei mit jener Zeile fortgesetzt, die auf »EndSKIP« folgt.

EVAL ermittelt einfache arithmetische und logische Ausdrücke

Schablone

Wert_1/a, Operand, Wert_2, TO/k, L=LFORMAT/k

Syntax

Eval <Wert_1 > [Operand] [Wert_2] [TO Datei] [LFORMAT Zeichenkette]

Mit »Eval« steht Ihnen ein elektronischer Abakus zur Verfügung. Einfachste arithmetische und logische Operationen mit maximal zwei Argumenten sowie Zahlenbasis-Transformationen sind mit »Eval« auf einfachste Weise durchführbar. Der große Vorteil dabei ist, daß diese Operationen auch innerhalb von Batch-Dateien erfolgen können und das Ergebnis dieser Operationen nicht nur auf dem aktuellen Ausgabegerät ausgegeben werden, sondern auch in ein zu definierendes File eingetragen werden kann.

»Eval« unterstützt die Zahlenbasen »dezimal«, »hexadezimal« und »oktal«. Auch Rechnungen mit alphanumerischen Zeichen sind ohne weiteres zulässig.

Die gewünschte Zahlenbasis wird dem »EVAL«-Befehl folgendermaßen bekanntgegeben:

Dezimal - ist die voreingestellte Basis des »EVAL«-Befehl, für dezimale Argumente ist keine besondere Kennzeichnung erforderlich.

Hexadezimal - muß mit, dem eigentlichen Argument vorangestellten, »0x« oder »#x« deklariert werden. Die richtige Bezeichnung für das Zahlensystem mit der Basis »16« wäre ja eigentlich »Sedezimal«, aber da sich der Ausdruck »Hexadezimal« schon so eingebürgert hat, soll er auch hier verwendet werden.

Oktal - Zahlenwerte mit oktaler Basis sind mit dem Prefix »0« oder »#« zu versehen.

Alphanumerische Zeichen - werden mit einem vorangestellten Hochkomma an »EVAL« übergeben. Dieses Hochkomma erhalten Sie bei eingeschalteter deutscher Tastatur mit <ALT> <ä>.

Folgende Operationen sind implementiert:

Addition	+
Subtraktion	-
Multiplikation	*
Division	/
Negation	
NOT	~
AND	&
ODER	
Exklusiv ODER	XOR
verschieben nach links	<<
verschieben nach rechts	>>
Modulo	MOD
Bitvergleich	eqv

Wahlschalter

TO - führt zur Umleitung der Ergebnisausgabe in ein File anstatt auf den Bildschirm.

L=LFORMAT - beeinflusst Zahlenbasis und Art der Ausgabe des Ergebnisses. Der Vorgabewert, der ohne die Anwahl von »LFORMAT« verwendet wird, ist das Dezimalsystem.

Mit »%X« wählen Sie die hexadezimale Zahlenbasis als Ausgabeform, mit %O die oktale, mit %N die dezimale und mit %C eine alphanumerische Ausgabe. Bei hexadezimaler oder oktaler Ausgabebasis muß die Anzahl der auszugebenden Stellen an »EVAL« übergeben werden. »%X4« begrenzt die Ausgabe auf vier Zeichen im Hexadezimalsystem.

Beispiel

```
eval 'A
ergibt die Ausgabe
```

```
65
also den ASCII-Wert von »A« in dezimaler Form.
```

Beispiel

```
eval 240 * 12 lformat="%N*N"
zeigt als Ergebnis
```

```
2880
```

Die Zeichenfolge » "%N*N" « bedeutet, daß die Ausgabe explizit im dezimalen Zahlensystem gewünscht wird. Der Cursor geht nach der Ausgabe in die nächste Zeile. Versuchen Sie bitte nie, eine dezimale Ausgabe mit einer Stellenanzahl zu übergeben! Die Befehlszeile

```
eval 240 * 12 lformat="%N5*N"
hätte
28805
```

zur Folge - absoluter Nonsens!

Beispiel

```
eval #xDe / 0xE lformat="%x3_
*n" to ram:evaldemo
schreibt in das File »evaldemo« den String
00F
```

Auch hier dürfen Sie auf keinen Fall mehr als neun Stellen für die Ausgabe vorschreiben!

Es scheint allerdings, als wäre der »EVAL«-Befehl noch im Software-Stadium »alpha« (voller Bugs, aber wenigstens vorhanden). Daß mit alphanumerischen Zeichen keine sinnvollen Operationen möglich sind, ist schon mehr als ein Schönheitsfehler! So führt die Eingabe von

```
eval 'K + 'A
ebenso zur Ausgabe von
75
```

wie die Eingabe von

```
eval 'K + %N6
```

Daß die »modulo«-Operation immer Null ergibt, spricht auch nicht

gerade für Commodore. Na, vielleicht wird's ja noch.

Anmerkung: An dieser Stelle muß betont werden, daß für diesen Artikel die Workbench_1.3.2, Version 34.28, als Basis diente, die als Letztstand gilt. Überraschenderweise hat der »EVAL«-Befehl der Workbench_1.3 Version 34.20, keine Probleme mit Zeichen- und Modulo-Verarbeitung!

EXECUTE bringt die in einer Batch-Datei eingetragenen Befehle zur Ausführung

Schablone

[Filename] [Argumente]

Syntax

Execute [Filename] [Argumente]

»Execute« veranlaßt die Abarbeitung der Befehle, die in dem Scriptfile <Filename> eingetragen sind. Nach dem letzten Befehl, also am Ende der Batch-Datei, wird die Kontrolle wieder an jenen Task zurückgegeben, von dem aus das »Execute«-Kommando gestartet wurde. Die Abarbeitung des Scriptfiles kann jederzeit durch die Tastenkombination <CTRL-D> abgebrochen werden.

»Execute« ist jedoch nicht nur ein stures Startkommando für Scriptfiles. Die in »Execute« implementierten Kommandos (auch »Direktiven« genannt) erlauben dem User, Parameter zu übergeben, Vorgabewerte zu setzen usw. Jedes dieser Kommandos beginnt mit einem Punkt, genannt »DOT«. Damit »Execute« seine Kommandos sicher erkennen kann, muß dieser Punkt am Anfang einer Zeile stehen. Unmittelbar an diesen Punkt anschließend steht dann das Kommandowort. Es ist wichtig, keinerlei Zwischenraum zwischen dem »DOT« und dem Kommandowort zu haben. »Remark«-Zeilen werden am einfachsten mit einem Semikolon (Strichpunkt) als erstes Zeichen in der Zeile eingeleitet.

In manchen Fällen braucht »Execute« die Möglichkeit, zu internen Zwecken ein temporäres File anzulegen, vorzugsweise in Directory oder Device »T.«.

Direktiven:

.k=key - Die »key«-Direktive und deren ebenso zulässige Kurzform »k« wird dazu benutzt, dem »execute«-Programm die Bezeichnung eines Parameters bekanntzugeben, der beim Start an das Programm übergeben wird. Der übergebene Parameter wird dann anstelle des im Scriptfile zwischen spitzen Klammern (»<« und »>«) stehenden Parameternamens eingetragen und kann damit beliebig verwendet werden. Werden Übergabeargumente erwartet, aber nicht übergeben, so ersetzt »execute« die Parameterplatzhalter im Script durch Leer-Strings. Die »key«-Anweisung muß als erste Zeile im Scriptfile eingetragen werden, andernfalls werden übergebene Parameter ignoriert.

.def=default - Wie der Name schon andeutet, setzt diese Direktive einen Vorgabewert. Wird eine mit »key« angemeldete und mit »def« vorbesetzte Variable nicht übergeben, wird anstelle des Parameternamens kein Leer-String wie bei »key« eingetragen, sondern eben der Vorgabewert. Wie bei »key« wird auch bei »def« das gesamte Script nach dem jeweiligen Parameternamen abgesucht und dieser Parameternamen überall gegen den Default-Wert ausgetauscht.

Eine zweite Art, Vorgabewerte einzutragen, ist das Dollarzeichen. Diese Methode unterscheidet sich in zwei Dingen vom Direktiv »DEF«. Erstens wirkt das Dollarzeichen nur lokal, während die »DEF«-Anweisung für das ganze Script gilt. Zweitens hat die »DEF«-Anweisung eine höhere Priorität und überlagert eine eventuelle »\$«-Definition gleichen Namens.

Kommandos zum Ändern interner Steuerzeichen:

Um die Flexibilität zu erhöhen, wurde von den Commodore-Programmierern die Möglichkeit vorgesehen, bei Bedarf die »Execute«-internen Steuerzeichen ».«, »\$« usw. frei wählen zu können. Folgende Anweisungen sind dafür vorgesehen:

.bra

Mit diesem Kommando-Paräthen lassen sich anstelle der vorgegebenen spitzen Klammern beliebige andere Zeichen für die Eingrenzung von Parametervariablen definieren. »bra« modifiziert " < ", »ket« ändert " > ".

.dot - ersetzt das Direktiven-Prefix ».« durch ein beliebiges anderes Zeichen.

.dol=dollar - ersetzt »\$« durch ein anderes, frei wählbares Zeichen.

Änderungen dieser internen Steuerzeichen sind auch innerhalb eines Scripts erlaubt und gelten ab dem Moment der Änderung für den Rest des Scripts. **VORSICHT BEI VERWENDUNG VON »SKIP BACK«.**

Beispiel

```
.key schalte
if " <schalte> " EQ "schalter"
echo "schalter gesetzt"
else
echo " <schalte$Das> ist
falsch! Es muß SCHALTER heißen"
endif
```

Wenn diese Zeilen im Scriptfile »SCRIPT« stehen und Sie dieses File durch die Eingabe von `execute script kanal` abarbeiten lassen, wird »kanal« als Ersatz für <schalte> in der Batch-Datei (natürlich nur im RAM) eingetragen. Demzufolge ist die »IF«-Bedingung nicht erfüllt, und es erfolgt die Ausgabe

```
kanal ist falsch! Es muß SCHALTER heißen
```

Bei einem Aufruf ohne Übergabe eines Arguments, wenn Sie also lediglich

```
execute script
```

eingeben, tritt die Default-Funktion mit dem Dollarzeichen in Kraft. Natürlich ist die »IF«-Bedingung wieder nicht erfüllt, aber der Vorgabewert für »schalte« ist jetzt die Zeichenfolge nach dem »\$«-Zeichen. Daraus resultiert folgende Ausgabe:

```
Das ist falsch ! Es muß SCHALTER heißen
```

FAILAT setzt den Fehlerlevel auf den gewünschten Wert

Schablone

ReturnCodeLimit

Syntax

Failat <Fehler_Limit> Bei der Abarbeitung von Batch-Dateien führt je-

der Fehlerwert, der über dem momentan zulässigen Level liegt, zum vollständigen Abbruch der Stapelverarbeitung. Der Zahlenwert solcher Fehlerückmeldungen wird aber oft als Entscheidungskriterium verwendet, z.B. in »IF«-Abfragen. Es ist daher öfters notwendig, den aktuell zulässigen Fehlerlevel zu erhöhen, um einen Abbruch des »Execute«-Programms zu vermeiden und die bei gewollten Tests als Antwort zurückgegebenen Fehlerwerte zur Auswertung und Entscheidung benutzen zu können. Mit »FailAt <Wert>« setzen Sie den für die Abbruchentscheidung relevanten Fehlerlevel auf einen Wert Ihrer Wahl.

Beispiel

```
FailAt 21
```

setzt den Abbruchlevel auf den aktuellen Wert 21. Nur Fehlerwert-Rückgaben, die größer als der Wert »20« sind (Sintfluten, Erdbeben etc.), führen jetzt noch zu einem Abbruch der Stapelverarbeitung.

FAULT gibt den Klartext zu einer gegebenen Fehlernummer aus

Schablone

Fehler-Nummer/...

Syntax

Fault <Amiga-DOS-Fehlernummer> [...]

Jedem Fehler, den das Betriebssystem mit seinen Kontrollroutinen entdeckt, ist in Amiga-DOS eine Kennnummer zugeordnet. Einigen dieser Fehler wurde von den Amiga-Entwicklern eine Klartexterklärung mit auf den Weg gegeben. Diese erklärenden Texte lassen sich mit »Fault« abrufen.

FF beschleunigt Textausgabe

Schablone

Switch/s, Font/a

Syntax

FF [-0 I-N] [Font]

»FF« ist die Abkürzung von Fast-Fonts. »FF« ersetzt die Standardroutinen für die Textausgabe durch deutlich schnellere. Aktiviert werden diese schnellen Routinen mit

```
FF -0
```

Die »alten«, langsameren Routinen schalten Sie mit

```
FF -N
```

wieder ein. Was im Handbuch zur Workbench_1.3 nicht dokumentiert ist, ist die Funktion »Load Font«. Damit schalten Sie auf einen anderen Font um. Für den Font »opal« beispielsweise lautet die Syntax

```
FF opal.font
```

wenn sich der Font »opal« im »Fonts«-Directory der Boot-Diskette befindet. Sie können aber auch auf externe Fonts zurückgreifen:

```
FF df2:fontsammlung/myfont.  
font
```

Zurückschalten auf den »Original-Font« »topaz80« erfolgt durch die Eingabe des entsprechenden Font-Namens oder durch Anwahl eines nichtvorhandenen oder unzulässigen Fonts. Unzulässig sind alle Proportional-Fonts und überdies alle Fonts, deren Breite und Höhe ungleich »acht« ist und bei denen Abstand und Breite inklusive Zwischenraum nicht »acht« ist. Ein Font »paßt« dann, wenn im »FED« (»FontEditor«) folgende Werte eingestellt sind: »Kern«=00, »Space«=8, »Width«=8, »BLine«=6, »XSize«=8 und »Y-Size«=8. Wie gesagt, »Proportional« darf nicht eingeschaltet sein! Der Über-Gag daran ist, daß diese Font-Änderung für alle(!) Textausgaben gilt, also auch innerhalb von Programmen, die eine Font-Änderung gar nicht zulassen! Der Haken an der Sache ist, daß jeder neue Zeichensatz RAM okkupiert und erst wieder bei einem Systemneustart freigibt.

FILENOTE bindet einen Kommentar an ein File

Schablone

```
File/a, Comment/a
```

Syntax

```
Filenote <Filename> <"Kommentar-String">
```

Mit »FileNote« sind Sie in der Lage, jedem beliebigen File einen Kommentar-String zuzuordnen und diese Zuordnung auch wieder aufzuheben. Der Kommentar-String darf nicht länger als 79 Zeichen sein!

Beispiel

```
FileNote ram:Batch "Das ist  
eine Notiz"
```

bindet an das in RAM: stehende File »Batch« den kleinen Kommentar »Das ist eine Notiz«.

Beispiel

```
LIST ram:Batch
```

gibt jetzt nicht nur die gewohnten Daten zu »Batch« aus, sondern auch den Kommentar »Das ist eine Notiz«.

Beispiel

```
FileNote ram:Batch ""
```

entfernt die dem File »Batch« zugeordnete Notiz.

GETENV gibt den Inhalt einer Umgebungsvariablen aus

Schablone

```
Name/a
```

Syntax

```
GetEnv
```

```
[Name_der_Umgebungsvariable]
```

»GetENV« gibt den Inhalt einer angeforderten Umgebungsvariablen auf dem aktuellen Ausgabegerät aus, wenn der Aufruf aus dem CLI erfolgt. »GETENV« kann aber auch in Scriptdateien eingesetzt werden. Angenommen, die Datei »Dat1« hat folgenden Inhalt:

```
echo "ich setze die Umgebungs-  
variable >>ALPHA<< auf den  
Wert >>7<<"  
setenv alpha 7
```

Wenn Sie diese Minidatei abarbeiten lassen, legen Sie im »ENV:«-Handler (der zur Zeit ja leider noch durch ein Verzeichnis in »RAM:« substituiert werden muß) eine Variable namens »ALPHA« an und weisen dieser den Wert »7« zu. In einem zweiten Programm, nennen wir es »BETA«, lassen Sie je nach Inhalt dieser Variable gewisse Verzweigungen ausführen:

```
echo "ich prüfe den Inhalt der  
Variablen >>ALPHA<<"  
echo "ALPHA hat den Wert "  
noline  
getenv alpha  
if val $alpha ge 5  
echo "ALPHA ist größer als 4  
!"  
else  
echo "ALPHA ist kleiner als 5"  
endif
```

ICONX macht Programme »Work-Bench«-tauglich

Schablone

```
<Filename>
```

Syntax

```
ICONX <Filename>
```

Manche Programme haben kein Icon und können daher nicht von der Workbench aus durch einfaches Anklicken gestartet werden. Bei vielen Programmen genügt die Zuordnung eines passenden Icons, und die Sache ist geritzt. Ebenso viele Programme aber stellen bestimmte Ansprüche, die durch ein »Tool«-Icon alleine nicht zu befriedigen sind. Da hilft »ICONX«.

Um z.B. von der Workbench aus den CLI-Befehl »list« aufrufen zu können, genügt es nicht, ein »TOOL«-Icon ins »C«-Verzeichnis zu kopieren und in »list.info« umzubenennen (ganz abgesehen davon, daß Sie zuvor ein »Drawer«-Icon in das Hauptverzeichnis kopieren und in »C.info« umbenennen müßten, um überhaupt an das »list«-Icon herankommen zu können). Wenn Sie wollen, probieren Sie diesen Holzweg doch mal - GURU läßt grüßen. Wie Sie richtig vorgehen, ist am Ende des Artikels über die Workbench in diesem Heft ausführlich beschrieben.

IF Prüf- und Verzweigungsfunktion für Scripts bzw. Stapeldateien

Schablone

```
NOT/s, WARN/s, ERROR/s, FAIL/s, ,  
EQ/k, GT/k, GE/k, VAL/s, EXISTS/k
```

Syntax

```
IF [NOT] [WARN] [ERROR] [FAIL]  
<Zeichenfolge1> EQIGTGE  
<Zeichenfolge2>] [VAL] [EXISTS  
<Datei>]
```

Wenn die Überprüfung der angegebenen Parameter Gleichheit ergibt (TRUE), werden die Kommandos, die in den Zeilen unmittelbar nach »IF« stehen, ausgeführt. Eingegrenzt wird dieser Befehlsblock entweder durch ein »ELSE«- oder durch ein »ENDIF«-Kommando. Ergibt die Überprüfung der Parameter aber Ungleichheit (keine Übereinstimmung - FALSE) und ist ein »ELSE«-Kommando in diesen »IF«-Block eingebunden, so wird der Befehlsblock zwischen »IF« und »ELSE« über-

sprungen und es werden jene Kommandos ausgeführt, die zwischen »ELSE« und »EndIF« stehen. Ist kein »ELSE« vorgesehen, wird die Abarbeitung der Scriptdatei mit jenem Befehl fortgesetzt, der an die »EndIF«-Anweisung anschließt.

Bei Vergleichen werden im Normalfall Zeichenketten miteinander verglichen, doch ist bei Anwendung des »VAL«-Kommandos auch ein Vergleich zwischen zwei Zahlenwerten zulässig. Der Inhalt von Umgebungsvariablen kann als Vergleichswert herangezogen werden. Ein Beispiel dafür und für die Schreibweise (dem Variablennamen ist ein Dollarzeichen voranzustellen) finden Sie bei »GETENV«.

WARN - ist TRUE, wenn das letzte abgearbeitete Kommando einen Return-Code größer/gleich 5 an das Betriebssystem übergab.

```
IF WARN
```

```
....
EndIF
```

ERROR ist »wahr«, wenn das letzte abgearbeitete Kommando einen Return-Code größer/gleich 10 zurückgab.

```
IF ERROR
```

```
.....
EndIF
```

FAIL ist TRUE bei einem Return-Code größer/gleich 20.

```
IF FAIL
```

```
....
EndIF
```

EQ vergleicht zwei Argumente, die durch die Option »EQ« verbunden werden. Sind die Argumente gleich, wird das Ergebnis TRUE, bei Ungleichheit FALSE.

```
IF <Argument_1> EQ <Argument_2>
```

```
....
EndIF
```

GE vergleicht zwei Argumente und wird TRUE, wenn »Argument_1« größer oder zumindest gleich groß ist wie »Argument_2«. Ist »Argument_2« aber größer als »Argument_1«, so ist das Ergebnis FALSE.

```
IF <Argument_1> GE <Argument_2>
```

```
....
```

```
EndIF
```

GT überprüft zwei Argumente und wird nur dann »wahr«, wenn »Argument_1« größer als »Argument_2« ist.

```
IF <Argument_1> GT <Argument_2>
```

```
.....
EndIF
```

EXISTS dient der Überprüfung, ob ein bestimmtes Device oder Directory vorhanden (also dem Betriebssystem bekannt) ist. Ist das Device angemeldet, gibt »EXISTS« den Wert TRUE zurück.

```
IF <Device:> EXISTS
```

```
.....
EndIF
```

Modifikatoren

Um möglichst alle Zustände ohne große Umwege abfragen zu können, wurden dem »IF«-Befehl noch zwei »Modifiers« zur Seite gestellt:

NOT - invertiert die Aussage der Überprüfung. Aus einem FALSE wird ein TRUE und natürlich umgekehrt.

```
IF <Argument_1> NOT GT <Argument_2>
```

```
Mounted disks:
```

Unit	Size	Used	Free	Full	Errs	Status	Name
DF3:	880K	1259	499	71%	0	Read/Write	Whoopie_Backup
DF2:	880K	1603	155	91%	0	Read/Write	Whoopie
DF1:	880K	1705	53	96%	0	Read/Write	Workbench1.3D
DF0:	880K	1602	156	91%	0	Read Only	AMIGA-SH
RAM:	217K	434	0	100%	0	Read/Write	RAM DISK

```
Volumes available:
```

```
Workbench1.3D [Mounted]
RAM DISK [Mounted]
Whoopie_Backup [Mounted]
Whoopie [Mounted]
AMIGA-SH [Mounted]
```

```
.....
EndIF
```

VAL - Vergleiche werden im Normalfall zwischen Zeichenketten angestellt. Mit dem »VAL«-Modifikator wird auf »Zahlenvergleich« umgeschaltet.

```
IF VAL <Argument_1> EQ <Zahl>
```

```
.....
EndIF
```

INFO zur Ausgabe von Informationen über angemeldete Devices
Schablone
Device

Syntax

Info [Device]

Das Kommando »Info« zeigt wichtige Informationen zu jedem angemeldeten Device (Diskettenstationen, Hard-Disks, RAM-Disks etc.). »Info« gibt die unformatierte Kapazität an, also sozusagen den Rohwert. »Unit« enthält die Laufwerksbezeichnung, »Size« die maximale Speicherkapazität des nebenstehenden Laufwerks in KBytes. »Used« listet die Anzahl der durch Dateien und Verwaltungstabellen belegten Blöcke (1 Block = 512 Byte), »Free« gibt die Anzahl der noch nicht belegten Blocks an. Die beiden Boot-Blocks werden dabei nicht berücksichtigt. »Full« gibt den Füllgrad der Diskette in Prozent an, »Errs« soll (!) die Anzahl der seit der letzten Formatierung festgestellten Soft-Errors enthalten - war bei mir noch immer Null. »Status« und »Name« sind klar.

Beispiel

```
info
```

zeigt Informationen über alle verfügbaren Volumes und Laufwerke inklusive RAM-DISKS:

Beispiel

```
info df2:
```

zeigt folgende Informationen an:

```
Mounted disks:
```

Unit	Size	Used	Free	Full	Errs	Status	Name
DF2:	880K	1603	155	91%	0	Read/Write	Whoopie

INSTALL zur Installation und Überprüfung des Boot-Blocks

Schablone

Drive/a, NOBOOT/s, CHECK/s

Syntax

Install [DRIVE] <df0: ldf1: ldf2: ldf3:> [NOBOOT|CHECK]

»Install« ist jenes Programm, das dem Amiga-User den bequemen Zugriff auf den Boot-Block der Diskette ermöglicht. In erster Linie dient es dazu, eine Diskette »bootfähig« zu machen, dient aber auch zum gezielten Löschen und Überprüfen von Bootblöcken.

Wahlschalter

NOBOOT - löscht den Inhalt des Boot-Blocks der spezifizierten Diskette und trägt nur die »DOS«-Kennung, nicht aber den Boot-Code für den Systemstart ein.

CHECK - vergleicht den Inhalt des Boot-Blocks Byte für Byte mit dem Originalcode, der bei einem »Install«-Kommando auf die Diskette geschrieben wird. »Check« ist recht nützlich zur schnellen Überprüfung eines Boot-Blocks auf Viren. Die Option »Check« liefert einen »Return-Code«, der in Batch-Dateien abgefragt werden kann:

Standard-Boot-Block - Fehlercode = 0

kein Boot-Code - Fehlercode = 5

Non-Standard-Bootblock - Fehlercode = 10

Beispiel

Install df2:

schreibt den Boot-Code auf die Diskette in df2:

Beispiel

Install df0: noboot

überschreibt den Inhalt des Bootblocks der Diskette in DF0: mit einer »DOS«-Kennung, nicht aber mit dem Boot-Code.

Beispiel

Install df1: check

überprüft den Boot-Block der Diskette in df1: und meldet das Ergebnis dieser Überprüfung an den User.

JOIN verbindet ASCII-Files

Schablone

Files/..., TO=AS/a

Syntax

Join <Datei1 > <Datei2 > [...] <AS Summen-File >

Dieses Programm verbindet bis zu 15 kleine ASCII-Files zu einem großen, wobei anstelle eines Summenfiles auch ein Device (z.B. der Drucker) angewählt werden kann. Leider ist »Join« nicht in der Lage, Jokerzeichen zu verarbeiten - Sie müssen jede Datei mit vollem Namen angeben. Sollten die zu verbindenden Dateien nicht im aktuellen Directory abgelegt sein, muß auch noch bei jedem einzelnen File der volle Pfad angegeben werden.

Beispiel

```
Join File_4 File2 Testfile18
as ram:Summenfile
```

verbindet die Files »File_4«, »File2« und »Testfile18« und legt das neue Summenfile in der Datei »Summenfile« in RAM: ab. Die jeweiligen Quellfiles bleiben natürlich in ihrem Originalzustand erhalten.

LAB definiert ein Label, das von »SKIP« verwendet werden kann

Schablone

Label__Name

Syntax

LAB Label__Name

»LAB« dient zur eindeutigen Deklaration eines Labels innerhalb einer Stapel-Datei (Batch-Datei). Das damit gekennzeichnete Label dient dann als Ansprungs-Ziel für den »SKIP«-Befehl.

Beispiel

```
.....
SKIP Label__Name
echo "Dieser String wird nicht angezeigt"
LAB Label__Name
echo "von SKIP hierher"
```

LIST zeigt Informationen zum Inhalt eines Directory

Schablone

Dir, P=PAT/k, KEYS/s, DATES/s, NO-DATES/s, SUB/k, SINCE/k, TO/k, UPTO/k, QUICK/s, BLOCKS/s, NO-HEAD/s, FILES/s, DIRS/s, LFOR-MAT/k

Syntax

List [Directory |File-Name] [PAT Muster] [KEYS] [DATES] [NODATES] [SUB String] [SINCE Datum] [TO Da-

tei] [UPTO Datum] [QUICK] [BLOCK] [NOHEAD] [FILES] [DIRS] [LFOR-MAT string]

»List« zeigt den Inhalt eines Directory - entweder nur des einen, dessen Name beim Aufruf übergeben wurde oder des aktuellen. Bereits der Umfang von »Schablone« und »Syntax« sagt eindeutig: »LIST« ist einer der umfangreichsten und mächtigsten Befehle der Amiga-DOS.

»List« ohne weiteren Zusatzbefehl listet die Einträge des gewünschten Directory folgendermaßen auf:

Voller Name des gelisteten Directory, Wochentag und Datum der Durchführung, Filename, Filegröße in Blocks, Schutzbits, Datum und Uhrzeit des letzten Schreibzugriffs: File-Note

Directory-Name, Directory-Kennung »Dir«, Schutzbits, Datum und Uhrzeit des letzten Schreibzugriffs auf eine der beinhalteten Dateien

Anzahl Dateien, Anzahl Directories, Anzahl Blöcke, Anzahl Byte

Wahlschalter

PAT=P - gibt ein Muster vor, nach dem die zu listenden Files und Directories selektiert werden.

SUB - sucht nach dem Auftreten des Muster-Strings irgendwo in File- oder Directory-Namen innerhalb des durchsuchten Directory und listet die entsprechenden Einträge auf.

SINCE - zeigt nur jene Einträge an, deren letzter Schreibzugriff nicht vor dem angegebenen Datum erfolgte. Beachten Sie bitte, daß eine Änderung der Filenotes nicht zu einer Änderung des Filedatums führt.

UPTO - ist das Gegenstück zu »SINCE«. Mit »UPTO« werden nur jene Einträge gelistet, deren letzter Schreibzugriff spätestens zum angegebenen Zeitpunkt erfolgte.

FILES - zeigt nur Fileeinträge an und unterdrückt die Anzeige von Sub-Directories

DIRS - hingegen zeigt nur Sub-Directory-Header an und unterdrückt die Ausgabe von Filedaten.

TO - Mit »TO« wird ein Device oder eine Datei angegeben, an die das Ergebnis des »LIST«-Kommandos umgeleitet wird.

list df1: TO RAM:listfile

gibt die Informationen zum Root-Directory der Diskette in df1: nicht auf dem Monitor aus, sondern trägt sie in das File »listfile« als ASCII-Text ein. Existiert das Zielfile noch nicht, wird es automatisch angelegt; ist es bereits vorhanden, wird sein Inhalt überschrieben.

DATES - schaltet das Ausgabeformat des »Last Write«-Datums auf das aktuelle Format um. Das heißt, daß Aussagen wie »Yesterday« oder »Sunday« durch das entsprechende Datum ersetzt werden.

NODATES - unterdrückt die Ausgabe des Datum- und Zeitstempels.

QUICK - beschränkt die Ausgabe auf die File- und Directory-Namen.

KEYS - zeigt zu allen anderen Informationen auch noch die Blocknummer des File- oder Directory-Headers an (sehr nützlich für das Bearbeiten von Disketten mit einem Disk-Monitor).

BLOCK - Die Voreinstellung des »LIST«-Befehls sieht vor, die Größe von Files in Bytes anzugeben. Die »BLOCK«-Option bewirkt, daß diese Größenangabe des Files in Blöcke umgerechnet wird.

NOHEAD - unterdrückt die Ausgabe der Kopf- und Fußzeile des »List«-Ergebnisses.

LFORMAT - Mit »LFORMAT« wird der Inhalt der »List«-Ausgabe beeinflusst. Dies ist besonders dann von Vorteil, wenn mittels »LIST« ein Batch-File angelegt werden soll, die Ausgabe also in ein ASCII-File umgeleitet wird. Mit »LFORMAT« können beliebige Pre- und Postfix-Zeichenketten automatisch an den Filenamen angehängt werden. Um den Luxus noch auf die Spitze zu treiben, steht Ihnen die Wahl frei, aus dem »List«-Prozeß nur die Filenamen zu übernehmen oder ob der komplette Pfad zur gelisteten Datei in das ASCII-File eingetragen werden soll. Dazu steht die »%S«-Anweisung zur Verfügung, die folgendermaßen anzuwenden ist:

```
"%S"      Filename
"%S%S"    Pfadangabe/File-
           name"
"%S%S%S"  Pfadangabe/File-
           namePfadangabe"
"%S%S%S%S" Pfadangabe/File-
           namePfadangabe/
           Filename"
```

»%S« steht jeweils für Pfad- oder Filename, je nach Position in der Reihe. Mehrere »%S« können durch beliebige Zeichenketten getrennt werden. Um mit dieser Option etwas anfangen zu können, muß die Ausgabe natürlich entweder in ein File umgeleitet oder mit einer »PIPE«-Anweisung weiterverarbeitet werden.

Beispiel

```
list df1: pat "List#?"
```

```
Directory "df1:" on Sunday 22-
Apr-90
ListingSpalten 134 ----rwd
04-Aug-89 21:26:17
ListKopf 19 ----rwd
19-Jul-89 22:16:24
2 files - 4 blocks - 153 bytes
```

Beispiel

```
list df1: sub "alt"
Directory "df1:" on Sunday 2
2-Apr-90
TextSpalten 159 ----rwd
04-Aug-89 18:58:15
: Das ist ein Kommentar zur
Datei >>TextSpalten<<
ListingSpalten 134 ----rwd
04-Aug-89 21:26:17
2 files - 4 blocks - 293 bytes
```

Beispiel

```
list df1: since "04-aug-89"
Directory "df1:" on Sunday 2
2-Apr-90
.info 16 ----rwd
10-apr-90 20:49:04
TextSpalten 159 ---rwd
04-Aug-89 18:58:15
: Das ist ein Kommentar zur Datei
»TextSpalten«
ListingSpalten 134 ---rwd
04-Aug-89 21:26:17
AMIGA-Sonderheft 12 Dir---rwd
Today 01:59:12
Konklu 8037---rwd
29-Aug-89 23:01:28
: Das ist ein Kommentar zur Datei
»Konklu«
5 files - 1 directories - 28
blocks - 9085 bytes
```

Beispiel

```
list df1: upto "04-aug-89"
Directory "df1:" on Sunday 22-
Apr-90
TextSpalten 159 ----
rwd 04-Aug-89 18:58:15
: Das ist ein Kommentar zur
Datei >>TextSpalten<<
ListingSpalten 134 ----
rwd 04-Aug-89 21:26:17
Disk.info 606 ----
rwd 23-Jun-89 23:32:43
ListKopf 19 ----
rwd 19-Jul-89 22:16:24
4 files - 9 blocks - 918 bytes
```

Beispiel

```
list df2:fonts/diamond
LFORMAT "Rename %S%S TO
%S%S.c" TO RAM:RenameBatch
```

legt in RAM: das File »Rename-Batch« an, in dem dann folgende Zeilen stehen:

```
Rename df2:fonts/diamond/12 t-
o df2:fonts/diamond/12.c
Rename df2:fonts/diamond/20
to df2:fonts/diamond/20.c
```

womit komfortabel ein Batchfile zum Umbenennen der beiden Files »12« und »20« entstanden wäre. Daß der Komfort durch die Umleitung in die Pipeline noch zu steigern ist, bedarf wohl keiner weiteren Erwähnung.

Beispiel

```
list df1: blocks nodates
nohead
```

ergibt eine Auflistung aller Files und Directories, ohne Kopf- und Fußzeile, ohne Datumsstempel, mit Filegröße in Block

```
c Dir ---rwd
devs Dir ---rwd
fonts Dir ---rwd
l Dir ---rwd
libs Dir ---rwd
POPCLI 10 ---rwd
Prefs Dir ---rwd
s Dir ---rwd
Shell 1 ---rwd
system Dir ---rwd
t Dir ---rwd
Utilities Dir ---rwd
```

LOADWB lädt und startet die Workbenchoberfläche des Amiga Schablone DELAY/s, -DEBUG/s

Syntax

LOADWB [DELAY] [-DEBUG]

»LoadWB« ist jenes Steuerprogramm, das die Intuition-Workbench auf den Bildschirm bringt. Wenn die Workbench einmal geladen und gestartet ist, kann sie mit keinem Kommando mehr abgeschaltet werden. Die von der Workbench belegten Ressourcen (Speicher, Libraries, ...) werden erst mit dem nächsten Systemneustart wieder freigegeben.

Wahlschalter

DELAY - gibt die Kontrolle an das aufrufende Programm (z.B. die »Startup-Sequence«) erst dann zurück, wenn der Initialisierungsprozeß der Workbench (Icons einlesen und anzeigen,...) komplett abgeschlossen ist. Ohne den Zusatz »DELAY« geht besagte Kontrolle bereits unmittelbar nach Laden und Starten retour.

DEBUG - spendiert der Workbench ein zusätzliches (viertes) Menü in der Menü-Leiste, das aber keinen Namen hat. Dieses vierte Menü ist wirklich nur für Programmierer interessant. Der erste Sub-Menüpunkt namens »Debug« aktiviert die Verbindung zu einem an die RS232-Schnittstelle angeschlossenen Terminal. Sollte keines angeschossen sein, dürfen Sie den berühmten »Dreifingergriff« üben. Der zweite Menüpunkt ist »Flushlibs«. Damit werden im Speicher residente Libraries entfernt.

LOCK Schreibschutz für FFS-Festplatten

Schablone

Laufwerk/a, ON/s, OFF/s,
<Paßwort>

Syntax

LOCK <Laufwerk> [ON | OFF]
[<Paßwort>]

Das Kommando »Lock« ist nur auf Laufwerke anwendbar, die mit dem »FastFilesystem« (»FFS«) betrieben werden. Dieses Filesystem wurde zwar ursprünglich nur für Hard-Disks entwickelt, kann aber auch für Disketten-Laufwerke und RAM-Disks verwendet werden. Mit »Lock« wird ein Schreibschutz aktiviert oder

deaktiviert, wobei zusätzlich noch ein vier Zeichen langes Paßwort als weitergehender Zugriffsschutz zur Verfügung steht. Wird der Schreibschutz unter Angabe eines vier Zeichen langes Paßwortes aktiviert, muß dieses Passwort auch für die Deaktivierung eingegeben werden. Beachten Sie bitte, daß es sich bei »Lock« nur um einen »Software-Schreibschutz« handelt - beim nächsten System-Reset ist »LOCK« auf jeden Fall wieder deaktiviert!

Beispiel

LOCK RAD: OFF HTBL

deaktiviert den Software-Schreibschutz für die resetfeste RAM-Disk »RAD:«, wenn diese mit FFS betrieben wird und wenn beim Setzen des Schreibschutzes das Kennwort »HTBL« angegeben wurde.

MAKEDIR eröffnet Unterverzeichnisse auf dafür geeigneten Devices

Schablone

Dir/a

Syntax

MakeDir <Directory_Name>

Mit »MakeDir« werden in einem Verzeichnis weitere Unterverzeichnisse angelegt. Es ist nicht möglich, ein Sub-Directory anzulegen, ehe das nächste übergeordnete Directory eingerichtet wurde! Beachten Sie bitte, daß es nicht zulässig ist, Sub-Directories anzulegen, die denselben Namen tragen sollen wie ein im entsprechenden Verzeichnis bereits vorhandenes File.

Beispiel

MakeDir df2:Drawer/SubDrawer

MOUNT veranlaßt die Einbindung eines neuen Devices in die Verwaltungslisten des Betriebssystems

Schablone

Device/..., FROM/k

Syntax

Mount <Device_Name> [...] [FROM MountList_File_Name]

Nicht alle Devices, die für den Amiga existieren, sind Bestandteil des Betriebssystems bzw. werden von diesem automatisch geladen und eingebunden. Ein typisches Beispiel dafür sind »SPEAK:« und »PIPE:«, die in der Startup-Sequence explizit bekanntgegeben werden müssen, wenn sie verwendet werden sollen. Der für diese Anmeldung benötigte Befehl ist »Mount«.

»Mount« führt zwar die Anmeldung durch, braucht aber noch ganz bestimmte, standardisierte Kenndaten des anzumeldenden Devices. Diese Kenndaten müssen in einem ASCII-File vermerkt sein. Im Normalfall ist dieses File die Datei »MountList« aus dem »Devs«-Verzeichnis. Durch die Angabe »FROM Dateiname« wird »Mount« angewiesen, die Device-Daten nicht aus dem File »MountList« zu holen (das ist nur der Default-Eintrag), sondern aus jenem File, dessen Name nach »FROM« steht.

Wenn von der resetfesten RAM-Disk RAD: gebootet werden soll, muß im Eintrag für »RAD:« in der Mountlist des »Devs«-Verzeichnis unbedingt »BOOTPRI = 0« stehen!

Beispiel

Mount SPEAK:

meldet das Device »SPEAK« beim Betriebssystem an.

Mount QUASI: from dh0:Test/MODO

meldet das Device »QUASI« dem Betriebssystem, holt sich die Device-Parameter aber nicht aus der Datei »MountList«, sondern aus dem File »MODO«.

NEWCLI ruft einen neuen unabhängigen CLI-Task auf

Schablone

Window, From/k

Syntax

NewCLI

[Window_Handler:x1/y1/x2/y2/Fenstername] [FROM Filename]

»NewCLI« aktiviert ein neues Kommandofenster des echten CLI-Typs, also ein Ein-/Ausgabe-Window ohne »Shell«-Support.

Wahlschalter

Window - mit dieser Option ist die Übergabe definierter Fensterparameter an das neue Window ermöglicht. Die Aufrufkonvention lautet:

»Console-Handler:X-Koordinate links oben/Y-Koordinate links oben/X-Koordinate rechts unten/Y-Koordinate rechts unten/Fenster-Name«.

FROM - übergibt Pfad und Namen des Scriptfiles, das alternativ zum voreingestellten Scriptfile »CLI-Startup« nach dem CLI-Start abgear-

beitet werden soll. In diesem Scriptfile sind alle auch sonst in Scripts erlaubten Aktivitäten gestattet.

Beispiel

```
NewCLI CON:20/40/100/200/Test-
Fenster
```

startet einen neuen CLI-Task und öffnet das zugehörige Fenster 20 Pixel vom linken Bildschirmrand entfernt auf Höhe der 40. Pixel-Reihe. Das Fenster selbst ist 100 Pixel breit und 200 Pixel hoch.

NEWSHELL ruft einen neuen Shell-Task auf

Schablone

Window, From/k

Syntax

NewShell

[Window_Handler:x1/y1/x2/y2/Fenstername] [FROM Filename]

»NewShell« aktiviert ein neues Kommandofenster, das von den Features der »Shell« profitiert. Wie auch bei »NewCLI« ist es zulässig, ein Steuerfile beliebigen Namens anzugeben, das nach Aktivierung des neuen Shell-Tasks abgearbeitet wird. Die Voreinstellung arbeitet den Inhalt des Scriptfiles »Shell-Startup« im »S«-Verzeichnis ab.

Wahlschalter

Window – mit dieser Option wird die Übergabe definierter Fensterparameter ermöglicht. Die Aufrufkonvention lautet:

»Console-Handler:X-Koordinate links oben/Y-Koordinate links oben/X-Koordinate rechts unten/Y-Koordinate rechts unten/Fenster-Name«.

FROM – übergibt Pfad und Namen des Scriptfiles, das alternativ zum voreingestellten Scriptfile »Shell-Startup« nach dem Shell-Start abgearbeitet werden soll. In diesem Scriptfile sind alle auch sonst erlaubten Aktivitäten gestattet.

Beispiel

```
NewShell "NewCON:10/10/100/
100/ATARI"
```

eröffnet ein neues Shell-Fenster mit den unpraktischen Dimensionen 100 Punkte lang, 100 Punkte hoch. Die

linke obere Ecke dieses Fensters steht an der Bildschirmposition X=100/Y=100, und das Fenster trägt in seiner Titelleiste die fremdartige Bezeichnung »ATARI«.

Beispiel

```
NewShell from df2:Test/Steuer-
file
```

öffnet ein neues Shell-Fenster. Typ, Name und Dimension entsprechen den Voreinstellungen. Nach der Öffnung des Fensters wird die Scriptdatei »Steuerfile« abgearbeitet, die sich im Directory »Test« jener Diskette befindet, die aktuell im Laufwerk »df2:« eingelegt ist. Wenn die Scriptdatei im Directory »SYS:s« abgelegt wurde, kann eine Pfadangabe entfallen.

PATH setzt den Suchpfad, der vom Betriebssystem auf der Suche nach einer pfadlos eingegebenen Datei abgegangen wird

Schablone

Dir/..., ADD/S, SHOW/s, RESET/s, QUIET/s

Syntax

Path [Directory] [...] [ADD] [SHOW] [RESET] [QUIET]

Die Suche nach einem File beschränkt sich im allgemeinen auf das aktuelle Directory und auf das Directory »c« der Boot-Diskette. Soll ein File auch noch in anderen Bereichen (Devices, Sub-Directories etc.) gesucht werden, ehe die Suche mit einer Fehlermeldung abgebrochen wird, so ist dieser »Suchpfad« dem Betriebssystem mittels »Path« bekanntzugeben. Die Eingabe mehrerer Directories mit einem »Path«-Kommando ist erlaubt. Es können maximal zehn Verzeichnisse automatisch »abgegrast« werden.

Wahlschalter

ADD – fügt das/die eingegebenen Directories dem bestehenden Suchpfad hinzu. Da dies aber der Voreinstellung des »Path«-Kommandos bei übergebenen Directories entspricht, darf diese Option weggelassen werden.

SHOW – zeigt den aktuellen Suchpfad auf dem aktiven Ausgabegerät. »SHOW« können Sie sich sparen, weil »Path« ohne weiteren Zusatz dieselbe Wirkung hat.

RESET – überschreibt die alte Pfadangabe und läßt nur die Verzeichnisse »Current Direwctory« und »SYS.C« als Suchpfad bestehen.

QUIET – Normalerweise prüft das Betriebssystem bei einem »PATH«-Kommando, ob die angegebenen Pfade auch beschreibbar sind und verlangt bei Bedarf nach der benötigten Diskette. Die Option »QUIET« unterdrückt System-Requester, wenn ein Pfad auf eine Diskette gerichtet wird, die nicht eingelegt ist.

Beispiel

```
Path df1: df2:
```

befiehlt dem Betriebssystem, auch in den Root-Directories der Disketten in DF1: und DF2: nach einem File oder Directory zu suchen, aber nur wenn dieses im aktuellen Directory und im Directory »c« der Boot-Diskette nicht zu finden ist.

```
PATH df1: df2: reset
```

löscht diese Anordnung wieder.

PROMPT setzt den Prompt für ein CLI- oder Shell-Fenster

Schablone

[Prompt_Zeichenkette]

Syntax

Prompt [Zeichenkette]

Mit dem »Prompt«-Kommando wird jene Zeichenkette definiert, die anstelle des Standard-Prompt-Zeichens »>« auszugeben ist. Während in einem normalen CLI-Fenster nur diskrete Zeichenketten erlaubt sind, kann in einem Shell-Fenster auf die Platzhalter »%S« und »%N« zurückgegriffen werden. Anstelle von »%S« wird im Prompt-String der volle Pfad ausgegeben, anstelle von »%N« die Task-Nummer des aktivierten Fensters. Ebenso können in einem Shell-Window für den Prompt-String Umgebungsvariablen herangezogen werden. Die Anwendung von Bildschirmsteuerzeichen (ES-CAPE-Befehle) ist auch zulässig.

Beispiel

```
Prompt "TaskNummer:%N akt.Di-
rectory:%S"
```

ergibt z.B. folgenden Prompt:

```
Tasknummer:4 akt.Directory:
Word:
```

Dies gilt natürlich nur, wenn es wirklich der Task mit der Nummer 4 ist und das aktuelle Directory (Schulblade) »Word« heißt.

PROTECT setzt die Schutzbits von Dateien und Directories

Schablone

File/a, Flags, ADD, SUB,

Syntax

Protect <File-od. Directory-Name>
[[[+I]][[ADDISUB]][rlweldlalsplh]

»Protect« setzt den Schutzlevel von Dateien und Directories. Allerdings sind es seit WB_1.3 nicht mehr lediglich Schutzbits, die gesetzt werden, sondern es wurden Funktionen eingebaut, die mit Schutz eigentlich nichts mehr zu tun haben; »Kennbits« oder »Infobits« wäre der richtige Ausdruck.

Hier eine Auflistung der vorgesehenen Bits und deren Bedeutung; bis auf das »h«-Bit werden alle vom Betriebssystem in der Version 1.3 unterstützt, wobei es aber noch Unterschiede gibt, in welcher Umgebung (Shell etc.) man sich befindet.

h h-----

hidden File wird bei »List« oder »Dir« nicht angezeigt. Diese Funktion wird noch nicht genutzt, das Flag ist aber bereits setz- und löscherbar.

s s-----

script kennzeichnet ein »Script«-File, also eine Batch-Datei, die unter WB_1.3 und deren Shell ohne »execute« aufgerufen und abgearbeitet werden kann.

p --p-----

pure Programm ist resident verwendbar

a --a-----

archieve Archiv-Flag. Wird von verschiedenen Backup-Programmen ausgewertet und zur Selektion benutzt.

r ---r---

read File darf gelesen werden

w ---w---

write File darf beschrieben werden (aber nicht unbedingt ersatzlos gelöscht)

e -----e

execute File ist ein ausführbares Programm.

d -----d

delete Datei darf gelöscht werden

Ein gesetztes Flag (=Schutzbit) bedeutet, daß die zugehörige Aktion zulässig ist, ein gelöschttes Flag unterbindet sie. Der Zustand der Schutzbits kann nur mit dem »List«-Kommando standardmäßig abgefragt und ausgegeben werden.

ACHTUNG! Die Eingabe von »Protect Filename« ohne weiteren Zusatz löscht sämtliche Schutzbits des Files »Filename«.

Wahlschalter

ADD - Die angegebenen Flags sind zusätzlich zu den bereits vorhandenen Flags einzutragen. Dies hat dieselbe Wirkung wie ein dem Kennbuchstaben vorangestelltes »+«. »ADD« oder »t« beides geht.

SUB - Die angegebenen Flags sind aus dem Fileeintrag zu entfernen. Dies hat dieselbe Wirkung wie ein dem Protect-Kennbuchstaben vorangestelltes »-«.

Beispiel

```
protect turbo.c -d
```

löscht beim File »Turbo.c« im aktuellen Directory das Delete-Schutzbit.

Beispiel

```
protect df0:libsig +hsard
```

setzt für das File »df0:libsig« die Flags »h«, »s«, »a«, »r« und »d«. Die Flags »p«, »w« und »e« bleiben davon unberührt.

QUIT beendet die Abarbeitung eines Batch-Files

Schablone

ReturnCode

Syntax

Quit [RC]

»Quit« beendet bedingungslos die Abarbeitung eines Scriptfiles. Als Information für das den Scriptlauf aufrufende Programm kann ein Return-Code (RC) definiert werden, der dann vom aufrufenden Programm ausgewertet werden kann.

Beispiel

```
.....
```

```
QUIT
```

beendet die Abarbeitung der Batch-Datei.

Beispiel

```
.....
```

```
QUIT 10
```

beendet die Abarbeitung der Scriptdatei und übergibt dem aufrufenden Programm einen neuen Fehlercode für »ERROR«-Abfrage.

RELABEL ändert den Namen eines Datenträgers

Schablone

Drive/a, Name/a

Syntax

Relable <DiskDevice:>
<Neuer_Name>

Mit »Relable« werden Diskettenamen umbenannt (ähnlich »Rename« für Dateien).

Beachten Sie bitte, daß nicht der alte Name anzugeben ist, sondern die Device-Bezeichnung (»DF1:«, »DF2:«, etc.). Weiters ist unbedingt zu beachten, daß zwar der neue Name auf der Diskette und in den Verwaltungstabellen des Betriebssystems eingetragen wird, die Workbench-Arbeitsoberfläche diese Änderung aber nicht zur Kenntnis nimmt! Wenn Sie die Diskette »Alpha« mit »Relable« in »Beta« umbenennen, ändert sich der auf der Workbench unter dem zugehörigen Disketten-Icon angezeigte Name nicht. Da hilft auch kein »Diskchange«. Das bedeutet, daß Sie das zu jenem Disketten-Icon gehörende Fenster nicht mehr öffnen können. Es hilft nur noch, die Diskette aus dem Laufwerk zu entnehmen und wieder einzulegen. Daß dies bei »RAM-DISK« oder einer Festplatte leichter gesagt als getan ist, läßt sich leider nicht ändern.

Beispiel

```
Relable DFI: "Neuer_Name"
```

ändert den Namen der Disk in DF1: in »Neuer_Name«.

RemRAD löscht die resetfeste RAM-Disk und entfernt sie aus dem System des Amigas

Syntax

RemRAD

Dieser Befehl löscht den gesamten Inhalt der Ram-Disk »RAD« unwiederbringlich (ein Software-Hufeisenmagnet) und entfernt auch die zugehörigen Verwaltungstabellen etc. Bis auf geringe Restmengen werden alle von der »RAD« belegten Speicherbereiche zur anderweitigen Nutzung freigegeben.

RENAME ändert den Namen von Files und Directories

Schablone

Form/a, TO=AS/a

Syntax

Rename <Name__alt> [AS:TO]
<Name__neu>

»Rename« ist nicht über Devices hinweg arbeitsfähig, sondern sein Wirkungsbereich ist auf ein Device beschränkt. Es sind keine Jokerzeichen zulässig.

Beispiel

rename altesFile to geschafft
gibt dem File »altesFile« im aktuellen Directory den neuen Namen »geschafft«. Die gleiche Wirkung hätten die Eingaben

```
rename altesFile geschafft
rename altesFile as geschafft
```

RESIDENT Programme resident machen, residente Programme auflisten oder entfernen

Schablone

[ResidentName] Filename, ADD/s, REMOVE/s, REPLACE/s, PURE/s, SYSTEM/s

Syntax

Resident [*<ResidentName>*]
<Filename> [REMOVE] [ADD]
[REPLACE] [PURE] [SYSTEM]

Mit »resident« werden Kommandofiles in den Speicher geholt und stehen dort beliebig vielen anderen Programmen und Tasks simultan zur Verfügung. Die Liste der residenten Kommandos wird noch vor dem aktuellen oder dem »C:«-Directory auf Vorhandensein eines in der Kommandozeile eingegebenen Namens abgefragt. Der Hauptvorteil residenter Kommandos ist nicht so sehr die bei Aufruf entfallende Wartezeit, bis das gewünschte Kommando von der Disk in den Hauptspeicher eingeladen ist – da könnte die RAM-Disk durchaus mithalten. Nein, der große Vorteil liegt darin, daß ein solches Programm nur einmal den von ihm benötigten Speicher belegt, auch wenn es gleichzeitig von z.B. vier verschiedenen Tasks benutzt wird. Rufen Sie hingegen aus vier Tasks jeweils einmal ein nicht-residentes Programm auf, so wird dieses Pro-

gramm viermal in den Speicher geladen, belegt also die vierfache Menge an Speicherplatz.

Ein residentes Kommando kann also zeitgleich mehrfach aktiv sein; daraus folgt, daß ein resident-taugliches Programm gewisse Anforderungen bezüglich Struktur und Programmablauf erfüllen muß, kann es doch zeitgleich mehrfach aufgerufen werden. Commodore kennzeichnet solche resident-tauglichen Files durch ein gesetztes »PURE«-Bit. Sie können natürlich mit dem Kommando »protect« jedes beliebige File als »resident-tauglich« kennzeichnen, doch sollten Sie dabei äußerste Vorsicht walten lassen – »Big GURU is watching You«.

Die Eingabe von »Resident« ohne weiteren Filenamen bewirkt die Ausgabe aller residenten Kommandos auf dem Monitor und wie oft sie aufgerufen wurden. Wurde ein Kommando mit aktiviertem Wahlschalter »SYSTEM« resident geladen, erscheint es nur dann ebenfalls in dieser Auflistung, wenn der gleiche Wahlschalter angegeben wird.

Wahlschalter

ResidentName – Im Normalfall wird ein Programm unter seinem Originalnamen in die Resident-Liste eingetragen. Werden jedoch beim »resident«-Aufruf zwei Namen angegeben, so gilt der erste nach »Resident« angeführte Name als Bezeichnung für das Kommando in der Resident-Liste, während der zweite Name erst das resident zu machende File bezeichnet.

REMOVE – entfernt ein residentes Programm aus der Liste und aus dem Speicher. Wurde ein Kommando mit aktiviertem Wahlschalter »SYSTEM« resident gemacht, kann es nicht mehr entfernt werden.

ADD – fügt ein weiteres Programm einer bestehenden Reihe residenter Kommandos hinzu. Diesen Wahlschalter können Sie getrost vergessen, da er die Voreinstellung des »Resident«-Befehls darstellt.

REPLACE – ersetzt ein residentes Programm durch ein anderes gleichen (Resident-)Namens. Das funktioniert aber nur, wenn zum Zeitpunkt der Befehlserteilung das »alte« residente Kommando nicht gerade bei einem Programm im Einsatz ist.

PURE – veranlaßt »resident«, die Überprüfung des Schutzbits »p« (»p« wie »pure«) zu unterlassen. Wird »PURE« nicht angegeben, weigert sich der »Resident«-Befehl, ein File in die Resident-Liste aufzunehmen, dessen »p«-Bit nicht gesetzt ist. Mit Angabe von »PURE« übernehmen Sie die Verantwortung für die Residentfähigkeit eines Programms - bitte nur mit großer Sorgfalt anwenden!

SYSTEM – Ist dieser Wahlschalter aktiviert, wird das jeweilige Kommando unwiderruflich in den Speicher geholt und in die Liste der System-Residents eingetragen. Solche Residents können nur mit einem Systemneustart wieder aus dem Speicher entfernt werden. Ein Beispiel für ein solcherart resident gemachtes Programm ist »Shell-Seg«, in dem der Programmcode für die »Shell« abgelegt ist.

Beispiel

Resident Übersicht c/dir pure
lädt das File »dir« unabhängig vom Zustand des Schutzbits »p« in den Speicher und macht »dir« unter dem neuen Namen »Übersicht« resident. Sonderzeichen wie z.B. »Ü« in »Übersicht« sind zwar zulässig, führen aber zu »Case sensitivity«, müssen also immer genauso geschrieben werden wie beim ersten Aufruf (Groß-/Kleinschreibung).

Beispiel

Resident SYSTEM

ergibt folgende Ausgabe:

Name	UseCount
----	-----
dir	4
cd	9
SHELL-SEG	SYSTEM
FileHandler	SYSTEM
Restart	SYSTEM

Beispiel

Resident cd remove

entfernt das residente File »cd« aus der Resident-Liste und aus dem Speicher des Amigas.

Anmerkung

Residente Kommandos können nur dann aus dem Speicher entfernt werden, wenn auf sie aktuell kein Zugriff erfolgt. Wenn Sie »Resident« selbst resident gemacht haben, können Sie »Resident« nur auf eine Art wieder aus dem Speicher entfernen:

```
SYS:c/Resident Resident remove
```

RUN startet einen Hintergrundprozeß für ein Programm

Schablone

```
<Programm_Name>
```

Syntax

```
Run <Programm_Name> [Argumente für Programm_Name]
```

»Run« wird dazu benutzt, einen Hintergrund-Task zu starten. »Run« selbst übernimmt keinerlei Information aus der Kommandozeile. Der Aufruf des nach »Run« angeführten Programms erfolgt durch den Hintergrund-Task, der den Inhalt der Eingabezeile (exklusive »Run«) einfach übernimmt.

Um mehrere Kommandos im aufgerufenen Background-Task abarbeiten zu können, ohne eine Scriptdatei dafür bemühen zu müssen, erlaubt »Run« die Übergabe mehrerer Kommandozeilen, die mit einem Pluszeichen enden müssen.

Ein mit »Run >NIL: <Befehl>« gestarteter Prozeß blockiert zwar nicht die Weiterarbeit im aufrufenden Window, doch kann es vorkommen, daß sich der CLI-Task, aus dem heraus ein Programm mit »RUN« gestartet wurde, erst dann beenden läßt, wenn auch das aufgerufene Programm beendet ist.

Beispiel

```
run copy df2:libs/ar#? to  
ram:
```

startet das »Copy«-Kommando in einem separaten Task, so daß im aktuellen Task weitergearbeitet werden kann. Das aktuelle Fenster bleibt aber der Ausgabekanal für »Copy« (für Fehlermeldungen usw.).

Beispiel

```
run copy df2:libs/ar#? ram:+  
<RETURN>  
run execute Batchfile2+ <RE-  
TURN>  
echo "das war's !" <RETURN>
```

Beachten Sie das Pluszeichen in der ersten und zweiten Zeile vor <RETURN>! Damit blockieren Sie einstweilen die Ausführung von »Run...«. Erst das letzte <RETURN>, dem kein »+«-Zeichen mehr vorangestellt ist, startet die Kommandofolge (die dann mit der Abarbeitung der ersten Zeile beginnt).

SEARCH durchsucht Directories oder Files nach einer gegebenen Zeichenkette

Schablone

```
From/a, Search, ALL/s, NONUM/s,  
QUIET/s, QUICK/s, FILE/s
```

Syntax

```
Search <Dateiname ISTDIN>  
[Such-String] [ALL] [QUIET IQUICK]  
[FILE]
```

Mit »Search« suchen Sie in Dateien nach dem Vorkommen der an »Search« übergebenen Zeichenkette. »Search« zählt die Anzahl der »Carriage Return« mit und gibt bei Auffinden des gesuchten Strings die Zeilennummer und den Inhalt der Zeile auf dem aktuellen Ausgabegerät aus. Der Gebrauch von Jokerzeichen zur Fileselektion ist ebenso erlaubt wie das Einbinden von »Search« in einen Pipeline-Prozeß (dann muß allerdings als Filename »STDIN« angegeben werden). Im Dateinamen sind Jokerzeichen zugelassen, im Such-String werden sie nicht als Jokerzeichen erkannt. Falls im Such-String Leerzeichen enthalten sind, muß der Such-String wie gewohnt ganz in Anführungszeichen gesetzt werden.

Um »Search« in Batch-Files verwenden zu können, wurde folgende Rückmeldungsvereinbarung getroffen: findet »Search« den gesuchten String, so wird an das aufrufende Programm der Return-Code »NULL« zurückgegeben, andernfalls der Wert »5« für »WARN«.

Werden mehrere Files nach einem bestimmten String durchsucht, so kann die Suche im aktuell bearbeiteten File durch die Eingabe von <CTRL-D> abgebrochen werden. »Search« schließt dann ordnungsgemäß das aktuelle File und geht zum nächsten über. Allerdings kann das auch zum Abbruch eines ganzen Scriptprozesses führen. Den totalen Abbruch der Suchoperation erreichen Sie mit <CTRL-C>.

Wahlschalter

ALL – durchsucht alle Files innerhalb des angegebenen Directory und auch in den zugeordneten Unterverzeichnissen.

NONUM – unterdrückt die Ausgabe der Zeilennummern.

QUIET – unterdrückt sämtliche Ausgaben (nur sinnvoll, wenn der Return-Code ausgewertet wird, also in Scriptfiles).

QUICK – reduziert den Umfang der Ausgabe.

FILE – mit dieser Option sucht »Search« nach einem Filenamens, der zum gesuchten String paßt. Damit kann zwar auf Vorhandensein eines Files getestet werden, doch gibt »SEARCH« nicht den Pfad aus, der zu dem gesuchten File führt.

Beispiel

```
search df1:s/#?startup#?  
eside
```

durchsucht alle Files im Verzeichnis »s« auf der Diskette in Laufwerk DF1:, in deren Namen die Zeichenfolge »startup« enthalten ist, nach dem Vorkommen des Strings »eside«.

Beispiel

```
search df1:#? "Hilf Himmel"  
all file quick
```

durchsucht alle Verzeichnisse und Unterverzeichnisse der Diskette in Laufwerk DF1: nach Files, deren Name »Hilf Himmel« vorkommt.

Beispiel

```
search >ram:testfile s:resi-  
dent all
```

überprüft alle Files im Verzeichnis »S« der Boot-Diskette nach »resident« und schreibt anstatt auf den Bildschirm die gefundenen Zeilen in das File »Testfile« auf die RAM-Disk.

SETCLOCK Hardware-Uhr lesen oder setzen

Schablone

```
LOAD/s, SAVE/s, RESET/s
```

Syntax

```
SetClock [LOAD I SAVE I RESET]
```

»SetClock« ist das Bindeglied zwischen der (Software-)Systemuhr des Amigas und einer eventuell vorhandenen Hardware-Uhr. »SetClock«

prüft unmittelbar nach dem Start, ob auf den Adressen der Hardware-Uhr im Sekundentakt Uhrzeiteinformationen kommen. Ist dies nicht der Fall, gibt »SetClock« die Meldung »Battery Backed up Clock not found«. Für das Ausbleiben der Uhrzeiteinformationen gibt es (abgesehen von echten Hardware-Fehlern) zwei Ursachen: Entweder Sie haben keine Hardware-Uhr – dann dürfen Sie sich nicht wundern. Oder bei einem System-Crash wurde das Stop-Bit der Uhr gesetzt, und solange dieses Bit gesetzt ist, wird die Uhr angehalten. Um in letztgenannter Situation die Hardware-Uhr wieder zum Laufen zu bringen, wurde der Wahlschalter »Reset« eingebaut – allerdings hat es schon Fälle (und Uhren) gegeben, bei denen dieses Kommando versagte. Noch ein Hinweis: »SetClock« benutzt den Befehl »Date«, um das Datum »handeln« zu können. Findet »SetClock« das File »Date« nicht, wird die Aktion abgebrochen, was Sie während der Abarbeitung einer Batch-Datei vielleicht gar nicht bemerken.

Beispiel

```
SetClock load
```

setzt die Systemzeit des Amiga auf den Stand der Hardware-Uhr.

```
SetClock save
```

setzt die Hardware-Uhr auf den Stand der Systemuhrzeit.

Folgende Schreibweise ist ebenfalls zulässig:

```
SetClock opt save
```

```
Setclock opt r
```

```
SetClock l
```

SETDATE setzt Zeit- und Datumsstempel eine Files

Schablone

```
File/a, [DATE] [TIME]
```

Syntax

```
SetDate <Dateiname> [Datum]  
[Zeit]
```

Mit »SetDate« setzen Sie den Datums- und Zeiteintrag von Dateien. Diesen Eintrag zeigt Ihnen dann »List« an. Die Benutzung von Datumsangaben wie »YESTERDAY«, »TOMORROW«, »TODAY«, »MONDAY« usw. ist zulässig. Wenn Sie »Setdate« nur mit einem Filena-

men, nicht aber mit einer Datums- und Zeitangabe aufrufen, werden das aktuelle Systemdatum und die aktuelle Systemzeit eingetragen. Die Verwendung von Jokerzeichen im Filenamen wird von »SetDate« leider nicht unterstützt.

Beispiel

```
SetDate File_1 20-Aug-90 18:22
```

setzt den Datumsstempel des Files »File_1«.

SETENV setzt eine Umgebungsvariable im ENV:-Handler

Schablone

```
Name/a, Zeichenkette
```

Syntax

```
SetEnv <ENV: Name_der_Va-  
riablen> [Zeichenkette]
```

Dieses Kommando setzt eine Umgebungsvariable im ENV:-Handler (der zur Zeit ja bloß ein Directory in RAM: ist) auf den Inhalt der angegebenen Zeichenkette.

Beispiel

```
SetEnv ENV:meine_Variable Kal-  
kutta
```

setzt in ENV: die Variable »meine_Variable« auf den Inhalt »Kalkutta«. Auszulesen ist diese Variable aber nur mit »GetEnv«.

SETPATCH korrigiert System-Routinen

Schablone

```
R/s
```

Syntax

```
SetPatch [R]
```

»Nobody's perfect« und Commodore schon gar nicht (Perfektionismus und Genialität haben sich noch nie vertragen). In den Kickstart-Routinen der Versionen 1.2 und 1.3 stecken einige Bugs, die mit dem Befehl »SetPatch« ausgemerzt werden. Das kostet zwar ein bißchen RAM, aber das sollte es Ihnen wert sein. »SetPatch« muß die erste in der Startup-Sequenz durchgeführte Aktion sein. Die Position an erster Stelle in der Original-Commodore-Startup-Sequenz ist kein Zufall!

Wahlschalter

R – Wenn Sie Kickstart 1.2 verwenden oder nur 512 KByte Chip-RAM haben, dürfen Sie diesen Wahlschalter vergessen. Einer der Fehler von Kickstart 1.3 ist, daß die Routinen für den Betrieb der resetfesten RAM-Disk »RAD« nur auf 512 KByte Chip-RAM ausgelegt sind – bei 1 MByte Chip-RAM ist die resetfeste RAM-

Disk nach einem Reset futsch. Dieser Fehler kann ebenfalls mit »SetPatch« behoben werden.

SKIP Sprungbefehl in einer Batch-Datei

Schablone

```
LABEL, BACK/s
```

Syntax

```
SKIP Label_Name [BACK/s]
```

»Skip« durchsucht DEN REST DES SCRIPTFILES nach dem angegebenen Label und springt den ersten dem Label folgenden Befehl an, von wo aus die Bearbeitung der Scriptdatei fortgesetzt wird. Wird der Schalter »BACK« gesetzt, so wird die Suchrichtung umgedreht, »SKIP« sucht sein Label also in Richtung Fileanfang (Hallo Handbuch – habe ich Dich wieder mal erwischt...). Der »Rücklauf« wird allerdings spätestens beim nächsten »Execute« abgebrochen. Wird das gesuchte Label in der vorgegebenen Richtung nicht gefunden, wird die Bearbeitung der Batch-Datei abgebrochen. Dieser Abbruch kann durch Setzen des Fehlerlevels auf einen Wert größer 10 verhindert werden.

Stößt »Skip« während seiner Suche auf ein »EndSkip«-Kommando, so wird die Suche abgebrochen und die Bearbeitung der Batch-Datei mit dem ersten dem »EndSkip« folgenden Befehl fortgesetzt. Egal, aus welcher Richtung »EndSkip« erreicht wurde, die Bearbeitung der Batch-Datei wird in der »normalen« Richtung fortgesetzt.

Beispiel

```
SKIP Schleifenanfang
```

Beispiel

```
SKIP Schleifenanfang BACK
```

SORT sortiert den Inhalt eines Textfiles

Schablone

```
From, To, ColStart/k
```

Syntax

```
Sort [FROM] <Quell-Datei> TO  
<Ziel> [Start mit Spalte nn]
```

»Sort« ist ein Programm, mit dem der Inhalt von Textdateien sortiert wird. Die Angabe »FROM« kann entfallen, wenn das zu sortierende File als erstes genannt wird. Die Angabe von »TO« ist unbedingt erforderlich, auch wenn die Ausgabe des sortierten Textes auf dem Monitor erfolgt – in diesem Fall muß »*« für das aktuelle Ausgabegerät angegeben werden. Fairerweise weist das Handbuch darauf hin, daß vor Anwendung des »Sort«-Kommandos der Stack vergrößert werden soll. Aus eigener Erfahrung kann ich sagen: Soll ist noch geschmeichelt. Legen Sie den Stack auch bei kleinen Dateien lieber auf 10000, besonders dann, wenn Sie mit dem Wahlschalter »ColStart« operieren.

Wahlschalter

COLSTART – übergibt eine Zahl an »Sort«, mit der festgelegt wird, ab welcher Spalte der Sortiervorgang beginnen soll. Alle Zeichen links von dieser Spalte werden vom Sortiervorgang ausgeschlossen.

Beispiel

```
sort s/startupII to *
```

sortiert den Inhalt des Files »StartupII« und gibt den sortierten Inhalt auf dem Monitor aus.

Beispiel

```
list to pipe:a1 df2:
sort from pipe:a1 to pipe:a2c-
olstart 41
type from pipe:a2
```

gibt eine Liste der Files im Root-Directory der Diskette in Drive DF2: auf dem Schirm aus, sortiert nach dem Datum und der Uhrzeit des letzten Schreibzugriffs. Beachten Sie bitte die maximale Länge der PIPE – arbeiten Sie lieber mit mehreren Tasks!

STACK setzt oder zeigt die Stack-Größe des aktuellen Prozesses

Schablone

Stack-Größe in Bytes

Syntax

Stack [Stackgröße]

Manche Programme brauchen für ihre Arbeit einen größeren Stack, als er von Amiga-DOS bereitgestellt wird. Mit »Stack« kann die Stack-Größe gesetzt werden. Jeder neue Task, der aus dem aktuellen Task heraus gestartet wird, übernimmt die Stack-Größe des »Mutterprozesses«. Die Eingabe von »Stack« ohne Argument gibt die aktuelle Stack-Größe auf dem Monitor aus.

Beispiel

```
Stack 10299
```

setzt den Stack für den aktuellen Prozeß auf eine Größe von 10299 Byte.

STATUS gibt eine komplette Liste der aktuell laufenden CLI- und Shell-Prozesse aus

Schablone

Prozeß, F=FULL/s, TCB/s, CLI=ALL/s, COM=COMMAND/k

Syntax

Status [PROCESS #] [FULL I TCB I ALL] [COMMAND Name]

In erster Linie dient »Status« dazu, dem User einen Überblick über die aktuell laufenden CLI- oder Shell-Prozesse zu geben (»Status« macht keinen Unterschied zwischen CLI- und Shell-Prozessen, da letztendlich jeder Shell-Aufruf einen CLI-Task startet).

Wahlschalter

FULL – gibt alle Informationen aus, die »Status« zu bieten hat.

ALL – listet alle laufenden Prozesse auf, allerdings ohne Information über Stack-Größe, Priorität etc. Dies ist die Voreinstellung von »Status«.

PROCESS # – Damit kann Information über einen bestimmten CLI-Prozeß angefordert werden.

TCB – gibt nur die Informationen Prozeßnummer, Stack-Größe, Global Vector und Priorität aus.

COMMAND – sucht in der Prozeßliste nach einem bestimmten einzugebenden Befehl. Wird dieser Befehl in der Liste gefunden, endet »Status« mit der Rückgabe des Fehlercodes »NULL« und gibt die Nummer des Prozesses aus. Wird der Prozeß mit dem gesuchten Namen aber nicht gefunden, wird als Return-Code der Wert 5 (= WARN) zurückgegeben. Damit dürften die Hauptanwendungen der Option »COMMAND« klar sein: Stapeldateien und die Anwendung von »ChangeTaskPri«.

Beispiel

```
status tcb
```

ergibt

```
Process 1: stk 4000, gv 150,
           pri 0
Process 3: stk 4000, gv 150,
           pri 0
Process 4: stk 4000, gv 150,
           pri 0
```

Beispiel

```
status full
```

ergibt die Ausgabe aller Informationen:

```
Process 1: stk 4000, gv 150,
           pri 0 Loaded as command: SYS:
wp
Process 3: stk 4000, gv 150,
           pri 0 Loaded as command: Util-
ities/TimeMem
Process 4: stk 4000, gv 150,
           pri 0 Loaded as command: sta-
tus
```

TYPE gibt den Inhalt von Dateien auf dem aktuellen Ausgabegerät aus

Schablone

From/..., TO/s, OPT/k, HEX/s, NUMBER/s

Syntax

Type <Filename> [TO] [Zieldatei oder Device] [OPT H N] [HEX NUMBER]

Type gibt eine Datei auf dem aktuellen Window aus oder schreibt diese in das gewünschte Device. »Type« erlaubt keine Wildcards.

Wahlschalter

HEX – OPT H – zeigt den Inhalt des gewünschten Files in hexadezimaler Schreibweise und in ASCII-Form.

NUMBER – OPT N – numeriert bei der Ausgabe die Zeilen.

TO Filename – erlaubt die Definition einer Zieldatei oder eines Ziel-Device. Wird bei der Eingabe die Reihenfolge »Quelldatei« »Zieldatei« eingehalten, darf »TO« entfallen.

Beispiel

```
type s/startup-
sequence to ram:file number
bewirkt dasselbe wie
type s/startup-
sequence ram:file opt n
```

nämlich die Ausgabe des Files »Startup-Sequence« in die Datei »file« in »RAM:«, wobei die Zeilen durchnummeriert werden.

VERSION gibt die Versionsnummer der System-Software aus

Schablone

Name, VERSION, REVISION, UNIT

Syntax

Version [*<library name>*] [VERSION] [REVISION] [UNIT]

»Version« gibt mit seinen Default-Einstellungen die Versionsnummern von Kickstart und Workbench aus. Wird beim Aufruf ein Name angegeben, so wird konkret nach diesem Namen gesucht.

Wahlschalter

VERSION - ist gedacht für die Anwendung von »Version« in Batch-Prozessen. Mit dem Schlüsselwort »VERSION« gibt man einen Wert für die Versionsnummer vor. Stimmen Wert und Versionsnummer überein oder ist die gefundene Versionsnummer höher als die vorgegebene, so gibt das Programm »Version« den Return-Code »NULL« zurück. Bei Nichtübereinstimmung wird der Wert für »WARN« zurückgegeben.

REVISION - ist von der Funktion her gleichzusetzen mit der Option »VERSION«, nur daß mit »REVISION« die Revisionsnummer vorgegeben und abgefragt wird.

UNIT - ist nur für Devices sinnvoll anzuwenden, die aus mehreren Einzelgeräten bestehen können, wie dies z.B. bei SCSI-Devices und Omti-Controllern der Fall ist.

Beispiel

Die Eingabe von

```
Version
```

gibt folgende Informationen auf dem Monitor aus:

```
Kickstart version 33.180. Work-
bench version 34.28
```

Beispiel

```
version dos.library
```

ergibt

```
dos.library version 33.124
```

WAIT hält die Abarbeitung eines Batch-Files für eine vom User beliebige bestimmte Zeit an

Schablone

SEC=SECS/s, MIN=MINS, UNTIL/k

Syntax

Wait [*n SEC | SECS | IN | MINS*] [UNTIL *Zeit*]

»Wait« wartet eine vorgegebene Anzahl von Minuten oder Sekunden, oder bis eine bestimmte Systemuhrzeit erreicht ist.

Beispiel

```
wait 10 secs
```

hält die Abarbeitung des aktuellen Prozesses für 10 Sekunden an

Beispiel

```
wait 6 mins until 22:10
```

läßt den Prozeß warten, bis die Systemzeit 22 Uhr 10 Minuten und einige Sekunden erreicht. Interessanterweise reicht einige Sekunden von 10 bis 51 Sekunden - Atomwecker ist »wait« keiner.

WHICH zeigt den Pfadnamen eines Programms

Schablone

File/a, NORES/s, RES/s

Syntax

Which *<Programmname>* [RES | NOTRES]

»Which« durchsucht den Systempfad und die Resident-Liste nach dem gesuchten Programm. Die Default-Parameter von »Which« sind so gesetzt, daß zuerst Resident-Liste (vom Commodore-Befehl »Resident« angelegt) und dann der Systempfad (vom Kommando »Path« angelegt) durchsucht wird. Leider akzeptiert »WHICH« keinerlei Jokerzeichen.

Wahlschalter

RES - durchsucht nur die Liste der residenten Befehle

NORES - durchsucht nicht (!) die Liste der residenten Befehle

Beispiel

```
which dir
```

hat beispielsweise folgende Ausgabe zur Folge:

```
RESIDENT dir
```

wenn »dir« resident geladen ist.

WHY zeigt die Fehlernummer der letzten Aktion

Schablone

<RETURN>

Syntax

Why

»Why« wird ohne jeden Parameter aufgerufen. Falls das letzte Kommando mit einer Fehlermeldung abgebrochen wurde (z.B. »Can't

open ,,«), kann mit »Why« entweder eine »Klartext«-Erklärung oder zumindest der zugehörige Fehlercode angefordert werden, der dann auf dem Monitor ausgegeben wird. Dieser Fehlercode kann dann (meistens) mit »Fault <Fehlercode>« in Klartext umgewandelt werden.

Was ist ein Device?

Für das Betriebssystem des Amigas ist ein »Device« (=»Gerät«) eine Schnittstelle zu einem in das System eingebundenen Gerät, das für bestimmte zugewiesene Operationen zur Verfügung steht. Über diese Schnittstelle laufen alle Befehls- und Datentransfers. Amiga-DOS unterscheidet zwischen »physical devices«, »logical devices« und »volumes«. Angesprochen werden solche Geräte vom User immer mit deren Namen und einem folgenden Doppelpunkt. »Physical devices« sind alle körperlich vorhandenen Geräte, also z.B. Disketten-Laufwerke und Hard-Disks. Auch die Drucker- und die serielle Schnittstelle gehören zu den »> Physical devices«. Demgegenüber stehen die »logical devices«, also Geräte, die nicht als eigenständig körperlich vorhanden bezeichnet werden können. Es handelt sich dabei um Verzeichnisse, die wie ein Gerät angesprochen werden können, um den Zugriff darauf zu vereinfachen. Sieben Verzeichnisse jeder Boot-Diskette sind, sofern vorhanden, als logische Geräte voreingestellt, dazu kommen noch drei Verzeichnisse in der »RAM-Disk«. »Volumes« sind die Namen der vom System erkannten Disketten, RAM-Disks oder Hard-Disk-Partitions. Mit dem »assign«- und dem »mount«-Befehl kann die Liste der dem Betriebssystem bekannten Devices beliebig erweitert werden.

Wenn Sie mit der Workbench-Diskette das System namens Amiga gestartet haben und Sie über drei externe Laufwerke verfügen, stehen folgende »Devices«, »Directories« und »Volumes« für Sie bereit:

Volumes:

WorkBench1.3D [Mounted]

RAM DISK [Mounted]

Directories:

C WorkBench1.3D:c
CLIPS RAM DISK:clipboards
DEVS WorkBench1.3D:devs
ENV RAM DISK:env
FONTS WorkBench1.3D:fonts
L WorkBench1.3D:l
LIBS WorkBench1.3D:libs
S WorkBench1.3D:s
SYS WorkBench1.3D:
T RAM DISK:t

Devices:

AUX CON DFO DF1 DF2
DF3 NEWCON PAR PIPE
PRT RAM RAW SER SPEAK

Aus der Tatsache, daß unter »Volumes« nur zwei Namen angeführt sind, erkennen Sie, daß nur eine Diskette eingelegt ist, obwohl insgesamt vier Laufwerke zur Verfügung stehen (»df0« bis »df3«). Die »Directories«, die auf die Boot-Diskette zeigen, werden vom System automatisch zugewiesen, die Verzeichnisse in der RAM-Disk müssen gesondert (durch die »Startup-Sequence«-Datei) benannt werden. Ähnlich ist es bei den »Devices«, »PIPE«, »AUX«, »SPEAK« und »NEWCON« werden durch die Abarbeitung der »Startup-Sequence« und deren kleiner Schwester, der Datei »StartUpl«, dem System bekanntgegeben, die anderen stehen schon im Kickstart-ROM und werden automatisch zugewiesen. Übrigens fehlt in der Aufzählung der Devices ein wichtiges Geräte »NIL«. Es handelt sich dabei um ein Dummy-Gerät, an das man zwar Daten schicken kann, diese Daten aber im Nirwana verschwinden. Sagen Sie jetzt bitte nicht, daß das ja nur ein Blödsinn sein kann - Daten irgendwohin zu schicken, die in diesem irgendwo versickern. Es gibt immer wieder Fälle, in denen es von Vorteil ist, daß Ausgaben, Rückmeldungen etc. von Programmen nicht im aufrufenden CLI-Prozeß angezeigt werden. In diesen Fällen schicken Sie die Daten einfach mit dem Umleitungsfeil an das Device NIL: - fertig.

Was ist ein Directory?

Ein »Directory« hat auf dem Amiga wie auch auf vielen anderen Computern zwei Bedeutungen, die zwar eng miteinander verbunden, aber keineswegs verwandt sind.

Bedeutung 1: Das Betriebssystem des Amigas (Amiga-DOS) erlaubt die Unterteilung des Speicherplatzes der Speichermedien Diskette, RAM-Disk und Hard-Disk in beliebig große Teilstücke, wobei jedes Teilstück wieder in mehrere Teilstücke eingeteilt werden kann. In diese Teilstücke werden dann nach einem vom User gewählten oder vom Amiga verlangten System die einzelnen Dateien eingetragen. Dies erleichtert die strukturierte Ablage von Dateien. Ein »Directory« ist ein Teilstück, ein »Sub-Directory« ist ein Teilstück innerhalb eines Teilstücks. Man nennt solche Teilstücke auch »Ordner« oder »Schubladen«, und jedem Teilstück muß vom Benutzer ein Name, eine Bezeichnung zugeordnet werden.

Bedeutung 2: »Das Directory einer Diskette auflisten« bedeutet, daß alle Directories und Sub-Directories mit den darin enthaltenen Dateien namentlich aufgelistet werden. Ein Directory ist also auch das Inhaltsverzeichnis einer Diskette oder eines Ordners. Es ist keineswegs falsch, vom Directory eines Directory zu sprechen, wenn das Inhaltsverzeichnis einer »Schublade« damit gemeint ist.

Die wichtigsten ESCAPE-Sequenzen

Im Direktmodus des Command-Line-Interpreters wie auch aus Batch-Dateien heraus bieten die sog. »ESCAPE-Sequenzen« vielfältige Gelegenheit zur Beeinflussung des jeweiligen Ausgabe-Gerätes. Die meisten dieser Steuersequenzen und die damit erzielten Effekte sind durch den »ANSI-Standard« definiert. Manche Kommandos sind nur auf dem Monitor sinnvoll anzuwenden, andere nur in Zusammenhang mit einem Drucker, wieder andere sowohl als auch. Speziell bei Druckerbefehlen muß der Drucker natürlich auch zur Durchführung eines Befehls in der Lage sein. Es wird

wohl niemand annehmen, daß alleine durch das Senden des Befehls zum Umschalten auf Farbe Nummer 3 ein Schwarzweißdrucker zum Farbdrucker wird. Die Escape-Sequenz wird durch den in »Preferences« eingestellten Druckertreiber in für den jeweiligen Drucker verständliche Zeichenfolgen übersetzt. So wird z.B. aus der ESCAPE-Sequenz

»< ESCAPE > <[> <4 > <w >«
die für einen Epson-FX85 akzeptable Zeichenfolge »< ESCAPE > <E >«.

Beispiele und Anwendungshinweise für die Praxis

Die Eingabe der Escape-Sequenz »< ESCAPE >[3m« im CLI-Fenster erfolgt entweder durch Drücken der Tastenfolge <ESC> <ALT> + <Ü> <3> <m> oder durch die Eingabe von

*e[3m

In beiden Fällen ist die Eingabe mit <RETURN> abzuschließen. Die Zeichenfolge »*e« ist für das Amiga-Betriebssystem ein Synonym für »ESCAPE«. Im Direktmodus wird »ESCAPE« durch eine invertiert dargestellte eckige Klammer (»[«) erkenntlich. In Scriptdateien sollten Sie aus Gründen der besseren Lesbarkeit der Zeichenfolge »*e« den Vortzug geben.

Es ist zulässig, Escape-Sequenzen, die mit »m« enden, zusammenzufassen, wenn die einzelnen Codes durch ein Semikolon getrennt werden. Die Eingabe von

echo " *e[1;33;41m Testwort "

schaltet Fettschrift ein, macht Farbe 3 zur aktuellen Vordergrundfarbe, wechselt die Hintergrundfarbe auf 1 und gibt den Text »Testwort« aus. Auf der Servicediskette zu diesem Sonderheft finden Sie im File »ESCAPE-DEMO« eine Zusammenstellung verschiedener Escape-Sequenzen. Starten Sie dieses File mit

execute ESCAPE-DEMO

für die Darstellung auf dem Bildschirm. Für ein Demo der Druckerfunktionen starten Sie auf die gleiche Weise das Batch-File »ESCAPE-DEMO.prt«.

```

<ESCAPE>[Om -- Standard-Zeichenattributierung
<ESCAPE>[Op -- Cursor ausschalten
<ESCAPE>[p -- Cursor einschalten
<ESCAPE>[3m -- Kursiv-Schrift einschalten
<ESCAPE>[23m -- Kursiv-Schrift ausschalten
<ESCAPE>[4m -- Unterstreichen einschalten
<ESCAPE>[24m -- Unterstreichen ausschalten
<ESCAPE>[1m -- Fettschrift einschalten
<ESCAPE>[22m -- Fettschrift ausschalten (nicht auf Monitor!)
<ESCAPE>[nmm -- Vordergrundfarbe wählen nn=31 bis 34
<ESCAPE>[nmm -- Hintergrundfarbe wählen nn=41 bis 44
<ESCAPE>[nnt -- nn Zeilen im Fenster verfügbar
<ESCAPE>[nnu -- nn Zeichen pro Zeile
<ESCAPE>[0w -- Standard-Schriftart einschalten (meist "PICA")
<ESCAPE>[2w -- Schriftart ELITE einschalten
<ESCAPE>[1w -- Schriftart ELITE ausschalten
<ESCAPE>[4w -- Schmalschrift einschalten
<ESCAPE>[3w -- Schmalschrift ausschalten
<ESCAPE>[6w -- Breitschrift einschalten
<ESCAPE>[5w -- Breitschrift ausschalten
<ESCAPE>[6"z -- Schattenschrift einschalten
<ESCAPE>[5"z -- Schattenschrift ausschalten
<ESCAPE>[4"z -- Doppeldruck einschalten
<ESCAPE>[3"z -- Doppeldruck ausschalten
<ESCAPE>[2"z -- NLQ-Modus einschalten
<ESCAPE>[1"z -- NLQ-Modus ausschalten
<ESCAPE>[2v -- Superscript einschalten
<ESCAPE>[1v -- Superscript ausschalten
<ESCAPE>[4v -- Subscript einschalten
<ESCAPE>[3v -- Subscript ausschalten
<ESCAPE>[7m -- Zeichen invertiert darstellen
<ESCAPE>[nnH -- Cursor in Zeile nn setzen
<ESCAPE>[nn;mmH -- Cursor in Zeile nn / Spalte mm setzen
<ESCAPE>c -- Form Feed und Standard-Einstellungen
<ESCAPE>D -- Line Feed
<ESCAPE>E -- Carriage Return / Line Feed
<ESCAPE>M -- eine Zeile zurück
<ESCAPE>[nnA -- Cursor um nn Zeilen nach oben
<ESCAPE>[nnB -- Cursor um nn Zeilen nach unten
<ESCAPE>[nnC -- Cursor um nn Zeichen nach rechts
<ESCAPE>[nnD -- Cursor um nn Zeichen nach links
<ESCAPE>[J -- Bildschirm ab Cursorposition löschen
<ESCAPE>[nnL -- Bildschirm-Inhalt ab Cursor-
    Position um nn Zeilen nach unten scollen
<ESCAPE>[nnM -- Bildschirm-Inhalt ab Cursor-
    Position um nn Zeilen nach oben ziehen und die
    dazwischen liegenden nn Bildschirmzeilen löschen
<ESCAPE>[nnS -- scrollt Inhalt des Windows um nn Zeilen nach
    oben
<ESCAPE>[nnT -- scrollt Inhalt des Windows um nn Zeilen nach
    unten

```

Was ist eine Shell?

In der Computerei versteht man unter einer »Shell« eine Schale, die ein anderes Programm einhüllt. Die Kommunikation mit dem eingehüllten Programm erfolgt dann nur noch über diese Schale. Meist werden solche »Shells« entwickelt, um die Bedienung eines vorhandenen Programms zu erleichtern oder zumindest zu verbessern. Die »Shell« der Workbench 1.3 ist allerdings keine Schale, sondern ein vollwertiger und eigenständiger Ersatz für den normalen »CLI« und erleichtert die direkte Kommunikation mit dem Amiga ungemain. Mit den <Cursor links>- und <Cursor rechts>-Tasten kann der Cursor nun in der Eingabezeile frei bewegt werden, so daß das Ausbessern eines Tippfehlers nicht mehr einem Neuschreiben der ganzen Kommandozeile gleichkommt. Mit <Cursor rauf> und <Cursor runter> kann in den letzten eingegebenen Kommandozeilen geblättert werden, so daß z.B. nach einem Tippfehler, der zu »Unknown command ...« führte, die fehlerhafte Kommandozeile zurückgeholt und editiert werden kann. Wie viele Zeilen auf diese Art zurückgeholt werden können, ist abhängig von der Länge der Zeilen. Der Speicher reicht für insgesamt etwa 2000 Zeichen. Ist er voll, wird die älteste Kommandozeile überschrieben.

Editierfunktionen:

<CTRL><K> - löscht den Inhalt der Kommandozeile von der Cursorposition bis zum Zeilenende.

<CTRL><U> - löscht den Inhalt der Kommandozeile von der Cursorposition bis zum Zeilenanfang.

<CTRL><X> - löscht die gesamte Kommandozeile.

<CTRL><W> - setzt den Cursor auf die nächste Tabulatorposition (voreingestellt sind acht Zeichen).

<SHIFT><Cursor links> - setzt den Cursor an den Zeilenanfang.

<SHIFT> <Cursor rechts> - setzt den Cursor an das Ende der Kommandozeile.

<Cursor rauf> - »blättert« in der Kommandozeilen-History zurück.

<Cursor runter> - blättert in der History nach vorne.

<SHIFT> <Cursor rauf> - blättert zur ältesten Kommandozeile zurück.

<SHIFT> <Cursor runter> - blättert zur neuesten Kommandozeile.

Was ist booten?

Das »Booten« ist jener Vorgang, mit dem das Computersystem namens »Amiga« initialisiert wird. Wenn Sie den Amiga einschalten, reagiert er auf keine Tastatureingabe, auf keine Maus- oder Joystick-Aktion, auf rein gar nichts. Auf dem Monitor wird lediglich die zu Berühmtheit gelangte »Workbench-Hand« abgebildet, womit der Amiga zu verstehen gibt, daß er die mitgelieferte Diskette namens »Workbench 1.x« in das interne Laufwerk eingelegt haben möchte. Allerdings wird nicht nur das Einlegen dieser, sondern auch jeder anderen »bootfähigen« Diskette vom Amiga registriert und löst eine Aktion aus. Daß dies so ist, wird durch den Inhalt des Kickstart-ROMs bestimmt, das, mit Ausnahme des Amiga 1000 und Amiga 3000, fest in Ihren Computer eingebaut ist. Wenn Sie eine Diskette in das interne Laufwerk (das deshalb auch als »Boot-Laufwerk« bezeichnet wird) einlegen, werden die ersten beiden Blöcke dieser Diskette in den Arbeitsspeicher eingelesen. Diese beiden Blöcke werden der gebräuchlichen Numerierung folgend als Block 0 und Block 1 bezeichnet und befinden sich auf Spur 0, also am äußersten Rand der Magnetscheibe. Aufgrund ihrer Bedeutung für den Boot-Vorgang nennt man sie auch »Boot-Block«. »Boot-Blöcke« wäre zwar grammatikalisch richtiger, ist aber absolut ungebräuchlich. Ent-

```
echo >prt: "DAS IST EIN DEMO FUER DRUCKER- UND MONITOR-STEUERUNG"
echo >prt: "
=====*n*n"
wait 2
;
echo >prt: "Fenster wird geloescht"
echo >prt: "*ec"
;
echo >prt: "nach dieser Zeile kommt eine Leerzeile*eE"
echo >prt: "Es folgt ein reines LineFeed ohne Wagenruecklauf*eD und wo steht das?"
echo >prt: "Dummyzeile nach Leerzeile"
echo >prt: "*eMDummyzeile wird nur auf Monitor ueberschrieben"
echo >prt: "-----"
;
echo >prt: "Kursiv einschalten*e[3m"
echo >prt: "Dieser Text ist in Kursiv"
;
echo >prt: "*e[0mText ist wieder Standard"
echo >prt: "*e[3mKursiv ein - *e[23mKursiv aus auf dem Drucker"
echo >prt: "*e[0mStandardschrift fuer Monitor"
echo >prt: "*e[4mDieser Text ist unterstrichen"
echo >prt: "Auch dieser Text ist unterstrichen*e[24m aber dieser auf dem Drucker nicht-
mehr"
echo >prt: "*e[0mMit Escape [0 auch auf Monitor unterstreichen ausschalten"
;
echo >prt: "*e[1mDieser Text ist in Fettschrift gehalten"
echo >prt: "Fettschrift - *e[22mNormalschrift auf Drucker*e[0m und Monitor"
echo >prt: "Demo fuer Schrift in"echo >prt: "*e[31mFarbe 1 (ESC 31)"
echo >prt: "*e[32mFarbe 2 (ESC 32)"
echo >prt: "*e[33mFarbe 3 (ESC 33)"
echo >prt: "*e[34mFarbe 4 (ESC 34)"
echo >prt: "Demo fuer Hintergrundfarbe"
echo >prt: "*e[41mFarbe 1 (ESC 41)"
echo >prt: "*e[42mFarbe 2 (ESC 42)"
echo >prt: "*e[43mFarbe 3 (ESC 43)"
echo >prt: "*e[44mFarbe 4 (ESC 44)"
echo >prt: "*e[0mSchrift und Hintergrund wieder normal"
;
echo >prt: "Umschalten auf Elite - *e[2wELITE -
SCHRIFT und wieder zurueck : *e[1wSTANDARDSCHRIFT"
echo >prt: "*e[4wSTANDARDSCHRIFT SCHMAL - *e[3wSTANDARDSCHRIFT IST WIEDER NORMAL BREIT"
echo >prt: "*e[2wDas ist Elite normal *e[4wDas ist ELITE schmal -
*e[3wund das ist Elite normal"
echo >prt: "*e[6wBREITSCHRIFT EIN *e[5wund wieder aus"
echo >prt: "*e[6* "zSchattenschrift ein - *e[5* "zSchattenschrift aus"
echo >prt: "*e[4* "zDoppeldruck ein - *e[3* "zDoppeldruck aus"
;=====;
Beachten Sie bitte, dass das Anfuhrungszeichen ein Sonderzeichen darstellt !;Jedem An-
fuhrungszeichen innerhalb einer Zeichenkette, die Sie mit >>ECHO<<;ausgeben wollen,
muessen Sie ein Sternchen voranstellen !
;=====
echo >prt: "*e[2* "zNEAR LETTER QUALITY ein"
echo >prt: "Das ist eine Zeile in NLQ"
echo >prt: "Genug NLQ - *e[1* "zNLQ wieder abgeschaltet"
echo >prt: "Standard - *e[2vSuperscript - *e[1vStandard - *e[4vSubscript -
*e[3vStandard"
;
echo >prt: "Zeile A*nZeile B*nZeile C*nZeile D*nZeile E*nZeile F*nZeile G*nZeile H*nZe
ile I"echo >prt: "Nach dem fuenften Sternchen geht es weiter in Zeile C
! *****e[7A*****"
wait 3
echo >prt: "Machen wir Ordnung auf dem Bildschirm -
loeschen ab aktueller Cursorposition:*e[J"
wait 3
echo >prt: "Jetzt brauchen wir das ganze Fenster frei"
wait 2
echo >prt: "*ec"
echo >prt: "Zeile 1*nZeile 2*nZeile 3*nZeile 4*nZeile 5*nZeile 6*nZeile 7*nZeile 8*nZe-
ile 9"
echo >prt: "Cursor in >>Zeile 5<< setzen, Bildschirm um 3 Zeilen nach unten scroll
en und fuenf Sternchen ausgeben"
wait 2
echo >prt: "*e[6A*e[3L*****"
echo >prt: "Jetzt loeschen wir die naechsten drei Zeilen und ziehen den Rest nach oben"
wait 3
echo >prt: "*e[3M"
echo >prt: "Zeile 1*nZeile 2*nZeile 3*nZeile 4*nZeile 5*nZeile 6*nZeile 7*nZeile 8*n
Zeile 9"
echo >prt: "gleich wird der Bilschirminhalt um 10 Zeilen nach oben gerollt"
wait 2
echo >prt: "*e[12S"
wait 2
echo >prt: "In zwei Sekunden wird der Bilschirminhalt um 5 Zeilen nach unten scrollen"
wait 2
echo >prt: "*e[5T"
echo >prt: "das wars"
wait 10
```

**Ein Demo für die Druckeransteuerung
aus dem Amiga-DOS**

```

echo "DAS IST EIN DEMO FUER DRUCKER- UND MONITOR-STEUERUNG"
echo "=====*n*n"
wait 2
;
echo "Fenster wird geloesch"
echo "*ec"
;
echo "nach dieser Zeile kommt eine Leerzeile*eE"
echo "Es folgt ein reines LineFeed ohne Wagenruecklauf*eD und wo steht das?"
echo "Dummyzeile nach Leerzeile"
echo "*eMDummyzeile wird nur auf Monitor ueberschrieben"
echo "-----"
;
echo "Kursiv einschalten*e[3m"
echo "Dieser Text ist in Kursiv"
;
echo "*e[0mText ist wieder Standard"
echo "*e[3mKursiv ein - *e[23mKursiv aus auf dem Drucker"
echo "*e[0mStandardschrift fuer Monitor"
echo "*e[4mDieser Text ist unterstrichen"
echo "Auch dieser Text ist unterstrichen*e[24m aber dieser auf dem Drucker nicht mehr"
echo "*e[0mMit Escape [O auch auf Monitor unterstreichen ausschalten"
;
echo "*e[1mDieser Text ist in Fettschrift gehalten"
echo "Fettschrift - *e[22mNormalschrift auf Drucker*e[0m und Monitor"
echo "Demo fuer Schrift in"
echo "*e[31mFarbe 1 (ESC 31)"
echo "*e[32mFarbe 2 (ESC 32)"
echo "*e[33mFarbe 3 (ESC 33)"
echo "*e[34mFarbe 4 (ESC 34)"
echo "Demo fuer Hintergrundfarbe"
echo "*e[41mFarbe 1 (ESC 41)"
echo "*e[42mFarbe 2 (ESC 42)"
echo "*e[43mFarbe 3 (ESC 43)"
echo "*e[44mFarbe 4 (ESC 44)"
echo "*e[0mSchrift und Hintergrund wieder normal"
;
echo "Umschalten auf Elite - *e[2wELITE -
SCHRIFT und wieder zurueck : *e[1wSTANDARDSCHRIFT"
echo "*e[4wSTANDARDSCHRIFT SCHMAL - *e[3wSTANDARDSCHRIFT IST WIEDER NORMAL BREIT"
echo "*e[2wDas ist Elite normal *e[4wdas ist ELITE schmal -
*e[3wund das ist Elite normal"
echo "*e[6wBREITSCHRIFT EIN *e[5wund wieder aus"
echo "*e[6*"zSchattenschrift ein - *e[5*"zSchattenschrift aus"
echo "*e[4*"zDoppeldruck ein - *e[3*"zDoppeldruck aus"
;
=====
;Beachten Sie bitte, dass das Anfuhrungszeichen ein Sonderzeichen darstellt !;Jedem An-
fuhrungszeichen innerhalb einer Zeichenkette, die Sie mit >>ECHO<<
;ausgeben wollen, muessen Sie ein Sternchen voranstellen !
;=====
echo "*e[2*"zNEAR LETTER QUALITY ein"
echo "Das ist eine Zeile in NLQ"
echo "Genug NLQ - *e[1*"zNLQ wieder abgeschaltet"
echo "Standard - *e[2vSuperscript - *e[1vStandard - *e[4vSubscript - *e[3vStandard";
echo "Zeile A*nZeile B*nZeile C*nZeile D*nZeile E*nZeile F*nZeile G*nZeile H*nZeile I"
echo "Nach dem fuenften Sternchen geht es weiter in Zeile C ! *****e[7A*****"
wait 3
;=====
;Beachten Sie bitte, dass das Sternchen ein Sonderzeichen darstellt !
;Fuer jedes Sternchen, das Sie mit >>ECHO<< ausgeben wollen, muessen;Sie ein weiter-
es Sternchen eintragen !
;=====
echo "Machen wir Ordnung auf dem Bildschirm - loeschen ab aktueller Cursorposition:*e[J"
wait 3
echo "Jetzt brauchen wir das ganze Fenster frei"
wait 2
echo "*ec"
echo "Zeile 1*nZeile 2*nZeile 3*nZeile 4*nZeile 5*nZeile 6*nZeile 7*nZeile 8*nZeile 9"
echo "Cursor in >>Zeile 5<< setzen, Bildschirm um 3 Zeilen nach unten scrollen und
fuenf Sternchen ausgeben"
wait 2
echo "*e[6A*e[3L*****"
echo "jetzt loeschen wir die naechsten drei Zeilen und ziehen den Rest nach oben"
wait
;echo "*e[3M"
echo "Zeile 1*nZeile 2*nZeile 3*nZeile 4*nZeile 5*nZeile 6*nZeile 7*nZeile 8*nZeile 9"
echo "gleich wird der Bilschirminhalt um 10 Zeilen nach oben gerollt"
wait 2
echo "*e[12S"wait 2echo "In zwei Sekunden wird der Bilschirminhalt um 5 Zeilen nach un-
ten scrollen"
wait 2
echo "*e[5T"echo "das wars"
wait 10

```

Ein Demo für die formatierte Textausgabe am Monitor

halten die ersten 3 Byte die ASCII-Zeichen »DOS«, ist dies für das im Kickstart-ROM der Beweis, daß der eingelesene Boot-Block-Inhalt ein Maschinenprogramm darstellt, das nun, nachdem es in den Arbeitsspeicher (RAM) eingelesen wurde, gestartet werden muß. Enthalten die genannten ersten 3 Byte nicht diese Kennung, stellt sich der Amiga weiterhin taub und verlangt nach wie vor nach der »Workbench-Disk«. Das erwähnte Maschinenprogramm (sofern es sich um das Standardprogramm von Commodore handelt) bewirkt, daß der Amiga in die Lage versetzt wird, auf Tastatureingaben zu reagieren und einen CLI-Task zu starten. Danach sucht der Amiga auf der Boot-Disk nach einem Verzeichnis namens »s«. Ist dieses nicht vorhanden, wird der Boot-Vorgang beendet, und Sie befinden sich im CLI. Existiert dieses Verzeichnis aber, und enthält es überdies noch eine Datei namens »Startup-Sequence«, werden die in dieser Datei enthaltenen Anweisungen abgearbeitet. Die auf der Workbench-Disk enthaltene »Startup-Sequence« beispielsweise ist für den Start der Intuition-Arbeitsoberfläche ebenso verantwortlich wie für die deutsche Tastaturbelegung. Anstelle des Standard-Boot-Programms können auch beliebige andere Maschinenprogramme in diesen Boot-Blöcken eingetragen sein, die ebenfalls nach dem Einlegen der Diskette ausgeführt werden – vom Intro bis zum Virus.

**Programmier-
Wettbewerb Spiele**

30 000 Mark zu gewinnen!

1. PREIS 20 000 DM **2. PREIS 6 000 DM** **3. PREIS 4 000 DM**

Warum gute Ideen verschenken? Bei uns gibt es viel Geld dafür! Machen Sie mit beim großen Spiele-Programmierwettbewerb des AMIGA-Magazins. Insgesamt 30 000 Mark warten auf die besten Spieleprogrammierer. Mitmachen und mitgewinnen heißt das Motto: Programmieren Sie Ihr Traumspiel! Es gibt noch genug Ideen, die niemand in ein Spiel umgesetzt hat. In der Thematik sind Sie ganz ohne Beschränkung. Ob Rollenspiel, harte Action, knifflige Strategiespiele, Simulationen, Sport oder anderes - nur die Qualität allein zählt. Natürlich kann Ihr Spiel auch ein Adventure sein (dann aber bitte mit Lösungsweg für uns). Gute Spiele müssen übrigens nicht unbedingt lang sein. Auf die Idee und den Spielspaß kommt es an.



COUPON

SPIELE-PROGRAMMIER-WETTBEWERB

Für eine Teilnahme am Spiele-Programmier-Wettbewerb verwenden Sie bitte diesen Coupon. Programme ohne diesen Coupon nehmen nicht am Wettbewerb teil.

Name: _____ Vorname: _____

Straße: _____

PLZ, Wohnort: _____

Alter: _____ Beruf: _____

Name des Programms: _____

Hiermit erkläre ich mich mit den abgedruckten Teilnahmebedingungen einverstanden. Das oben genannte Programm soll am Markt & Technik-Spiele-Programmier-Wettbewerb teilnehmen. Ich habe es vollständig selbst erarbeitet und nicht, auch nicht teilweise, anderen Veröffentlichungen entnommen. Das Programm ist frei von Rechten anderer Personen und liegt zur Zeit keinem Dritten zur Veröffentlichung vor.

Ich bin damit einverstanden, daß die Markt & Technik Verlag AG das Programm in ihren Zeitschriften oder Büchern abdruckt sowie (beispielsweise durch die Herstellung von Disketten) vervielfältigt und verbreitet.

Bei Druck oder sonstiger Verwertung meines Programms erhalte ich ein entsprechendes Honorar.

_____, den _____

(Unterschrift)

Bei Minderjährigen muß diese Erklärung vom gesetzlichen Vertreter bestätigt werden:

_____, den _____

(Unterschrift)





SUPERCHANCE
Die 30 000 Mark in diesem Koffer
könnten bald Ihnen gehören.

TEILNAHME- BEDINGUNGEN

- Jede Art von Spiel ist erlaubt.
- Auf der Diskette (Amiga-Format, 3 1/2 Zoll) muß das lauffähige Programm sowie der Quellcode enthalten sein.
- Eine ausführliche Dokumentation muß auf Diskette und ausgedruckt vorliegen.
- Ihr Programm muß frei von Rechten Dritter und noch unveröffentlicht sein.
- Programme ohne ausgefüllten Coupon nehmen nicht am Spiele-Programmier-Wettbewerb teil.
- Mitarbeiter der Markt & Technik Verlag AG und deren Angehörige dürfen nicht teilnehmen.
- Der Rechtsweg ist ausgeschlossen.
- Einsendeschluß: 15. April 1991.
- Schicken Sie die Unterlagen an:

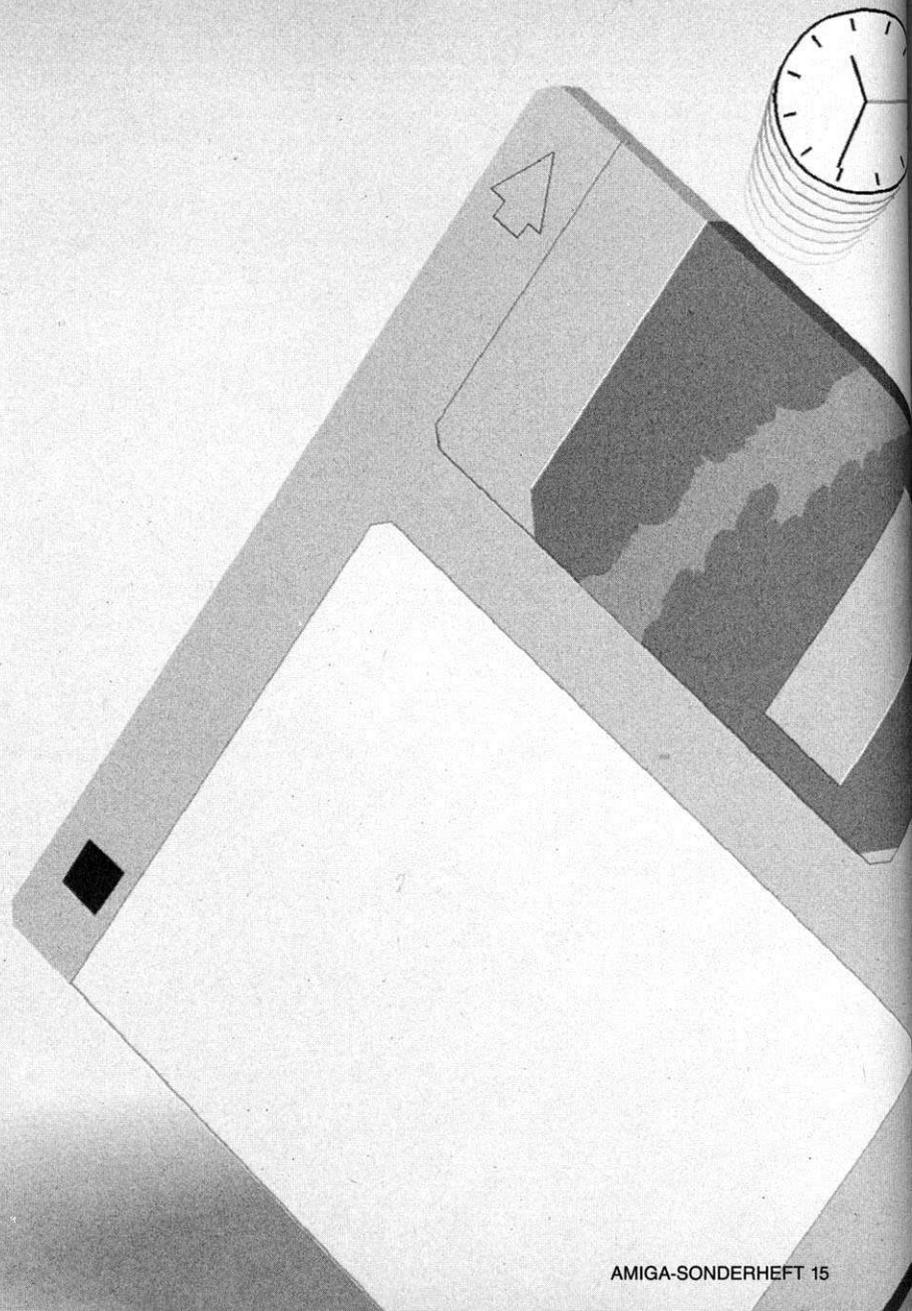
Markt & Technik Verlag AG
Redaktion AMIGA
Stichwort: 30 000 Mark
Hans-Pinsel-Straße 2
8013 Haar bei München

Workbench-Disketten

Entdeckungsfahrt durch die Workbench

Die Workbench ist für jeden Amiga-Besitzer eine Selbstverständlichkeit.

Jedermann verwendet sie mehrmals täglich, um mit seiner »Freundin« arbeiten zu können. Doch wissen Sie wirklich, was alles hinter dieser so einfach zu bedienenden Workbench steckt? Kommen Sie mit auf eine Entdeckungsfahrt durch die Workbench-Diskette und erleben Sie, wie flexibel und anpassungsfähig diese Arbeitsoberfläche ist.





von Hartwig Taubenberger

Jeder frischgebackene Amiga-Besitzer, der seinen Computer auspackt, entdeckt mindestens zwei Disketten. Die eine hat die Aufschrift »Workbench 1.3D«, die andere trägt eine Etikette, auf dem »Extras 1.3D« steht. Diese beiden Datenträger enthalten alle Daten, die für die Arbeit mit Ihrem Amiga wichtig sind. Im folgenden wollen wir den Inhalt dieser beiden Disketten Schritt für Schritt besprechen und aufzeigen, für welche speziellen Einsatzgebiete die jeweiligen Dateien und Directories notwendig sind.

Eine von ungeduldigen Anwendern oft mißmutig gestellte Frage. Dadurch ergibt sich eine relativ lange Wartezeit nach jedem Neustart des Rechners. Wäre es nicht einfacher und sinnvoller, die komplette Arbeitsoberfläche fix in den Computer ein-

zubauen, so daß sie gleich nach dem Einschalten zur Verfügung steht?

Diese Frage ist durchaus berechtigt, doch wird sie sich in diesem Artikel von selbst beantworten. Die Workbench als solche besteht nämlich nicht nur aus einem kurzen Programm, sondern aus einer Vielzahl von Einzelteilen, die jeweils verschiedene Aufgaben übernehmen. All diese Einzelteile werden laufend verbessert und optimiert. Um nun dem Anwender die Möglichkeit zu geben, mit der neuesten Version arbeiten zu können, ist es notwendig, die Arbeitsoberfläche auf Diskette zu halten. Es ist nämlich weit einfacher, eine Diskette zu kopieren, als einen ROM-Baustein auszutauschen.

Ein weiterer Grund liegt darin, daß dem Anwender auf diese Art und Weise die Möglichkeit gegeben wird, sich seine eigene, individuelle Werkbank zusammenzustellen, ohne ein Hardware-Bastler zu sein.

Nachdem Sie Ihren Computer aufgebaut und eingeschaltet haben, tut sich enttäuschend wenig. Auf dem Bildschirm erscheint eine stilisierte Hand mit einer Diskette, die dazu auffordert, die Workbench einzulegen. Das Laufwerk knattert in regelmäßigen Abständen vor sich hin, doch sonst ist von den sagenhaften Eigenschaften des Amiga nichts zu sehen. Der Grund dafür ist einfach, weil sich die Arbeitsoberfläche auf Diskette befindet. Der Computer weigert sich so lange Eingaben vom Benutzer anzunehmen, bis er die Workbench-Diskette zwischen die Schreib-/Leseköpfe seines Laufwerks bekommen hat.

HERMANN DER USER



I. Die Workbench 1.3

Legen Sie die Diskette mit der Aufschrift »Workbench 1.3D« in Ihr internes Laufwerk und warten Sie ab. (Bitte verwenden Sie immer eine Kopie und verwahren Sie die Originaldiskette an einem sicheren Ort.) Nach kurzer Zeit beginnt das Lämpchen des Disketten-Laufwerks zu leuchten und der Computer die Daten einliest. Es dauert nicht lange und auf dem Bildschirm erscheint eine Meldung, die Angaben über die Herkunft des Computers und das Erstellungsjahr Ihrer Kickstart enthält. Nachdem nun noch die Uhrzeit angezeigt wird, bzw. bei Computern ohne Speichererweiterung mit Echtzeituhr darauf hingewiesen wird, daß keine Uhr gefunden werden konnte, erscheint der Arbeitstisch auf dem Monitor.

In der rechten, oberen Ecke sehen wir zwei Bildchen, sog. Piktogramme, die zwei Disketten darstellen. Unter dem ersten steht der Name »RAM DISK«, unter dem zweiten »Workbench 1.3D«. Genau dieses zweite Symbol interessiert uns jetzt. Es symbolisiert jene Workbench-Diskette, die wir nach dem Einschalten unseres Amiga eingelegt haben.

Hinter diesem Piktogramm verbergen sich all jene Geheimnisse, die wir gemeinsam erkunden wollen. Klicken Sie dieses Bildchen zweimal an. Es öffnet sich ein Fenster, in dem alle direkt von der Workbench erreichbaren Dateien und Unterverzeichnisse angezeigt werden.

Sie müßten nun fünf Schubladen, einen Mülleimer und ein Bildchen, das ein Fenster darstellt, sehen. Unter den Schubladen stehen die Namen »Utilities«, »System«, »Expansion«, »Empty« und »Prefs«. Unter dem Abfalleimer steht »Trashcan« und das Fenster-Piktogramm trägt die Aufschrift »Shell«.

Die Entdeckungsreise beginnt

Das Inhaltsverzeichnis, das Sie vor sich sehen, nennt man Root- oder Hauptverzeichnis der Diskette, da es sozusagen das oberste Verzeichnis ist, von dem man sich weiter in die Unterverzeichnisse bewegen kann. Es wurde übrigens bewußt darauf hingewiesen, daß es sich bei den hier sichtbaren Dateien und Verzeichnissen nur um jene handelt, die direkt von der Workbench per Mausklick erreichbar sind. Es gibt jedoch noch eine Menge anderer, wichtiger Files und Directories, die kein Bildchen besitzen und deshalb für uns (noch) nicht sichtbar sind.

Starten Sie deshalb die Shell, indem Sie das zugehörige Piktogramm zweimal anklicken. Es erscheint ein weiteres Fenster, mit dem Titel »Amigashell«. Des weiteren se-

hen Sie eine Eingabeaufforderung, das sog. Prompt, das Ihnen das aktuelle Verzeichnis anzeigt. In unserem Fall sollte dies der Text »1.SYS: >« sein. Daneben sehen wir ein Rechteck, den Cursor, der auf die Stelle zeigt, wo wir unsere Eingaben per Tastatur machen können. Tippen Sie den Befehl »DIR« ein und drücken Sie die Taste <RETURN>. Es erscheinen nun alle Dateien und Verzeichnisse auf dem Bildschirm, die sich auf Ihrer Workbench-Diskette befinden. Wie Sie sehen, sind dies weit mehr Programme, als wir noch kurz zuvor in Piktogrammform gesehen haben. Um die komplette Liste überblicken zu können, sollten Sie das Shell-Fenster auf volle Bildschirmgröße bringen und den Befehl »DIR« erneut eingeben.

Beginnen wir unsere Entdeckungsreise gleich mit einem Verzeichnis, das wir von der Workbench aus nicht sehen können, das aber für die Arbeit mit dieser sehr wichtig ist.

1. Das Verzeichnis »S«

So unscheinbar der Name auch wirken mag: das Verzeichnis ist beim Umgang mit der Workbench von elementarer Bedeutung. Doch beginnen wir von vorne. Geben Sie den Befehl »DIR S« ein und drücken Sie <RETURN>. Sie sehen nun den Inhalt dieses Verzeichnisses.

Der Computerstart

1.1 Startup-Sequence

Zuallererst wollen wir unser Hauptaugenmerk auf eine Datei richten, die den Namen »Startup-Sequence« trägt. Sie stellt einen der wichtigsten Bestandteile der Workbench dar, ohne jemals wirklich in den Vordergrund zu treten. Wozu ist diese Datei nötig? Um dies zu zeigen, müssen wir den Startvorgang des Amiga bis ganz zum Anfang hin verfolgen. Wie bereits erwähnt, wartet der Amiga nach jedem Neustart darauf, daß eine Diskette eingelegt wird. Was geschieht jedoch konkret, wenn ein solcher Datenträger in das Laufwerk geschoben wird? Zuerst liest der Computer die ersten beiden Blöcke (den berühmten Bootblock) der eingelegten Diskette. Enthalten diese bestimmte Informationen, die anzeigen, daß von der Diskette gebootet werden kann, d.h., daß es sich um einen Datenträger handelt, von dem aus ein Startvorgang möglich ist, so beginnt der Rechner seine Arbeit. Sollte dies nicht der Fall sein, erscheint bereits nach kurzer Zeit wiederum das Einschaltbild, mit der Anforderung eine Workbench-Diskette in das Laufwerk einzulegen.

Nehmen wir an, eine Diskette, von der aus gestartet werden kann, wurde eingelegt. Das nächste, was der Amiga tut, ist nichts anderes, als das Unterverzeichnis »s«, und darin wiederum die Datei »Startup-Sequence« zu suchen. Findet der Amiga diese nicht, bricht er die Arbeit ab und es ist lediglich das Fenster mit den oben erwähnten Informationen über Kickstart und Hersteller zu sehen, und ein einfaches Prompt, das auf Eingaben wartet. Die Arbeitsoberfläche wurde jedoch nicht geladen, und Einstellungen, wie z.B. eine deutsche Tastaturbelegung sind ebenfalls nicht durchgeführt worden. Es wäre eine mühsame Arbeit für den Benutzer, müßte er jedesmal von Hand all diese Befehle eintippen, die

die Workbench aktivieren, die Uhr einstellen oder die deutsche Tastaturbelegung festlegen. Genau aus diesem Grunde wurde die »Startup-Sequence« eingeführt. Sie ist eine Textdatei, die all die Befehle ausführt, die nötig sind, um den Computer »hochzufahren«.

Da es sich hierbei um eine Datei handelt, in der eine Anweisung nach der anderen abgearbeitet wird, so als nehme man einen Befehl nach dem anderen von einem Stapel, nennt man sie Stapelverarbeitungsdatei oder, da Stapel auf Englisch »Batch« heißt, Batch-Datei. Sie möchten gerne wissen, was die Startup-Sequence Ihrer Workbench eigentlich tut? Nun, dann tippen Sie einfach »TYPE S/STARTUP-SEQUENCE« ein. Auf dem Bildschirm wird die Textdatei ausgegeben. Wenn Sie sich bereits mit dem CLI oder der Shell befaßt haben, so werden Sie lauter gute Be-

Die »Startup-Sequence«

kannte finden, denn Batch-Dateien setzen sich in der Regel aus den Befehlen des CLI zusammen. Mehr über die »Startup-Sequence« erfahren Sie in dem Artikel zu den Batch-Dateien ab Seite 82. Wenn Sie Ihre eigene Workbench zusammenstellen möchten, müssen Sie diese Datei auf jeden Fall anlegen oder direkt übernehmen. Welche Änderungen Sie vornehmen müssen, falls Sie bestimmte Dateien Ihrer Workbench weglassen, wird immer bei der Besprechung der entsprechenden Files angegeben. Eines ist jedoch wichtig. Sie sind nicht gezwungen, immer die original »Startup-Sequence« zu verwenden.

Neben der Startup-Sequence befinden sich noch weitere Batch-Dateien in diesem Verzeichnis, die wir uns nun kurz ansehen möchten.

1.2 StartupII

Diese Stapelverarbeitungsdatei ist ein Teil der »Startup-Sequence«. Sie wurde ausgelagert, da dadurch die eigentliche »Startup-Sequence« kürzer gehalten werden konnte. Sie dient dazu, Befehle resident zu machen, d.h. sie im Speicher zu halten. Damit sie nicht laufend von Diskette bzw. Festplatte geladen werden müssen, was Geschwindigkeitsverlust mit sich bringt.

1.3 Startup-Sequence.HD

Diese Batch-Datei benötigen Sie, falls Ihre Festplatte nicht automatisch bootet. Kopieren Sie die normale »Startup-Sequence« in das »S«-Verzeichnis Ihrer Hard-Disk und benennen Sie die »Startup-Sequence.HD« in »Startup-Sequence« um. Wenn Sie nun mit dieser Workbench-Diskette starten, wird die Kontrolle automatisch an Ihre angeschlossene Festplatte übergeben.

1.4 CLI-Startup, Shell-Startup

CLI bzw Shell-Startup sind Batch-Dateien, die ausgeführt werden, wenn Sie entweder das CLI oder die Shell starten. So ist es z.B. möglich, das Prompt festzulegen oder Befehle resident zu machen.

1.5 PCD, SPAT, DPAT

Mit diesen drei Dateien werden weitere Möglichkeiten der Nutzung des »S«-Verzeichnisses gezeigt. Normalerweise werden Stapelverarbeitungsdateien mit dem Befehl »EXECUTE« gestartet. Dieser Befehl sucht zuerst in jenem Verzeichnis, in dem Sie sich gerade befinden, nach der angegebenen Batch-Datei. Sollte er diese nicht finden, sieht er als nächstes im Verzeichnis »S« Ihrer Startdiskette nach. Dies ist vor allem dann sinnvoll, wenn Sie bereits eine ganze Sammlung solcher Batch-Dateien besitzen, die über die ganze Diskette verstreut sind. Wollen Sie nämlich eine ganz bestimmte Befehlsdatei aufrufen, müßten Sie jedes Mal ganz genau angeben, wo sich diese befindet, damit sie der »EXECUTE«-Befehl auch findet. Kopieren Sie dagegen all diese Stapelverarbeitungsdateien in Ihr »S«-Directory, so können Sie sich in Zukunft diese Arbeit ersparen. Sie schreiben nur noch »EXECUTE« und den Namen Ihrer Datei, den Rest erledigt der Computer.

1.6 Weitere Möglichkeiten

Das »S«-Directory wird häufig noch für andere Zwecke verwendet. Da sich die »Startup-Sequence«, die mit dem Start des Computers zu tun hat, in diesem Verzeichnis befindet, gibt es viele andere Programme,

die auch Ihre Starteinstellungen (sog. »Default-Werte«) in diesem Verzeichnis ablegen.

2. Das Verzeichnis »C«

Hier befinden sich all jene Befehle, die Sie vom CLI oder von der Shell aus nutzen, oder auf die die »Startup-Sequence« zurückgreift.

Ein Verzeichnis für alle Befehle: »C«

Tippen Sie einfach »DIR C« und <RETURN>. Nach kurzer Zeit erscheinen alle Befehle, die sich in diesem Verzeichnis befinden, auf dem Bildschirm. Diese einzeln zu besprechen, käme jedoch einer Einführung in die Arbeit mit dem CLI bzw. mit der Shell gleich, lesen Sie dazu den Artikel über das CLI ab Seite 34. Deshalb hier nur einige grundsätzliche Bemerkungen zu diesem Verzeichnis. Wenn Sie bei der Arbeit mit CLI/Shell einen Befehl eintippen, so sucht der Computer zuerst im aktuellen Verzeichnis (dies ist jenes Verzeichnis, das angezeigt wird, wenn Sie »DIR« eingeben). Findet er diesen nicht, sucht er als nächstes automatisch im Verzeichnis »C«. Der Vorteil dieser Methode liegt auf der Hand: Sie können alle Befehle, die Sie nachträglich hinzufügen, in dieses Directory kopieren und der Amiga wird sie immer finden, ohne daß Sie angeben müssen, wo sich diese befinden. Wenn Sie sich fragen, welche Befehle man hinzufügen könnte, sehen Sie sich einmal einige Public-Domain-Disketten an oder werfen Sie ein Blick auf die Tips&Tricks-Rubriken des AMIGA-Magazins. Sie werden sich wundern, wie viele Ergänzungen es für Ihr »C«-Verzeichnis gibt! Falls Sie eine eigene Workbench zusammenstellen, ist es nicht notwendig, daß Sie das komplette Verzeichnis übernehmen. Es genügt, nur jene Befehle zu kopieren, die Sie wirklich benötigen. Achten

Sie dabei darauf, daß Ihre »Startup-Sequence«, Ihre Batch-Dateien oder ihre »CLI- bzw. Shell-Startup«-Dateien nur Befehle enthalten, die Sie übernommen haben, da es sonst zu Fehlermeldungen kommt.

Eine Bemerkung noch an dieser Stelle. Es gibt nämlich noch eine weitere Möglichkeit, dem Computer mitzuteilen, wo er seine Befehle zu suchen hat, falls er sie weder im aktuellen, noch im C-Verzeichnis findet. Es existiert dazu das Programm »PATH«, das Sie auch in Ihrer »Startup-Sequence« wiederfinden können. Mit diesem ist es möglich, dem Amiga anzugeben, wo er nach dem eingegebenen Befehl suchen soll. Tippen Sie »TYPE S/STARTUP-SEQUENCE« ein, und suchen Sie nach jener Zeile, in der sich der Befehl »PATH« befindet. Wie Sie sehen, wird in diesem Fall nicht nur im »C«-Verzeichnis nach dem Befehl gesucht, sondern auch in den Directories RAM, Utilities, System, S und Prefs. Trotzdem sollten Sie sich aufgrund der Übersichtlichkeit angewöhnen, alle neuen Befehle in das »C«-Verzeichnis zu kopieren. Der Grund, warum der Befehl »PATH« an dieser Stelle erwähnt wird, ist jener, daß Sie die Startup-Sequence ändern müssen, falls Sie eines der oben erwähnten Verzeichnisse löschen. Ansonsten kommt es zu einer Fehlermeldung, da der Amiga dieses Directory nicht finden kann.

Der Datenpuffer

3. Das Verzeichnis »t«

Dies ist ein weiteres Directory, dessen Name keinerlei Rückschlüsse über die Verwendung zuläßt. Das »t« steht in diesem Fall für den englischen Begriff »temporary«, also zeitweilig. Hierher legen Programme Daten ab, die sozusagen nur zwischengespeichert, und meist bei Beendigung des Programms wieder gelöscht werden. Vor allem Textverarbeitungsprogramme bzw. Texteditoren (z.B. Wordperfect) machen davon Gebrauch. Um die Workbench trotzdem in schreibgeschütztem Zustand lassen zu können, wird dieses Verzeichnis bereits in der »Startup-Sequence« in die »RAM-Disk« ausgelagert, wodurch zwar Speicher verloren geht, die wertvolle Arbeitsdiskette aber geschützt ist. (Die Be-

fehle, die dies bewirken, befinden sich in der Datei »StartupII«.) Sollten Sie auf Ihrer eigenen Workbench das Verzeichnis nicht auslagern, so daß die Daten auf Diskette geschrieben werden (vor allem bei wenig Arbeitsspeicher interessant), so sollten Sie dieses Verzeichnis von Zeit zu Zeit überprüfen. Denn verlassen Sie ein Programm, das auf das Directory »t« zugreift, durch Ausschalten oder Reset, so kann es sein, daß sich noch Daten darin befinden, die nun wertlos sind und deshalb ersatzlos gelöscht werden können.

4. Das Verzeichnis »libs«

Der Name »libs« ist die Abkürzung für »libraries«, was auf deutsch soviel wie Bibliotheken bedeutet. Natürlich erhebt sich die Frage, wozu ein Computer wie der Amiga eine Bibliothek benötigt. Ein Computer wird laufend verbessert und weiterentwickelt. Trotzdem sollten alle Programme, die auf früheren Modellen liefen, auch auf den neuesten Versionen von Betriebssystem und Hardware problemlos funktionieren. Deshalb ist es nicht möglich, einfach, wie bei älteren Computern, z.B. dem C64, den Wert einer bestimmten Speicherstelle zu ändern, um die Bildschirmfarbe zu wechseln oder ähnliche Dinge auszuführen. Es kann zwar durchaus sein, daß in der derzeitigen Version des Amiga diese Speicherstelle für diesen speziellen Zweck benutzt wird. Dies bedeutet jedoch nicht, daß auch alle weiteren Versionen dabei bleiben. Unter Umständen ist in einer zukünftigen Weiterentwicklung des Amiga ausgerechnet dieser Wert für ganz andere Aufgaben gedacht. Die Folge ist, daß Programme, die direkt auf Speicherstellen zugreifen, Gefahr laufen, auf zukünftigen Amiga-Modellen nicht mehr zu funktionieren. Bestes Beispiel sind die vielen Spiele, die bereits heute ihren Dienst mit dem Amiga 3000 verweigern, da die Programmierer mit ihrem Wunsch, möglichst alles aus dem Computer herauszuholen, genau die oben erwähnten Techniken angewendet haben.

Die Bibliotheken des Amiga

Wie Sie sich sicher bereits denken können, sind jene Bibliotheken, die sich im Verzeichnis »libs« befinden, dazu da, die oben erwähnten Probleme zu vermeiden. Commodore stellt mit diesen Libraries dem Programmierer fertige Routinen zur Verfügung, mit denen er den kompletten Computer programmieren kann, ohne direkt auf die Hardware zugreifen zu müssen. Dies bedeutet konkret, daß, falls sich der Aufbau des Computers oder das Betriebssystem ändern, nicht alle Programme, sondern nur die Bibliotheken geändert werden müssen, was natürlich weit weniger Aufwand bedeutet.

Auf der Workbench-Diskette befinden sich bereits einige wichtige Bibliotheken, die nun kurz besprochen werden, wobei gleichzeitig darauf eingegangen wird, ob sie auf eigenen Workbench-Disketten notwendig sind oder nicht. Zuvor sollten Sie jedoch wissen, daß sich nicht alle Libraries auf Diskette befinden. Jene, die häufig benötigt werden, da sie die Ein-/Ausgaben, Grafikoperationen usw. steuern, sind fest in Ihrem Amiga eingebaut, da dadurch Geschwindigkeitsvorteile erzielt werden.

4.1 diskfont.library

Die »diskfont.library« wird immer dann benötigt, wenn Sie neben den im Betriebssystem eingebauten Schriften Topaz 6 bzw. Topaz 8 auch jene verwenden möchten, die sich auf Diskette befinden (dazu später mehr). Da relativ viele Programme auf diese Bibliothek zurückgreifen, sollten Sie sie auf Ihrer eigenen Workbench belassen.

4.2 icon.library

Diese Bibliothek steuert die Arbeit mit Piktogrammen und stellt viele weitere Funktionen zur Verfügung, die von der Workbench benutzt werden. Deshalb muß sich dieses »library« unbedingt auf Ihrer Workbench-Diskette befinden. Sollten Sie einmal darauf vergessen, dieses File in Ihr »libs«-Verzeichnis zu kopieren, so

wird dies mit einem »Recoverable Alert«, also einem Systemabsturz belohnt, sobald der Befehl »LOADWB« aufgerufen wird.

4.3 info.library

Die Bibliothek »info.library« wird vom Menüpunkt »Info« im Workbench-Menü verwendet. Damit können Angaben über Art, Größe usw. von Programmen oder Datenträgern gemacht werden. Befindet sich diese Bibliothek nicht auf Ihrer Workbench-Diskette, und Sie rufen den entsprechenden Menüpunkt auf, so erhalten Sie die Systemmeldung »Info not available«.

4.4 mathieedoubbas.library

Programme, die mit hoher Rechengenauigkeit arbeiten, benötigen diese Bibliothek. Sie stellt viele Rechenoperationen zur Verfügung, die in doppelter Genauigkeit ausgeführt werden. Sollten Sie also Programme (z.B. Ray-Tracer) verwenden, die aufwendige Berechnungen zu Verfügung stellen, so müssen Sie diese Bibliothek auf Ihre Workbench-diskette kopieren.

Für Mathematiker

4.5 mathtrans.library, mathieedoubtrans.library

Diese Libraries werden verwendet, um transzendente Funktionen wie Sinus, Cosinus usw. in einfacher bzw. doppelter Genauigkeit berechnen zu können. Programme, die solche Operationen durchführen, benötigen diese Bibliotheken unbedingt.

4.6 translator.library

Immer, wenn der Amiga sprechen soll, wird diese Bibliothek benötigt. Sie wandelt die eingegebenen Texte so um, daß Sie der Computer aussprechen kann. Falls Sie Programme benutzen, die Sprachausgabe unterstützen, sollten Sie diese Library auf Ihre Workbench übernehmen.

4.7 version.library

Die »version.library« ist eine eher selten benötigte Bibliothek, die nur von Interesse ist, falls Sie regelmäßig mit dem Befehl Version im Spezial-Menü der Workbench die aktuelle Versionsnummer Ihrer Arbeitsoberfläche abfragen möchten. Ansonsten erhalten Sie beim Aufruf dieses Menüpunkts die Angabe »Workbench Version -1.-1«.

5. Das Verzeichnis »l«

Dieses so unscheinbar aussehende Verzeichnis birgt einige Dateien in sich, die für die Arbeit mit der Workbench 1.3 fast überlebenswichtig sind. Trotzdem stellt gerade dieses Directory für viele Anwender ein Buch mit sieben Siegeln dar, da der Sinn der enthaltenen Programme oft nicht verstanden wird. Sehen Sie sich zuerst einmal den Inhalt des »l«-Verzeichnisses an, indem Sie »DIR L« und <RETURN> eintippen. Neben dem »Disk-Validator«, dem »Shell-Seg« und dem »FastFileSystem« finden Sie hier Dateien, die am Ende den Namen »-Handler« tragen.

Die Datenhändler

Welche Aufgaben erfüllen nun solche Handler? Nun, der Begriff selbst ist vom englischen Wort »handling« abgeleitet, was soviel bedeutet wie steuern oder handhaben. Es handelt sich also um Programme, die bestimmte Vorgänge im Amiga steuern. Welche dies genau sind, wird bei der folgenden Einzelbesprechung jeweils noch genauer ausgeführt. Soviel läßt sich jedoch allgemein sagen: alle diese Steuerprogramme stellen dem Benutzer sog. physische Geräte zur Verfügung. Das heißt Sie können diese Geräte so ansprechen, wie Sie es auch mit Disketten-Laufwerken tun. Sie haben also vor allem von der Shell aus die Möglichkeit, alle Umleitungsbeefehle (<, >, usw.) zu verwenden. Wichtig ist nur, daß all diese Geräte in der »Mountlist« eingetragen und mit dem Befehl »MOUNT« angemeldet werden. Betrachten wir die einzelnen Handler im Detail:

5.1 Aux-Handler

Der »Aux-Handler« bietet dem Anwender die Möglichkeit, Ein- und Ausgaben direkt über die serielle Schnittstelle zu machen, ohne vorher in einen Puffer zu schreiben. Mit diesem Programm ist es z.B. möglich, dem Amiga mitzuteilen, daß er Eingaben in Zukunft nicht mehr von Tastatur, sondern von der seriellen Schnittstelle annehmen soll, bzw. umgekehrt, alle Ausgaben nicht auf

dem Monitorbildschirm, sondern über den seriellen Port ausgehen soll. Falls Sie diese Möglichkeiten auf Ihrer eigenen Workbench-Diskette nicht nutzen wollen, können Sie diesen Handler ruhig weglassen, wobei Sie in der Datei »StartupII« im Verzeichnis »s« die Zeile »mount aux:« löschen müssen.

5.2 Newcon-Handler

Dieser Handler wird wohl am häufigsten benötigt, ohne, daß es dem Benutzer direkt auffällt. Er wird automatisch in der »Startup-Sequence« angemeldet und jedesmal aktiviert, wenn Sie die Shell aufrufen. Was tut dieses Steuerprogramm nun genau? Wenn Sie bereits mit dem CLI gearbeitet haben (d.h. Sie haben nicht das Piktogramm »Shell«, sondern das Bildchen »CLI« in der Schublade »System« angeklickt), so wird Ihnen der Unterschied zwischen der Arbeit mit der Shell und

Shell oder CLI

dem CLI sicherlich bereits aufgefallen sein. »NEWCON:« ersetzt nämlich den bisher verwendeten Handler »CON:« und bietet viele neue Möglichkeiten, die das Arbeiten mit der Shell stark vereinfachen. So ist es z.B. möglich, sich mit den Cursor-Tasten in einer Zeile zu bewegen, um Tippfehler auszubessern oder Befehlszeile einzufügen. Früher mußte man dazu die gesamte Zeile bis zu der Stelle, an der sich der Fehler befand, löschen. Des weiteren ist es nun möglich, eingetippte und mit <Return> abgeschlossene Befehlszeilen wieder aufzurufen, indem man mit der Taste Cursor-Up alle bisher eingetippten Befehle »durchblättert«. Wie angenehm dies ist, kann sich sicherlich jeder vorstellen, der bereits einmal eine lange Befehlszeile eingetippt hat, und erst, nachdem er die <Return>-Taste betätigt hat, merkt, daß sich ein Tippfehler in dieser Zeile befindet. Während es früher notwendig war, den gesamten Befehl noch einmal zu tippen, genügt es nun, einfach auf die Cursor-Up-Taste zu drücken und den

Fehler auszubessern. Der Zwischenspeicher, der diese Zeilen aufnimmt, ist auf 2 KByte beschränkt. Dies dürfte allerdings auch dem anspruchsvollsten Anwender genügen und verhindert, daß unnötig Speicherplatz verlorengeht. Dieser Handler sollte sich auf jeden Fall auf Ihrer Workbench befinden, da es sonst nicht möglich ist, die Shell zu verwenden. Falls Sie diesen trotzdem weglassen, müssen Sie in der »Startup-Sequence« die Zeile »mount newcon:« löschen, da es sonst bestimmt zu einer Fehlermeldung kommt.

5.3 Pipe-Handler

Der »Pipe-Handler« ist ein Ein-/Ausgabeprogramm, das die Kommunikation zwischen Programmen ermöglicht. Das bedeutet, daß Sie, ohne eine Datei auf Diskette anzulegen, Daten von einem Programm an ein anderes übergeben können, indem Sie es einfach in das Geräte »PIPE:« schreiben, und von dem anderen Programm lesen. Ein Beispiel: Sie haben zwei Shell-Prozesse gestartet, und möchten nun Daten von der ersten Shell an die zweite übergeben. Dazu tippen Sie einfach in der einen Shell »COPY Daten PIPE:Daten« und nun können Sie von der zweiten diese Daten z.B. mit »TYPE PIPE:Daten« weiterverwenden.

Die Pipeline

Die Größe des Zwischenspeichers ist auf 4 KByte beschränkt, was für die meisten Anwendungen reichen dürfte. Auch dieser Handler ist nur dann auf Ihrer Workbench notwendig, wenn Sie planen, diese Möglichkeiten zu nutzen. Ansonsten müssen Sie in der Datei »StartupII« im »s«-Verzeichnis der Workbench die Zeile »mount pipe:« löschen.

5.4 Port-Handler

Dieser Handler steuert die Ausgaben über die Geräte »PAR:« (paralleler Port), »SER:« (serieller Port) und »PRT:« (Ausgabe über den Drucker-treiber), die wohl häufig verwendet werden. Deshalb sollten Sie diesen unbedingt auf Ihrer Workbench belassen, da sonst die Ausgabe auf den Drucker und andere Peripheriegeräte nicht möglich ist.

5.5 RAM-Handler

Die »RAM-DISK« wird von diesem Steuerprogramm verwaltet. Wenn sie vorhaben, diese zu benutzen, müssen Sie diesen Handler auf Ihre Workbench übernehmen.

5.6 Speak-Handler

Der »Speak-Handler« wird selten benutzt, bietet aber einige interessante Möglichkeiten. Mit ihm kann man Daten, die normalerweise auf den Bildschirm ausgegeben werden, vom Computer »vorlesen« lassen. Probieren Sie es aus, indem Sie folgendes eintippen: »COPY S/STARTUP-SEQUENCE TO SPEAK:«. Falls Sie diesen Handler auf Ihrer Workbench weglassen, müssen Sie die Zeile »mount speak:« in der Datei »StartupII« im »s«-Verzeichnis Ihrer Diskette löschen.

Der Amiga spricht

Sollten Sie auf Ihrer eigenen Workbench die original »Startup-Sequence« verwenden, brauchen Sie alle bisher erwähnten Geräte nicht mit dem Befehl »MOUNT Gerätenamen:« anmelden, da dies bereits in der Datei »StartupII« getan wird, die beim Start des Computers ausgeführt wird. Ansonsten dürfen Sie dieses Anmelden nicht vergessen, denn nur dann ist es möglich, auf diese Geräte zuzugreifen.

Im Verzeichnis »I« befinden sich noch einige weitere wichtige Dateien, die nach Möglichkeit auf Ihrer Workbench verbleiben sollten.

5.7 Disk-Validator

So unscheinbar dieses Programm auch aussehen mag, ohne ist es nicht möglich, auf einen Datenträger zu schreiben. Es überprüft bei jedem Diskettenwechsel die neue Diskette und gibt sie erst nachdem festgestellt wurde, daß diese in Ordnung ist, für Schreibzugriffe frei. Die Systemmeldung »Error validating Disk« stammt übrigens von diesem »Disk-Valida-

tor« und bedeutet, daß ab sofort auf diese Diskette nicht mehr geschrieben werden kann, da der Amiga einen Fehler auf der Diskette vermutet. Sie sehen also, daß dieses Programm von größter Bedeutung für die Arbeit mit dem Computer ist, und deshalb auf keiner Diskette fehlen darf. Wichtig ist hierbei noch, daß der Disk-Validator unbedingt in dem Verzeichnis »I« stehen muß, da er sonst nicht gefunden wird!

5.8 Fast-File-System

Das »Fast-File-System« (FFS) ist eine Verbesserung des alten »File-Systems« und ermöglicht einen schnelleren Zugriff auf Datenträger. In der Version 1.3 unterstützt das »FFS« nur Festplatten. Ab der Version 2.0, die mit dem Amiga 3000 ausgeliefert wird, unterstützt es auch Disketten-Laufwerke.

5.9 Shell-Seg

Dieses Programm sorgt dafür, daß immer, wenn die Shell aufgerufen wird, diese mit den neuen Vorzügen der Shell ausgestattet wird und nicht den alten, unkomfortablen Befehlsprozessor des CLI verwendet. Deshalb ist dieses Programm immer dann wichtig, wenn Sie die Shell verwenden möchten.

Vorsicht DEVS

6. Das Verzeichnis »devs«

Dieses Verzeichnis enthält sog. Gerätetreiber. Diese sind ebenso wichtig wie die Handler des »I«-Verzeichnisses und erfüllen auch ähnliche Aufgaben. Auch befinden sich in diesem Directory drei Unterverzeichnisse mit den Namen »keymaps«, »printers« und »clipboards«.

6.1 clipboard.device

Dieses Device stellt einen Zwischenspeicher für Daten zur Verfügung, der über den Namen CLIPS: angesprochen werden kann. Allerdings wird dies äußerst selten verwendet. Falls Sie dieses Device weglassen, müssen Sie in der Datei »StartupII« die Zeilen »makedir ram:clipboards« und »assign CLIPS: ram:clipboards« löschen.

6.2 narrator.device

Das englische Wort »narrator« bedeutet soviel wie Erzähler. Damit dürfte klar sein, wozu dieser Gerätetreiber gut ist. Er wird benötigt, um dem Amiga das Sprechen beizubringen. Ist dieses Device nicht vorhanden, verweigert der Computer sozusagen jede Aussage.

6.3 parallel.device

Die Steuerung der parallelen Schnittstelle wird von diesem Device übernommen. Alle Ein-/Ausgaben wie die direkte Ansteuerung eines Druckers, der an der parallelen Schnittstelle angeschlossen ist, laufen über diesen Gerätetreiber.

Schwarz auf weiß

6.4 printer.device

Das »printer.device« dient zur Ausgabe von Daten an den Drucker, wobei Rücksicht auf alle in den Preferences gemachten Einstellungen genommen wird. Verwenden Sie also in der Shell das Gerät »PRT:«, so wird dieses Device angesprochen. Es sollte sich auf jeder Workbench-Diskette befinden, falls ein Drucker verwendet wird.

6.5 ramdrive.device

Wenn Sie eine »RAM-Disk« anlegen möchten, so benötigen Sie diesen Gerätetreiber, der die Verwaltung derselben übernimmt.

6.6 serial.device

Für dieses »device« gilt dasselbe wie beim »parallel.device«, nur daß diesmal die serielle Schnittstelle angesprochen wird.

6.7 MountList

In dieser Textdatei, die Sie mit dem Befehl »TYPE DEVS/MOUNTLIST« einsehen können, werden die Eigenschaften all jener Geräte festgelegt, die von den Handlern und Devices gesteuert werden. Immer, wenn Sie den Befehl »MOUNT« anwenden, wird auf diese Liste zurückgegriffen. In ihr stehen genaue Informationen über die einzelnen Geräte, wie die Anzahl der Blöcke auf der Festplatte oder die Angabe, ob es sich bei einem 5,1/4-Zoll-Laufwerk um eines mit 40 oder 80 Spuren handelt. Diese Liste sollten Sie auf jeden Fall auf Ihre eigene Workbench übernehmen und bei Bedarf an Ihre Bedürfnisse anpassen.

6.8 system-configuration

Wenn Sie sich gefragt haben, wohin das Preferences-Programm seine Daten abspeichert, so haben Sie nun die Antwort gefunden. In der Datei »system-configuration« stehen all jene Werte, die Sie in den Preferences einstellen, wie z.B. den verwendeten Druckertreiber, die Farben der Workbench usw. Wenn Sie dieses File in das »devs«-Verzeichnis einer anderen Diskette kopieren, so hat diese dann dieselben Einstellungen übernommen. (Genau das macht übrigens auch das Programm »CopyPrefs« im »Prefs«-Verzeichnis.)

6.9 printers

Im Unterverzeichnis »printers« stehen die momentan verfügbaren Druckertreiber. Falls Sie immer mit dem selben Drucker arbeiten, genügt es, wenn Sie nur den von Ihnen benötigten Druckertreiber von der »Extras1.3D«-Diskette in dieses Verzeichnis kopieren und alle anderen Treiber löschen.

Tastaturbelegung

6.10 keymaps

Falls Sie mit dem Befehl »SETMAP« aus dem »System«-Verzeichnis eine andere Tastaturbelegung einstellen möchten, muß sich hier der entsprechende Tastaturreiber befinden, sonst kommt es zu einer Fehlermeldung. Das heißt falls Sie auf Ihrer Workbench mit dem Befehl »SYSTEM/SETMAP D« die deutsche Tastaturbelegung einstellen möchten, muß sich die Datei »D« im Verzeichnis keymaps befinden.

6.11 clipboards

Dieses Verzeichnis ist als externer Datenspeicher für das »Clipboards.device« angelegt. Es wird dann gebraucht, wenn ein Programm Daten an das Device sendet, aber diese nicht mehr abrufen kann. Dann werden diese Daten hier gesichert.

7. Das Verzeichnis »fonts«

Viele Programme bieten die Möglichkeit, neben den beiden Standardschriften Topaz 6 und Topaz 8 weitere Zeichensätze von Diskette nachzuladen (z.B. Notepad oder DPaint III). Damit der Amiga diese Schriften auch findet, müssen sie sich im Verzeichnis »fonts« befinden. Jeder Zeichensatz besteht aus zwei Teilen. Der Schrift selbst, die die Endung ».font« besitzt, und einem Unterverzeichnis, das alle Schriftgrößen enthält, die zur Verfügung stehen. Es steht Ihnen frei, so viele oder wenige Schriftarten wie Sie wollen auf Ihre Workbench zu kopieren, je nachdem, ob Sie Wert auf eine große Auswahl an verschiedenen Zeichensätzen legen oder nicht.

Die Systemfonts

Nachdem wir nun die Tiefen unserer Workbench-Diskette erforscht haben, kehren wir wieder an die Oberfläche zurück und betrachten in etwas kürzerer Form jene Dateien, die mit einem Piktogramm versehen sind und sich deshalb direkt von der Workbench aus aufrufen lassen. Sollten Sie sich noch in der Shelloberfläche befinden, tippen Sie einfach »ENDSHELL« und drücken die <Return> Taste.

Falls Sie sich eine eigene Workbench zusammenstellen, und Sie lassen eines der drei folgenden Verzeichnisse (Prefs, Utilities und System) weg, sollten Sie in Ihrer »Startup-Sequence« die drittletzte Zeile, die den Befehl »PATH« enthält, dahingehend ändern, daß dieses Verzeichnis nicht mehr erscheint. Wenn Sie die Schublade »Prefs« völlig löschen, müssen Sie auch die Zeichenfolge »SYS:PREFS« in der oben erwähnten Zeile entfernen, sonst kommt es zu einer Fehlermeldung.

8. Das Verzeichnis »Prefs«

8.1 Preferences

Dieses Programm dient zu Einstellung der Startwerte des Computers, d.h. Dinge, wie die Art des angeschlossenen Druckers, die Farben

und Auflösung der Workbench, die Form des Mauszeigers, Datum und Systemzeit und vieles mehr, werden hier festgelegt. Sollten Sie auf Ihrer eigenen Workbench-Diskette nicht vorhaben, die Grundeinstellungen zu verändern, können Sie dieses Programm ruhig weglassen.

8.2 Printer, Serial, Pointer

Diese drei Piktogramme tun nichts anderes, als das Programm Preferences aufzurufen, wobei sie direkt in den jeweils gewählten Einstellungsbereich springen. Das heißt, wenn Sie das Icon »Printer« anklicken, öffnet sich sofort der Bildschirm zum Einstellen des jeweiligen Druckertreibers.

8.3 CopyPrefs

Mit diesem Programm ist es möglich, die Einstellungen Ihrer Workbench-Diskette auf andere Disketten zu kopieren.

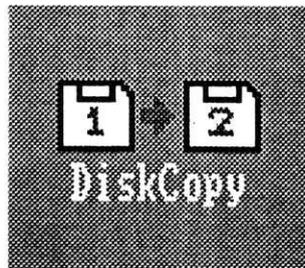
9. Das Verzeichnis »System«

9.1 CLI

Dieses Piktogramm gibt Ihnen die Möglichkeit, auf die Annehmlichkeiten der Shell zu verzichten und mit dem äußerst unkomfortablen Ur-CLI zu arbeiten. Einziger Vorteil: das CLI benötigt etwas weniger Speicherplatz zum Starten.

9.2 DiskCopy

Direktes Anklicken dieses Programms bewirkt eine Systemmeldung, die darauf hinweist, daß es nur mit dem Menüpunkt Duplicate des Workbench-Menüs aufrufbar ist. Es dient zum Kopieren von Disketten.



Sollten Sie vorhaben, diesen Menüpunkt auf Ihrer eigenen Workbench zu verwenden, so muß sich dieses Programm unbedingt auf Ihrer Boot-Diskette befinden.

9.3 FastMemFirst

Der Commodore Amiga teilt seinen Speicher in zwei Bereiche ein: das Chipmem und das Fastmem. Auf das Chipmem können neben dem Hauptprozessor (CPU - bei ei-

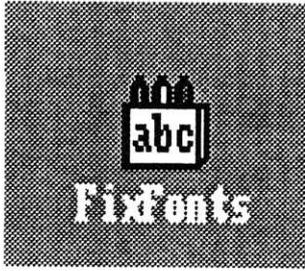
nen normalen Amiga der 68000er Prozessor von Motorola) auch alle Custom-Chips (spezielle von Commodore entwickelte Bausteine mit ganz spezifischen Aufgaben), die z.B. für Grafik- oder Soundausgabe und Speicherverwaltung zuständig sind, zugreifen. Dadurch kommt es hin und wieder zu »Stauungen«, wenn alle Chips gleichzeitig auf diesen Speicher zugreifen möchten. Dies brems natürlich den Hauptprozessor. Das Fastmem dagegen ist nur dem 68000er vorbehalten, wo-



durch er auf dieses ungehindert zugreifen kann, was durchaus Geschwindigkeitsvorteile bringt. Das Programm FastMemFirst veranlaßt, daß alle Daten, die nicht ausdrücklich in das Chipmem gehören, in das Fastmem abgelegt werden. Sollten Sie diesen Befehl löschen, müssen Sie auch die entsprechende Zeile in der »Startup-Sequence« entfernen, da dort dieses Programm schon beim Systemstart aufgerufen wird (SYS:SYSTEM/FASTMEMFIRST).

9.4 FixFonts

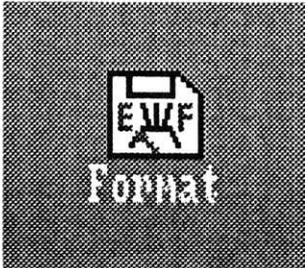
Dieses Programm dient dazu, die Einträge in Ihrem »fonts«-Verzeichnis zu aktualisieren. Das heißt, wenn Sie zu der Schrift »ruby«, die bereits in den Größen 8, 12 und 15 vorhanden ist, noch die Größe 24 dazukopieren, so wird dies erst erkannt, nachdem Sie dieses Programm aufgerufen ha-



ben. Dasselbe gilt auch, wenn Sie die Datei 12 im Unterverzeichnis »ru-by« löschen. Auch hier wird dies erst akzeptiert, wenn Sie das Programm »FixFonts« aufrufen.

9.5 Format

Für diesen Befehl gilt dasselbe wie für »DiskCopy«. Er dient zum Formatieren von Disketten und kann nur vom Menü »Disk« mit dem Menü-



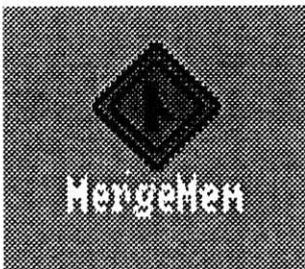
punkt »Initialize« aufgerufen werden.

9.6 InitPrinter

Dieses Programm übergibt Ihren Drucker die neuen Einstellungen, wenn Sie in den Preferences etwas verändert haben.

9.7 MergeMem

Wenn in einem Amiga mehrere Speichererweiterungsplatinen vor-



handen sind, oder Platinen, die den Hauptspeicher um mehr als 2 MByte erweitern, so trägt die Autokonfigurationsroutine beim Systemstart jede

einzelne Speichererweiterung bzw. jeden Speicherblock in die System-speicherliste als separaten Eintrag ein. »MergeMem« versucht nun diese einzelnen Speicherblöcke zu einem großen Block zu verbinden, um es Programmen zu ermöglichen, auch sehr große Speicherblöcke vom Betriebssystem an einem Stück anzufordern.

9.8 NoFastMem

Bei älteren Programmen kann es vorkommen, daß Daten, die für die Grafik- bzw. Soundchips gedacht



sind, in das Fastmem gelangen, wodurch sie für diese Customchips nicht mehr erreichbar sind. Das Ergebnis ist meistens ein Absturz des Computers. Um solche Programme trotzdem verwenden zu können, belegt »NoFastMem« den Fastmem-Bereich, wodurch nur noch Chipmem zur Verfügung steht.

9.9 SetMap

Mit diesem Programm können Sie die Tastaturbelegung ändern, wobei



sich die entsprechende Belegungsdatei im Verzeichnis »devs/key-maps« befinden muß.

10. Das Verzeichnis Utilities

10.1 Calculator

Ein kleiner Taschenrechner, der nur die normalen Grundfunktionen bietet. Wer ihn nicht benötigt, kann ihn ruhig weglassen.

10.2 Clock

Die berühmte Uhr, die sich jeder Größe anpaßt. Ein nettes, aber nicht unbedingt notwendiges Programm.

10.3 ClockPtr

Dieses Programm verwandelt Ihren Mauszeiger in eine Uhr, sofern kein Fenster aktiviert ist. Dieses nette Tool wird auch nicht unbedingt benötigt und kann gelöscht werden.

10.4 CMD

Sollten Sie Daten, die z.B. an den Drucker geschickt werden, in eine Datei umlenken wollen, können Sie dieses Programm verwenden.

10.5 GraphicDump

Mit diesem Programm können Sie den gerade aktuellen Bildschirm auf dem Drucker ausgeben. Nachdem Sie dieses Programm gestartet haben, bleiben Ihnen etwa zehn Sekunden Zeit, jenen Bildschirm in den Vordergrund zu bringen, den Sie ausdrucken möchten.

10.6 InstallPrinter

Da es sinnlos wäre, alle vorhandenen Druckertreiber auf der Workbench zu haben, wurden diese auf die Extras-Diskette ausgelagert. Um nun den von Ihnen benötigten Treiber auf Ihre Workbench zu kopieren, starten Sie einfach dieses Programm und wählen den benötigten Druckertreiber aus einer Liste aus.

10.7 More

Dieses Programm dient zur komfortablen Anzeige von Textdateien. Sie können den Text vorwärts- und rückwärtsblättern und haben viele weitere Möglichkeiten.

10.8 Notepad

Ein kleiner Texteditor, der alle Amiga-Fonts unterstützt. Für kurze Schriftstücke durchaus geeignet.

10.9 PrintFiles

Mit diesem Programm können Sie beliebige Textdateien auf dem Drucker ausgeben. Klicken Sie das Piktogramm der ersten Datei, die Sie drucken möchten, einmal an. Halten Sie ab nun die <Shift>-Taste gedrückt. Klicken Sie alle weiteren zu druckenden Dateien einmal an und danach das Programm »PrintFiles« zweimal. Nach kurzer Zeit beginnt der Ausdruck.

10.10 Say

Wenn Sie Ihren Amiga sprechen hören möchten, können Sie dies mit

diesem Programm erreichen. Probieren Sie es ruhig aus.

11. Das Verzeichnis »Expansion«

Dieses Verzeichnis dient zur Aufnahme von Treiber-Software von Erweiterungskarten, wie z.B. der PC-Brückenkarten für den Amiga 2000.

12. Das Verzeichnis »Empty«

Dieses Verzeichnis ist, wie der Name bereits sagt, leer. Es dient dazu, auf einfachste Weise eine neue Schublade anlegen zu können. Klicken Sie einfach »Empty« einmal an und wählen Sie den Menüpunkt »Duplicate« im »Workbench«-Menü. Schon haben Sie eine leere Schublade, die Sie beliebig verschieben und umbenennen können.

13. Das Verzeichnis »Trashcan«

All jene Dateien, die Sie nicht mehr benötigen, können Sie in diesen Mülleimer werfen. Danach wählen Sie den Menüpunkt »Empty Trash« im Disk-Menü. Schon sind all diese Dateien gelöscht.

Nachdem nun alle Programme der Workbench-Diskette ausführlich besprochen wurden, kommen wir zur zweiten beigelegten Diskette, den Extras 1.3D.

II. EXTRAS 1.3D

Die Extras-Diskette

1. Das Verzeichnis »FD1.3«

Dieses Verzeichnis ist für all jene Basic-Programmierer interessant, die von Amiga-Basic aus die Betriebssystemroutinen der Bibliotheken benutzen möchten. Amiga-Basic benötigt hierzu sog. »bmap«-Dateien, die Sie aus den hier befindlichen »_lib.fd«-Dateien mit dem Basic-Programm »ConvertFD« aus dem »BasicDemos«-Verzeichnis erzeugen können.

2. Das Verzeichnis »devs«

2.1 keymaps

Hier finden Sie weitere Tastaturbelegungstabellen. Sollten Sie diese mit dem »SETMAP«-Befehl nutzen wollen, müssen Sie diese zuerst in das »devs/keymaps«-Verzeichnis Ihrer Startdiskette kopieren.

2.2 printers

In diesem Directory befindet sich eine große Auswahl an Druckertreibern. Kopieren Sie jenen, den Sie benötigen, entweder mit dem Programm »InstallPrinter« oder von der Shell aus in das Verzeichnis »devs/printers« Ihrer Startdiskette.

3. Das Verzeichnis »Tools«

Dieses Directory enthält einige interessante Hilfsprogramme. Genaue Anleitungen zu diesen Hilfsprogrammen zu geben, würde den Rahmen dieses Artikels sprengen. Deshalb wird zu jedem Tool nur eine kurze Bemerkung über das jeweilige Anwendungsgebiet abgegeben.

3.1 Fed

Dieser Zeichensatzeditor dient zum Verändern bzw. Neuerstellen von Schriften (Fonts).

3.2 FreeMap

Dieses Programm zeigt grafisch die Belegung des Chipmem und Fastmem, außerdem gibt es Auskunft über den derzeitigen Speicherverbrauch Ihres Amigas.

3.3 IconEd

Um für Ihre Programme eigene Piktogramme erstellen zu können, benötigen Sie dieses Tool.

3.4 IconMerge

Falls Sie möchten, daß sich das Icon Ihres Programms beim einmaligen Anklicken ändert, können Sie dies mit diesem leicht bedienbaren Tool bewerkstelligen.

3.5 KeyToy2000

Dieses Programm zeigt Ihnen die derzeitige Tastaturbelegung Ihres Amiga grafisch an.

3.6 MEmacs

Dieser Texteditor wird fast professionellen Ansprüchen gerecht. Er läßt sich sowohl als Editor zum Programmieren von eigenen Programmen als auch als Textverarbeitung für kurze Schriftstücke einsetzen.

3.7 Palette

Ohne das Preferences-Programm aufrufen zu müssen, können Sie mit diesem Tool die Farben Ihrer Workbench verändern.

3.8 PerfMon

Wenn Sie schon immer einmal wissen wollten, wie weit der Hauptprozessor ausgelastet ist, ist PerfMon das Richtige für Sie. Dieses Programm zeigt grafisch an, wie weit der 68000er von den Programmen in Anspruch genommen wird.

3.9 Terminal

Ein Terminalprogramm, das in Verbindung mit einem Modem zur Datenfernübertragung verwendet werden kann.

4. Das Verzeichnis »fonts«

In diesem Verzeichnis finden Sie weitere Schriftarten. Wenn Sie diese von Ihrer Workbench aus nutzen wollen, kopieren Sie sie in das Verzeichnis »fonts« Ihrer Workbench-Diskette.

5. Das Verzeichnis »Basic Demos«

In diesem Verzeichnis finden Sie eine ganze Reihe von Beispielprogrammen, die Ihnen die Fähigkeiten der Programmiersprachen Amiga-Basic demonstrieren sollen.

6. Das Programm »Amiga Basic«

Dieses Programm stellt eine vollständige Programmiersprache auf Interpreterbasis dar. Sie besitzt eine Reihe von mächtigen Befehlen, die es ermöglicht, auf einfache Art und Weise erstaunlich professionelle Programme zu schreiben.

7. Das Programm »Linie2«

Ein kurzes, aber äußerst eindrucksvolles Demo der Fähigkeiten des Commodore Amiga stellt dieses Programm dar. Es zeigt sowohl die Farbenpracht als auch die Geschwindigkeit Ihres Computers.

Des Weiteren befindet sich noch ein kurzer Text mit dem Titel »Hinweise« auf der Extras 1.3D-Diskette, der einige Erläuterungen zu dem Umgang mit dieser Diskette enthält.

Somit wären wir am Ende unserer Entdeckungsreise angelangt. Es gab viel Neues zu erforschen, viel Interessantes zu entdecken. Unter Umständen haben Sie auch neue Erkenntnisse erworben, die Ihnen die Arbeit mit Ihrem Amiga in Zukunft noch einfacher machen werden. Eines ist jedoch ganz sicher. Es gibt nicht viele Computer, bei denen eine Entdeckungsreise derartig viele Möglichkeiten eröffnet. Denn er ist und bleibt ein flexibler, leistungsfähiger Computer, dessen Fähigkeiten den Anwender immer wieder ins Staunen versetzen können. In diesem Sinne: Viel Spaß und Glück bei weiteren Entdeckungsreisen durch Ihren Computer.

pe

von Arnd Goebel

Der Amiga ist in verschiedenen Modellen zu haben. Jeder kann sich sein System nach seinen Ansprüchen aussuchen: Da gibt's den Amiga 500, A1000, A2000, A2500 oder einen A3000, das jüngste Kind der Reihe. Da sie untereinander nahezu kompatibel sind, wird im folgenden schlicht vom »Amiga« die Rede sein.

Hervorragend geeignet ist der Amiga für Grafik, Sound und Animation. Seine Leistungen überbieten weit den Standard, der normalerweise in dieser Preisklasse besteht. Der Amiga verfügt über Multitaskingfähigkeiten, d.h. er kann mehrere Prozesse gleichzeitig bearbeiten.

1. Grundlagen

Ein Computer ist grundsätzlich eine recht teure Angelegenheit. Die Anschaffungskosten bewegen sich für einen Mittelklassecomputer z. Zt. (1990) zwischen 2500 und 10000 Mark. Für das kleinste Gerät des Amiga, den Amiga 500 ohne Zubehör, bezahlt man heute in etwa 800 bis 900 Mark. Man sollte sich vor dem Kauf eines Computers also darüber im klaren sein, was man genau mit ihm anfangen will. Der Verwendungszweck entscheidet über die Kosten der Anschaffung und über die Notwendigkeit von Zusatzanschaffungen. Was bringt einem Studenten ein Computer mit einer enormen Arbeitsgeschwindigkeit und einem ebenso enormen Preis, wenn er nur Textverarbeitung betreiben will?

Auch beim Amiga sollte man sich vor dem Kauf überlegen, ob man damit professionell arbeiten will, oder nur hin und wieder einen Brief schreibt, spielt oder ein Liedchen komponiert. Jedes Modell des Amiga bietet spezifische Möglichkeiten. Vor allem sollte man wissen, daß Erweiterungen für den Amiga 500 teurer sind als für den Amiga 2000. Das wird dann klar, wenn man sich vor Augen hält, daß in den Amiga 2000 Erweiterungen eingebaut werden. Im Amiga 500 dagegen ist kein Platz für z.B. ein Genlock-Interface oder eine Festplatte. Hier muß immer noch

Amiga Ein Bummel durch Ihren Computer

Aus welchen Teilen besteht der Computer? Wie arbeiten diese zusammen? Welche Peripherie ist nötig, um individuell sinnvoll arbeiten zu können? Wenn Sie die folgenden Seiten gelesen haben, dürften Sie kaum mehr Fragen haben.

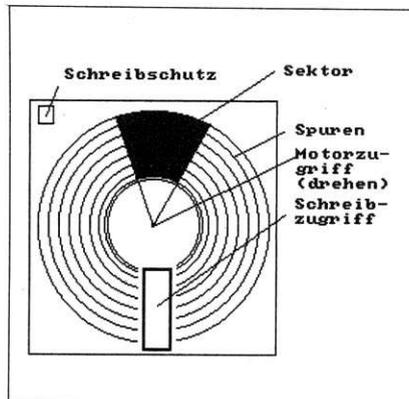


Bild 1. Der schematische Aufbau einer Amiga-Diskette

ein Zusatzgehäuse mitgeliefert werden. Wer professionell arbeiten will, entscheidet sich lieber gleich für einen Amiga 2000. Denn der kann praktisch ohne Begrenzung erweitert werden.

1.1 Der Computer

Als Computer bezeichnet man den meist nüchtern designten Kasten, in dem alle Bauteile, ICs, Quarze etc. untergebracht sind. Auf das Innenleben kommen wir später noch zurück. Der Computer ist das Zentrum der Anlage und verwaltet die Daten, den Text, die Grafik oder einfach nur ein Programm. Über ihn läuft die Kommunikation. Alle Geräte sind an ihn angeschlossen.

1.2 Die Peripherie

Der Computer ist das Gehirn der angeschlossenen Geräte, die im Gesamtbegriff Peripherie zusammengefaßt werden. Dazu gehört die Grundausstattung exklusiv Computer.

1.2.1 Das Disketten-Laufwerk

Das Disketten-Laufwerk gewährleistet einen relativ schnellen Zugriff auf die Daten. Die im Computer gespeicherten Daten müssen nämlich abgespeichert werden, da sonst mit dem Ausschalten alle Daten gelöscht werden. Der Computer gibt die Daten beim Speichern an das Laufwerk weiter und dieses speichert sie auf einer Magnetplatte. Das Prinzip ist ähnlich dem Aufzeichnen von Musik auf einer Kassette. Hier erfolgt das Abspeichern statt auf einem Band, auf einer runden Scheibe, die in Sektoren aufgeteilt ist (Bild 1). Der Vorteil liegt darin, daß man sofort auf eine beliebige Stelle zugreifen kann, da eine runde Platte konzentrische Kreise besitzt und diese symmetrisch zum Mittelpunkt angeordnet sind. So ist kein langes Spulen notwendig. Auf der Diskette ist ebenfalls ein sog. Nulltrack (Nummer 0, unterschiedlich bei den Systemen) enthalten, auf dem genau steht, wo das jeweilige Programm der Diskette zu finden und der Diskettenkopf zu positionieren ist. Vor jedem Laden wird dieser Index- oder Directorytrack (Directory = Inhaltsverzeichnis) abgefragt. Bei den Laufwerken hingegen fragt der Computer nach dem Ein-

schalten über den sog. Bootblock (to boot = starten) nach Informationen zum Starten von Arbeitsprogrammen, die ein Arbeiten am Computer erst zulassen. Dazu aber später.

Disketten haben verschiedene Formate und Eigenschaften. Es gibt einseitige, doppelseitige, große (5¹/₄ Zoll) und kleine (3¹/₂ Zoll) Disketten (Zoll = 2,54 cm). Die Handhabung ist nahezu bei allen Disketten gleich.

1.2.2 Der Drucker

Der Drucker bringt die Daten oder den gespeicherten Text aufs Papier. Hier kann nur eine Ausgabe erfolgen, während ein Disketten-Laufwerk auch Daten vom Computer empfangen kann.

Drucker unterscheiden sich in der Qualität und Geschwindigkeit des Ausdrucks und natürlich im Preis.

1.2.3 Der Monitor

Der Monitor ist nichts weiter, als ein Fernseher, der nicht Sender empfängt, sondern die Daten, die ihm der Computer übermittelt, auf dem Bildschirm darstellt. Ein Monitor ist allerdings für die Augen um ein Vielfaches angenehmer, da er eine höhere Bildwiederholfrequenz hat, d.h. das Bild wird pro Sekunde häufiger wiederholt und erscheint ruhiger. Somit ist es möglich, auch näher am Monitor zu sitzen, als das bei einem Fernseher möglich wäre. Ein Fernseher kann meist auch am Computer angeschlossen werden. Es ist aber nicht empfehlenswert, da das Bild unruhig verschwommen oder beides ist und häufig Kopfschmerzen verursacht.

Die normalen Amiga-Monitore haben eine Bildwiederholfrequenz von 50 Hz. Das entspricht dem Bild eines Fernsehers, da aber die Auflösung höher ist, erscheint das Bild ruhiger. Für die Augen ist es besser, wenn man Monitore verwendet, deren Bildwiederholfrequenz höher als 50 Hz liegt. Ab ca. 63 Bildwiederholungen pro Sekunde, also 63 Hz flimmert das Bild (fürs Auge) nicht mehr. Ab 70 Hz erscheint Schrift auf dem Monitor wie Schrift auf Papier. Je nachdem, welche Auflösung Sie

beim Amiga gewählt haben, wird das Flimmern bei den 50-Hz-Monitoren sehr deutlich, besonders im Interlace-Modus, bei dem die Bildwiederholfrequenz halbiert wird. Bei geringer Auflösung läßt es sich aber sehr gut arbeiten. Außerdem sind Hardware-Erweiterungen erhältlich, die das Flimmern unterdrücken. Informationen darüber entnehmen Sie am besten den jeweils aktuellen AMIGA-Magazin-Ausgaben.

Der Computer steht in ständigem Austausch mit der Peripherie, also mit den eben beschriebenen Geräten. Ein Arbeiten am Computer ohne Peripherie wäre sinnlos.

Aber warum brauchen wir das Wunderding Computer? Wozu taugt dieses Gerät überhaupt?

Computer haben sich in den vergangenen zehn Jahren unaufhaltsam etabliert und sind heute nicht mehr wegzudenken. Sie machen sich im Haushalt, im Büro, im Krankenhaus etc breit.

Das einfachste Beispiel ist eine Digitaluhr oder eine Waschmaschine. Woher sollte eine Waschmaschine wissen, was sie tun soll? Wie lange sie Wasser einlaufen lassen soll, zu welchem Zeitpunkt das Waschmittel eingespült wird oder wann sie wieder abzuschalten hat. Sie ist programmiert. Und es gibt nicht nur ein Waschprogramm, sondern mehrere. Folglich sitzt irgend etwas in ihr, das sie steuert und erkennt, welches Programm auszuführen ist. Es ist ein Computer. So finden wir ihn fast überall im Alltag wieder.

2. Aufbau des Computers

Für viele ist der Computer ein Buch mit sieben Siegeln: Der eine oder andere hat Begriffe wie RAM oder ROM vielleicht schon gehört.

Ein Computer hat eine zentrale Einheit, die CPU (Central Processing Unit). Sie heißt beim Amiga MC68000 bzw. beim Modell Amiga 3000 MC68030. Diese Einheit kontrolliert alle Vorgänge. Ohne sie läuft im Computer nichts.

Weiter besteht ein Computer aus einem Schreib-/Lese-Speicher RAM (Random Access Memory) genannt, einem Nur-Lese-Speicher, auch als ROM (Read Only Memory) bezeichnet und einer Ein- und Ausgabeeinheit (I/O-Einheit = Input/Output).

2.1 Der Mikroprozessor (CPU)

Der Mikroprozessor ist das Herzstück in einem Computer. Es gibt viele Arten und Typen davon. Die neuen Computer haben eine CPU Motorola MC68000, Weiterentwicklungen wie 68020, 68030, eine CPU Intel 80286/80386/80486 oder andere.

Prozessoren sind unterschiedlich schnell, d.h., sie können mehr Zyklen pro Zeiteinheit abarbeiten und jeweils mehr Speicher verwalten. Der Mikroprozessor enthält die ganze »Intelligenz« des Computers. In ihm gehen die algebraischen Funktionen wie Addieren, Subtrahieren etc. vor sich. Er kann addieren, subtrahieren, Zahlen vergleichen und die anderen Bauteile des Computers steuern. Er kann den Speicher auslesen und auch Daten absenden, z.B. zum Laufwerk.

Wir brauchen, um die CPU näher zu erläutern, noch die bereits erwähnten Begriffe, RAM und ROM.

Dem Amiga greifen bei seiner Arbeit Custom-Chips unter die Arme, um so die Rechengeschwindigkeit zu erhöhen. Diese Chips sind für besondere Bereiche im Amiga zuständig, auf die man sich bei diesem Computer spezialisierte. Um nur einige Namen der Custom-Chips zu nennen: Denise, Paula, Fat Agnus etc.

Weiterhin gibt es sog. Coprozessoren wie den MC68881 oder MC68882, die aber nur beim Amiga 3000 integriert sind, bei den anderen Modellen jedoch nachgerüstet werden können. Das sind Arithmetikprozessoren, die die Rechenarbeit der CPU übernehmen und so erheblich zur Entlastung der CPU beitragen. Besonders bei rechenintensiven Operationen wird dies deutlich.

2.2 RAM und ROM

Die Benutzung eines Computers wäre sinnlos, wenn er sich nichts merken könnte. Bei einer Addition muß der Computer sich z.B. die erste Zahl merken, um dann die Zweite dazuaddieren. Diesen Speicher nennen wir RAM. Das ist der Schreib-/Lese-Speicher. Hier kann der Mikroprozessor Daten ablegen, um sie später wieder abzurufen. Er muß die Daten weiterleiten, um weiterarbei-

ten zu können. Bildlich kann man sich das folgendermaßen vorstellen. Man hat irgendwelche Akten und legt sie in einer Schublade in einem großen Aktenschrank ab. Bei Bedarf zieht man die Schublade auf und holt sich das Aktenbündel wieder heraus.

Beim Amiga unterscheiden wir zwischen zwei Arten von RAM, das Chip- und das Fast-RAM. Das Chip-RAM steht beiden zur Verfügung, sowohl der CPU, als auch den Custom-Chips. Das Fast-RAM ist nur für die CPU reserviert, und ist somit schneller, da die CPU beim Zugriff nicht von anderen Chips gestört wird.

Das ROM ist der Nur-Lese-Speicher und enthält wichtige Daten, die zum Arbeiten mit dem Computer benötigt werden. Hier werden keine Daten gespeichert, weil sie bereits vorhanden sind und zwar konstant. Sie sind auch nach dem Ausschalten des Computers präsent. Hier kann die CPU nur lesen. Das ist der entscheidende Unterschied zum RAM.

Das ROM ist vergleichbar mit einem Standardlexikon. Hier stehen Grundinformationen, deren Änderung nicht sinnvoll wäre, da diese Daten immer wieder gebraucht werden und zwar unverändert.

2.3 CPU, RAM und ROM

Der Mikroprozessor arbeitet sehr schnell Befehle ab, die wir ihm über das Betriebssystem geben. Wenn ein Computer eingeschaltet ist, arbeitet er immer, auch wenn man auf dem Monitor nichts oder nur ein blinkendes Rechteck sieht. Er überprüft ständig, ob vielleicht eine Diskette eingelegt oder die Maus bewegt wurde. Die Geschwindigkeit eines Computers wird in MIPS (Millionen Instruktionen pro Sekunde) gemessen. Das sind die Befehlsabarbeitungen pro Sekunde.

Der Amiga würde nahezu zum Stehen kommen, wären ihm nicht der Blitter, der Copper und die ande-

ren Custom-Chips an die Seite gestellt. Grafische Benutzeroberflächen wie die Workbench haben erstens eine große Datenflut zu bewältigen und zweites wird sehr viel Rechenzeit benötigt. Stellen Sie sich vor, Sie verschieben ein Fenster (Window) auf der Workbench. Es muß der ganze Inhalt, wie Icons und Directories relativ dazu verschoben und wieder plaziert werden.

Dies würde die CPU allein zu sehr belasten und die Arbeitsgeschwindigkeit stark senken. Diese Arbeit übernehmen andere Chips.

Im ROM befindet sich das gesamte Betriebssystem. Es stellt die Benutzerschnittstelle dar. Hier bekommt der Computer die für ihn verständlichen übersetzten Informationen, um den Benutzer verstehen zu können. Durch das Betriebssystem ist es möglich, mit dem Computer zu kommunizieren. Es hat quasi eine Übersetzerfunktion.

RAM ist der variable Speicher. Hier werden sog. Durchgangsdaten gespeichert, die nach Belieben auch wieder gelöscht werden können. Wird der Computer ausgeschaltet, sind auch die gesamten Daten im RAM gelöscht.

RAM und ROM stehen im ständigen Austausch mit der CPU.

Sehr vorteilhaft beim Amiga ist, daß die Custom-Chips direkten Zugriff auf den Speicher haben. Man nennt das auch Direct Memory Access (DMA). So muß nicht immer ein Umweg über einen Puffer und die CPU genommen werden um Daten auszutauschen.

2.4 Die Rohrpost

Zwischen den Bausteinen liegen Verbindungen. Man hat sie sich als Röhren vorzustellen, durch die Informationen geschickt werden. Diese Verbindungen werden einzeln angesprochen. Der Adreßbus kann jede einzelne Speicherzelle aufrufen. Er gibt der CPU die Stellen (Adressen) an, wo die Daten abgelegt werden müssen. Der Speicher des Computers ist durchnummeriert. So kann jede Stelle exakt abgefragt werden. Durch die Schaltzustände (Strom, kein Strom) werden die Adressen dargestellt. Außer dem Adreßbus gibt es noch den Datenbus.

Die CPU kennt das Ziel der Daten, aber noch nicht die Daten selbst. Hierfür ist der Datenbus zuständig. Bei einem 8-Bit-Computer (Bit = kleinste Einheit) besteht er aus acht Leitungen. Bei einem 16-Bit-Computer aus 16 Leitungen. Die Speicherzelle gibt ihren Inhalt auf z.B. 16 Datenleitungen, so daß die CPU die Werte lesen kann.

2.5 Die Arbeitsweise

Ein Computer kennt von sich aus eigentlich nur zwei Zahlen. Das heißt der Computer erkennt, ob durch eine Leitung Strom fließt und ordnet dieser Leitung dann den Wert »1« zu. Fließt dagegen kein Strom, wird dieser Leitung der Wert »0« zugeordnet.

Der Entwickler des ersten Computers, Konrad Zuse, erkannte, daß für die Entwicklung des Computers nur ein bestimmtes Zahlensystem in Frage kommt. Das Binärsystem (binär = aus zwei Einheiten oder Teilen).

Das Binärsystem (Zweiersystem) besteht statt wie unser Dezimalsystem (Zehnersystem) aus zehn (0,1, 2,3,...,9), nur aus zwei Ziffern (0,1). Mit dem Binärsystem arbeiten alle Computer. »1« heißt Strom fließt, »0« heißt kein Strom fließt.

Um mit dem Computer zu arbeiten, gibt es noch das Hexadezimalsystem (hexadezimal = 6+10 =16), das Sechzehnersystem.

2.6 Das Binär- und Hexadezimalsystem

Unser Zehnersystem besteht nur aus den Ziffern 0 bis 9. Damit können wir alle Zahlen, auch Bruchzahlen darstellen. Bei der Addition von 9 und 1 können wir das Ergebnis nicht mehr mit einer Ziffer allein darstellen. Deswegen müssen wir eine Zehnerpotenz weiter nehmen.

Potenz	10^6	10^5	10^4	10^3	10^2	10^1	10^0
Zahl							
9	0	0	0	0	0	0	9
10	0	0	0	0	0	1	0
11	0	0	0	0	0	1	1
459	0	0	0	0	4	5	9
100000	0	1	0	0	0	0	0
476746	0	4	7	6	7	4	6
1553000	1	5	5	3	0	0	0

So schreiben wir Zahlen in unserem Zehnersystem. Gehen wir zurück in die Mathematik, dann wissen wir, jede Zahl mit der Hochzahl 0 ist eins (1). Das ist eine Definition. Jede Zahl mit der Hochzahl 1 ist die Zahl der Basis selbst.

Da es beim Computer aber nur zwei Zustände gibt, ist nur ein Zahlensystem mit zwei Ziffern sinnvoll. Also wird statt mit Zehner- mit Zweierpotenzen gearbeitet.

Potenz 2^6 2^5 2^4 2^3 2^2 2^1 2^0
Zahl

2 0 0 0 0 0 1 0
10 0 0 0 1 0 1 0

Das Hexadezimalsystem arbeitet genauso, nur mit Zahlen zu Basis 16.

2.7 Bits und Bytes

Bit ist die Abkürzung von den amerikanischen Ausdruck **B**inary **d**igit.

Jede der schon erwähnten Datenleitungen im Computer stellt ein Bit dar. Ein Bit ist die kleinste Einheit und kann im Binärsystem den Wert »1« oder »0« annehmen (Strom/kein Strom). Jeweils 8 Bit ergeben 1 Byte. Aus 8 Bit ergeben sich $2_8 = 256$ Schaltzustände. Es gibt Binär also die Zahlen %00000000-%11111111. Dezimal sind das die Zahlen 0 bis 255. Hexadezimal geht die Zahlenskala von \$00 bis \$FF.

Um Binärzahlen zu kennzeichnen schreibt man ein %-Zeichen davor.

Hexadezimalzahlen (kurz Hex-Zahlen) erkennt man an einem vorgestellten \$-Zeichen.

Dezimalzahlen werden nicht gesondert gekennzeichnet.

Doch warum ist Binär %11111111=\$FF=255? Hier kommt die Auflö-

$$\begin{aligned} \%11111111 &= 1 \times 2^0 = 1 \\ &1 \times 2^1 = 2 \\ &1 \times 2^2 = 4 \\ &1 \times 2^3 = 8 \\ &1 \times 2^4 = 16 \\ &1 \times 2^5 = 32 \\ &1 \times 2^6 = 64 \\ &1 \times 2^7 = 128 \end{aligned}$$

= 255

$$\begin{aligned} \$FF &= 15 \times 16^0 = 15 \\ &15 \times 16^1 = 240 \end{aligned}$$

= 255

Das heißt 255 Zustände + dem Nullzustand %00000000 (= \$00=0) ergibt 256 Zustände.

Der Amiga ist ein 16-Bit-Rechner. Das heißt er stellt 16 Datenleitungen zur Verfügung. Wir bekommen also 65536 Zustände. Hexadezimal sind das \$FFFF = 216 = %11111111 11111111. Der Amiga kann aber auch mit 32-Bit-Werten rechnen.

Auf die Feinheiten der Rechnung gehen wir an dieser Stelle nicht ein, um die Sache nicht zu komplizieren.

Sie kennen jetzt den groben Aufbau des Amigas bzw. eines beliebigen Computers. Egal, welches Modell Sie besitzen, der Aufbau ist bei allen nahezu derselbe. Wenn Sie das Prinzip der einfachen Schaltungen, der Zustände 0 und 1 verstanden haben, dann haben Sie die notwendigen Computergrundkenntnisse erworben. Alles andere sind komplizierte Schaltungen, Transistoren, Widerstände und Leitungen.

2.8 Der Amiga speziell

Durch die Hilfe des Blitters und Coppers, die die zeitintensivsten Arbeiten des Amiga übernehmen, ist die enorme Geschwindigkeit dieses Computers möglich.

Der Amiga zeichnet sich durch seine offene Systemarchitektur (OSA) aus. Wenn Sie ihn erweitern, haben die CPU und alle Custom-Chips direkten Zugriff auf diese Erweiterung. Nehmen wir an, sie möchten eine neue CPU in das System integrieren. Die Custom-Chips müssen nicht erst über die alte CPU operieren, sondern haben direkten Zugriff zum Erweiterungsslot, an dem Anschlüsse möglich sind. Auch hat der Anwender die Möglichkeit, sich das Betriebssystem selbst so zu gestalten, daß es seinen Vorstellungen entspricht. Dies geschieht nicht zuletzt durch die Programmierung eigener Befehle in der Sprache C.

Die Kombination von Spezialchips und OSA öffnet dem Amiga die Türen für die Zukunft. Er kann aufgerüstet werden und bleibt auf dem neuesten Stand, während andere Computer schon längst in Vergessenheit geraten sein werden.

Der Amiga bietet dem Anwender drei verschiedene Operationsfelder. Das Kernel, das Amiga-DOS und die Workbench.

Direkt mit der Hardware ist das Kernel verbunden. Durch das Kernel wird eine Verbindung zwischen Anwender und Prozessor bzw. Maschine hergestellt. Für uns ist das Kernel nicht von größerer Bedeutung, weil wir als Einsteiger eigentlich nur mit dem CLI und der Workbench Berührungspunkte mit dem Amiga haben.

Es reicht, wenn wir wissen, daß das Kernel die innerste Ebene des Betriebssystems darstellt und die Steuerung nach außen übernimmt. Hier werden Speicherorganisation, Bibliotheken, Nachrichtenübermittlung usw. behandelt.

Für uns ist das CLI und die Workbench von größerer Bedeutung. Beide werden vom Kernel unterstützt. Mit dem CLI (Command Line Interpreter) ist es möglich, wie bei herkömmlichen Betriebssystemen Befehle einzugeben.

Dieses CLI macht es möglich, die Befehle des Amiga-DOS auszuführen. DOS heißt Disk Operating System, d. h. ein System, um mit den Disketten-Laufwerken zu operieren. Dazu gehören auch Hard-Disk-Laufwerke und anderes.

Das DOS besteht aus Befehlen, die sie eingeben können und die dann ausgeführt werden.

Faßt man derartige Befehle zusammen, entsteht eine Batch-Datei (Batch = Stapel). Solch eine Reihe von Befehlen in einer Batch-Datei stellt die Startup-Sequence dar. Sie finden Sie im S-Verzeichnis ihrer Workbench-Disketten.

Nehmen wir an, Sie wollen ein Programm von einer Diskette laden. Das Betriebssystem steht im ROM und ist bereits geladen. Nur wollen Sie nicht immer dasselbe Programm von der einen Diskette laden, sondern auch auf eine ganz andere Disketten zurückgreifen.

Also ist es notwendig, dem Computer jedesmal spezifische Informationen für die unterschiedlichsten Starts mitzuteilen, also welches Programm bei einem Neustart geladen werden soll. Vor dem Start müssen auch noch einige Voraussetzungen geschaffen werden. Zum Beispiel soll die interne Uhr abgefragt werden oder eine resetfeste RAM-Disk angelegt werden. Diese Aufgabe übernimmt die Startup-Sequenz. Hier können Sie alle Befehle eintragen, die jedesmal ausgeführt werden sollen, wenn sie von dieser Diskette starten. Es geschieht dann alles von allein, und sie müssen nicht jede Zeile jedesmal einzeln eingeben. Zum Beispiel brauchen Sie sich nicht mehr zu ärgern, wenn die Tastaturbelegung wieder einmal keine Sonderzeichen bereitstellt, da Sie in der Startup-Sequenz den Befehl SET-MAP D ausgeführt haben.

Die Befehle des DOS müssen in einem bestimmten Verzeichnis namens »C« einer Diskette stehen. Diese Befehle können auch jederzeit durch modifizierte ersetzt werden.

Die Workbench stellt eine weitere Möglichkeit dar, mit dem Amiga zu arbeiten. Sie müssen hier keinerlei Vorkenntnisse haben, da Sie einfach mit der Maus auf das Disketten-Icon klicken und das Programm durch nochmaliges Klicken auf das Icon in der geöffneten Schublade starten.

Die Workbench ist allerdings auch die langsamste Art und bietet nicht die große Befehlsvielfalt z. B. des Amiga-DOS.

Aber sie ist für den User komfortabel zu handhaben.

Man muß sich nicht stundenlang mit dem Betriebssystem befassen, sondern kann einfach anfangen, Farben zu verändern, den richtigen Druckertreiber anzuwählen, die Auflösung zu bestimmen und vieles mehr. Die Leichtigkeit und das Wegfallen lästiger Tipparbeit erspart viel Zeit und Geduld.

Erst seit ein paar Jahren gibt es grafische Benutzeroberflächen. Und sie haben sich durchgesetzt. Wer den Amiga kennt, arbeitet sicher bald mehr mit der Shell. Aber beson-

Lexikon	
Begriff	Erklärung
Arbeitsspeicher	siehe RAM
Assembler	Maschinensprache, die der Prozessor direkt versteht
Basic	Beginner's All Purpose Symbolic Instruction Code früher einfache, heute durch viele Dialekte stark entwickelte, recht anspruchsvolle, weit verbreitete Anfängersprache
Betriebssystem	Systemprogramm, anhand dessen ein Computer bedient werden kann
Binärsystem	Zweiersystem (0,1). System zur Basis 2
Bit	kleinste im Computer darstellbare Einheit
Bus	Leitungssystem zwischen den Baugruppen im Computer
Byte	1 Byte = 8 Bit / Darstellung erfolgt hexadezimal
CPU	Central Processing Unit, Zentraleinheit
EDV	Elektronische Datenverarbeitung Erfassung von Daten in elektronischen Rechenmaschinen mit Massenspeicher
Hardware	alle Computerbauteile
Interpreter	Programm, das die Befehle einer Programmiersprache in Assembler übersetzt (assembliert)
Maschinensprache	siehe Assembler
Peripherie	alle am Computer angeschlossenen Geräte, die vom Computer angesteuert werden
Programm	erstellte Routine, die der Computer schrittweise abarbeitet
programmieren	Lösen von Problemen der Mathematik
Programmiersprache	Sprachen mit unterschiedlichen Schlüsselwörtern, die ein Interpreter in Assembler umsetzt. Ziel: Vereinfachung der Programmierung
RAM	Random Access Memory Arbeitsspeicher des Computers. Der Arbeitsspeicher kann eingelesen und überschrieben werden.
Rechenmaschine	Computer
ROM	Read Only Memory. Das Betriebssystem oder andere feste Programme, die nur eingelesen werden können
Software	Programme für einen Computer
System	Computer + Peripherie (Hardware) + Software
Verarbeitung	Erfassung, Absicherung, Abrufbereitschaft

ders bei Mal-, DTP- oder CAD-Programmen wird der Vorteil der Mausbedienung deutlich. Es müssen keine komplizierten Tastaturkombinationen vorgenommen werden, bei denen man sich die Finger verrenkt – ein einfacher Mausklick genügt, und der Befehl wird ausgeführt.

CLI und Workbench sind voneinander abhängig. Sie können die Workbench durch das CLI aktivieren und auch eine Shell durch die Workbench öffnen. Die Shell ist nichts anderes als eine Verbesserung des CLI in wichtigen Punkten. Shell und

Workbench sind die beste Kombination und bieten beide für den spezifischen Fall die idealen Möglichkeiten. Nähere Informationen zu beiden Benutzeroberflächen finden Sie in den Workshops auf Seite 6 bzw. Seite 34.

ag

Ein ausführliches Lexikon über alle Begriffe, die in der Computerwelt und speziell beim Amiga auftreten, finden Sie in unserem AMIGA-Sonderheft 8 »Starthilfe«.

Hoch- und andere Stapeldateien

Was Sie kennen Batch-Processing nicht? Batch-Processing ist die bequemste und leichteste Art, mit dem Amiga komplexe Befehlsfolgen abzuarbeiten. Wozu man das braucht und wie das geht, zeigt dieser Artikel.

von Thomas Hauser

Der Ausdruck »Batch Processing« stammt ebenso wie sein Ableiter »Stapelverarbeitung« aus den Anfangszeiten der automatischen Datenverarbeitung. Als noch ganze Stapel von Lochkarten Stück für Stück abgearbeitet wurden, eine Lochkarte nach der anderen. Verlangte das Ergebnis einer Operation das Überspringen einiger Befehle, wurden die entsprechenden Lochkarten eben vom Stapel abgezogen, ohne die zugehörigen Operationen durchzuführen. Die moderne Analogie zu diesen Kartenstapeln sind Stapeldateien, in denen Kommandosequenzen sozusagen »im Klartext« eingetragen sind. Jede Zeile einer solchen Datei entspricht im übertragenen Sinn einer Lochkarte, und wie bei den Lochkartenstapeln werden bei Batch-Dateien die Zeilen Stück für Stück abgearbeitet, von der ersten bis zur letzten Zeile (es sei denn, es werden Sprünge verlangt). Am Beispiel der Dateien im »S«-Directory der Workbench-Diskette wollen wir Ihnen das Wesen und die Arbeitsweise der Amiga-Stapelverarbeitung etwas näherbringen. Zusätzlich sollen ausführliche Kommentare

die Vorgänge bei der Abarbeitung dieser Dateien verdeutlichen.

Zunächst einmal - warum gerade die Files im »S«-Directory? Die Entwickler des Amiga-Betriebssystems legten fest, daß Stapeldateien (sie werden im englischsprachigen Raum »Script«-Dateien genannt) bevorzugt in diesem »Script-Directory« abzulegen sind. Die Eintragung von Script-Files ist zwar grundsätzlich überall zulässig,

Die Script-Files im »S«-Directory der Workbench-Diskette

Startup-Sequence StartupII Startup-Sequence.HD CLI-Startup Shell-Startup SPAT DPAT PCD

doch sprechen zumindest zwei Fakten dafür, sich an diese Vorgabe der Systementwickler zu halten. Faktum 1 betrifft Betriebssystemvorgaben. Die Datei »Startup-Sequence« wird vom Betriebssystem **nur** im »S«-Directory gesucht, gleiches gilt für »CLI« und »SHELL-STARTUP«. Eine »Startup Sequence« außerhalb des »S«-Directory wird beim Systemstart nicht erkannt und demzufolge auch nicht abgearbeitet. Ein »NewCLI« ohne explizite Angabe eines Steuerfiles sucht immer im »S«-Directory nach »CLI-Startup«. Faktum 2 hängt mit dem Aufruf einer Batch-Datei zusammen, also mit der Aufforderung, den Inhalt abzuarbeiten. Mit dem CLI-Kommando »Execute« starten Sie

die Durchführung der in einer Stapeldatei eingetragenen Befehle. Mit

```
execute cli-startup
```

könnten Sie beispielsweise jene Kommandofolge durchführen lassen, die üblicherweise beim Aufruf eines neuen CLI-Task zur Anwendung kommt. Jede Batch-Datei wird mittels »Execute« gestartet (zum »Script«-Flag und seiner Bedeutung kommen wir später) und das Kommando »Execute« sucht die auf das Kommandowort folgende Datei vorrangig im »S«-Directory. Das hat zur Folge, daß wie im obigen Beispiel die Pfadangabe, die zum gewünschten Script-File weist, entfallen darf, sofern sich das Script-File im Verzeichnis »S:« befindet. Ergänzend sei noch bemerkt, daß Sie Script-Dateien mit jedem beliebigen Texteditor schreiben und verändern können, vorausgesetzt, es werden keine Steuerzeichen mitgespeichert, sondern nur der reine ASCII-Text. »Ed« und »Edit« sind für solche Zwecke ebenso geeignet wie »MEMACS« von der »Extras«-Diskette, »Notepad« hingegen nicht!

Sehen wir uns zunächst den Inhalt des Files »Startup Sequence« an (die Zeilenummerierung wurde von uns zugefügt, um im Text leichter auf bestimmte Zeilen und Kommandos verweisen zu können):

```
10      c:SetPatch >NIL:
20      Addbuffers df0: 10
30      cd c:
40      echo "A500/A2000
      Workbench disk. Re-
      lease 1.3.2 Version
      34.28*N"
50      Sys:System/FastMem
      First
60      BindDrivers
70      SetClock load
80      FF >NIL: -0
90      resident CLI L:Shell-
      Seg SYSTEM pure add
100     resident c:Execute pu-
      re
110     mount newcon:
120     failat 11
130     run execute s:Startup
      II
140     wait >NIL: 5 mins
150     SYS:System/SetMap d
```

```
160 path ram: c: sys:util-
ities sys:system s:
sys:prefs add
170 LoadWB delay
180 endcli >NIL:
```

In Zeile 10 werden zunächst einige fehlerhafte Routinen korrigiert, die in den Kickstart-ROMs der Betriebssystemversionen 1.2 und 1.3 verewigt wurden. Der Zusatz `>NIL:` schickt die von »SetPatch« ausgegebenen Meldungen an das fiktive Device »NIL:«. Ohne diesen Zusatz würden Sie auf dem Bildschirm folgendé Meldungen sehen:

```
Alert patched Layers patched
Exec 020/881 exceptions patch-
ed AllocEntry patched
```

In Zeile 20 wird dem Laufwerkspuffer des internen Laufwerks ein zusätzlicher Speicherblock von 5 KByte zugewiesen, um den Datendurchsatz zu erhöhen. Zeile 30 wechselt das aktuelle Directory nach »C«, Zeile 40 gibt eine kurze Information aus.

»Startup-Sequence«

Der Befehl in Zeile 50 muß mit etwas Skepsis betrachtet werden. »FastMemFirst« weist das Betriebssystem an, geladene Daten soweit zulässig im »Fast Memory« abzulegen. Wie Sie sicher wissen, unterteilt man den Speicher des Amiga in »Chip Memory« und »Fast Memory«. Auf das Chip-Memory greift nicht nur der 68000-Prozessor zu, sondern auch die Custom-Chips. Während ein Custom-Chip auf das Chip-Memory zugreift, kann der 68000 verständlicherweise nicht mit diesem Bereich hantieren – er muß warten. Anders im Fast-RAM – darauf hat einzig und allein der Hauptprozessor Zugriff, wird also nicht von den Custom-Chips gebremst. Die Anweisung, das Fast-Memory bevorzugt mit Daten zu belegen und ChipMem nur, wenn es sich nicht vermeiden läßt, ist also durchaus sinnvoll und die für die Abarbeitung des Befehls »FastMemFirst« benötigte Zeit scheint sinnvoll genutzt. Die Betonung liegt aller-

dings auf »scheint«, denn die wenigsten unter Ihnen werden »echtes« FastMem ihr eigen nennen. Dieses muß nämlich im Adreßbereich \$200000 bis \$9FFFFFF liegen, während die meisten Speichererweiterungen (besonders jene, die im Amiga 500 in den internen Erweiterungs-Slot gesteckt werden) bei Adresse \$C00000 ansetzen. Dort liegt zwar kein »Chip-Memory« mehr, trotzdem wird durch die Daten- und Adreßbus-Struktur des Amiga der Hauptprozessor bei Zugriffen auf diesen Bereich von Paula & Co. gebremst. Sollte Ihr System nicht über »echtes« Fast Memory verfügen, ist es also Schade um die Zeit und Sie sollten die Zeile 50 löschen oder zumindest durch ein Semikolon »totlegen«:

```
50 Sys:System/FastMemFirst
```

Ähnliches gilt für Zeile 60. Wenn das »Expansion«-Directory Ihrer Boot-Diskette leer ist, ist es reine Zeitvergeudung, »Binddrivers« aufzurufen, denn nur Treiber, die im genannten Directory abgelegt sind, werden von »Binddrivers« ins System eingebunden. Falls Ihr Computer über eine Hardware-Uhr verfügt, wird deren Zählerstand (Uhrzeit und Datum) in Zeile 70 ausgelesen, auf dem Bildschirm ausgegeben und in das System übertragen. Falls Sie

Die Zeitmaschine

diese Uhr nicht besitzen, kann dem Befehl »SetClock load« nicht Folge geleistet werden und es wird eine entsprechende Fehlermeldung ausgegeben. Zeile 80 beschleunigt die Textausgabe – die Rückmeldung des Programms »FF« wird wieder ins Leere geschickt.

Der Inhalt der Zeile 90 rechtfertigt einen neuen Absatz, ist er doch maßgeblich an einer der wichtigsten Verbesserungen der Workbench 1.3 beteiligt: dem Ersatz des »alten« Kommando-Prozessors »CLI« durch die neue »SHELL«. Der neue Kommando-Prozessor wurde – wohl um den Einsteiger möglichst lange im Ungewissen zu lassen – entgegen aller Gewohnheit im Verzeichnis »L« der Boot-Diskette unter der Bezeichnung »Shell-Seg« abgelegt. Damit jeder »gewöhnliche« CLI-Aufruf auf den neuen Kommandoprozessor umge-

leitet wird, trägt man »Shell-Seg« in der Resident-Liste mit dem Resident-Namen »CLI« ein. Ebenso in die Resident-Liste wird »Execute« in Zeile 100 übernommen. Der zur »Shell« gehörende Handler »Newcon«, mit dem erst Leistungsmerkmale wie die History-Funktion und die Zeileneditierung möglich werden, wird in Zeile 110 eingebunden. Zeile 120 erhöht den Fehlerlevel auf 11, verhindert also, daß der laufende Batch-Prozeß »wegen jeder Kleinigkeit« abgebrochen wird. Zeile 130 startet einen neuen Prozeß, der die Batch-Datei »StartupII« abgearbeitet.

Stop! Bitte warten

Der »WAIT«-Befehl in Zeile 140 hat den Sinn, die weitere Abarbeitung der Datei »Startup Sequence« so lange zu stoppen, bis entweder fünf Minuten vergangen sind oder ein Befehl zum Weitermachen eintrifft. Sie werden bei der Besprechung von »StartupII« sehen, daß das Kommando »Weitermachen« von dieser Datei gesendet wird. Sollte jedoch, aus welchem Grund auch immer, dieser Fortsetzungsbefehl nicht kommen, wird die Abarbeitung von »Startup-Sequence« allerspätestens nach fünf Minuten fortgesetzt.

Die Fortsetzung sieht dann so aus, daß die deutsche Tastaturbelegung aktiviert wird (Zeile 150). Bis zu diesem Zeitpunkt ist die amerikanische Tastaturbelegung gültig, die sich im Kickstart-ROM befindet. Danach wird für das Betriebssystem ein erweiterter Dateisuchpfad definiert (Zeile 160). Weiter geht's in Zeile 170 mit Laden und Starten der Intuition-Arbeitsoberfläche namens »Workbench«. Die weitere Bearbeitung der Stapeldatei wird so lange gestoppt, bis die Workbench komplett initialisiert ist. Das letzte Kommando der Batch-Datei »Startup Sequence« beendet den aktuellen Task, wobei die Ausführungsrückmeldung wieder ins Nirwana geschickt wird.

Soweit zur Stapeldatei »Startup Sequence«. Wie erwähnt, wird

»Startupll« aus »Startup Sequence« heraus aufgerufen. Warum hat Commodore diese Unterteilung vorgenommen und nicht einfach die Befehle, die in »Startupll« stehen, an entsprechender Stelle in »Startup Sequence« eingetragen? Der Grund ist folgender: Der Prozeß, innerhalb dessen die Startup Sequence abgearbeitet wird, ist immer ein »normaler« CLI-Prozeß ohne die erweiterten Fähigkeiten der »Shell«. Beispiels-

Startup die Zweite

weise können residente Kommandos nur aus einem Shell-Task heraus benutzt werden, nicht aber aus einem CLI-Task. Um die Abarbeitung zu beschleunigen, wird deshalb nach der Installation der Shell und des zugehörigen Handlers »Newcon« ein Shell-Task aufgerufen. In diesem ist es dann möglich, die Fähigkeiten der Shell zu nutzen, um die Abarbeitung zu beschleunigen. Im gegenständlichen Fall bringt das eine Reduktion der Wartezeit bis zur Abarbeitung beider Dateien von rund 12 Prozent, was eine Zeiterparnis von etwa zehn Sekunden bedeutet. Sehen wir uns die »Startupll«-Datei etwas näher an.

```

0      resident c:Resident
      pure
20     resident c:List pure
      ; LIST und CD resident
      machen
30     resident c:CD pure
40     resident c:Mount pure
50     resident c:Assign pure
60     resident c:Makedir pure
70     ; IF, ENDIF, ELSE,
      SKIP, ENDSKIP und
      ECHO resident, falls
80     ; Sie viele Scripts
      benutzen und sich
      den RAM-Bedarf lei-
      sten können.
90     ; Dazu Failat, WAIT
      und ENDCLI resident
      machen, wenn IconX
      oft benutzt.
100    makedir ram:t
110    assign T: ram:t
120    makedir ram:env
130    assign ENV: ram:env

```

```

140    makedir ram:clipboards
150    assign CLIPS: ram:
      clipboards
160    mount speak:
170    mount aux:
180    mount pipe:
190    resident Mount remove
      ; Bei ausreichend RAM
      resident lassen,
200    resident Assign remove
      ; durch Löschen dieser
      Zeilen.
210    resident Makedir remove
220    ;
230    break 1 C ; Signal an
      anderen Prozeß, daß
      er abschließen kann

```

In den Zeilen 10 bis 60 werden einige Befehle in die Resident-Liste und damit in den Speicher eingetragen. Dem »guten Rat« in Zeile 70, 80 und 90 sollten Sie Folge leisten – die 9 KByte können Sie sich sicher leisten, und der erzielte Geschwindigkeitsgewinn rechtfertigt diese »RAM-Verschwendung« allemal. Die Zeilen 100 und 110 legen in der RAM-Disk ein Verzeichnis an, in dem verschiedene Batch-Dateien (auf die wir noch zu sprechen kommen) Hilfs-Files anlegen. Die Zeilen 120 und 130 tun ähnliches für die »Environment-Variables« – der ENV-Handler war zum Zeitpunkt der Veröffentlichung der Workbench-Version 1.3 noch

»SPEAK«, »AUX« und »PIPE«

nicht fertig und wurde deshalb durch dieses Verzeichnis in der RAM-Disk substituiert. Die Einrichtung des Directory »CLIPS« (das u.a. von »Notepad« benutzt wird) in der RAM-Disk sowie das zugehörige Assignment erledigen die Zeilen 140 und 150. Die Aktivierung der logischen Devices »SPEAK«, »AUX« und »PIPE« (Zeile 160, 170 und 180) kostet wirklich nicht viel Speicher und bei Bedarf stehen die genannten Devices prompt zur Verfügung. Während wir in der AMIGA-Redaktion für »SPEAK« nur selten und für »AUX« so gut wie nie Verwendung fanden, ist »PIPE« bei uns ständig im Einsatz, weshalb wir Ihnen das »mounten« speziell dieses Devices nur empfehlen können. In den Zeilen 190 bis 210 werden residente Befehle wieder aus dem Spei-

cher entfernt. Unserer Meinung nach völlig zu Recht – es gibt wesentlich nützlichere Kommandos, die Sie resident halten sollten. Das Ende von »Startupll« bildet Zeile 230. Damit wird der »WAIT«-Befehl im anderen Task abgebrochen. Wenn Sie Zeile 230 löschen, geht die Kontrolle nach Zeile 220 zwar ebenso wieder an den Mutterprozeß (in dem »Startup Sequence« läuft) zurück wie nach Zeile 230, doch wartet dann besagter Prozeß geduldig auf den Ablauf der erwähnten fünf Minuten. Das in Zeile 230 an den Prozeß Nummer 1 geschickte Signal »Control-C« beendet nicht etwa die Abarbeitung der Stapeldatei »Startup Sequence« sondern nur das in diesem Prozeß gerade laufende Programm »WAIT«. Stapel-Dateien werden immer mit »Control-D« abgebrochen – auf diese Weise lassen Sie den eben laufenden Befehl sein Werk noch verrichten und beenden erst danach die Abarbeitung des Stapels. Das ist die »seriöse« Art, einen Batch-Prozeß vorzeitig abzubrechen.

Für alle Hard-Disk-Besitzer interessant ist die Batch-Datei »Startup-Sequence.HD«.

```

10     Setpatch
20     SYS:System/FastMem
      First
30     binddrivers
40     assign >NIL: DH0: ex-
      ists
50     IF NOT WARN
60     assign sys: dh0:
70     assign c: SYS:c
80     assign L: SYS:l
90     assign FONTS: SYS:
      fonts
100    assign S: SYS:s
110    assign DEVS: SYS:devs
120    assign LIBS: SYS:libs
130    makedir ram:tr
140    assign t: ram:tr
150    execute s:Startup-
      Sequence
160    ELSE
170    execute s:Startup-
      Sequence.f
180    ENDIF

```

Diese Stapeldatei ist für all jene gedacht, deren Festplatte nicht auto-bootfähig ist, die also mit einer Boot-Diskette das System aus seinem Reset-Dornröschenschlaf erwecken müssen. Die Zeilen 10 bis 30 haben wir schon bei »Startup Sequence« kennengelernt. In Zeile 40 wird die Einsatzbereitschaft der Hard-Disk (hier mit der Gerätebezeichnung »dh0«) überprüft. Ist der »ASSIGN«-Befehl erfolgreich (Zeile 50), meldet sich also die Festplatte, werden die System-Zuweisungen, die normalerweise auf die Boot-Diskette und deren Verzeichnisse weisen, auf die entsprechenden Verzeichnisse der

Booten mit Hard-Disk

Hard-Disk umgeleitet (Zeile 60 bis 120). Warum Commodore das RAM-Diskverzeichnis »tr« in Zeile 130 anlegt und dann in Zeile 140 dem logischen Device »t« zuweist, ist allerdings nicht ganz klar, wird doch in Zeile 150 die »Standard«-Startup-Sequence-Datei aufgerufen, deren »kleine Schwester« »Startupll« diese Zuweisung wieder zunichte macht (siehe Zeile 110 der »Startupll«-Datei). Damit hätten Sie ein Directory namens »tr« in der RAM-Disk angelegt, auf das kaum ein Programm zugreifen wird. Die Zeilen 160 und 170 kommen nur dann zum Einsatz, wenn die Festplatte keine Betriebsbereitschaft an den »ASSIGN«-Befehl aus Zeile 40 zurückmeldet. In diesem Fall wird die Stapeldatei »Startup Sequence.f« aufgerufen. Damit ist die »Standard«-Startup-Sequence-Datei gemeint, die Sie in »Startup-Sequence.f« umbenennen sollten, ehe Sie die Datei »Startup-Sequence.hd« abarbeiten lassen – nur für den Fall der Fälle. Übrigens müssen Sie die Datei »Startup-Sequence.hd« in »Startup Sequence« umbenennen, wenn Sie Ihre Hard-Disk für den Systemstart heranziehen wollen. Tun Sie dies aber erst, nachdem Sie die Original-Startup-Sequence-Datei in »Startup-Sequence.f« umbenannt haben, sonst ist sie futsch. Daß vor der ersten Ab-

arbeitung von »Startup-Sequence.hd« der Inhalt der Directories »C«, »FONTS«, »S« etc. in die entsprechenden Directories der Festplatte kopiert werden muß, sei hier nur der Vollständigkeit halber erwähnt.

Der Trick mit der Hard-Disk-Abfrage in Zeile 40 funktioniert übrigens mit allen logischen und physischen Devices, also auch mit Disketten-Laufwerken. Mit

```

10      assign >NIL: DF1:
        exists
20      IF NOT WARN
30      Addbuffers df1: 10
40      echo " *e[42;1;7m
        Laufwerk-Puffer von
        Drive #1 vergrößert
        *e[0m"
50      ELSE
60      echo " *e[33;1;7m
        Drive #1 nicht eingeschaltet ! *e[0m"
70      ENDIF

```

fragen Sie die Verfügbarkeit von Laufwerk »df1:« ab (Zeile 10). Ist df1: vorhanden (Zeile 20), geht es weiter mit Zeile 30, in der Sie den Pufferspeicher des Laufwerks um rund 5000 Byte vergrößern, und Zeile 40, die Sie über das Vorhandensein dieses Laufwerks informiert (die Escape-Sequenzen im Text sind nur schmückendes Beiwerk). Ist das Laufwerk aber nicht eingeschaltet, geht es nach Zeile 10 weiter bei Zeile 50. In diesem Fall brauchen Sie keinen Speicher zu vergeuden und werden in Zeile 60 von der Nichtverfügbarkeit des Laufwerks informiert. Zeile 70 dient nur der »Einklammerung« des gesamten IF-Blocks.

Falls Sie eine resetfeste RAM-Disk einrichten wollen (»RAM:« ist **nicht resetfest**!), fügen Sie bitte folgende Zeilen in der Startup Sequence ein:

```

10      mount RAD:
20      assign >NIL: RAD: EX-
        ISTS
30      IF NOT WARN
40      copy s:RAD-Icon.info
        to RAD:Disk.info
50      ELSE
60      ECHO "Sorry - zu-
        wenig Speicher frei"
70      ENDIF
80      cd RAD:

```

In Zeile 10 richten Sie die RAM-Disk »RAMB0« ein. Sollte nicht genug Speicher frei sein (die Voreinstellung in der Mountlist belegt rund 200 KByte), wird die RAM-Disk nicht angelegt. Damit die folgenden Kommandos nicht »ins Leere greifen«, wird die erfolgreiche Installation mit Zeile 20 abgefragt. Hat es geklappt (Zeile 30), wird in Zeile 40 ein eigenes »RAD-Icon« in die neue RAM-Disk kopiert. Falls Sie mit dem vom Betriebssystem vorgegebenen Icon zufrieden sind, kann diese Zeile entfallen. Falls Sie aber ein eigenes Icon der resetfesten RAM-Disk zuordnen wollen, müssen Sie ein Icon vom Typ »Disk« dafür nominieren. Die Bezeichnung »RAD-Icon« in Zeile 40 ist ebenso wahlfrei wie der »Aufbewahrungsort« dieses Icons im Verzeichnis »S«. Wichtig ist nur, daß Sie das neue Icon in die RAM-Disk unter dem Namen »Disk.info« kopieren, ehe Sie »LoadWB« aufrufen. Gleichermaßen verfahren Sie, wenn Sie der RAM-Disk »RAM:« ein eigenes Icon spendieren wollen. Wurde keine RAM-Disk »RAD:« angelegt, werden die Zeilen 30 und 40 übersprungen und man konfrontiert Sie in Zeile 60 mit dem Fehlschlag. Damit das Icon der »RAD« auf der Workbench erscheint, müssen Sie einen Zugriff auf »RAD« vornehmen lassen. Gleiches gilt für das Icon der RAM-Disk

Das »RAD:« neu erfunden

»RAM:«. Ein Beispiel für einen solchen Zugriff ist in Zeile 80 eingetragen. Noch eine Anmerkung zu »RAD:«. Sie sollten den obigen Kommandoblock soweit wie möglich an den Anfang der »Startup Sequence« stellen. Sie laufen sonst Gefahr, daß beim nächsten Warmstart die Daten der RAM-Disk versehentlich überschrieben werden.

Wenden wir uns nun den beiden Script-Files zu, die in engem Zusammenhang mit Kommandofenstern zu sehen sind: »CLI-Startup« und »Shell-Startup«. Immer dann, wenn

Sie mit »NewCLI« einen CLI-Task starten und kein spezielles Steuer-File im Aufruf angeben, sucht der Befehlsprozessor im »S«-Verzeichnis nach einem File namens »CLI-Startup«. Findet er ein solches File, wird nach dem Start des neuen Prozesses zuerst die Anweisungsliste im File »CLI-Startup« abgearbeitet, ehe die »Kommandogewalt« an Sie übergeben wird. Existiert kein File des gesuchten Namens, steht Ihnen der Prozeß sozusagen »roh« zur Verfügung. Das Gesagte gilt übrigens analog für »NewShell« und die Datei »Shell-Startup«. Der Inhalt von »CLI-Startup« besteht lediglich aus

```
prompt "%N> "
```

womit als Eingabe-Prompt die Tasknummer ausgegeben wird, gefolgt von einem »>«. Wie Sie sicher bereits erkannt haben, wird durch das (notwendige) Laden des »Prompt«-Befehls von Diskette etwas Zeit verbraucht, ehe Sie den neuen CLI-Prozeß »in Besitz« nehmen können. Abhilfe würde hier eine Aufnahme des »Prompt«-Kommandos in die Resident-Liste schaffen (oder Sie löschen einfach das Batch-File »CLI-Startup« – steht Ihnen frei).

Etwas mehr steht da schon in der Stapeldatei »Shell-Startup«:

```
10 Prompt "%N.%S> "
20 alias xcopy copy []
   clone
30 alias endshell endcli
40 alias pro execute s:
   spat protect []
50 alias sdate execute
   s:spat setdate []
60 alias ren execute s:
   dpat rename []
70 alias clear echo "*E[
   0;0H*E[J"
80 alias reverse echo "*
   E[0;0H*E[41;30m*E[J"
90 alias normal echo "*
   E[0;0H*E[40;31m*E[J"
```

Die Zeilennummern wurden wieder von uns eingefügt – wären sie Bestandteil des Script-Files, würde sich die »Shell« schön bedanken. In Zeile 10 sehen Sie wieder den schon von »CLI-Startup« bekannten »Prompt«-

Befehl, diesmal aber erweitert um die Zeichenfolge ».%S«. Mit »%S« erzwingen Sie die Ausgabe des aktuellen Directory als Bestandteil des Prompt – eine feine Sache. Die folgenden Zeilen »erweitern den Wortschatz« des Command-Line-Interpreter. »Befehlsmakro« nennt Commodore diese Erweiterungen, die nur mit der Shell, nicht aber mit dem Standard-CLI möglich sind, da der Befehl »alias« fixer und integrierter Bestandteil der Shell sein muß. Zeile 20 versetzt die Shell in die Lage,

Kopieren mit »clone«

»XCOPY« als gültiges Kommando und als Synonym für den Aufruf des »COPY«-Befehls mit aktiviertem Wahlschalter »clone« anzusehen. Die beiden eckigen Klammern umschließen einen Bereich, in den nach Aufruf von »XCOPY« die in der Kommandozeile übergebenen Filenamen eingetragen werden.

```
XCOPY devs: ram: all
```

wird von der Shell in die Kommandozeile

```
Copy devs: ram: all clone
```

umgeformt, ehe diese an den CLI weitergereicht wird. Sie sehen, daß der im Script vorgesehene eckig geklammerte Bereich nun die Namen von Quelle und Ziel enthält. Mit solchen Befehlsmakros können Sie sich eine Menge Tipparbeit ersparen. Zeile 30 weist das System an, auf das Kommando »EndShell« so zu reagieren, als wäre »EndCLI« eingegeben worden. Genausogut könnten Sie beispielsweise anstelle von »EndShell« das wesentlich kürzere »WB« eintragen. In den Zeilen 40, 50 und 60 kommen die Scripts »SPAT« und »DPAT« zum Einsatz. An dieser Stelle sei lediglich angemerkt, daß durch den Eintrag in Zeile 40 der Shell ein Befehl implementiert wird, mit dem der »Protect«-Befehl wild-cardfähig gemacht wird. Während bekanntlich die Befehlszeile

```
protect devs:#?.device +h
```

zur Rückmeldung

```
Protect failed
```

führt und die Nachfrage des Users mit »Why« die Aussage

```
Last command failed because
object not found
```

und damit die Erinnerung an die Joker-Unfähigkeit des »Protect«-Kommandos ans Tageslicht bringt, schafft die Befehlszeile

```
pro devs:#?.device +h
```

das Gewünschte problemlos. Ähnliches gilt für das Makro »sdate«, mit dem der »SetDate«-Befehl Jokerfähig wird. Diese Makroliste können Sie frei nach Ihren Wünschen und Bedürfnissen erweitern. Ähnlich verhält es sich mit dem Makro in Zeile 60, nur daß dort nicht »SPAT« zur Anwendung kommt, sondern »DPAT«. Damit wird die Pattern-Unverträglichkeit des »Rename«-Befehls teilweise behoben. Mit

```
ren Artikel/Seite#? Artikel/
Buch_A/
```

werden alle Files im Verzeichnis »Artikel«, die mit »Seite« beginnen, so umbenannt, daß sie danach im Unterverzeichnis »Buch_A« zu finden sind. Beachten Sie bitte die etwas exotische Schreibweise für das zu wählende Ziel-Directory. Und noch etwas: verwenden Sie **nicht** die Schlüsselworte »FROM« und »TO«! »DPAT« benötigt Quelle und Ziel in der obigen Reihenfolge. Egal, was Sie als zweites Argument übergeben (»FROM«, »TO«, ...), es wird immer als Zielangabe interpretiert.

Die Zeilen 70, 80 und 90 vereinfachen die Zeichenattributierung im CLI und die Ausgabe von »ESCAPE«-Sequenzen. Die Alias-Namensgebung umschreibt die Funktionen unserer Meinung nach ausreichend, und was es mit diesen ESCAPE-Sequenzen auf sich hat, lesen Sie an anderer Stelle in diesem Heft – dabei wollen wir es belassen.

Wenn Sie sich mehr mit den Möglichkeiten der Stapelverarbeitung befassen, werden Sie sicher noch viele weitere Anwendungen entwickeln. Wenn es soweit ist und Sie eine tolle Batch-Datei entwickelt haben, denken Sie bitte an die AMIGA-Sonderheft-Redaktion. Wir veröffentlichen gerne interessante und nützliche Batch-Dateien. Viel Spaß und Erfolg beim Stapeln. ■

Eingabehilfe

CHECKSUMMER

Mit unserem Checksummer »Checkie 42« ist die Zeit der Tippfehler vorbei. Alle unsere Listings, egal welche Programmiersprache, können Sie mit ihm eingeben. Nutzen Sie diese Vorteile und sparen Sie Zeit bei der Suche nach Fehlern in den von uns abgedruckten Listings.

Ein längeres Listing ohne Fehler abzutippen ist (fast) unmöglich. Aus diesem Grund gibt es zum Abtippen der Listings aus dem AMIGA-Magazin seit der Ausgabe 3/88 eine Eingabehilfe – den Checksummer »Checkie 42«. Hier noch einmal für alle neuen Leser dieses unentbehrliche Werkzeug. Damit möglichst viele unserer Leser das Programm auch tatsächlich anwenden, haben wir es kurz gehalten und in einer Sprache programmiert, die alle Abtipper besitzen: Amiga-Basic. Die Listingzeilen im AMIGA-Magazin bestehen aus einer vierstelligen Zeilennummer, der zwei- bzw. dreistelligen Prüfsumme und der eigentlichen Programmzeile. Beispiel:

```
1 TTo print "Hallo!"
```

Nach einer Leerstelle im Anschluß an die Zeilennummer stehen bis zu drei Zeichen Prüfcode. Die einzelnen Zeichen können sein eine Ziffer (»0« bis »9«), ein kleiner Buchstabe (»a« bis »z«) oder ein Großbuchstabe (»A« bis »Z«). Die ersten beiden Zeichen der Prüfsumme sind der eigentliche Prüfcode. Im dritten Zeichen ist die Spaltenposition der ersten »Nicht-Leerstelle« verschlüsselt. Das ist für diejenigen Anwender interessant, welche die Struktur des Listings, also die Einrückungen durch Leerzeichen, übernehmen wollen. Ist dies nicht Ihre Absicht, können Sie die Eingabe der Checksummer schon nach den ersten beiden Zeichen mit RETURN abschließen. Checkie 42 ist »case sensitive«, das heißt, daß die Groß- und Kleinschreibung so wie im Listing abgedruckt übernommen werden muß. Nach dem Start fragt das Programm nach einem Dateinamen. Unter dem angegebenen Namen speichert Checkie 42 die eingegebenen Listingzeilen. Existiert bereits eine Datei mit diesem Namen auf der Diskette, so haben Sie zwei Möglichkeiten:

1. Ausgabe der Datei mit Checksumme auf den Bildschirm oder den Drucker.
2. Umleitung der Eingabe von der Tastatur auf diese Datei.

Beide Alternativen sind gedacht für Anwender, die ein Listing nicht mit dem Zeileneditor des Checkie, sondern mit einem schnelleren und/oder komfortableren Editor ihrer Wahl – z.B. dem Editor von Amiga-Basic (mit »..«, »a« speichern) erfaßt haben. Checkie 42 errechnet Ihnen bei der Alternative 1 die Prüfsummen Ihres Textes und Sie können diese dann mit dem Listing im AMIGA-Magazin vergleichen. Bei der Ausgabe auf den Bildschirm schreibt das Programm die Programmzeilen inklusive Checksummen zusätzlich in eine Datei mit dem Zusatz ».chk«. Diese können Sie dann später mit dem Systembefehl TYPE ohne erneute Berechnung der Prüfsumme noch einmal ausgeben. Wollen Sie dem Programm den Vergleich überlassen, so beantworten Sie die Frage »Eingabe aus Datei« mit »j«. Dann brauchen Sie nur noch die Checksummen eingeben. Der Checksummer holt sich die Zeile aus der angegebenen Datei statt von der Tastatur. Entspricht die eingegebene Prüfzahl nicht der errechneten, kann die Zeile gleich korrigiert werden. Beantworten Sie obige Frage mit »n«, zählt Checkie die in der Datei vorhandenen Zeilen und wartet mit der Zeilennummer »Anzahl+1« auf die Eingabe der nächsten Zeile. Alle weiteren Eingaben hängt das Programm dann an die Datei an. Diese Funktion ist sinnvoll, wenn Sie ein Listing in mehreren Teilen abtippen wollen.

Nach Erledigen der »Dateiverwaltung« schlägt Checkie 42 eine Zeilennummer vor und wartet auf die Prüfsumme. Nach Eingabe derselben taucht der Cursor zwischen den zwei Trennstrichen auf. Dort muß nun die Zeile »ohne« Zeilennummer und Prüfsumme eingegeben werden. Nach Betätigen der RETURN-Taste berechnet Checkie die Prüfsumme. Leerstellen vor und hinter der Programmanweisung werden ignoriert. Stimmen Programmzeile und Prüfsumme mit derjenigen im Listing überein, speichert der Checksummer die Eingabe und wartet auf die nächste Zeile. Wahrscheinlich wird eine abgetippte Zeile mal einen Fehler enthalten. Checkie 42 positioniert den Cursor dann an den Anfang der Zeile und wartet auf die korrekte Eingabe. Korrekturen lassen sich mit der Backspace- oder Delete-Taste durchführen. Um Zeichenfolgen einzufügen, kann kurzfristig mit

<F2> der Einfügemodus eingeschaltet werden. Dieser Modus sollte allerdings nach der Fehlerkorrektur wieder ausgeschaltet werden, da er die Eingabe verlangsamt.

Möchten Sie eine Kommentarzeile nicht »original« übernehmen, läßt sich unter Nichtbeachtung der falschen Prüfsumme eine Übernahme der Zeile mit der Funktionstaste <F6> erzwingen. Sie können damit aber auch falsche Programmzeilen übernehmen. Verwenden Sie deshalb die Taste <F6> nicht gewohnheitsmäßig. Der Checksummer teilt Ihnen nach Beenden des Programms mit, wieviel Zeilen er ungeprüft übernommen hat.

Natürlich kann es auch vorkommen, daß die Programmzeile zwar richtig abgetippt wurde, sich bei der Prüfsumme aber ein Fehler eingeschlichen hat. Nach Betätigen von <F1> kann die Prüfsumme korrigiert werden. Während der Eingabe der Prüfsumme läßt sich mit <F7> die vom Programm vorgeschlagene Zeilennummer verändern. Damit können Sie gezielt nur bestimmte Teile eines Listings übernehmen. Haben Sie eine mit einem anderen Editor geschriebene Programmdatei überprüft und nur in wenigen Zeilen Fehler festgestellt, lassen sich durch Vorgabe der Nummern diese Zeilen gezielt ändern. Bei Angabe der Zeilennummer in aufsteigender Reihenfolge benötigt das Programm übrigens erheblich weniger Zeit für die Suche der Zeilen in der jeweiligen Datei. Um die versehentliche Übernahme fehlerhafter Zeilen zu verhindern, sperrt das Programm bei fehlender Übereinstimmung der Prüfsummen die Änderung der Zeilennummer mit der Taste <F7>.

Die Kombination <Ctrl> beendet den Programmlauf nach vollständiger Eingabe des Listings oder für eine Unterbrechung.

Am Schluß noch ein Tip für diejenigen Leser, denen unser Basic-Editor zu langsam ist: Die Berechnung der Prüfsumme geschieht im Unterprogramm »CalcSumme«. Dieser Teil ist sehr einfach in schnelleren Sprachen, wie C, umsetzbar.

Natürlich werden wir Anregungen von Ihnen gerne aufnehmen und »Checkie 42« laufend verbessern.

Wer schon einmal Fehler in einem abgetippten Listing gesucht hat, weiß, wie frustrierend diese Tätigkeit ist. Nutzen Sie deshalb Checkie 42. Sie sparen damit viel Zeit für die vielleicht vergebliche Suche nach tückischen Fehlern.

Dieter Behlich/pa/rb

```

1 Cq0 REM *****
2 KF REM *** Checksummer: Checkie 42 ***
3 7g REM *** Version 1.2 ***
4 Ft REM *****
5 Z7 Start:
6 op2 CLEAR,82000&:GOSUB Init
7 Pn GOSUB OpenDate1
8 mb IF dn$="" THEN Ende
9 b2 GOSUB Bild
10 Ow0 NeueZeile:
11 K42 GOSUB loeschen
12 zT GOSUB EingabeSumme
13 pt IF FEnde=wahr THEN Ende
14 Mv0 Wiederholung:
15 sQ2 GOSUB EingabeZeile
16 sw IF FEnde=wahr THEN Ende
17 mO GOSUB CalcSumme
18 fB IF FZok = falsch THEN GOTO Wiederholung
19 oC GOSUB Uebernahme
20 iv GOTO NeueZeile
21 ma0 Ende:
22 7S2 GOSUB fertig
23 72 END
24 UG0 Init:
25 fD2 wahr=-1
26 fy falsch=0
27 Kh LZeile=240 : REM Anzahl Zeichen/Zeile
28 28 LBZeile=60 : REM Anzahl Zeichen/Bildschirmzeile
29 OG AnzBZeilen=LZeile/LBZeile
30 N1 zx=6 : zy=14 : REM Position Zeile
31 PC sx=20 : sy=5 : REM Position Checksumme
32 mF AnzCsZ=3 : REM Anzahl Ziffern/Checksumme
33 QF DIM z(LZeile)
34 O5 DIM cs(AnzCsZ*2)
35 iz cs(AnzCsZ)=0 : REM Zeilenstart
36 gv a=0 : b=0 : c=0 : REM Hilfsvariablen
37 AI i=0 : j=0 : k=0 : REM Zählvariablen
38 XL FZok = wahr : FCz3=100
39 Wn Checkfile=0 : Zeile=1
40 Bb READ Faktor(i)
41 X1 WHILE Faktor(i) <> 0
42 gs4 i=i+1
43 cm READ Faktor(i) : REM Faktorenreihe
44 9x2 WEND
45 Gm AnzFak=i
46 RB DATA 2,3,4,5,6,0

```

Listing

Mit »Checkie 42«
können Sie Listings
fehlerfrei abtippen


```

224 fY      GOSUB backup
225 Xa      OPEN dn$+".bak" FOR INPUT AS #1
226 T1      OPEN dn$ FOR OUTPUT AS #2
227 9t      Zeile=1
228 oV6     END IF
229 Yp      WHILE (NeuZeile > Zeile) AND (NOT EOF(1))
230 fQ8      LINE INPUT #1,e$
231 mn      PRINT #2,e$
232 l7      Zeile=Zeile+1
233 CO6     WEND
234 cR      IF EOF(1) THEN
235 bL8      CLOSE 1
236 8i      NeuZeile=Zeile
237 uN      LOCATE zy,1:PRINT USING "# # # #";NeuZeile;
238 9G      Checkfile=0
239 ng6     END IF
240 oh4     END IF
241 vV      Zeile=NeuZeile
242 qJ2     END IF
243 Ht0     RETURN
244 lH      EingabeZeile:
245 n72     x=cs(AnzCsZ)
246 Bw0     weiter:
247 ga2     cy=zy+INT(x/LBZeile):cx=zx+(x MOD LBZeile)
248 vy      LOCATE cy,cx
249 o6      COLOR 0,1
250 mI      PRINT CHR$(z(x));
251 y1      LOCATE cy,cx
252 68      IF x>apos THEN apos=x
253 xx      IF Checkfile AND FZok THEN
254 w14      IF EOF(1) THEN
255 OR6      Checkfile=0 : CLOSE 1
256 jS4      ELSE
257 ix6      e$=INPUT$(1,1)
258 6z4      END IF
259 mV2     ELSE
260 Q24     e$=INKEY$
261 922     END IF
262 gp4     IF e$="" THEN weiter
263 OI      COLOR 1,0
264 OW      PRINT CHR$(z(x));
265 CF      LOCATE cy,cx
266 8G      e=ASC(e$)
267 UR2     IF ((e AND 127)<32) OR e=127 THEN Controlcode
268 op      IF imode THEN GOSUB insert
269 I6      PRINT e$
270 K1      z(x)=e : e=30
271 3D0     Controlcode:
272 gF2     IF e=13 OR e=10 THEN
273 lN4     RETURN
274 O92     ELSEIF e=30 THEN
275 2p4     a=1
276 Ea2     ELSEIF e=29 THEN
277 O34     a=LBZeile
278 6G2     ELSEIF e=31 THEN
279 DD4     a=-1
280 Cb2     ELSEIF e=28 THEN
281 Rg4     a=-LBZeile
282 9s2     ELSE
283 UX4     GOTO noCrs
284 WP2     END IF
285 hz      x=x+a
286 5Z      IF x>=0 AND x<LZeile THEN weiter
287 tD      x=x-a
288 HB      GOTO weiter
289 kEO     noCrs:
290 dN2     IF e=8 THEN
291 lS4     IF x>0 THEN
292 Kk6     x=x-1
293 AM      LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
294 Bb      FOR i=x TO apos
295 788     z(i)=z(i+1)
296 Ja      PRINT CHR$(z(i));
297 Z0      IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
298 FV6     NEXT i
299 pr      z(apos)=32 : PRINT " "
300 yR      apos=apos-1
301 ng4     END IF
302 H12     ELSEIF e=127 THEN
303 Kk4     FOR i=x TO apos
304 CH6     z(i)=z(i+1)
305 Sj      PRINT CHR$(z(i));
306 I9      IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
307 Oe4     NEXT i
308 y0      z(apos)=32 : PRINT " "
309 7a      apos=apos-1
310 Vy2     ELSEIF e=129 THEN

```

```

311 oI4     GOSUB EingabeSumme
312 sC      x=cs(AnzCsZ)
313 9U2     ELSEIF e=130 THEN
314 5I3     imode=imode XOR 1
315 tU5     LOCATE 7,28
316 yZ3     IF imode THEN
317 IT5     PRINT "aus"
318 jS3     ELSE
319 YV5     PRINT "ein"
320 6z3     END IF
321 Kg2     ELSEIF e=131 THEN
322 L54     GOSUB loeschen
323 3N      x=cs(AnzCsZ)
324 7Y2     ELSEIF e=134 THEN
325 bD4     RETURN
326 PR2     ELSEIF e=5 THEN
327 ZT4     FEnde=wahr
328 eG      RETURN
329 F82     END IF
330 xr      GOTO weiter
331 IF0     insert:
332 oJ2     IF apos>x THEN
333 3o4     FOR i=apos TO x STEP -1
334 Qt6     z(i+1)=z(i)
335 q64     NEXT i
336 PH      z(x)=32
337 Ny      apos=apos+1
338 Kp      IF apos=LZeile THEN apos=apos-1:z(LZeile)=32
339 uK      FOR i=x TO apos
340 lI6     PRINT CHR$(z(i));
341 Hi      IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
342 xD4     NEXT i
343 yA      LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
344 UN2     END IF
345 vX      RETURN
346 cX0     CalcSumme:
347 3s2     a=0 : b=0 : c=0
348 z2      IF e=134 THEN
349 pv4     FZok=wahr
350 WX      FF6=FF6+1
351 Gz2     ELSE
352 hv4     WHILE z(apos)=32 AND apos>0
353 pi6     apos=apos-1
354 9x4     WEND
355 Ni      IF apos>0 THEN
356 O36     WHILE z(c)=32
357 Bt8     c=c+1
358 D16     WEND
359 jc4     END IF
360 EJ      FOR i=c TO apos
361 Y26     j=(i-c) MOD AnzFak
362 Qs      k=(i+1-c) MOD AnzFak
363 s1      a=a+((z(i) AND 127)-32)*Faktor(j)
364 2E      b=b+((z(i) AND 127)-32)*Faktor(k)
365 Ka4     NEXT i
366 pA      cs(4)=a+Zeile-INT((a+Zeile)/62)*62
367 4N      cs(5)=b+Zeile-INT((b+Zeile)/62)*62
368 4Q      FZok=(cs(1)=cs(4)) AND (cs(2)=cs(5))
369 tm2     END IF
370 Kw      RETURN
371 pM0     uebernahme:
372 xD2     FOR i=0 TO apos
373 Ih4     PRINT #2,CHR$(z(i));
374 Tj2     NEXT i
375 VZ4     PRINT #2,""
376 5R2     Zeile=Zeile+1
377 R3      RETURN
378 N90     fertig:
379 MJ2     IF Checkfile THEN
380 Z24     WHILE NOT EOF(1)
381 6r6     LINE INPUT #1,e$
382 DE      PRINT #2,e$
383 cQ4     WEND
384 Ok      CLOSE 1
385 922     END IF
386 5q      CLOSE 2
387 nt      CLS
388 Rg      LOCATE 12,35
389 WL      PRINT "F E R T I G !!!"
390 I9      LOCATE 20,1
391 D1      IF FF6<>0 THEN
392 C14     PRINT "ACHTUNG!!! ";
393 hL      PRINT FF6;" Zeile(n) wurde(n) ungeprüft gespeichert."
394 IB2     END IF
395 jL      RETURN

```

(C) 1989 M&T

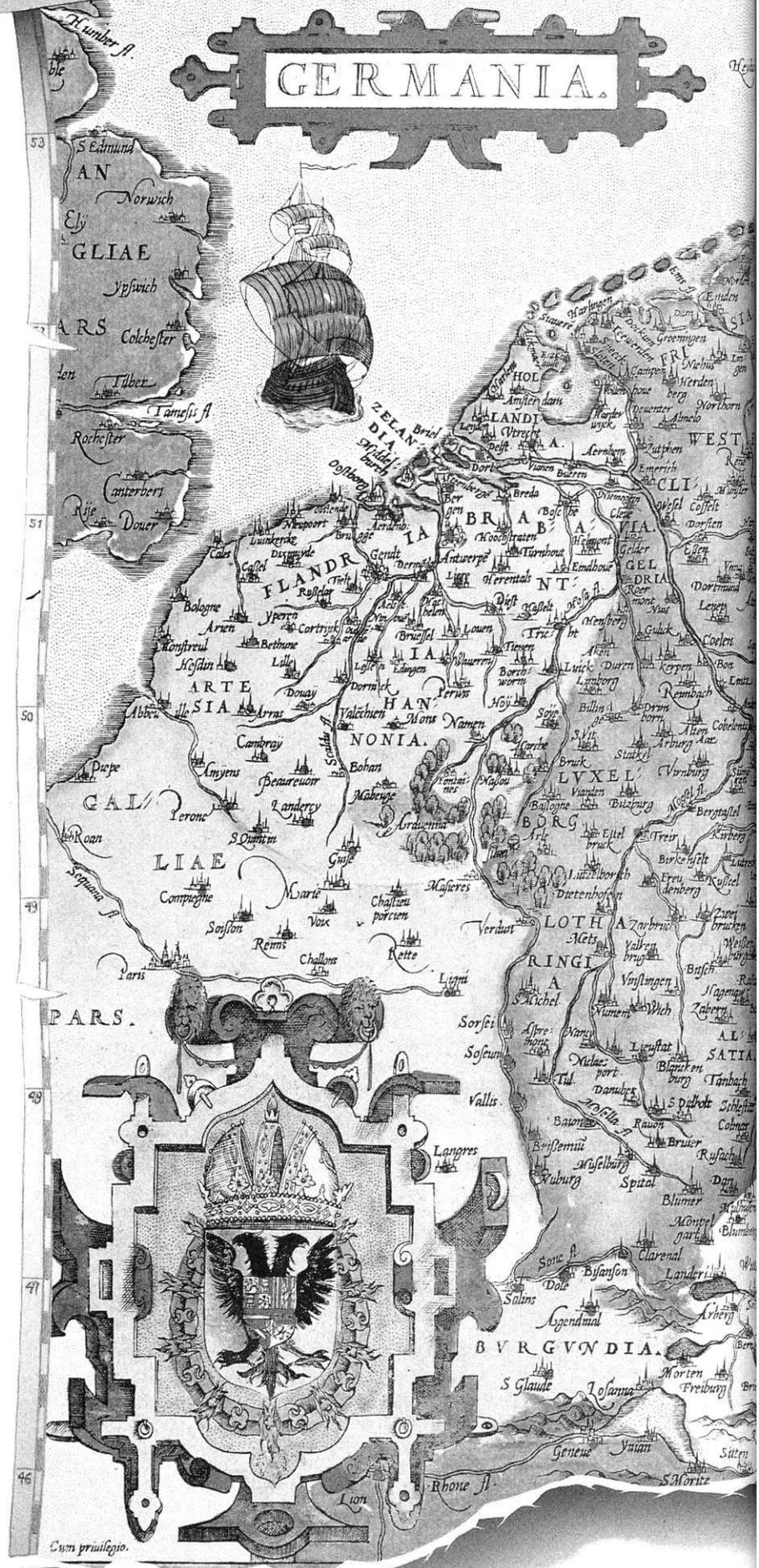
Listing

Mit »Checkie 42«
können Sie Listings
fehlerfrei abtippen

Handelssimulation

Herr der Meere

Es ist ein harter Weg zum erfolgreichen Reeder. Die Umstände machen es Ihnen nicht leicht. Aber mit Glück, List und Tücke ernennt man Sie sogar zum Bürgermeister. Bis es soweit ist benötigen Sie noch viel Erfahrung mit dem Umgang von Waren und Handelsschiffen.





von Martin Jobst

Keine leichte Aufgabe, sicher als Reeder und Handelsunternehmer in einem heißumkämpften Markt zu profilieren: Waren einkaufen, Schiffe entsenden, Handelshäuser eröffnen, Image pflegen... »Reeder« ist eine komplexe Handelssimulation für bis zu vier Spieler, die viel strategisches Geschick erfordert. Auf einer grafisch ansprechenden Oberfläche ziehen Sie Ihre Fäden und Stricke im harten Seehandelsbusiness.

Die Kommando-zentrale

Ihre Kommandozentrale ist das Hauptmenü (Bild 1), von hier aus können Sie alle gewünschten Aktionen tätigen. Die Menüstruktur von Reeder ermöglicht eine unkomplizierte Bedienung über die Tastatur. Zur Anwahl der einzelnen Punkte genügt ein Druck auf die entsprechende Zifferntaste.

Nach dem Start von »Reeder« müssen Sie sich etwas gedulden, bis das Titelintrö aufgebaut wird. Abgebrochen wird dieses Intro mit der <Space>-Taste. Nach der Angabe der Anzahl der Mitspieler und deren

Die Steuerung

Namen erscheint das Hauptmenü. Von dort aus gelangen Sie in folgende Spielabschnitte:

0 Nächstes Jahr:

Der Spielzug eines Spielers ist beendet, ein neues Jahr wird eingeläutet. Die getätigten Aktionen bestimmen die weitere Entwicklung - hoffentlich zu Ihren Gunsten.

Nehmen mehrere Spieler bei Reeder teil, kommen zunächst diese an die Reihe, bevor dem alten Jahr ade gesagt wird.

1 ver/kaufen:

Hier können Sie - sofern es Ihr Goldbeutel zuläßt, Waren aus dem Angebot einkaufen. Sehen Sie zu, daß genügend Lagerplatz (siehe »5 Lager«) für Ihre Waren zur Verfügung

steht. Wenn Sie Kapital benötigen und die Preise günstig stehen, können Sie auch Waren verkaufen.

2 Schiffe:

Schiffe sind Ihr Kapital bei Reeder (Bild 2). Mit ihnen steht und fällt der Handel. Ohne Sie kommt der Markt zum Stillstand. Die riesigen Handelsschiffe werden natürlich nicht im Laden zum Mitnahmepreis angeboten. Sie müssen erst in Auftrag gegeben werden und stehen dann nach einem Geschäftsjahr zur Verfügung.

Aber natürlich nagt der Zahn der Zeit an Ihren Schiffen. Sie sollten daher von Zeit zu Zeit repariert werden. Sinkt nämlich der Zustandwert eines Schiffes unter 50 Prozent, besteht akute Sinkgefahr - also Vorsicht. Eine solche Havarie kostet Sie nicht nur das Schiff, sondern natürlich auch die mitgeführten Waren.

Sie können Ihre Schiffe auch verkaufen, wegen ihres niedrigen Wiederverkaufspreises ist das aber nur in Notfällen zu empfehlen.

3 Schiffe senden:

Damit Sie mit Ihren Schiffen Handel tätigen können, müssen sie zu Handelsstädten entsandt werden. Sie können nur mit jenen Städten Handel betreiben, in denen Sie ein Kontor (Handelshaus; siehe »4 Kontor eröffnen«) besitzen. Zu Beginn besitzen Sie nur in Rostock ein Kontor. Die entsandten Schiffe kehren nach einem Jahr automatisch wieder zurück und bringen Waren mit.

Schiffe können nur dann auf die Reise geschickt werden, wenn folgende Bedingungen erfüllt sind:

- Es liegen Schiffe im Hafen.
- Sie können die Heuer bezahlen.
- Sie haben Salz eingekauft.
- Es ist ausreichend Platz für das Salz vorhanden.
- Es steht ein Kontor zur Verfügung.

4 Kontor eröffnen:

Hiermit können Sie die Anzahl Ihrer Handelshäuser vergrößern, um Ihre Geschäfte auszudehnen.

Zu Beginn des Spiels steht Ihnen nur ein Kontor - in Rostock - zur Ver-

fügung. Dieses Handelshaus kommt Sie zwar billig, und die Heuer kostet wenig, andererseits bringen aber auch die Waren nicht viel ein. Teure Kontore in anderen Städten sind finanziell wesentlich interessanter.

5 Lager:

Es genügt nicht, Waren einzukaufen, sie benötigen auch jede Menge Platz. Sehen Sie vor dem Einkauf zu, daß auch genügend Lagerplatz zur Verfügung steht. Waren, für die kein Platz ist, verderben und gehen verloren. Jedes Lager, das Sie erwerben, bietet Platz für 500 Wareneinheiten.

Erworbene Lagerplätze können – mit geringem Goldverlust – auch wieder verkauft werden.

6 Soldaten:

Wollen Sie vor Angriffen gewappnet sein, sollten Sie sich eine kleine Armee zusammenstellen. Soldaten kosten aber Geld, ohne den geforderten Sold desertieren sie. Kanonen tun das nicht, sie kosten aber auch wesentlich mehr als Soldaten.

Es könnte ja sein, daß Waldemar von Dänemark Lübeck angreift und es auf Ihre Lagerwaren abgesehen hat. Spätestens dann machen sich die Kosten für Ihre kleine Privatarmee bezahlt.

7 Laden/Speichern:

Reeder ist ein komplexes Spiel. Damit Sie Ihre Errungenschaften nicht nach jeder Spiel-Session wieder verlieren, kann der Spielstand von bis zu zehn Spielen gespeichert werden. Reeder legt die Dateien automatisch auf jener Diskette an, von der Sie das Programm gestartet haben. Die Diskette darf natürlich nicht schreibgeschützt sein.

Später kann ein Spielstand über diesen Menüpunkt wieder geladen und das Spiel fortgesetzt werden.

8 Börse:

Eine interessante Bereicherung von Reeder ist die eingebaute Börse (Bild 3): Jeder Spieler stellt eine Aktiengesellschaft dar, die auf der Börse vertreten ist. Dort wird mit Anteilsprozenten der verschiedenen Ge-

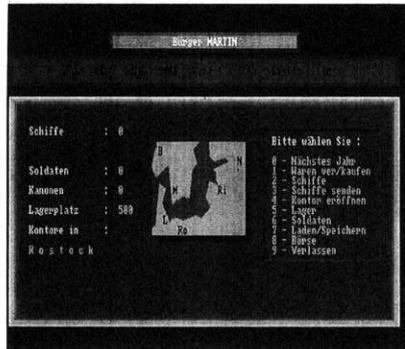


Bild 1. Von der Kommandozentrale aus ziehen Sie die Fäden

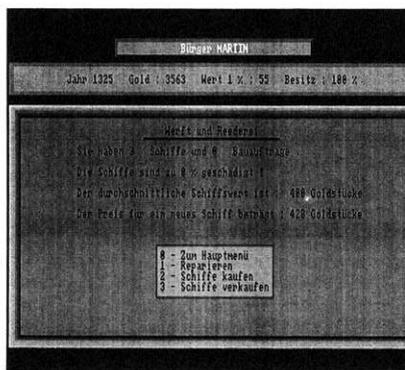


Bild 2. Erst wenn Sie Schiffe besitzen, können Sie Handel treiben

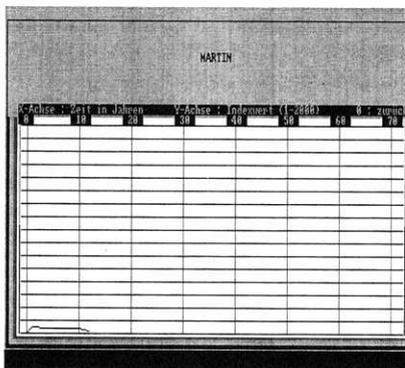


Bild 3. An der Börse können Sie den Wert Ihrer Aktien ablesen

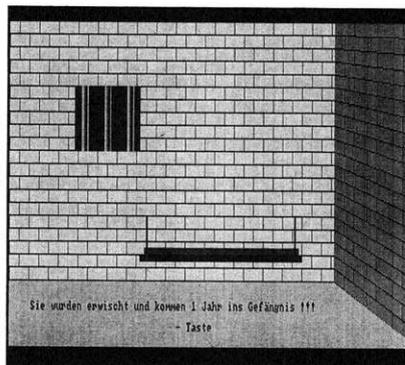


Bild 4. Ein mißglücktes Schmuggelgeschäft hat den Handelsmann hinter Gitter gebracht

sellschaften gehandelt. Der Aktienkurs richtet sich nach dem momentanen Vermögen (Kapital/Wert) der einzelnen Spieler. Zusätzlich steigert ein Index für begangene gute Taten den Wert der einzelnen Aktien.

Mit dem Wert der Aktien Ihrer Handelsgesellschaft steigt zugleich auch Ihr gesellschaftlicher Status: vom Bürger über den Großhändler bis zum Bürgermeister.

9 Beenden:

Mit dieser Option wird das Programm beendet, und Sie gelangen auf die Workbench.

Zwischenfälle

Wie im richtigen Leben geht natürlich auch bei Reeder kein Jahr ereignislos vorüber: Ein Brand kann ausbrechen, und Sie verlieren einen Großteil der wertvollen Waren. Der Bischof von Köln bittet Sie um eine Spende zum Bau des Doms – gut für die Imagepflege. Sie werden von einer schlimmen Krankheit befallen, und nur eine teure Behandlung bringt Sie wieder auf die Beine. Ein vielversprechendes Schmuggelgeschäft bietet sich an, kann Sie allerdings auch ins Gefängnis (Bild 4) bringen. Ein Krieg bricht aus, und Sie müssen mit Ihrer Armee um die nackte Existenz kämpfen etc.

Für Abwechslung und Spannung ist in Reeder mit Sicherheit gesorgt. Aber nur gewiefte Händler schaffen es bis zum Bürgermeister...

Eingabehinweise

Um in den Genuß von Reeder zu kommen, müssen Sie die Listings »Loader«, »Reeder« und »Sieg« mit Hilfe von Checkie 42 eingeben und speichern. Zum Start von Reeder genügt ein Doppelklick auf das Icon »Loader«. Bitte beachten Sie, daß sich sowohl »Loader«, »Reeder« und »Sieg« als auch Amiga-Basic in jenem Verzeichnis befinden müssen, von dem Sie Reeder starten. pe

Programmname:	Sieg
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic
Bemerkung:	siehe Text
Programmautor: Paul Schokal ----- <pre> 1 EJO FOR i=0 TO 15:PALETTE i,0,0,0:NEXT 2 5s CLS:COLOR ,3 3 ck LINE (10,15)-(620,235),6,bf 4 d7 FOR i%=20 TO 230 5 jE x=RND*15000:PATTERN x:LINE (20,i%)-(610,i%),5:NEXT 6 ax FOR i%=275 TO 355 7 vp x=RND*5000+10000:PATTERN x:LINE (i%,20)-(i%,215),4:NEXT 8 Oj FOR i=1 TO 4 9 bY LINE (313+i,21)-(313+i,230),1:NEXT 10 JF PATTERN -1:COLOR ,0 11 ec FOR i%=20 TO 50 STEP 5 12 8J LINE (i%,20)-(i%,240-i%/2),1 13 gV LINE (630-i%,20)-(630-i%,240-i%/2),1 14 pP LINE (i%,240-i%/2)-(630-i%,240-i%/2),1 15 KP NEXT 16 sV FOR i%=1 TO 30 STEP 5 17 7N SCROLL (305-i%,0)-(325+i%,250),0,1:NEXT 18 vK COLOR 1,3:LOCATE 5,9:PRINT "Im Jahre";Jahr;" wurde der" 19 yn PRINT TAB(9) "Großbreeder ";Na\$(Pla) 20 tk PRINT TAB(9) "zum Bürgermeister von der" 21 u4 PRINT TAB(9) "Hansestadt Lübeck gewählt." 22 iL PRINT 23 tL PRINT TAB(9) "Er war ein anständiger" 24 Iz PRINT TAB(9) "Bürger,der Armen und" 25 BA PRINT TAB(9) "Waisen half,die Kirche" 26 E5 PRINT TAB(9) "unterstützte und gute" 27 Hi PRINT TAB(9) "Dienste leistete im Kampf" 28 9Q PRINT TAB(9) "gegen den König von" 29 Vw PRINT TAB(9) "Dänemark, Waldemar den IV." 30 qT PRINT 31 lA PRINT TAB(9) "Durch schwere Arbeit," 32 dH PRINT TAB(9) "und durch seinen guten" 33 60 PRINT TAB(9) "Geschäftssinn,arbeitete er" 34 yR PRINT TAB(9) "sich langsam hoch ,bis er," 35 pU PRINT TAB(9) "an der Spitze seiner" 36 GV PRINT TAB(9) "Erfolge,vom dankbaren Volk" 37 de PRINT TAB(9) "zum Bürgermeister ernannt" 38 AK PRINT TAB(9) "wurde ." 39 Tt LOCATE 5,47 40 pv IF Hei(Pla)>0 THEN 41 vY1 PRINT "Bürgermeister ";Na\$(Pla) 42 JA PRINT TAB(47) "wurde am 13.1. 1301 " 43 JI PRINT TAB(47) "geboren und heiratete im" 44 fe PRINT TAB(47) "Jahre ";Hei(Pla);" die Tochter" 45 fD PRINT TAB(47) "einer wohlhabenden Familie" 46 6j PRINT 47 ha0 END IF 48 Ls PRINT TAB(47) "Möge er noch lange" 49 r6 PRINT TAB(47) "ein guter Bürgermeister" 50 6K PRINT TAB(47) "für unsere Hansestadt" 51 CF PRINT TAB(47) "Lübeck sein !" 52 Cp PRINT 53 ly PRINT TAB(60) "" 54 sn LOCATE 26,47:PRINT "Taste drücken und warten !" 55 n6 PALETTE 3,.9,.6,.4:PALETTE 4,.6,.4,.3 56 tz PALETTE 5,.7,.5,.4:PALETTE 6,.6,.2,0 57 Jh REM WHILE INKEY\$<>"":WEND:d\$="" 58 z8 REM WHILE d\$="" :d\$=INKEY\$:WEND 59 Af FOR i=0 TO 15:PALETTE i,0,0,0:NEXT 60 G5 COLOR 4,4:CLS 61 HL CIRCLE (315,95),100,6:PAINT (315,95),6 62 Op CIRCLE (315,80),50,1,3,7,5,7 63 qT LINE (273,92)-(356,92),1:LINE (287,98)-(343,98),1 64 BL PAINT (315,97),1:LINE (335,88)-(356,92),1,bf 65 zv LINE (273,92)-(260,91),1 66 Bw LINE (295,92)-(296,80),1,bf:LINE (320,92)-(321,75),1,bf 67 WB LINE (0,100)-(630,180),3,bf 68 Ih LINE (0,181)-(630,250),5,bf 69 XH COLOR 6,3:PATTERN 4228:LINE (0,99)-(630,99),6:PATTERN -1 70 mb AREA (109,100):AREA (521,100):AREA (630,145) </pre>	

<pre> 71 J2 AREA (630,180):AREA (0,180):AREA (0,145):AREAFILL 72 GA x%=-95 73 tr FOR i=110 TO 520 STEP 1.5 74 PT LINE (i,100)-(x%,180),3 75 JN x%=x%+3:NEXT 76 Gv FOR i%=100 TO 180 77 fn x%=RND*150 78 3Y LINE (x%,i%)-(x%+1,i%),6 79 gk LINE (x%+480,i%)-(x%+481,i%),6 80 NS NEXT 81 fU FOR i%=100 TO 170 STEP 5 82 4S SCROLL (0,i%)-(630,i%+4),RND*40-20,0 83 QV NEXT 84 lK x%=-315 85 AB FOR i%=0 TO 630 STEP 20 86 pO LINE (i%,181)-(x%,250),0 87 6i x%=x%+40:NEXT 88 wU LINE (535,221)-(585,223),1,bf 89 Ug AREA (540,220):AREA (545,200):AREA (535,200):AREA (532,198) 90 kK AREA (532,196):AREA (535,194):AREA (570,194):AREA (580,220) 91 rn COLOR 7:AREAFILL 92 8t COLOR 8:LOCATE 1,1:PRINT "THE END" 93 So FOR i%=1 TO 63 94 KO FOR x%=0 TO 6 95 6b IF POINT (i%,x%)=8 THEN PUT (i%*11+5,x%*10+80),t%,PSET 96 Jm NEXT:NEXT:LINE (0,0)-(64,7),4,bf 97 ue PALETTE 3,0,0,.5:PALETTE 4,0,0,1:PALETTE 5,.4,.3,0 98 oL PALETTE 6,1,.3,0:PALETTE 7,.4,.4,.4 99 qg ENDE: 100 25 FOR i%=0 TO 7 101 tD PALETTE f1%(i%),.1,.75,0:PALETTE f2%(i%),.93,.68,0 102 JI PALETTE f3%(i%),.87,.62,0:PALETTE f4%(i%),.81,.56,0 103 N8 PALETTE f5%(i%),.75,.5,0:PALETTE f6%(i%),.68,.44,0 104 A0 PALETTE f7%(i%),.62,.38,0:PALETTE f8%(i%),.56,.3,0 105 mr NEXT 106 7c GOTO ENDE </pre>	<p>Listing 1. Nur wenn Sie's bis zum Bürgermeister schaffen, benötigen Sie dieses Listing</p> <p>(C) 1990 M&T</p>
---	--

Programmname:	Loader
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic
Bemerkung:	siehe Text
Programmautor: Paul Schokal ----- <pre> 1 b00 REM Reeder loader by Paul Schokal in 1989 2 Fv CLEAR ,76000& 3 lO LOAD "REEDER",r </pre> (C) 1990 M&T	
<p>Listing 2. Das Ladeprogramm muß im selben Verzeichnis wie »Reeder« und »Sieg« stehen</p>	

Programmname:	Reeder
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic
Bemerkung:	siehe Text
Programmautor: Paul Schokal ----- <pre> 1 nVO REM ----- 2 56 REM --- Reeder V 3.01 --- 3 K8 REM --- by Paul Schokal in 1990 --- 4 qY REM ----- 5 Cr WINDOW CLOSE 1 6 5d SCREEN 2,641,280,4,2 </pre>	

```

7 zf t$=SPACE$(15)+" - Reeder - programmed in 1989 by PAUL SCHO
KAL"
8 z2 WINDOW 2,t$(0,0)-(630,250),0,2
9 KT RANDOMIZE TIMER
10 ze GOSUB schwarz
11 1F DEFINT m1,m2,oh
12 Va DIM f1$(7),f2$(7),f3$(7),f4$(7),f5$(7),f6$(7),f7$(7),f8$(7)
)
13 dg FOR i%=0 TO 7
14 JW READ f1$(i%),f2$(i%),f3$(i%),f4$(i%),f5$(i%),f6$(i%),f7$(i
%),f8$(i%)
15 KP NEXT
16 aC DIM t$(50):x%=8
17 WQ FOR i%=20 TO 40 STEP 2
18 1h LINE (0,i%)-(i%,0),x%
19 Zd LINE (0,i%+1)-(i%+1,0),x%
20 HV x%=x%+1:IF x%=16 THEN x%=8
21 QV NEXT
22 7v GET (10,10)-(20,20),t%
23 rD CLS:LOCATE 1,5:PRINT "Reeder":x%=0
24 5s FOR i%=10 TO 290 STEP 36
25 PO AREA (0,0):AREA (i%,150):AREA (600-i%,150):COLOR x%:AREAFI
LL
26 QS x%=x%+1:NEXT
27 m1 FOR i%=33 TO 79
28 GK FOR x%=0 TO 6
29 Yn IF POINT (i%,x%)=1 THEN PUT (i%*11-284,x%*10+80),t%,PSET
30 Ae NEXT:NEXT
31 kJ d$="Reeder by PAUL SCHOKAL in 1990 !!!":x%=8
32 Mg FOR i%=1 TO 34
33 FH COLOR x%:LOCATE 4,23+i%:PRINT MID$(d$,i%,1):x%=x%+1
34 BX IF x%=16 THEN x%=8
35 ej NEXT
36 zF FOR i%=33 TO 25 STEP -1:SCROLL (176,i%)-(460,38),1,1:NEXT
37 7e d$=" Copyright by Markt & Technik

38 eS x%=8:FOR i%=58 TO 1 STEP -1
39 Qp COLOR x%:LOCATE 29,10+i%:PRINT MID$(d$,i%,1):x%=x%+1
40 Hd IF x%=16 THEN x%=8
41 kp NEXT
42 nv x=0:FOR i%=0 TO 7:PALETTE i%,0,0,x:x=x+.0625:NEXT
43 gB x=0:i1=d$=""
44 LO WHILE d$=""
45 vW d$=INKEY$
46 AD FOR i%=0 TO 7
47 Et PALETTE f1$(i%),1,1,0:PALETTE f2$(i%),.93,0,0
48 lz PALETTE f3$(i%),.87,0,0:PALETTE f4$(i%),.81,0,0
49 GQ PALETTE f5$(i%),.75,0,0:PALETTE f6$(i%),.68,0,0
50 96 PALETTE f7$(i%),.62,0,0:PALETTE f8$(i%),.56,0,0
51 40 Weg%=Weg%+i1
52 ER IF Weg%=70 THEN i1=-1
53 xd IF Weg%=0 THEN i1=1
54 Ag SCROLL (0,80)-(610,80+Weg%),i1*5*SIN(.4*Weg%),0
55 y3 NEXT
56 L9 WEND
57 Ox FOR i%=0 TO 7:PALETTE i%,0,0,0:NEXT
58 aC FOR Weg%=0 TO 140 STEP 2
59 L7 SCROLL (0,80)-(600,80+Weg%),i1*5*SIN(.4*Weg%),-4
60 7c NEXT:CLS
61 gB Anfang:
62 yS GOSUB farben
63 1A LOCATE 3,16:COLOR 9,0:PALETTE 9,0,0,0
64 vf PRINT "Wieviele Spieler wollen ins Geschäft einsteigen ?"
65 Hy LOCATE 9,18:Spieler=1
66 fj PRINT "Drücken Sie die Tasten 1-4 und dann SPACE "
67 ny LINE (306,36)-(326,50),15,bf
68 oP LINE (308,38)-(324,48),12,bf
69 OZ COLOR 8,12:LOCATE 6,40:PRINT "1"
70 Sk FOR i=0 TO 1 STEP .00625:PALETTE 9,0,i,0:NEXT
71 Xt key1:
72 J6 GOSUB Taste
73 rH IF ASC(d$)>48 AND ASC(d$)<53 THEN
74 ET2 FOR i=0 TO 9:SCROLL (308,38)-(324,48),0,-1:NEXT
75 f1 LOCATE 6,40:PRINT d$:Spieler=VAL(d$)
76 A30 END IF
77 O7 IF d$<>" " THEN key1
78 N4 FOR i%=0 TO 80:LINE (0,i%)-(600,i%),0:NEXT
79 hF GOSUB Init
80 IJ zug=1:Jahr=zug+1321:g$="Goldstücke"
81 5x FOR i%=1 TO Spieler
82 o4 wh1:
83 9D COLOR 9,0:CLS:LOCATE 8,18:PRINT "Name des Spielers ";i%;
84 D1 COLOR 8:INPUT Na$(i%)
85 Fa IF LEN(Na$(i%))=0 OR LEN(Na$(i%))>10 THEN wh1
86 HJ Na$(i%)=UCASE$(Na$(i%)):Ti$(i%)="Bürger"
87 Hh alt$(i%)=60+INT(RND*10)
88 EH ap$(i%,i%)=100
89 gJ La(i%)=1
90 ZJ Kon$(i%,5)=1
91 AM Kont$(i%)=1
92 bb Gold(i%)=5000
93 U9 kurve$(i%,0)=240
94 iJ GOSUB preise
95 wn GOSUB Index
96 di NEXT
97 JM ST$(1)="Malmö":ST$(2)="Bergen":ST$(3)="Riga"
98 4Q ST$(4)="Nowgorod":ST$(5)="Rostock"
99 yY War$(1)="Töpfe Honig":War$(2)="Kisten Fische"
100 M2 War$(3)="Tonnen Getreide":War$(4)="Stapel Pelze"
101 4K War$(5)="Tonnen Eisen"
102 ZC Kon$(1)="M a l m ö":Kon$(2)="B e r g e n":Kon$(3)="R i g a
"
103 eU Kon$(4)="N o w g o r o d":Kon$(5)="R o s t o c k"
104 Vr GOSUB schwarz:COLOR ,5:CLS
105 KN AREA (410,90):AREA (410,196)
106 Dw AREA (412,195):AREA (412,91)
107 uF AREA (598,91):AREA (600,90)
108 qg COLOR 1:AREAFILL
109 sw AREA (411,196):AREA (412,195)
110 Na AREA (598,195):AREA (598,91)
111 TY AREA (600,90):AREA (600,196)
112 3w COLOR 4:AREAFILL
113 iR COLOR 10:LOCATE 13,54
114 cw PRINT "Bitte wählen Sie ":COLOR 11
115 pQ LOCATE 15,54:PRINT "0 - Nächstes Jahr"
116 A8 LOCATE 16,54:PRINT "1 - Waren ver/kaufen"
117 QE LOCATE 17,54:PRINT "2 - Schiffe"
118 eN LOCATE 18,54:PRINT "3 - Schiffe senden"
119 AB LOCATE 19,54:PRINT "4 - Kontor eröffnen"
120 5c LOCATE 20,54:PRINT "5 - Lager"
121 DS LOCATE 21,54:PRINT "6 - Soldaten"
122 uM LOCATE 22,54:PRINT "7 - Laden/Speichern"
123 ZN LOCATE 23,54:PRINT "8 - Börse"
124 My LOCATE 24,54:PRINT "9 - Verlassen"
125 oC LINE (240,105)-(390,180),1,bf
126 Yv LINE (230,100)-(380,170),8,bf
127 86 COLOR 9
128 gH FOR i%=1 TO 22 STEP 2
129 Li READ x%,i1:AREA (x%,i1):NEXT:AREAFILL
130 sV FOR i%=1 TO 24 STEP 2
131 Nk READ x%,i1:AREA (x%,i1):NEXT:AREAFILL
132 oQ FOR i%=1 TO 32 STEP 2
133 Pm READ x%,i1:AREA (x%,i1):NEXT:AREAFILL
134 N1 COLOR 1,9:LINE (230,170)-(380,175),9,bf
135 1P LINE (236,103)-(237,104),6,bf:LOCATE 14,31:PRINT "B"
136 06 LINE (269,147)-(270,148),6,bf:LOCATE 21,32:PRINT "L"
137 Qy LINE (258,163)-(259,164),6,bf:LOCATE 22,35:PRINT "Ro"
138 BJ LINE (265,161)-(266,162),6,bf:LOCATE 18,34:PRINT "M"
139 gG LINE (334,133)-(335,134),6,bf:LOCATE 18,43:PRINT "Ri"
140 Vh LINE (368,122)-(369,123),6,bf:LOCATE 15,47:PRINT "N"
141 kD GET (230,90)-(600,196),bild$:Pla=1
142 bP NaeJahr:
143 tM IF Na$(Pla)="" THEN Weg
144 yF IF zug>alt$(Pla)-1 THEN Tot
145 JY GOSUB preise:i%=Pla:GOSUB Index1
146 33 IF w<10 THEN
147 IA2 a$="Ein Jahr Knast , weil Sie versagt und das Geschäft r
uinert haben !"
148 FS GOSUB Gef:Gold(Pla)=5000
149 LEO END IF
150 NW IF ASC(LEFT$(Na$(Pla),1))=0 THEN
151 ph2 Na$(Pla)=RIGHT$(Na$(Pla),LEN(Na$(Pla))-1)
152 px GOTO Weg
153 PIO END IF
154 pB Haupt:
155 kZ GOSUB schwarz
156 kX COLOR 1,5:CLS
157 Q9 LINE (0,0)-(630,30),1,bf
158 a9 LINE (0,56)-(630,63),1,bf
159 Ao LINE (163,12)-(468,27),13,bf
160 i6 AREA (163,27):AREA (167,24):AREA (463,15)
161 Gh AREA (468,12):AREA (468,27):COLOR 15:AREAFILL
162 Ar LINE (167,15)-(463,24),12,bf
163 mk COLOR 3,12:x%=LEN(Ti$(Pla))+1+LEN(Na$(Pla)):x%=(80-x%)/2

```

```

164 As LOCATE 3,x%:PRINT Ti$(Pla)+" "+Na$(Pla)
165 9z GOSUB Head:x=10
166 R4 FOR i%=0 TO 15 STEP 3
167 UX x%=64+i%/2:i1=249-i%/2
168 EB LINE (i%,x%)-(i%+2,i1),x,bf
169 6p LINE (630-i%,x%)-(628-i%,i1),x,bf
170 A9 LINE (i%,x%)-(630-i%,x%+1),x,bf
171 KS LINE (i%,i1)-(630-i%,i1-1),x,bf
172 Ev x=x+1:NEXT
173 gA COLOR 10:LOCATE 12,5:PRINT "Schiffe      : ";
174 Ao COLOR 11:PRINT Schiffe(Pla)
175 Vf IF we(Pla)>0 THEN
176 cJ1 COLOR 10:LOCATE 14,5:PRINT "Schiffszustand : ";COLOR 11
177 O8 GOSUB SWert:PRINT INT(SW/4);"% "
178 oh0 END IF
179 PZ COLOR 10:LOCATE 16,5:PRINT "Soldaten      : ";
180 2I COLOR 11:PRINT Sol(Pla)
181 9S COLOR 10:LOCATE 18,5:PRINT "Kanonen      : ";
182 GC COLOR 11:PRINT Kan(Pla)
183 lI GOSUB LaCheck
184 1B COLOR 10:LOCATE 20,5:PRINT "Lagerplatz   : ";
185 TS COLOR 11:PRINT Lap(Pla)
186 7A COLOR 10:LOCATE 22,5:PRINT "Kontore in    : "
187 6q x%=24:COLOR 11
188 OQ FOR i%=1 TO 5
189 EE IF Kon$(Pla,i%)=1 THEN LOCATE x%,5:PRINT Kon$(i%):x%=x%+1
190 9E NEXT
191 pv PUT (230,90),bildg,PSET
192 4Y GOSUB farben
193 ax key2:
194 H4 GOSUB Taste
195 Mx ON VAL(d$)+1 GOTO OK,Kau,Sch,Sen,Kon,Lag,Sol,Lad,Boe,Quit
196 np GOTO key2
197 st Quit:
198 px COLOR 8:LOCATE 30,17
199 yY PRINT "Wollen Sie Reeder wirklich verlassen ??? (J/N)"
200 mA key3:
201 OB GOSUB Taste
202 jX d$=UCASE$(d$):IF d$="J" THEN quit1 ELSE quit2
203 z2 GOTO key3
204 zW quit1:
205 Ey SYSTEM
206 7a quit2:
207 2I LOCATE 30,17:PRINT SPACE$(46):GOTO key2
208 TO Kau:
209 vd LINE (18,74)-(612,240),5,bf
210 GT COLOR 1,5:LOCATE 11,28:PRINT "Markt - An und Verkauf"
211 b6 LINE (175,89)-(425,90),1,bf
212 iy LOCATE 13,17
213 BM PRINT "Schiffsladungen ( 100 Kisten Salz ) : "
214 p6 LOCATE 14,17
215 9T PRINT "Freier Lagerplatz";TAB(55) " : "
216 DG LOCATE 16,17:COLOR 7
217 YL PRINT "Drücken Sie :           Aktuelle Preise :   Ihr
      Vorrat : "
218 D3 COLOR 1
219 1T LOCATE 17,17:PRINT "0 - zurück zum Hauptmenü"
220 r1 LOCATE 18,5:PRINT "ANKAUF :   1 - von Lübecker Salz"
221 1A LOCATE 20,5:PRINT "VERKAUF :   2 - Honig aus Malmö"
222 h6 LOCATE 21,17:PRINT "3 - Fisch aus Bergen"
223 gp LOCATE 22,17:PRINT "4 - Getreide aus Riga"
224 WE LOCATE 23,17:PRINT "5 - Pelze aus Nowgorod"
225 LW LOCATE 24,17:PRINT "6 - Eisenerz aus Rostock"
226 Tx LOCATE 25,17:PRINT "7 - Alle Vorräte EINER Ware"
227 6n LOCATE 26,17:PRINT "8 - Alle Vorräte"
228 aI Kau1:
229 Ff GOSUB LaCheck:COLOR 1
230 6f LINE (448,96)-(500,111),5,bf
231 h1 LOCATE 13,56:PRINT INT(Wa(Pla)/100)
232 zD LOCATE 14,56:PRINT Lap(Pla)
233 O9 LINE (20,208)-(600,240),5,bf
234 OX LINE (408,136)-(600,190),5,bf
235 4a LOCATE 18,52:PRINT wap;:PRINT TAB(68) Wa(Pla)
236 AC FOR i%=1 TO 5
237 zn LOCATE 19+i%,52:PRINT PR$(i%);:PRINT TAB(68) War$(Pla,i%):
      NEXT
238 Rj GOSUB Head
239 Ut key4:
240 NO GOSUB Taste:COLOR 7
241 BJ IF ASC(d$)<48 OR ASC(d$)>56 THEN key4
242 N2 IF d$="0" THEN Haupt
243 4X IF d$="1" THEN Kauf

244 MC IF d$="7" THEN
245 C52 LOCATE 28,17:PRINT "Alles verkaufen ? - Welche Ware ? (2
      -6)"
246 Bu GOSUB Taste:IF ASC(d$)<50 OR ASC(d$)>54 THEN clean1
247 iE LOCATE 30,17:i=VAL(d$)-1
248 hn IF War$(Pla,i)>0 THEN
249 Cw4 Gold(Pla)=Gold(Pla)+War$(Pla,i)*PR$(i):War$(Pla,i)=0
250 Tk PRINT "Die ganze Ware ist verkauft !":GOTO Kau1
251 zs2 END IF
252 N4 GOTO clean1
253 iu0 END IF
254 aR IF d$="8" THEN
255 r52 LOCATE 28,17:PRINT "Wirklich ALLES verkaufen ? (J/N)"
256 W9 GOSUB Taste:IF d$<>"J" THEN clean1
257 So FOR i=1 TO 5
258 L5 Gold(Pla)=Gold(Pla)+War$(Pla,i)*PR$(i):War$(Pla,i)=0
259 GL NEXT
260 20 LOCATE 30,17:PRINT "In Ordnung - Ich habe Alles verkauft
      !"
261 He GOTO Kau1
262 A30 END IF
263 OM LOCATE 28,17
264 lq i=VAL(d$)-1
265 y4 IF War$(Pla,i)>0 THEN
266 Gt2 PRINT "Wieviel ";War$(i);" verkaufen Sie ";:INPUT anz
267 24 anz=INT(anz):IF anz<1 THEN clean1
268 O2 IF anz>War$(Pla,i) THEN
269 cf4 LOCATE 30,17:COLOR 6
270 iE PRINT "Sie haben nicht so viele ";War$(i);" !!! - Tas
      te "
271 JA GOSUB Taste:GOTO clean1
272 KD2 END IF
273 3q Gold(Pla)=Gold(Pla)+anz*PR$(i):War$(Pla,i)=War$(Pla,i)-a
      nz
274 Ur GOTO Kau1
275 NGO END IF
276 UJ clean1:LINE (20,208)-(600,240),5,bf:GOTO key4
277 69 Kauf:
278 Fb LOCATE 28,17
279 q3 INPUT "Wieviele Kisten Salz wollen Sie kaufen ";anz
280 FH anz=INT(anz):IF anz<1 THEN clean1
281 H2 IF Gold(Pla)-anz*wap<0 THEN
282 ps1 LOCATE 30,17:COLOR 6
283 gt PRINT "Soviel Gold haben Sie nicht ! - Taste"
284 wN GOSUB Taste:GOTO clean1
285 XQ0 END IF
286 Th IF Lap(Pla)<anz THEN
287 ux1 LOCATE 30,17:COLOR 6
288 xI PRINT "Ihr Lagerplatz reicht nicht ! - Taste"
289 iS GOSUB Taste:GOTO clean1
290 eV0 END IF
291 ii Gold(Pla)=Gold(Pla)-anz*wap:Wa(Pla)=Wa(Pla)+anz
292 m9 GOTO Kau1
293 Mq Sch:
294 IO LINE (18,74)-(612,240),5,bf
295 yx COLOR 9:LOCATE 11,31:PRINT "Werft und Reederei"
296 F5 LINE (206,89)-(416,90),1,bf
297 w9 LOCATE 13,14
298 hn PRINT "Sie haben Schiffe und Bauaufträge ."
299 Vf IF we(Pla)>0 THEN
300 z51 GOSUB SWert:x=INT(100-(SW/4))
301 AP LOCATE 15,14
302 Ey PRINT "Die Schiffe sind zu";x;"% geschädigt !"
303 Md LOCATE 17,14
304 iD PRINT "Der durchschnittliche Schiffswert ist : ";SW;g$
305 rk0 END IF
306 Zs LOCATE 19,14
307 O3 PRINT "Der Preis für ein neues Schiff beträgt : ";Schp;g$
308 iC key6:
309 YP LINE (32,160)-(610,240),5,bf
310 MR LINE (225,171)-(407,212),14,bf
311 HK LINE (228,173)-(404,210),12,bf
312 Hq COLOR 15,12
313 aE LOCATE 23,30:PRINT "0 - Zum Hauptmenü"
314 2K LOCATE 24,30:PRINT "1 - Reparieren"
315 GE LOCATE 25,30:PRINT "2 - Schiffe kaufen"
316 Kp LOCATE 26,30:PRINT "3 - Schiffe verkaufen"
317 JL COLOR 9,5:LOCATE 13,23

```

Listing 3. »Reeder« entschädigt Sie sicher für die Tipparbeit

```

318 5c PRINT Schiffe(Pla);" ";TAB (39) BauA(Pla)
319 I5 GOSUB Taste
320 px IF VAL(d$)=0 AND d$<>"0" OR VAL(d$)>3 THEN key6
321 kb LINE (32,160)-(610,240),5,bf
322 cj ON VAL(d$)+1 GOTO Haupt,Rep,SchiKa,SchiVk
323 Q3 Rep:
324 fF IF x<1 or Schiffe(Pla)=0 THEN key6
325 IU LOCATE 21,14
326 CC PRINT "Die Reparatur pro Prozent des Zustandes kostet :";(
Schiffe(Pla))*5
327 kV LOCATE 23,14:INPUT "Wieviel wollen Sie reparieren";anz
328 eB anz=INT(anz):IF anz<1 THEN key6
329 Ne IF anz>x THEN
330 nr2 LOCATE 25,14:COLOR 6
331 zu PRINT "So sehr sind Ihre Schiffe nun auch nicht beschädi
gt ! - Taste"
332 LH GOSUB Taste:GOTO key6
333 JCO END IF
334 V5 IF anz*Schiffe(Pla)*5>Gold(Pla) THEN
335 sw2 LOCATE 25,14:COLOR 6
336 2V PRINT "Tja,da reicht das Gold aber nicht !!! - Taste"
337 QM GOSUB Taste:GOTO key6
338 OH0 END IF
339 ID Gold(Pla)=Gold(Pla)-anz*Schiffe(Pla)*5
340 9o we(Pla)=we(Pla)+anz*Schiffe(Pla)*4
341 CA GOSUB Head:GOTO Sch
342 SO SchiKa:
343 Pz LOCATE 21,14:INPUT "Wieviele Schiffe sollen gebaut werden
";anz
344 uR anz=INT(anz):IF anz<1 THEN key6
345 PC IF Gold(Pla)<anz*Schp THEN
346 wG2 COLOR 6:LOCATE 23,14
347 I9 PRINT "Doch dazu fehlt Ihnen leider etwas Gold ! - Taste
"
348 bX GOSUB Taste:GOTO key6
349 ZSO END IF
350 iy Gold(Pla)=Gold(Pla)-anz*Schp
351 pM BauA(Pla)=BauA(Pla)+anz
352 AP GOSUB Head:GOTO key6
353 1P SchiVk:
354 c0 IF Schiffe(Pla)=0 THEN key6
355 my LOCATE 21,14
356 lz PRINT "Auf dem Markt bekommen Sie nur den Schiffswert aus
ezahlt !"
357 cU LOCATE 23,14:INPUT "Wieviele Schiffe wollen Sie verkaufen "
;anz
358 8f anz=INT(anz):IF anz<1 THEN key6
359 Dk IF anz>Schiffe(Pla) THEN
360 Gc2 COLOR 6:LOCATE 25,14
361 CZ PRINT "Soviele Schiffe haben Sie gar nicht ! - Taste"
362 pl GOSUB Taste:GOTO key6
363 ng0 END IF
364 iS Gold(Pla)=Gold(Pla)+anz*SW
365 bM we(Pla)=we(Pla)-anz*SW
366 aQ Schiffe(Pla)=Schiffe(Pla)-anz
367 Pe GOSUB Head:GOTO key6
368 3f Sen:
369 Dy LINE (0,64)-(630,250),5,bf
370 BS FOR i=0 TO 9 STEP 2:LINE (i*2,64+i)-(630-i*2,250-i),1,b:NE
XT
371 9a COLOR 1:LOCATE 11,35:PRINT "Hafenbüro"
372 8d LINE (236,89)-(382,90),1,bf
373 6a IF Kont%(Pla)<1 THEN
374 ub1 COLOR 6:LOCATE 13,7
375 9W PRINT "Leider haben Sie gar keinen Kontor,Sie können kein
e Schiffe senden !"
376 s3 PRINT:PRINT TAB(36) "- Taste":GOSUB Taste:GOTO Haupt
377 1u0 END IF
378 ln LOCATE 17,6
379 9Z PRINT "Wählen Sie eine Stadt mit der jeweiligen Zahl !
0 - zum Hauptmenü"
380 x1 LOCATE 19,6
381 4a PRINT "Ihre Kontore :";TAB(34)"Heuer pro Schiff :";
382 mL PRINT TAB (59) "Gesandte Schiffe : "
383 PX COLOR 7:LOCATE 21
384 Ya FOR i%=1 TO 5
385 N1 IF Kon%(Pla,i%)=1 THEN PRINT TAB (5) i%;" - ";Kon$(i%);T
AB (33) He$(i%);" ";g$
386 JO NEXT
387 xC LOCATE 13,25:PRINT "Es liegen Schiffe im Hafen"
388 a6 LOCATE 15,22:PRINT "Sie haben Schiffsladungen Salz"
389 9b key7:
390 ZU LINE (18,208)-(612,240),5,bf
391 lR COLOR 7:a$=""
392 kC LINE (264,96)-(311,103),5,bf:LOCATE 13,34:PRINT Schiffe(P1
a)
393 Yd LINE (240,112)-(287,119),5,bf:LOCATE 15,31
394 dg x=INT(Wa(Pla)/100):PRINT x
395 HQ LOCATE 21,11
396 km FOR i%=1 TO 5
397 RQ IF Kon%(Pla,i%)=1 THEN PRINT TAB (65) FaSch%(Pla,i%)
398 Va NEXT
399 za GOSUB Taste:IF ASC(d$)<48 OR ASC(d$)>53 THEN key7
400 va IF d$="0" THEN Haupt
401 ID i=VAL(d$):IF Kon%(Pla,i)=0 THEN key7
402 r6 LOCATE 27,20
403 pR PRINT "Wieviele Schiffe sollen nach ";ST$(i);" ";:INPUT an
z
404 93 anz=INT(anz)
405 CN IF anz<1 THEN key7
406 yV IF anz>Schiffe(Pla) THEN
407 bl2 COLOR 6:LOCATE 29,18
408 pc PRINT "Sie haben nicht genug Schiffe !!! - Taste"
409 eZ GOSUB Taste:GOTO key7
410 YRO END IF
411 K1 IF anz*He$(i)>Gold(Pla) THEN
412 vk2 COLOR 6:LOCATE 29,8
413 Ht PRINT "Ihr Gold reicht nicht,um der Mannschaft die Heuer
zu zahlen !"
414 VM PRINT TAB(36) "- Taste":GOSUB Taste:GOTO key7
415 dW0 END IF
416 wU IF Wa(Pla)<anz*100 THEN
417 zf2 COLOR 6:LOCATE 29,10
418 zL PRINT "Sie haben nicht genug Lübecker Handelsware für Ih
re Schiffe !"
419 bl PRINT TAB(15) "Sie brauchen 100 Kisten Salz pro Schiff !
- Taste"
420 nk GOSUB Taste:GOTO key7
421 jc0 END IF
422 k6 Gold(Pla)=Gold(Pla)-anz*He$(i)
423 VL Schiffe(Pla)=Schiffe(Pla)-anz
424 GO FSch%(Pla)=FSch%(Pla)+anz
425 jM FaSch%(Pla,i)=FaSch%(Pla,i)+anz
426 RR Wa(Pla)=Wa(Pla)-anz*100
427 he GOSUB LaCheck
428 Wx LOCATE 29,37:PRINT "O.K. !"
429 Vg GOSUB Head:GOTO key7
430 Hv Kon:
431 Q7 LINE (18,74)-(612,240),4,bf
432 UM COLOR 1,4:LOCATE 11,35:PRINT "Maklerbüro"
433 6M LINE (200,89)-(430,90),1,bf
434 2C LOCATE 14,10
435 tB PRINT "Kontor kaufen : Preise";
436 Vq PRINT TAB (45) "Kontor verkaufen : Werte"
437 fx PRINT TAB (10) "-----";TAB(45) "-----"
438 Nj FOR i=1 TO 5
439 eV LOCATE 16+i,30:PRINT KP$(i);"G."; TAB(66) KP$(i)-300;"G."
440 BG NEXT
441 6W LOCATE 23,23:COLOR 1:PRINT "0 - Verdammte Makler ! Ich geh
e !"
442 G5 Kon1:
443 Uk LINE (72,128)-(223,167),4,bf
444 oQ LINE (352,128)-(504,167),4,bf
445 DA COLOR 8
446 Vr FOR i=1 TO 5
447 CA LOCATE 16+i,9+Kon%(Pla,i)*35:PRINT i;"- ";Kon$(i):NEXT
448 v1 COLOR 1
449 H1 key9:
450 LR LINE (18,192)-(612,240),4,bf
451 QD GOSUB Taste
452 1a IF d$="0" THEN COLOR ,5:GOTO Haupt
453 lw IF ASC(d$)<49 OR ASC(d$)>53 THEN key9
454 1f i=VAL(d$)
455 EF IF Kon%(Pla,i)=1 THEN
456 s22 Kon%(Pla,i)=0:Gold(Pla)=Gold(Pla)+KP$(i)-300
457 KX1 LOCATE 25,15:Kon%(Pla)=Kon%(Pla)-1
458 Dk PRINT "Sie haben gerade Ihren Kontor in ";ST$(i);" verkau
ft !"
459 fh GOSUB Head:COLOR ,4
460 5W LOCATE 27,36:PRINT "- Taste":GOSUB Taste:GOTO Kon1
461 NGO END IF
462 dp LOCATE 25,10
463 6t PRINT "Sie wollen den Kontor in ";ST$(i);" kaufen ..."

```

```

464 4n IF Gold(Pla)<KP%(i) THEN
465 3M1 LOCATE 26,16
466 6P PRINT "Tja , wenn man nur genug Gold hätte ... - Taste"
467 cb GOSUB Taste:GOTO key9
468 UNO END IF
469 wP Gold(Pla)=Gold(Pla)-KP%(i):Kon%(Pla,i)=1
470 iR LOCATE 26,40:PRINT "O.K. Ist gekauft ! - Taste"
471 nJ Kont%(Pla)=Kont%(Pla)+1:GOSUB Head:COLOR ,4
472 7K GOSUB Taste:GOTO Kon1
473 sC Lag:
474 Cu LINE (18,74)-(612,240),5,bf
475 Lk COLOR 7,5:LOCATE 11,32:PRINT "Lübecker Lager"
476 7T LINE (210,89)-(390,90),1,bf
477 k1 LINE (225,163)-(407,196),14,bf
478 fu LINE (228,165)-(404,194),12,bf
479 yX COLOR 15,12
480 hA LOCATE 22,30:PRINT "0 - Weg"
481 Yh LOCATE 23,30:PRINT "1 - Lager kaufen"
482 Vf LOCATE 24,30:PRINT "2 - Lager verkaufen"
483 tt COLOR 7,5:LOCATE 13,23
484 Of PRINT "Sie besitzen Lübecker Lager "
485 BR LOCATE 15,15
486 jM PRINT "Ihr freier Lagerplatz beträgt : Quadratmet
er !"
487 Nf LOCATE 17,15
488 tP PRINT "Der aktuelle Preis für ein Lager :";Lapr;g$
489 Zt LOCATE 19,15
490 oO PRINT "Der Verkaufspreis beträgt :";Lapr-300;g$
491 zO lag1:
492 i1 GOSUB Head:COLOR 7,5
493 OM LINE (272,96)-(304,103),5,bf:LOCATE 13,35:PRINT La(Pla)
494 mJ GOSUB LaCheck
495 PY LINE (384,112)-(432,119),5,bf:LOCATE 15,49:PRINT Lap(Pla)
496 M1 key11:
497 Gt LINE (20,200)-(600,240),5,bf
498 By GOSUB Taste
499 WB IF d$="0" THEN Haupt
500 9o IF d$="1" THEN Lk
501 DY IF d$="2" THEN LV
502 ID GOTO key11
503 qB Lk:
504 dv LOCATE 26,15
505 iR INPUT "Wieviele neue Lager wollen Sie erwerben ";anz
506 nh anz=INT(anz)
507 pB IF anz<1 THEN key11
508 j3 IF anz*Lapr>Gold(Pla) THEN
509 uF2 COLOR 6:LOCATE 28,15
510 DA PRINT "Ihr Gold reicht nicht aus ,um das zu bezahlen !!!
"
511 kC PRINT:PRINT TAB(36) "- Taste":GOSUB Taste:GOTO key11
512 C50 END IF
513 7x Gold(Pla)=Gold(Pla)-anz*Lapr
514 Eh La(Pla)=La(Pla)+anz
515 GV LOCATE 28,15:PRINT "O.K. Lager gekauft !!! - Taste"
516 97 GOSUB Taste:GOTO lag1
517 33 LV:
518 g8 IF La(Pla)<1 THEN key11
519 B8 GOSUB LaCheck
520 tB LOCATE 26,15
521 xp INPUT "Wieviele Ihrer Lager wollen Sie verkaufen ";anz
522 3x anz=INT(anz)
523 5R IF anz<1 THEN key11
524 Tx IF anz>La(Pla) THEN
525 AV2 COLOR 6:LOCATE 28,15
526 QU PRINT "Hey - Ich kann rechnen ! Können Sie nicht lesen ?
"
527 oM PRINT TAB(15) "Sie haben nicht so viele Lager ! - Tast
e"
528 AI GOSUB Taste:GOTO key11
529 TMO END IF
530 mv IF Lap(Pla)-500*anz<0 THEN
531 EY2 LOCATE 28,15
532 vh PRINT "Sie können die Lager nicht verkaufen , weil Sie s
onst"
533 px PRINT TAB(15) "einige Waren nicht mehr unterbringen könn
en !!! - Taste"
534 GO GOSUB Taste:GOTO key11
535 ZSO END IF
536 o1 Gold(Pla)=Gold(Pla)+anz*(Lapr-300):La(Pla)=La(Pla)-anz
537 K1 LOCATE 28,15:PRINT "Lager verkauft !!! - Taste"
538 VT GOSUB Taste:GOTO lag1
539 Au Sol:
540 Gy LINE (18,74)-(612,240),5,bf
541 k3 LOCATE 11,36:PRINT "Soldaten"
542 Zy LINE (225,89)-(405,90),1,bf
543 41 LOCATE 13,5
544 T8 PRINT "Sie brauchen Soldaten und Kanonen zur Verteidigung
gegen Dänemark !"
545 QR LOCATE 17,5
546 m2 PRINT "Preise : für 1 Soldat";TAB(44);"für 1 Kanone"
547 Sf PRINT TAB(5)"Kaufpreis -";Sok;g$;TAB(44)"- ";Sok*10;g
$
548 vm PRINT TAB(5)"Sold pro Jahr - 1 ";g$;CHR$(8)
549 Dq PRINT
550 nJ PRINT TAB(5)"Ihre Soldaten Ihre Kanonen Anzuheue
rnde Sol. Bestellte Kan."
551 69 PRINT TAB(10)Sol(Pla);TAB(29)Kan(Pla)
552 mz LINE (225,179)-(407,212),14,bf
553 68 LINE (228,181)-(404,210),12,bf
554 Bk COLOR 15,12
555 4Z LOCATE 24,30:PRINT "0 - Weg"
556 PG LOCATE 25,30:PRINT "1 - Soldaten kaufen"
557 vv LOCATE 26,30:PRINT "2 - Kanonen kaufen"
558 o9 wh6:
559 o1 COLOR 11,5
560 Ir LINE (300,168)-(612,175),5,bf
561 71 LOCATE 22,48:PRINT SAuf(Pla);TAB(68)KAuf(Pla)
562 HB LINE (18,216)-(612,240),5,bf
563 6d GOSUB Taste:COLOR 1
564 ZE IF d$="0" THEN Haupt
565 KP IF d$<>"1" AND d$<>"2" THEN wh6
566 i5 IF d$="2" THEN Kanonen
567 WG Sold:
568 wz LOCATE 28,5
569 V7 INPUT "Wieviel Soldaten werben Sie an ";anz
570 pJ anz=INT(anz)
571 PP IF anz<1 THEN wh6
572 HN IF anz*Sok>Gold(Pla) THEN
573 H32 COLOR 6:LOCATE 29,5
574 It PRINT "Die Soldaten wollen alle Sold haben !"
575 fB PRINT TAB(5) "Sie können aber nicht genug Gold vorweisen
! - Taste"
576 6F GOSUB Taste:GOTO wh6
577 F80 END IF
578 Sh SAuf(Pla)=SAuf(Pla)+anz
579 SX Gold(Pla)=Gold(Pla)-Sok*anz
580 xF GOSUB Head
581 lf GOTO wh6
582 Ya Kanonen:
583 BE LOCATE 28,5
584 Pu INPUT "Wieviel Kanonen wollen Sie bestellen ";anz
585 4y anz=INT(anz)
586 ee IF anz<1 THEN wh6
587 nr IF (anz*Sok*10)>Gold(Pla) THEN
588 W12 COLOR 6:LOCATE 29,5
589 lC PRINT "Die Rüstungsfirmen wollen Bares sehen !"
590 uQ PRINT TAB(5) "Sie können aber nicht genug Gold vorweisen
! - Taste"
591 LU GOSUB Taste:GOTO wh6
592 UNO END IF
593 BA KAuf(Pla)=KAuf(Pla)+anz
594 MY Gold(Pla)=Gold(Pla)-(Sok*10*anz)
595 CU GOSUB Head
596 Ou GOTO wh6
597 gx Lad:
598 GC LINE (0,0)-(630,250),8,bf:COLOR 3,8
599 LF LINE (176,18)-(460,37),3,bf
600 G6 LINE (180,20)-(456,35),8,bf
601 FY LOCATE 4,25:PRINT "REEDER - DATEN LADEN / SPEICHERN"
602 4X DirLesen:
603 qE F=0:CLOSE
604 NC OPEN "R",#1,"Reeder.dir",34
605 cx FIELD #1,34 AS a$
606 Rm IF LOF(1)=0 THEN
607 oJ2 COLOR 6:LOCATE 22,11
608 Vm PRINT "ACHTUNG ! Ich lege Dateien für Reeder auf der Di
skette an."
609 Kk FOR i=1 TO 9
610 aO b$=" 0 OReeder by PAUL SCHOKAL YEAH !"
611 YU LSET a$=b$:PUT #1,i
612 J3 nam$="Reeder."+STR$(i)
613 J6 OPEN nam$ FOR OUTPUT AS 2

```

Listing 3. Fortsetzung

```

614 aW PRINT #2, "Hi Freaks !"
615 mX CLOSE 2
616 wO NEXT:CLOSE
617 9v LINE (20,168)-(612,240),8,bf
618 uw GOTO DirLesen
619 voO END IF
620 kQ LINE (20,58)-(612,158),3,b
621 3e LINE (24,60)-(608,156),3,b
622 tF COLOR 7,8:LOCATE 7,23
623 kt PRINT "Die gespeicherten Spiele von REEDER : "
624 eD LOCATE 9,10
625 iI PRINT "Spiel   Spieler   Jahre   Kommentar"
626 B9 COLOR 9
627 jP FOR i%=1 TO 9
628 4f GET #1,i%
629 en d$=SPACE$(9)
630 Hc b$=STR$(i%)+d$+MID$(a$,1,2)+d$+MID$(a$,3,2)+d$+MID$(a$,5,3
0)
631 GB LOCATE 10+i%,11:PRINT b$
632 me NEXT:CLOSE
633 l2 wh2:
634 i7 LINE (45,80)-(67,150),8,bf
635 Mz LINE (0,160)-(630,240),8,bf
636 SS COLOR 3,8:LOCATE 23,32
637 ux PRINT "0 - Weg hier !"
638 R1 PRINT TAB(32)"1 - Spiel laden"
639 iI PRINT TAB(32)"2 - Spiel speichern"
640 SI key19:
641 UH GOSUB Taste
642 pU IF d$="0" THEN Haupt
643 v1 IF d$<>"1" AND d$<>"2" THEN key19
644 NS i1=VAL(d$)
645 Dr LINE (20,160)-(612,240),8,bf
646 uH IF i1=1 THEN LOCATE 22,36:PRINT "Spiel laden"
647 6W IF i1=2 THEN LOCATE 22,34:PRINT "Spiel speichern"
648 UA PRINT:i%=1
649 X6 PRINT TAB(18) "Drücken Sie die Tasten 1-9 und dann SPACE
"
650 3C PRINT:PRINT TAB(33)"Raus mit - 0 - !!!"
651 qH key20:
652 dP COLOR 6:x%=i%*8
653 T2 LINE (45,74+x%)-(60,76+x%),6,bf
654 hR AREA (60,72+x%):AREA (67,75+x%)
655 Lf AREA (60,78+x%):AREAFILL
656 jW GOSUB Taste
657 1b IF d$="0" THEN wh2
658 e5 IF d$=" " AND i1=1 THEN
659 Pb2 OPEN "Reeder."+STR$(i%) FOR INPUT AS 1
660 EI INPUT #1,a$
661 iZ IF a$="Hi Freaks !" THEN
662 ev4 CLOSE:LOCATE 30,22
663 cv PRINT "Diese Datei ist unbenutzt !!! - Taste"
664 hP GOSUB Taste:LINE (20,232)-(612,240),8,bf
665 uu GOTO key20
666 g22 END IF
667 4M COLOR 6:LOCATE 30,34:PRINT "Bitte warten ..."
668 DG INPUT #1,Spieler,zug,Jahr,Pla
669 ZR FOR i%=1 TO Spieler
670 tM INPUT #1,Na$(i%)
671 kX INPUT #1,Ti$(i%)
672 Yz INPUT #1,alt$(i%),Hei(i%),In(i%)
673 Pg INPUT #1,Gold(i%),Schiffe(i%),FSch$(i%),we(i%)
674 ED INPUT #1,BauA(i%),SAuf(i%),KAuf(i%)
675 Tq INPUT #1,Sol(i%),Kan(i%),Wa(i%),La(i%),Kont$(i%)
676 es FOR i1=1 TO 5
677 UC INPUT #1,War$(i%,i1),FaSch$(i%,i1),Kon$(i%,i1):NEXT
678 73 FOR i1=1 TO Spieler:INPUT #1,ap$(i%,i1):NEXT
679 pQ FOR i1=0 TO zug:INPUT #1,curve$(i%,i1):NEXT
680 4D NEXT:CLOSE:GOTO Nae,jahr
681 voO END IF
682 6Y IF d$=" " AND i1=2 THEN
683 ln2 LOCATE 28,4
684 hL PRINT "Geben Sie einen Kommentar ein (max 30 Zeichen) un
d drücken Sie RETURN !"
685 wA COLOR 8,1:LOCATE 30,4:LINE (24,231)-(272,240),1,bf
686 li LINE INPUT ">";b$:IF b$="" THEN wh2
687 Hq IF LEN(b$)>30 THEN b$=LEFT$(b$,30)
688 Pn b$=RIGHT$(STR$(Spieler),2)+RIGHT$(STR$(zug),2)+b$
689 Pp b$=b$+SPACE$(34-LEN(b$))
690 la OPEN "R",#1,"Reeder.dir",34
691 Ru FIELD #1,34 AS a$:LSET a$=b$:PUT #1,i%:CLOSE
692 Ab a$="Reeder."+STR$(i%)
693 NR OPEN a$ FOR OUTPUT AS 1
694 F2 PRINT #1,"O.K."
695 33 PRINT #1,Spieler,zug,Jahr,Pla
696 Os FOR i%=1 TO Spieler
697 j9 PRINT #1,Na$(i%)
698 ak PRINT #1,Ti$(i%)
699 Om PRINT #1,alt$(i%),Hei(i%),In(i%)
700 FT PRINT #1,Gold(i%),Schiffe(i%),FSch$(i%),we(i%)
701 40 PRINT #1,BauA(i%),SAuf(i%),KAuf(i%)
702 Jd PRINT #1,Sol(i%),Kan(i%),Wa(i%),La(i%),Kont$(i%)
703 5J FOR i1=1 TO 5
704 Kz PRINT #1,War$(i%,i1),FaSch$(i%,i1),Kon$(i%,i1):NEXT
705 xq FOR i1=1 TO Spieler:PRINT #1,ap$(i%,i1):NEXT
706 cj FOR i1=0 TO zug:PRINT #1,curve$(i%,i1):NEXT
707 zr NEXT:CLOSE
708 M0 GOTO DirLesen
709 NGO END IF
710 nB IF ASC(d$)<49 OR ASC(d$)>57 THEN key20
711 8q LINE (45,72+x%)-(67,78+x%),8,bf
712 uG i%=VAL(d$):GOTO key20
713 yK Boe:
714 L0 GOSUB schwarz
715 gx LINE (18,0)-(612,240),1,bf:LINE (0,0)-(630,70),1,bf
716 Cb LOCATE 1,33:COLOR 8,12:PRINT " B Ö R S E "
717 4X FOR i%=0 TO 16 STEP 2:SCROLL (220,i%)-(410,30),0,1:NEXT
718 UA a$="Inhaber : Wert 1% Anteil : noch käuflich : I
hr Besitz : "
719 4E COLOR 6,0:LOCATE 7,8:PRINT a$
720 MW LINE (225,115)-(407,156),14,bf
721 HP LINE (228,117)-(404,154),12,bf
722 tS COLOR 15,12
723 sO LOCATE 16,30:PRINT "0 - Weg"
724 yz LOCATE 17,30:PRINT "1 - Aktien kaufen"
725 u5 LOCATE 18,30:PRINT "2 - Aktien verkaufen"
726 fY LOCATE 19,30:PRINT "3 - Indekskurven sehen"
727 hB GOSUB farben
728 25 boe1:
729 17 LINE (200,31)-(430,40),0,bf
730 zH COLOR 7,0:LOCATE 5,27:PRINT "Sie haben";Gold(Pla);g$
731 ZR FOR i%=1 TO Spieler
732 aZ COLOR 10+i%:LOCATE i%+8,7:GOSUB Index1
733 aH PRINT i%;Na$(i%);TAB(26) INT(w);"G.";
734 WT x=100:FOR i1=1 TO Spieler:x=x-ap$(i1,i%):NEXT
735 ad PRINT TAB(47) x;"% ";
736 e1 PRINT TAB(64) ap$(Pla,i%);"% "
737 y3 NEXT
738 m8 wh7:
739 j6 LINE (18,157)-(612,240),0,bf
740 5s GOSUB Taste
741 qX IF VAL(d$)=0 AND d$<>"0" THEN d$="4"
742 67 ON VAL(d$)+1 GOTO Haupt,AKau,AVkau,Kurse
743 RM GOTO wh7
744 ig AKau:
745 Og COLOR 12,0
746 Q3 IF Spieler>1 THEN
747 5H2 LOCATE 21,23
748 P1 PRINT "Wessen Aktien wollen Sie kaufen ?"
749 pX GOSUB Taste:IF ASC(d$)<48 OR ASC(d$)>48+Spieler THEN A
Kau
750 GV IF d$="0" THEN wh7 ELSE i%=VAL(d$)
751 iRO ELSE
752 yy2 i%=1
753 5yO END IF
754 Qn d$="Sie wollen also Aktien von "+Na$(i%)+ " kaufen ."
755 7N LOCATE 23,1+(79-LEN(d$))/2:PRINT d$
756 LL x=0:FOR i1=1 TO Spieler:x=x+ap$(i1,i%):NEXT
757 ix IF x=100 THEN
758 qB2 LOCATE 25,19
759 Oa PRINT "Diese Aktien sind alle in festen Händen !!!"
760 lQ PRINT:PRINT TAB(36) "- Taste":GOSUB Taste:GOTO wh7
761 D6O END IF
762 Ob LOCATE 25,22:INPUT "Wieviel Prozent Anteil kaufen Sie ";anz
z
763 aG anz=INT(anz):IF anz<1 THEN wh7
764 jg IF anz>100-x THEN
765 4Q2 LOCATE 27,18
766 Pn PRINT "So viel Aktien sind nicht mehr zu bekommen !"
767 BX PRINT:PRINT TAB(36) "- Taste":GOSUB Taste:GOTO wh7
768 KDO END IF
769 d1 GOSUB Index1
770 4V IF Gold(Pla)<anz*w THEN
771 6R2 LOCATE 27,26

```

```

772 99 PRINT "Spielen Sie sich nicht so auf !":PRINT
773 2o PRINT TAB(20) "So viel Geld haben Sie gar nicht ! - Taste
e"
774 0T GOSUB Taste:GOTO wh7
775 RKO END IF
776 bZ Gold(Pla)=Gold(Pla)-INT(anz*w)
777 nh ap%(Pla,i%)=ap%(Pla,i%)+anz
778 0i GOTO boel
779 Oz AVkau:
780 xF COLOR 12,0
781 zc IF Spieler>1 THEN
782 bm2 LOCATE 21,22
783 kd PRINT "Von wem wollen Sie Aktien verkaufen ?"
784 06 GOSUB Taste:IF ASC(d$)<48 OR ASC(d$)>48+Spieler THEN A
Kau
785 p4 IF d$="0" THEN wh7 ELSE i%=VAL(d$)
786 H00 ELSE
787 XX2 i%=1
788 eX0 END IF
789 ss IF ap%(Pla,i%)=0 THEN
790 nk2 x=LEN(Na$(i%))/2:COLOR 6:LOCATE 23,21-x
791 kf PRINT "Von "Na$(i%);" haben Sie keine Aktien ! - Taste"
792 gl GOSUB Taste:GOTO wh7
793 je0 END IF
794 LP d$="Sie wollen also Aktien von "Na$(i%)+ " verkaufen ."
795 ef LOCATE 23,(80-LEN(d$))/2:PRINT d$
796 Mf LOCATE 25,17
797 yh INPUT "Wieviele Prozent Anteil wollen Sie abstoßen ";anz
798 9p anz=INT(anz):IF anz<1 THEN wh7
799 T9 IF anz>ap%(Pla,i%) THEN
800 w02 COLOR 6:LOCATE 27,19
801 QT PRINT "So viel Anteil haben sie doch nicht ! - Taste"
802 qv GOSUB Taste:GOTO wh7
803 tm0 END IF
804 m3 IF i%=Pla AND ap%(Pla,Pla)-anz<51 THEN
805 vf2 COLOR 6:LOCATE 27,5
806 kz PRINT "Sind Sie verrückt ??? Wollen Sie etwa Ihre";
807 Zv PRINT " Aktienmehrheit verlieren ?"
808 JK PRINT TAB (30) "Is nich !!! - Taste"
809 x2 GOSUB Taste:GOTO wh7
810 0t0 END IF
811 JR GOSUB Index1
812 WS ap%(Pla,i%)=ap%(Pla,i%)-anz
813 06 Gold(Pla)=Gold(Pla)+INT(anz*w)
814 aI GOTO boel
815 9q Kurse:
816 RH LINE (0,0)-(630,73),1,bf
817 IP LINE (20,74)-(610,240),15,bf
818 p2 FOR i%=30 TO 590 STEP 80:LINE (i%,74)-(i%,240),10:NEXT
819 K1 FOR i%=80 TO 240 STEP 10:LINE (21,i%)-(609,i%),10:NEXT
820 Xx FOR i%=3 TO 73 STEP 10:LOCATE 10,i%:PRINT i%-3:NEXT
821 HJ COLOR 12,0:LOCATE 9,3:PRINT "X-Achse : Zeit in Jahren
";
822 Nx PRINT "Y-Achse : Indexwert (1-2000) 0 : zurück"
823 RV FOR i1=1 TO Spieler
824 cp x=1+INT((79-LEN(Na$(i1)))/2)
825 7m LOCATE 3+i1,x:COLOR 5+i1,0:PRINT Na$(i1)
826 rn FOR i%=1 TO zug
827 zc LINE ((i%-1)*8+30, kurve%(i1,i%-1)-(i%*8+30, kurve%(i1,i%))
,i1+5
828 2W NEXT:NEXT
829 sF GOSUB Taste:GOTO Boe
830 fX OK:
831 7S GOSUB schwarz:COLOR 4,1:CLS
832 xh LINE (250,44)-(375,59),2,bf
833 03 AREA (251,59):AREA (253,57):AREA (372,47):AREA (375,44)
834 xz AREA (375,59):AREAFILL:LINE (253,46)-(372,57),3,bf
835 pm COLOR 5,3:LOCATE 7,35:PRINT "Ereignis :":COLOR 3,1
836 PJ PALETTE 2,1,.75,0:PALETTE 3,.6,.45,0
837 Cu PALETTE 4,.4,.2,0:PALETTE 5,1,.75,.2
838 0C x%=RND*11+1:x1=x%*4*FSch%(Pla)
839 tW we(Pla)=we(Pla)-x1
840 B1 LOCATE 10,3:PRINT "Es war Sturmstärke";x%;
841 yy PRINT ". Der Gesamtwert Ihrer Schiffe sank um";x1;" ";g$
842 0L GOSUB LaCheck
843 qL IF Lap(Pla)<0 THEN
844 v02 LOCATE 12,19
845 4P PRINT "Ihre Lager können nicht alle Waren fassen !"
846 lx LOCATE 14,21
847 0F PRINT "Sie verlieren ";-Lap(Pla); " Wareneinheiten !!!"
848 r5 GOSUB LaCheck1
849 ho PRINT:PRINT TAB(36) "- Taste":GOSUB Taste
850 eX0 END IF
851 cz IF we(Pla)>0 THEN GOSUB SWert
852 3F IF FSch%(Pla)>0 AND SW<200 THEN
853 kz2 x=INT(RND*FSch%(Pla)):LOCATE 16,11
854 iD PRINT "Da Ihre Schiffe stark beschädigt sind , sanken";x
;"von ihnen."
855 8v we(Pla)=we(Pla)-x*SW
856 ot FSch%(Pla)=FSch%(Pla)-x
857 h6 PRINT:PRINT TAB(36) "- Taste":GOSUB Taste
858 mf0 END IF
859 61 x=0:IF Gold(Pla)<0 THEN x=Gold(Pla):Gold(Pla)=0
860 YQ Gold(Pla)=Gold(Pla)-Sol(Pla)
861 fW IF Gold(Pla)<0 THEN
862 JX2 LOCATE 18,15:PRINT "Sie können nicht allen Soldaten den
Sold bezahlen !"
863 k2 PRINT:PRINT TAB (21) "Deshalb desertieren ";-Gold(Pla); " v
on Ihnen !"
864 im Sol(Pla)=Sol(Pla)+Gold(Pla):Gold(Pla)=0
865 je PRINT:PRINT TAB (36) "- Taste":GOSUB Taste
866 un0 END IF
867 T7 Gold(Pla)=Gold(Pla)+x
868 S6 LINE (18,88)-(612,240),1,bf
869 25 x%=RND*100
870 u5 IF x%<6 AND Hei(Pla)=0 AND zug>10 THEN Hei ELSE IF x%<6
THEN weiter
871 tk IF x%<16 AND zug>10 THEN kri ELSE IF x%<16 THEN weiter
872 8q IF x%<26 THEN wai
873 yI IF x%<37 THEN Dom
874 mE IF x%<50 THEN schmuggel
875 r8 IF x%<60 AND La(Pla)>0 THEN Brand ELSE IF x%<60 THEN we
iter
876 x5 IF x%<70 THEN Audienz
877 Fo IF x%<80 THEN Bergen
878 Xq IF x%<90 THEN raub
879 zy GOTO krank
880 Pk weiter:
881 Zb FOR i%=1 TO 5
882 R5 IF FaSch%(Pla,i%)>0 THEN
883 3x2 x%=(50+RND*100)*(FaSch%(Pla,i%)*1.2)
884 QA War%(Pla,i%)=War%(Pla,i%)+x%:FaSch%(Pla,i%)=0
885 D60 END IF
886 NS NEXT
887 eQ Schiffe(Pla)=Schiffe(Pla)+FSch%(Pla):FSch%(Pla)=0
888 Mh we(Pla)=we(Pla)+BauA(Pla)*400
889 26 Schiffe(Pla)=Schiffe(Pla)+BauA(Pla):BauA(Pla)=0
890 wg Sol(Pla)=Sol(Pla)+SAuf(Pla):SAuf(Pla)=0
891 JD Kan(Pla)=Kan(Pla)+KAuf(Pla):KAuf(Pla)=0
892 Bk Weg:
893 YJ Pla=Pla+1:IF Pla<Spieler+1 THEN Naejahr
894 mo zug=zug+1:Jahr=Jahr+1
895 RU FOR i%=1 TO Spieler:Pla=i%:GOSUB Index:NEXT
896 N4 Pla=1:GOTO Naejahr
897 K schwarz:FOR i%=0 TO 20:PALETTE i%,0,0,0:NEXT:RETURN
898 xt farben:
899 G8 FOR i%=0 TO 2:PALETTE i%,0,0,0:NEXT
900 Wa PALETTE 3,1,1,1
901 Su PALETTE 4,.56,.3,.1
902 9g PALETTE 5,.5,.2,0
903 eo PALETTE 6,1,0,0
904 vC PALETTE 7,.7,.7,0
905 u1 PALETTE 8,0,0,1
906 z2 IF x<>13.1 THEN PALETTE 9,0,1,0 ELSE PALETTE 9,0,.06,0
907 rg x=-8:FOR i%=10 TO 15:PALETTE i%,x,x,x-x-1:NEXT
908 Oc RETURN
909 8g Taste:
910 1M WHILE INKEY$<>"":WEND:d$=""
911 So WHILE d$="" :d$=INKEY$:WEND
912 AW d$=UCASE$(d$):RETURN
913 Ym Head:
914 lf LINE (7,34)-(623,52),4,bf
915 mu a$="Jahr "+STR$(Jahr)+" Gold : "+STR$(Gold(Pla))
916 rx i%=Pla:GOSUB Index1
917 Cq a$=a$+" Wert 1 % : "+STR$(INT(w))
918 tj a$=a$+" Besitz : "+STR$(ap%(Pla,Pla))+" %"
919 7o x%=(80-LEN(a$))/2
920 4z COLOR 1,4:LOCATE 6,x%:PRINT a$:COLOR ,5
921 Dp RETURN
922 DM SWert:
923 ZJ IF we(Pla)>0 THEN
924 xN1 x%=Schiffe(Pla)+FSch%(Pla)

```

Listing 3. Fortsetzung

```

925 IQ we(Pla)=INT (we(Pla)/x%)*x%
926 Tw SW=we(Pla)/x%
927 Jv RETURN
928 ZIO ELSE
929 Kh1 SW=0:RETURN
930 wp0 END IF
931 7t Init:
932 KQ DIM bild%(10280)
933 MU DIM kurve%(4,70)
934 bt DIM ap%(4,4)
935 UF DIM ST$(5),War$(5),PR$(5),War%(4,5),He$(5),KPr$(5)
936 tq DIM Na$(4),alt%(4),Ti$(4),Hei(4),In(4)
937 1W DIM Schiffe(4),FaSch%(4,5),FSch%(4)
938 Ed DIM BauA(4),SAuf(4),KAuf(4)
939 5U DIM Kont%(4),Kon%(4,5),Kon$(5)
940 BY DIM Gold(4),Ges(4),Sol(4),Kan(4),we(4)
941 nu DIM Wa(4),La(4),Lap(4)
942 DT DIM Ka1$(4),Ka2$(4,5),Ka3$(4,5),Ka4$(4,5)
943 bu REM DIM xr%(9),yr%(9) ??????????
944 ac RETURN
945 g2 Hei:
946 Na LOCATE 12,15
947 us PRINT "Wollen Sie eine reiche schöne Frau heiraten ??? (J/N)"
948 k2 key12:
949 SF GOSUB Taste
950 vU IF d$="J" THEN Heirat
951 ha IF d$="N" THEN weiter ELSE key12
952 ZP Heirat:
953 3z x=3+INT(RND*7):x1=x*1000+INT(RND*990)+zug*300
954 pz LOCATE 14,20:COLOR 7:PALETTE 7,1,0,0
955 S1 PRINT "Das brachte Ihnen";x1;"Goldstücke ein !!!"
956 M1 Gold(Pla)=Gold(Pla)+x1:Hei(Pla)=Jahr:In(Pla)=In(Pla)+3
957 ap LOCATE 16,13:COLOR 3
958 7b PRINT "Außerdem sparen Sie natürlich in Zukunft die Putzfrau !!!"
959 cP GOSUB Taste
960 71 GOTO weiter
961 GJ wai:
962 dq LOCATE 12,15
963 DE PRINT "Geben Sie ein Fest für arme Waisenkinder ??? (J/N)"
964 6K key13:
965 iV GOSUB Taste
966 ui IF d$="J" THEN WaFest
967 Vi IF d$="N" THEN
968 t81 LOCATE 14,15
969 bg PRINT "Sie werden mit faulen Tomaten beworfen ! - Taste"
970 U5 GOSUB Taste:GOTO weiter
971 bU0 END IF
972 4r GOTO key13
973 di WaFest:
974 Sx Gold(Pla)=Gold(Pla)-(200+zug*50)
975 br In(Pla)=In(Pla)+1
976 CC LOCATE 14,7
977 ws PRINT "Die Waisenkinder freuen sich ! Sie kostet der Spaß";(200+zug*20);"Goldstücke !"
978 32 LOCATE 16,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
979 qM Dom:
980 yC LOCATE 12,16
981 rx PRINT "Spenden Sie 500 Goldstücke zum Dombau ??? (J/N)"
982 Ue key14:
983 On GOSUB Taste
984 Wf IF d$="J" THEN
985 rY1 Gold(Pla)=Gold(Pla)-500:In(Pla)=In(Pla)+1
986 Cx LOCATE 14,21:PRINT "Die Kirchenorganisation bedankt sich !"
987 CB LOCATE 16,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
988 s10 END IF
989 r4 IF d$="N" THEN
990 Wf1 IF Gold(Pla)<2000 THEN
991 1B2 LOCATE 14,10
992 9K PRINT "Bei Ihrer finanziellen Lage (!) ist das verständlich... - Taste"
993 rS GOSUB Taste:GOTO weiter
994 yr1 END IF
995 WM LOCATE 14,31:PRINT "!!! GEIZHALS !!!":GOSUB Taste
996 hb GOTO weiter
997 1u0 END IF
998 aJ GOTO key14
999 wE schmuggel:
1000 OL LOCATE 12,6
1001 bA PRINT "Ihnen wird ein Schmuggelgeschäft angeboten. Nehmen Sie an ??? (J/N)"
1002 u0 key15:
1003 K7 GOSUB Taste
1004 OI IF d$="J" THEN SCHM
1005 4C IF d$="N" THEN weiter
1006 oT GOTO key15
1007 JO SCHM:
1008 Uy x=INT(RND*100)
1009 ou IF x>59 THEN
1010 EW1 In(Pla)=In(Pla)-1
1011 eK a$=" Sie wurden erwischt und kommen 1 Jahr ins Gefängnis !!!"
1012 qo GOSUB Gef:GOTO weiter
1013 HAO END IF
1014 g3 x%=RND*100*50+zug*200
1015 Ss LOCATE 14,25:PRINT "Die Aktion verlief erfolgreich !"
1016 p6 LOCATE 16,15
1017 v1 PRINT "Ihnen werden zum Schluß";x%;"Goldstücke ausgehändigt."
1018 9s Gold(Pla)=Gold(Pla)+x%
1019 st LOCATE 18,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
1020 Qd Brand:
1021 dr LOCATE 12,16
1022 1A PRINT "Mitten in der Nacht brennt eines Ihrer Lager ab."
1023 PF LOCATE 14,20:PRINT "Und mit ihm die dort gelagerten Waren !"
1024 f8 La(Pla)=La(Pla)-1:GOSUB LaCheck1
1025 on LOCATE 16,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
1026 K4 LaCheck:
1027 O6 x=Wa(Pla):FOR i=1 TO 5:x=x+War%(Pla,i):NEXT
1028 UB Lap(Pla)=La(Pla)*500-x:RETURN
1029 tu LaCheck1:
1030 MO GOSUB LaCheck:IF Lap(Pla)>-1 THEN RETURN
1031 KJ x=-Lap(Pla)
1032 MG IF Wa(Pla)>0 THEN
1033 IO2 IF Wa(Pla)>x THEN Wa(Pla)=Wa(Pla)-x:GOTO LaCheck1
1034 Y4 x=x-Wa(Pla):Wa(Pla)=0
1035 dW0 END IF
1036 1N FOR i=1 TO 5
1037 QW IF War%(Pla,i)>0 THEN
1038 de2 IF War%(Pla,i)>x THEN War%(Pla,i)=War%(Pla,i)-x:GOTO LaCheck1
1039 Uo x=x-War%(Pla,i):War%(Pla,i)=0
1040 ib0 END IF
1041 sx NEXT
1042 gG GOTO LaCheck1
1043 j5 Bergen:
1044 r2 LOCATE 12,13
1045 xt PRINT "Versuchen Sie ein gesunkenes Schiff zu bergen ??? (J/N)"
1046 ik key16:
1047 2p GOSUB Taste
1048 yE IF d$="J" THEN try
1049 mu IF d$="N" THEN weiter
1050 cD GOTO key16
1051 TO try:
1052 Cg x=INT(RND*100)
1053 w2 IF x<51 THEN
1054 WH2 x=50+zug*INT(RND*50):Gold(Pla)=Gold(Pla)-x
1055 ok LOCATE 14,25:PRINT "Es hat leider nicht geklappt !!!"
1056 Tk LOCATE 16,15
1057 8R PRINT "Die Bergungskosten belaufen sich auf";x;"Goldstücke"
1058 VW LOCATE 18,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
1059 1u0 END IF
1060 cN x=50+zug*INT(RND*50):Gold(Pla)=Gold(Pla)-x
1061 GO Schiffe(Pla)=Schiffe(Pla)+1:we(Pla)=we(Pla)+400
1062 SO LOCATE 14,3
1063 1A PRINT "Geschafft ! Sie konnten das Schiff bergen und erneut fahrtüchtig machen."
1064 bs LOCATE 16,15
1065 p3 PRINT "Jedoch kostete die Bergung";x;"Goldstücke ! - Taste"
1066 2d GOSUB Taste:GOTO weiter
1067 KS raub:
1068 ZB x=INT(RND*5+1):a$="-"
1069 Wm IF x=1 AND Wa(Pla)>0 THEN Wa(Pla)=0:a$="Lübecker Salzkisten"
1070 Ho IF x=2 AND Ho(Pla)>0 THEN Ho(Pla)=0:a$="Malmöer Honigpötte"
1071 JU IF x=3 AND Le(Pla)>0 THEN Le(Pla)=0:a$="Bergener Fische"
1072 Ez IF x=4 AND Hol(Pla)>0 THEN Hol(Pla)=0:a$="Tonnen Getreide"

```

```

1073 bh IF x=5 AND Pe(Pla)>0 THEN Pe(Pla)=0:a$="Nowgoroder Pelze"
1074 XS IF x=6 AND Ei(Pla)>0 THEN Ei(Pla)=0:a$="Rostocker Eisener
ze"
1075 pN IF a$="-" THEN weiter
1076 Hz LOCATE 12,30:PRINT "Sie wurden beraubt !!! "
1077 Ds LOCATE 14,15:PRINT "Man hat Ihnen Ihre gesamten ";a$;
1078 BG PRINT " geraubt ! ":LOCATE 16,3
1079 PE PRINT "Es tut mir leid , aber es gibt nun einmal so viele
schlechte Menschen !!!"
1080 3F PRINT:PRINT TAB (36)"- Taste":GOSUB Taste:GOTO weiter
1081 7y Audienz:
1082 oo LOCATE 12,9
1083 Ca PRINT "Geben Sie eine Audienz für den Herzog von Niederbay
ern ? (J/N)"
1084 WQ key18:
1085 eR GOSUB Taste
1086 ux IF d$="N" THEN In(Pla)=In(Pla)-2:GOTO weiter
1087 qd IF d$<>"J" THEN key18
1088 jw LOCATE 14,13
1089 Eb PRINT "Wieviel ";g$;" geben Sie für den Empfang aus ";
1090 yP INPUT anz
1091 oO anz=INT(anz):IF anz<1 THEN weiter
1092 CR IF anz>Gold(Pla) THEN
1093 BB2 LOCATE 16,5
1094 SW PRINT "Bleiben Sie doch einmal ernst ! So viel Gold habe
n Sie nicht !!! - Taste"
1095 V6 GOSUB Taste:GOTO weiter
1096 oV0 END IF
1097 D4 Gold(Pla)=Gold(Pla)-anz
1098 U5 IF anz>zug*90+INT(RND*100) THEN
1099 ev2 In(Pla)=In(Pla)+2
1100 BS LOCATE 16,15
1101 fr PRINT "Der Herzog hat sich königlich amüsiert !!! - Tas
te"
1102 N60 ELSE
1103 j12 In(Pla)=In(Pla)-1
1104 QS LOCATE 16,7
1105 j9 PRINT "Der Herzog ist verärgert ! Sie haben ihm zu wenig
geboten - Taste"
1106 mf0 END IF
1107 hI GOSUB Taste:GOTO weiter
1108 4C kri:
1109 Vg IF Sol(Pla)<Kan(Pla) THEN
1110 l12 pow1=Sol(Pla)+Sol(Pla)*7
1111 WFO ELSE
1112 ff2 pow1=Sol(Pla)+Kan(Pla)*7
1113 tm0 END IF
1114 RK pow2=zug*150+INT(RND(zug*100))
1115 w1 IF pow2<pow1 THEN
1116 OB2 LOCATE 12,22
1117 PN PRINT "Der dänische König erwog einen Angriff."
1118 7I LOCATE 14,11
1119 wA PRINT "Ihre Streitmacht hat ihn aber noch einmal davon a
bgehalten !"
1120 lg PRINT:PRINT TAB(36)"- Taste":GOSUB Taste:GOTO weiter
1121 gP0 ELSE
1122 Ko2 x=INT(RND*100)
1123 nz IF x<pow1/pow2*100 THEN
1124 OB4 LOCATE 12,10
1125 zh PRINT "Der dänische König Waldemar IV hat es gewagt ,
Sie anzugreifen,"
1126 Uk LOCATE 14,16
1127 Q1 PRINT "doch ihre Mannen schlugen tapfer sein Heer zurd
ck !"
1128 FE x%=RND*(Sol(Pla)/2):In(Pla)=In(Pla)+2
1129 Xq Sol(Pla)=Sol(Pla)-x%:LOCATE 16,11
1130 LB PRINT "Sie haben allerdings Verluste von ";x%;" Soldat
en zu beklagen !"
1131 gh LOCATE 18,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
1132 ra2 ELSE
1133 IT4 LOCATE 12,13
1134 ZU PRINT "Lübeck wurde von Waldemar IV von Dänemark angeg
riffen !!!"
1135 X1 LOCATE 14,14
1136 9g PRINT "Trotz tapferen Kämpfens verlor Lübeck diese Sch
lacht !!!"
1137 9P x=INT(RND*(Sol(Pla))):x1=INT(RND*(Kan(Pla)))
1138 wG LOCATE 16,18
1139 4b PRINT "Sie verloren ";x%;" Soldaten und ";x1%;" Kanonen
!"
1140 iU Sol(Pla)=Sol(Pla)-x:Kan(Pla)=Kan(Pla)-x1
1141 79 i1=INT((RND+1)*La(Pla)/2):La(Pla)=La(Pla)-i1
1142 vY GOSUB LaCheck1:LOCATE 18,8
1143 2m PRINT "Außerdem brannten ";i1;" Ihrer Lager nieder , -
mitsamt den Waren !"
1144 JD LOCATE 20,36:PRINT "- Taste":GOSUB Taste:GOTO weiter
1145 PI2 END IF
1146 QJO END IF
1147 Tt krank:
1148 Zb LOCATE 12,26:PRINT "Sie sind schwer erkrankt !!!"
1149 eb LOCATE 14,5:x=2000+(INT(RND*50)+50)*zug
1150 Bj PRINT "Zahlen Sie";x;" ";g$;" für eine ärztliche Behandlu
ng ??? (J/N)"
1151 VT key17:
1152 jW GOSUB Taste
1153 FO IF d$="J" THEN
1154 Y81 LOCATE 16,20:PRINT "Sie haben die Krankheit überstanden !
!!"
1155 Zf PRINT:PRINT TAB(35)"- Taste":GOSUB Taste
1156 lL Gold(Pla)=Gold(Pla)-x:GOTO weiter
1157 bUO END IF
1158 H7 IF d$<>"N" THEN key17
1159 vP x=INT(RND*100)
1160 BM IF x<41 THEN Tot
1161 O0 LOCATE 16,20:PRINT "Sie sind mit viel Glück wieder genesen
!"
1162 88 LOCATE 18,35:PRINT "- Taste":GOSUB Taste:GOTO weiter
1163 mf Tot:
1164 Ph GOSUB schwarz:COLOR ,1:CLS
1165 6J PATTERN 4228:x=0
1166 N6 FOR i%=100 TO 0 STEP -1
1167 gW LINE (0,i%-x)-(630,i%-x),15:x=x+.2:i%=i%-x:NEXT
1168 Ac CIRCLE (550,50),30,10:PAINT (550,50),10
1169 lO COLOR 0:FOR i%=0 TO 550 STEP 50
1170 eZ x=50+RND*20:x1=RND+20+RND*20
1171 hf AREA (i%,100):AREA (i%+x,100):AREA (i%+x1,100)-(15+RND*10))
1172 D6 AREA:PAINT:NEXT
1173 bo LINE (0,100)-(630,250),9,bf:x%=0
1174 S5 FOR i%=-330 TO 930 STEP 20:LINE (x%,100)-(i%,250),8:x%=x%+
10
1175 27 NEXT
1176 qJ AREA (195,250):AREA(465,250):AREA (415,130):AREA(245,130)
1177 RE COLOR 15:AREA:PAINT
1178 VO AREA (210,245):AREA(450,245):AREA (405,135):AREA(255,135)
1179 7x COLOR 1:AREA:PAINT
1180 JP LINE (370,160)-(290,164),10,bf
1181 j1 AREA (327,143):AREA (333,143):AREA (337,235):AREA (323,235
)
1182 Cu COLOR 10:AREA:PAINT
1183 LV LINE (245,50)-(415,130),15,bf:LINE (247,51)-(413,129),1,bf
1184 d8 LOCATE 9,37:PRINT "Hier ruht :"
1185 fU LOCATE 11,38:PRINT Na$(Pla)
1186 8R LOCATE 13,36:PRINT "* 13.1. 1301"
1187 ui LOCATE 15,36:PRINT "+ 26.2.";STR$(Jahr)
1188 JJ CIRCLE (390,132),7,1:PAINT (390,132),1
1189 5p PSET (387,125),6:PSET (391,125),6
1190 Ln x=13.1:GOSUB farben
1191 wF PALETTE 8,0,.2,0:PALETTE 9,0,.06,0:PATTERN -1
1192 NA GOSUB Taste
1193 mQ FOR i=1 TO 16
1194 kd PALETTE 10,1-1*.0625,1-1*.0625,1-1*.0625
1195 ya PALETTE 8,0,.2-.0125*1,0:PALETTE 6,1-1*.0625,0,0
1196 vt PALETTE 15,.2-.0125*1,.2-.0125*1,.2-.0125*1
1197 bh PALETTE 9,0,.06-.00375*1,0
1198 PU NEXT
1199 Zv Na$(Pla)="" : Gold(Pla)=0 : La(Pla)=0 : Schiffe(Pla)=0 : In(Pla)=0
1200 kb FOR i%=1 TO 5 : Kon%(Pla,i%)=0 : War%(Pla,i%)=0 : NEXT
1201 ml x=0 : FOR i%=1 TO Spieler : IF Na$(i%) > "" THEN x=1 : NEXT
1202 oL IF x=0 THEN SYSTEM
1203 2w GOTO weiter
1204 Is Sieg:
1205 D5 CHAIN "Sieg",,ALL
1206 aw Urkunde:
1207 3K GOSUB schwarz:COLOR ,0:CLS
1208 EJ COLOR ,4
1209 4Y FOR i%=20 TO 230
1210 Fw PATTERN INT(RND*65500&)
1211 25 LINE (100,i%)-(500,i%),5
1212 Id NEXT:PATTERN -1:COLOR ,0
1213 bP FOR i%=50 TO 550 STEP 3
1214 BT SCROLL (i%,20)-(i%+3,230),0,RND*2-1

```

Listing 3. Fortsetzung

```

1215 gl NEXT
1216 lr FOR i%=20 TO 230 STEP 3
1217 2A SCROLL (50,i%)-(550,i%+3),RND*4-2,0
1218 jo NEXT
1219 9e COLOR 1:LOCATE 1,1:PRINT "URKUNDE"
1220 69 FOR i%=0 TO 7
1221 yK FOR x%=0 TO 53
1222 Pe IF POINT (x%,i%)=1 THEN LINE (x%*5+160,i%*2+45)-(x%*5+163,
i%*2+46),0,bf
1223 ot NEXT
1224 pu NEXT
1225 sE COLOR 0,5
1226 Iy LOCATE 11,25:PRINT "Der Bürger ";Na$(Pla);" wurde"
1227 Yb LOCATE 13,25:PRINT "im Jahre";Jahr;"als ";Ti$(Pla)
1228 G7 LOCATE 15,25:PRINT "der Hansestadt L Ü B E C K"
1229 Lm LOCATE 17,25:PRINT "im Stadtgeschichtsbuch mit"
1230 Mz LOCATE 19,25:PRINT "großen Ehren eingetragen ."
1231 bl LOCATE 22,39:PRINT "gezeichnet : "
1232 oQ LOCATE 24,30:PRINT "Oberbürgermeister Klausen"
1233 BO LOCATE 26,30:PRINT "Lübeck den 2. August";Jahr
1234 JO AREA (495,213):AREA (545,250)
1235 NS AREA (538,230):AREA (570,240):AREA (510,208)
1236 QO COLOR 9:AREAFILL
1237 xH CIRCLE (500,210),50,8
1238 ZF CIRCLE (500,210),50,9:PAINT (500,210),9
1239 P2 CIRCLE (500,210),40,8:PAINT (500,210),8
1240 gr CIRCLE (500,210),40,0
1241 K1 CIRCLE (500,210),35,9
1242 az LINE (480,210)-(520,220),9,bf
1243 XN CIRCLE (500,220),6,8,0,3.14,.5
1244 yB PAINT (500,219),8
1245 fY AREA (479,210)
1246 UW AREA (486,200):AREA (492,210):AREA (500,207)
1247 hW AREA (508,210):AREA (514,200):AREA (521,210)
1248 Gy COLOR 10:AREAFILL
1249 Wq LINE (492,210)-(492,220),10
1250 Rh LINE (508,210)-(508,220),10
1251 w2 PALETTE 2,1,0,0
1252 Xp PALETTE 3,1,0,0:PALETTE 4,.7,.5,.3
1253 zC PALETTE 5,.75,.6,.3:PALETTE 6,.56,.44,.3
1254 Eg PALETTE 7,.8,.7,.4:PALETTE 8,.5,0,0
1255 Xc PALETTE 9,.6,0,0:PALETTE 10,.5,.3,0
1256 PC GOSUB Taste
1257 dF RETURN
1258 Vn Gef:
1259 Go GOSUB schwarz:COLOR ,15:CLS
1260 7I FOR i%=0 TO 239 STEP 10:LINE (0,i%)-(630,i%),1:NEXT
1261 Bj FOR i%=0 TO 630 STEP 30:LINE (i%,0)-(i%,240),1:NEXT
1262 Fm FOR i%=1 TO 239 STEP 20:SCROLL (0,i%)-(630,i%+8),15,0:NEXT
1263 v8 LINE (100,50)-(200,100),3,bf
1264 Vt FOR i%=6 TO 15
1265 vp LINE (105+i%,50)-(105+i%,100),i%
1266 lZ LINE (140+i%,50)-(140+i%,100),i%
1267 NS LINE (175+i%,50)-(175+i%,100),i%:NEXT
1268 zx AREA (500,0):AREA (500,201):AREA (630,249):AREA (630,0)
1269 rd COLOR 14:AREAFILL
1270 Yo FOR i%=0 TO 200 STEP 10:LINE (500,i%)-(630,i%*.25),1
1271 cc1 IF i%/20 =INT(i%/20) THEN x%=0 ELSE x%=15
1272 hY LINE (530-x%,i%*(1.06-x%*.002))-(530-x%,i%*(1.06-x%*.002)
+10),1
1273 cL LINE (565-x%,i%*(1.13-x%*.002))-(565-x%,i%*(1.13-x%*.002)
+10),1
1274 kg LINE (605-x%,i%*(1.21-x%*.002))-(605-x%,i%*(1.21-x%*.002)
+10),1
1275 eJ0 NEXT
1276 sJ AREA (0,201):AREA (500,201):AREA (630,250):AREA (0,250)
1277 Q9 COLOR 2,0:AREAFILL
1278 Qs LINE (200,180)-(450,185),4,bf
1279 bm x%=12:FOR i%=175 TO 179
1280 vJ LINE (207,i%)-(445,i%),x%,bf:IF x%=12 THEN x%=3 ELSE x%=12
1281 kp NEXT
1282 xn LINE (210,150)-(212,180),10,bf
1283 y3 LINE (440,150)-(438,180),10,bf
1284 Ts PATTERN 1073:LINE (211,150)-(211,180),10
1285 PY LINE (439,150)-(439,180),10:PATTERN -1
1286 u1 x=.8:FOR i%=6 TO 15:x=x-.0625:PALETTE i%,x,x,x:NEXT
1287 e6 PALETTE 2,.2,.1,0:PALETTE 3,0,0,1:PALETTE 4,.3,.2,0
1288 2c Na$(Pla)=CHR$(0)+Na$(Pla)
1289 43 LOCATE 28,2:COLOR 1,2:PRINT a$
1290 D1 LOCATE 30,33:PRINT "- Taste":GOSUB Taste
1291 o9 GOSUB schwarz:COLOR 0,0:CLS:RETURN
1292 FK preise:
1293 XG wap=INT(1+zug/6):wap=INT(wap/2+RND*(wap/2))+1
1294 PH PR$(1)=3*(1+zug/6):PR$(1)=PR$(1)/2+RND*(PR$(1)/2)
1295 XS PR$(2)=4.1*(1+zug/8):PR$(2)=PR$(2)/2+RND*(PR$(2)/2)
1296 2e PR$(3)=3.5*(1+zug/7):PR$(3)=PR$(3)/2+RND*(PR$(3)/2)
1297 R4 PR$(4)=4.5*(1+zug/8):PR$(4)=PR$(4)/2+RND*(PR$(4)/2)
1298 Px PR$(5)=2.5*(1+zug/6):PR$(5)=PR$(5)/2+RND*(PR$(5)/2)
1299 6U Sok=INT(1+zug/6):Sok=INT(Sok/2+RND*(Sok/2))+1
1300 lc Schp=400*INT(1+zug/10):Schp=Schp+INT(RND*100)
1301 OB Lapr=600*INT(1+zug/10):Lapr=INT(Lapr/4+RND*(Lapr/4))
1302 8E He$(1)=200*(1+zug/10):He$(1)=He$(1)/2+RND*(He$(1)/2)
1303 u5 He$(2)=350*(1+zug/10):He$(2)=He$(2)/2+RND*(He$(2)/2)
1304 9I He$(3)=250*(1+zug/10):He$(3)=He$(3)/2+RND*(He$(3)/2)
1305 BG He$(4)=400*(1+zug/10):He$(4)=He$(4)/2+RND*(He$(4)/2)
1306 hn He$(5)=150*(1+zug/10):He$(5)=He$(5)/2+RND*(He$(5)/2)
1307 R4 KP$(1)=3000*(1+zug/25):KP$(1)=KP$(1)/3+RND*(KP$(1)/3)
1308 Ob KP$(2)=7000*(1+zug/25):KP$(2)=KP$(2)/3+RND*(KP$(2)/3)
1309 Be KP$(3)=5000*(1+zug/25):KP$(3)=KP$(3)/3+RND*(KP$(3)/3)
1310 g6 KP$(4)=8000*(1+zug/25):KP$(4)=KP$(4)/3+RND*(KP$(4)/3)
1311 bp KP$(5)=2000*(1+zug/25):KP$(5)=KP$(5)/3+RND*(KP$(5)/3)
1312 W8 RETURN
1313 Yh Index1:
1314 Dz w=Gold(i%)
1315 pu IF we(i%)>0 THEN GOSUB Swert:
w=w+(Schiffe(i%)+FSch$(i%))*SW
1316 p0 w=w-La(i%)*(Lapr-300)
1317 zD FOR i1=1 TO 5
1318 bI w=w+PR$(i1)*War$(i%,i1)
1319 Va w=w+(KP$(i1)-300)*Kon$(i%,i1)
1320 NS NEXT
1321 tg w=w+In(i%)*1000
1322 gS w=w/(100+(100-ap$(i%,i%)))
1323 hJ RETURN
1324 ld Index:
1325 bj GOSUB Index1
1326 F9 kurve$(i%,zug)=240-INT(w/12.5)
1327 cP x=240-kurve$(i%,zug)
1328 hN IF Ti$(i%)="Großreeder" AND x>155 THEN Sieg
1329 z7 IF Ti$(i%)="Reeder" AND x>130 THEN
1330 dl2 Ti$(i%)="Großreeder"
1331 bW GOTO Urkunde
1332 QJ0 END IF
1333 Cw IF Ti$(i%)="Großhändler" AND x>100 THEN
1334 gr2 Ti$(i%)="Reeder"
1335 Fa GOTO Urkunde
1336 UNO END IF
1337 Vp IF Ti$(i%)="Händler" AND x>70 THEN
1338 Mt2 Ti$(i%)="Großhändler"
1339 Je GOTO Urkunde
1340 YRO END IF
1341 OO IF Ti$(i%)="Bürger" AND x>40 THEN
1342 ES2 Ti$(i%)="Händler"
1343 ni GOTO Urkunde
1344 eV0 END IF
1345 3f RETURN
1346 zo DATA 8,9,10,11,12,13,14,15
1347 h0 DATA 9,10,11,12,13,14,15,8
1348 vJ DATA 10,11,12,13,14,15,8,9
1349 7I DATA 11,12,13,14,15,8,9,10
1350 FR DATA 12,13,14,15,8,9,10,11
1351 lc DATA 13,14,15,8,9,10,11,12
1352 U9 DATA 14,15,8,9,10,11,12,13
1353 vE DATA 15,8,9,10,11,12,13,14
1354 HQ DATA 238,100,230,120,236,128,257,118
1355 Ew DATA 263,150,279,144,290,145,294,123
1356 TC DATA 302,117,296,106,305,100
1357 oH DATA 230,170,236,168,242,134,252,138
1358 e8 DATA 246,150,258,165,265,161,280,164
1359 HJ DATA 300,160,310,162,325,153,325,170
1360 nZ DATA 325,170,325,153,320,148,317,144
1361 tZ DATA 323,132,335,135,334,129,334,125
1362 So DATA 330,122,355,116,348,112,330,115
1363 e4 DATA 324,112,323,100,380,100,380,170
(C) 1990 M&T

```

Listing 3. Schluß

von Maurizio Lotauro und
Thomas Taubenberger

Das Amiga-Basic gehört sicherlich zu den komfortablen Interpretern und unterstützt den Programmierer in vielerlei Hinsicht. Doch einige Funktionen nützen den Computer nicht in dem Umfang aus, wie dies möglich wäre. Zu diesen Funktionen gehört auch die Einrichtung eines Menüs in Amiga-Basic. Darum möchten wir Ihnen eine praktische Erweiterung der Menüanweisung vorstellen.

Neben dem Listing »SuperMenuII« finden Sie in diesem Sonderheft auch die Listings zu den Basic-Programmen »Image«, »Brush2BasicImage«, »ShowBasicImage« und »SuperMenuIIDemo«. Diese Programme unterstützen Sie bei dem Erstellen eigener Grafiken, die Sie mit den »SuperMenuII«-Routinen in Ihre Menüzeile einbinden können.

Selbstverständlich sind alle Programme auch auf unserer Programmservice-Diskette zu finden.

Doch nun zu »SuperMenuII«.

Nach dem Abtippen muß das Programm als ASCII-File gespeichert werden. Eine Speicherung im ASCII-Format erreichen Sie, indem Sie »,A« am SAVE-Befehl des Amiga-Basic anfügen (nicht vom Menü aus).

```
SAVE "SuperMenuII",A
```

Um das Programm zu testen, können Sie das kleine Beispielprogramm »SuperMenuDemo« benutzen. Nach dem Eingeben des Demo-Programms laden Sie einfach die Menürountinen von »SuperMenuII« dazu.

```
Merge "SuperMenuII"
```

Wenn Sie nun das Programm starten, ertönt nach einigen Sekunden ein akustisches Signal und folgende Mitteilung zeigt die erfolgreiche Installation an:

```
SuperMenues bereit!
```

Bei gedrückter rechter Maustaste erscheint nun links an der Menüleiste das erste Menü, eine Auswahl von »Item A«, »Item B« und »Quit«.

Eigene Menüs in Basic-Programmen

Das Supermenü

Wer jetzt an ein festliches Mahl denkt, liegt völlig daneben, obwohl die Leistung des Programms einem auf der Zunge zergeht. Mit »SuperMenuII« haben Sie die Möglichkeit, in Ihren Basic-Programmen eine Menüleiste zu erstellen, wie sie sonst nur in professionellen Programmen zu finden ist.

Daneben befindet sich das zweite Menü, das verschiedene Unterpunkte zu dem bekannten Amiga-Logo enthält. Beim Anwählen von z.B. »Item A« gibt Ihnen das Programm die Werte der Variablen »MenuNum%«, »ItemNum%« und »SubNum%« aus. Mit diesen Werten können Sie auswerten, welches Item angewählt wurde.

Wenn Sie »SuperMenuII« in Ihr eigenes Programm einbinden möchten, sollten Sie folgende Routinen vor dem Aufruf mit »Initialize« in Ihr Programm einfügen.

```
LIBRARY "graphics.library"
LIBRARY "intuition.library"
DECLARE FUNCTION AllocRemember&() LIBRARY
DECLARE FUNCTION TextLenght%
() LIBRARY
```

Achten Sie unbedingt darauf, daß die Dateien »INTUITION.bmap« und »GRAPHICS.bmap« im LIBS-Directory Ihrer Diskette vorhanden sind.

Vor dem Programmende sollte »SMENUOFF« und »LIBRARY CLOSE« aufgerufen werden. Gestalten Sie Ihr Basic-Programm auf alle Fälle so, daß immer bei auftretenden Fehlern die Routine »SMENUOFF« aufgerufen wird. Dies verhindert, daß der zugeordnete Speicherplatz beibehalten und erst nach einem RESET wieder frei wird.

Für den Fall, daß ein falscher Wert den Routinen zugeführt wird, übergeben diese die Führung der Routine »MENUERROR«. Diese zeigt die Ursache der Fehler und ruft »SMENUOFF« auf. Anschließend gibt sie noch folgende Fehlermeldungen aus:

- 7 Speicherplatz nicht ausreichend.

- 200 Es wurde eine Menü-Item- oder Subitem-Nummer eingegeben, der kein Wert vorangeht. Ein solcher Fehler tritt dann auf, wenn man z.B. versucht das Item 5 zu definieren, ohne daß Item 4 festgelegt wurde.

- 210 Es wurde ein Parameter eingegeben, der außerhalb dem erlaubten Bereich liegt oder das Feld (Array), das eben eingelesen wurde, beinhaltet fehlerhafte Werte.

- 220 Es wurde ein Wechselitem verwendet, das nicht zum Hauptitem paßt (Strings von abweichender Länge oder Grafiken, die größtmäßig nicht reinpassen).

Die letzten drei Fehler sind sog. »unprintable Errors«. Weitere Details dazu finden Sie in Ihrem Handbuch.

Da für die Fehlerverwaltung, zur Vermeidung von Konflikten mit gleichnamigen Meldungen im Hauptprogramm, nicht »ON ERROR GOTO« verwendet wurde, kann als einziger der Fehler, 5 nicht verwaltet werden. Er tritt auf, wenn man versucht, ein Item an ein Menü anzuhängen, das nicht existiert.

Hier noch Tips zu dem Programm:

Verändern Sie auf keinen Fall die »SHARED«-Variablen im Unterprogramm »INITIALIZE«, vor allem nicht die Variable »RememberKey«. Diese Konstanten verursachen unvorhersehbare Resultate, wenn sie abgeändert wurden.

Verwenden Sie Strings gleicher Länge. Füllen Sie eventuelle Leerstellen mit Leerzeichen aus. Ein kurzer String läßt sich mit dem Mauszeiger nur sehr schwer direkt anwählen.

Programmname:	SuperMenusII		
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3		
Sprache:	Amiga-Basic		
Bemerkung:	siehe Text		
Programmautor: Maurizio Lotauro <pre> 1 DUO '***** 2 Ef '* 3 zc '* Routinen fuer das Menue-Management in Amigabasic * 4 Gh '* 5 ER '* verwendet : intuition.library * 6 tA '* graphics.library * 7 Jk '* 8 Kb '***** 9 Y7 ' Initialisierungsroutine 10 Zm SUB INITIALIZE STATIC 11 6p2 SHARED check%,command%,toggle%,enabled%,checked% 12 Gz SHARED selectimage%,comp%,box%,none% 13 b8 SHARED RememberKey& 14 ME SHARED item1&,Item2&,Item3&,item4&,item5&,item6&,item7&,item8&,item9&,item10& 15 MV SHARED item11&,item12&,item13&,item14&,item15&,item16&,item17&,item18&,item19&,item20& 16 k5 SHARED item21&,item22&,item23&,item24&,item25&,item26&,item27&,item28&,item29&,item30&,item31& 17 Zx0 ' Initialisierung der Flags 18 i82 check%=1 19 AQ command%=4 20 nr toggle%=8 21 kL enabled%=16 22 mf checked%=256 23 Iv selectimage%=0 24 ZH comp%=64 25 D8 box%=128 26 fg none%=192 27 240 ' Initialisierung der Flags gegenseitiger Ausschließung 28 sV2 item1&=1 29 15 Item2&=2 30 s1 Item3&=2^2 31 7L item4&=2^3 32 KQ item5&=2^4 33 XV item6&=2^5 34 ka item7&=2^6 35 xf item8&=2^7 36 Ak item9&=2^8 37 GZ item10&=2^9 38 pH item11&=2^10 39 vP item12&=2^11 40 iX item13&=2^12 41 7f item14&=2^13 42 Dn item15&=2^14 43 Jv item16&=2^15 44 P3 item17&=2^16 45 VB item18&=2^17 46 bJ item19&=2^18 47 Tz item20&=2^19 48 7W item21&=2^20 49 De item22&=2^21 50 Jm item23&=2^22 51 Pu item24&=2^23 52 V2 item25&=2^24 53 bA item26&=2^25 54 hI item27&=2^26 55 nQ item28&=2^27 56 tY item29&=2^28 57 lE item30&=2^29 58 P1 item31&=2^30 59 O0U ' Initialisierung des Pointers der Remember-Struktur 60 8R2 RememberKey&=0 61 131 END SUB 62 aK0 ' Textmanagement-Routine 63 PF SUB SMENUTEXT(xpos%,ypos%,spos%,flags%,mexclude&,key\$,title\$,fpen%,bpen%,title2\$,fpen2%,bpen2%,xset%,yset%) STATIC 64 Qx2 SHARED RememberKey& 65 up IF xpos%<1 OR xpos%>10 THEN MENUERROR "SMENUTEXT", "xpos%",210 66 bd IF ypos%<1 OR ypos%>19 THEN MENUERROR "SMENUTEXT", "ypos%",210 67 kw IF spos%<0 OR spos%>31 THEN MENUERROR "SMENUTEXT", "spos%",210 68 7o IF fpen%<0 OR fpen%>31 THEN MENUERROR "SMENUTEXT", "fpen%",210 69 Az IF bpen%<0 OR bpen%>31 THEN MENUERROR "SMENUTEXT", "bpen%",210 70 RO IF fpen2%<0 OR fpen2%>31 THEN MENUERROR "SMENUTEXT", "fpen2%",210 </pre>			

Kurzbeschreibung der Programmerroutinen:
INITIALIZE

Diese Routine wird nur einmal verwendet, und zwar von »SMENUTEXT« oder »SMENUIMAGE«. Dieses Programm initialisiert alle Variablen, die »SuperMenusII« braucht.

SMENUTEXT

Diese Routine ähnelt dem Basic-Befehl »Menu« und setzt sich bei jedem Aufruf erneut im Speicher fest. Hier nun die Syntax der Routine:

```
SMENUTEXT xpos%, ypos%, spos%
, flags%, mexclude&, key$, title$,
fpen%, bpen%, tite2$, fpen2%,
bpen2%, xset%, yset%
```

Die einzelnen Variablen haben folgende Bedeutung:

xpos%

ist die Position des Menüs mit einem Wert von 1 bis 10

ypos%

ist die Position des Ausdrucks im Menü, mit einem Wert von 1 bis 19.

spos%

ist die Position des Ausdruckes im Untermenü, mit einem Wert von 0 (für das Item) bis 31.

Die Bedeutung folgender Variablen ist für alle nachfolgenden Ausdrücke gleich.

flags%

beinhaltet ein oder mehrere Flags, die in der Flag-Tabelle angegeben sind.

mexclude&

beinhaltet die Wechselflags. Es wird im Zusammenhang mit den Flags »check%« und »toggle%« (siehe Flag-Tabelle) verwendet. Wenn man von »Item1« bis »Item31« wählt, löscht man hiermit die Markierung der anderen.

key\$

ist ein alphanumerischer Charakter, der mit der rechten Amiga-Ta-

ste zur Anwahl dieses Items verwendet wird. (Nur das erste ASCII-Zeichen wird verwendet.)

title\$

ist der Text für das Item.

fpen%

ist die Vordergrundfarbe für »title\$«. Sie ist normalerweise 0.

bpen%

ist die Hintergrundfarbe für »title\$«. Sie ist normalerweise 1.

title2\$

wird im Zusammenhang mit dem Flag »selectimage%« verwendet. Es ist der Text, der »title\$« ersetzt, wenn man mit dem Mauszeiger darüberfährt.

»Title2\$« muß dieselbe Länge aufweisen wie »title\$«.

xset% und yset%

sind die Koordinaten des linken oberen Ecks des Items, bezogen

```

71 cJ IF bpen2%<0 OR bpen2%>31 THEN MENUERROR "SMENUTEXT", "b
pen2%", 210
72 VK IF spos%=0 THEN ' wenn e
s kein Subitem ist, dann ...
73 eJ4 IF (flags% AND 16) THEN active%=1 ELSE active%=0
74 Wy MENU xpos%, ypos%, active%, title$
75 UB stall# = TIMER : WHILE TIMER < (stall#+1) : WEND ' Wa
rteschleife (ermöglicht dem Amigabasic die Menues aufzuz
etzen)
76 A33 END IF
77 9x0 ' Hier werden die Items und Subitems gesetzt
78 Xq ' WINDOW(7) giebt die Anfangsadresse der Window-Struktur au
s
79 Ob2 MenuAdd&=PEEK(WINDOW(7)+28) ' Adress
e Menue-Struktur
80 kq IF MenuAdd=0 THEN MENUERROR "SMENUTEXT", "", 200
81 KL Address&=MenuAdd&
82 ER FOR i%=1 TO xpos%-1
83 Mh4 Address&=PEEK(Address&)
84 Gs IF Address&=0 THEN MENUERROR "SMENUTEXT", "", 200
85 B13 NEXT ' Adress be
inhaltet am Schleifenende die Adresse des Menues xpos%
86 5p2 Address&=PEEK(Address&+18) ' Adresse e
rstes Item, des Menues
87 Jv IF Address&=0 THEN MENUERROR "SMENUTEXT", "", 200
88 Oc FOR i%=1 TO ypos%-1
89 Sn4 Address&=PEEK(Address&)
90 My IF Address&=0 THEN MENUERROR "SMENUTEXT", "", 200
91 Ju3 NEXT ' Adress be
inhaltet am Schleifenende die Adresse des Menues ypos%
92 Qd2 IF spos%>0 THEN ' Erstellt
die Adresse des Subitems
93 yJ4 Address&=Address&+28
94 6E FOR i%=1 TO spos%-1
95 Yt6 Address&=PEEK(Address&)
96 S4 IF Address&=0 THEN MENUERROR "SMENUTEXT", "", 200
97 PX5 NEXT ' Adress beinhaltet am Sch
leifenende die Adresse des Pointers zur MenuItem-Strukt
ur
98 AM4 TextAttr$(3)=0 ' Wird zur Erzeugung und D
imensionierung der Variable verwendet
99 NU CALL AskFont&(WINDOW(8), VARPTR(TextAttr$(0)))
100 cL height%=TextAttr$(2) ' Setzt die Höhe des aktue
llen Fonts
101 FM width%=TextLength$(WINDOW(8), SADD(title$), LEN(title$))
' Breite des Strings
102 rF title$=title$+CHR$(0)
103 vY title&=SADD(title$)
104 m30 ' Erzeugt die IntuiText-Struktur fuer den Namen
105 8w4 Intuit&=AllocRemember&(VARPTR(RememberKey&), 20, 65537&)
' Teilt der Struktur den nötigen Speicherplatz zu
106 FY IF Intuit&=0 THEN MENUERROR "SMENUTEXT", "", 7
107 H6 POKE Intuit&+2, 1 ' Setzt die Schreibart au
f JAM2
108 nK POKEL Intuit&+12, title& ' Setzt die Adresse des T
extes
109 7r0 ' Erzeugt die MenuItem-Struktur
110 e04 SubItem&=AllocRemember&(VARPTR(RememberKey&), 34, 65537&)
' Teilt der Struktur den noetigen Speicherplatz zu
111 Cs IF SubItem&=0 THEN MENUERROR "SMENUTEXT", "", 7
112 do POKEW SubItem&+8, width% ' Breite Schriftbox
113 Kk POKEW SubItem&+10, height% ' Hoehe Schriftbox
114 5K POKEW SubItem&+12, 82 ' flag=enabled+comp+
text
115 3M POKEL SubItem&+18, Intuit& ' Adresse IntuiText-
Struktur
116 eK POKEL Address&, SubItem& ' Speichert die Adresse
der letzten Struktur in die vorletzte
117 5y Address&=SubItem&
118 qj3 END IF
119 lw2 CALL ClearMenuStrip&(WINDOW(7)) ' Loescht
die Menues
120 Ih POKE PEEKL(Address&+18), fpen% ' Foregrou
nd color
121 aF POKE PEEKL(Address&+18)+1, bpen% ' Backgrou
nd color
122 t6 POKEW Address&+4, xset% ' Speicher
t horizontales Offset
123 Ig POKEW Address&+6, yset% ' Speicher
t vertikales Offset
124 A0 POKEW Address&+12, flags%+2 ' Speicher
t die Flags
125 TP POKEL Address&+14, mexclude& ' Speicher
t die Flags gegenseitiger Ausschließung
126 Ew IF key$ <> "" THEN POKE Address&+26, ASC(key$) ' Speich
ert die Befehlstaste
127 ks CALL SetMenuStrip&(WINDOW(7), MenuAdd&) ' Stellt d
ie Menues wieder auf

```

Listing 1. Fortsetzung

auf das linke obere Eck des Menüs oder des nächsthöheren Items.

SMENUIMAGE

Diese Routine ist »SMENUTEXT« sehr ähnlich und gilt für Grafiken. Hier die Syntax:

```
SMENUIMAGE xpos%, ypos%, spos%, flags%, mexclude&, key$, Image%(), Image2%(), xset%, yset%
```

Alle Argumente, die gleichnamig in »SMENUTEXT« auftauchen, haben auch die gleiche Bedeutung. Hier nun die Variablen:

Image%() und Image2%()

sind Arrays, die das Bild bzw. das Alternativbild beinhalten, welches mit dem normalen Basic-Befehl »GET« (siehe Basic-Handbuch) erzeugt wird. Sollte auch Image2%() Verwendung finden, so muß es die

Ausmaße von Image%() beinhalten.

SUBITEM

»SubNum%« stellt eine Vervollständigung der Basic-Funktionen »MENU(0)« und »MENU(1)« dar. Es ergibt den Wert des Subitems, das durch »SubNum%« gewählt wurde.

SMENU

Die Routine dient zur Aktivierung oder Desaktivierung der Items. Dabei gilt folgende Syntax:

```
-SMENU xpos%, ypos%, spos%, enab%
```

Die ersten drei Variablen haben wir schon weiter oben besprochen. Hier nun die Erklärung der neu hinzugekommenen:

enab%=0

zum Desaktivieren und »enab%=1« zum Aktivieren

SMENUOFF

soll am Ende des Programmes aufgerufen werden. Es befreit den

von »SuperMenuII« besetzten Speicherplatz und stellt die Originalmenüs wieder her.

Bedeutung der Flags:

check%

setzt das »CheckMark« links vom Namen.

command%

wird verwendet, wenn eine Taste dem Menü zugewiesen wurde.

toggle%

kommt in Verbindung mit dem Flag »check%« zur Verwendung. Mit diesem Flag setzen bzw. löschen Sie das »CheckMark« im Menü, sobald dasselbe angewählt wird.

enabled%

zeigt die Aktivierung des Begriffs.

checked%

wird in Verbindung mit dem Flag »check%« verwendet. Ist dieses Flag gesetzt, so erscheint ein »CheckMark« links im Menü.

Von den folgenden Flags darf ausschließlich nur eines verwendet werden. Sie dienen zur Festlegung der Evidenzierung des Menüs bei Kontakt mit dem Mauszeiger.

comp%

invertiert die Farbe.

box%

zeichnet einen Rahmen um das Menü

selectimage%

verwendet das Alternativbild.

Da nun in das Menü keine normalen Bilder mit eingebunden werden können, erweist sich das Programm »Image« (Listing 2) als sehr nützlich.

»IMAGE« ist ein kleiner Grafikeditor, mit dem Sie Bilder erstellen und dann in Ihre Menüliste (die mit »SuperMenüsll« erstellt wurde) einbinden können. Nach dem Start des Programms muß zunächst die Bildgröße eingegeben werden (max. 105 x 42 Bildpunkte), anschließend erscheint ein großes und ein kleines Window auf dem Bildschirm. Die erstellten Bilder werden als DATA-Sequenz im ASCII-Format abgespeichert und können problemlos an die Menüroutinen angefügt werden (Kommando »Merge«).

Bei »ShowBasicImage« handelt es sich um ein Programm, das Ihnen Bilder, die z.B. mit »Image« erstellt wurden, zur Ansicht auf dem Monitor darstellt. Nach dem Programmstart von »ShowBasicImage« erscheint eine Aufforderung zur Eingabe des Namens des darzustellenden Bildes. Nach dem Anwählen der gewünschten Bildschirmauflösung (durch Drücken der Zahlentasten eins bis vier) stellt das Programm das Bild auf dem Bildschirm (mit den momentan verwendeten Bitplans) dar. Mit dem Drücken der linken Maustaste endet das Programm.

Mit »Brush2BasicImage« haben Sie die Möglichkeit, einen IFF-Brush als ein IFF-Bild in eine »Image-Datei« umrechnen zu lassen. Die Umrechnung betrifft nur die Bitebene des Bildes, die jeweiligen Farben hängen von der Farbpalette des Screens ab, in dem das Bild später dargestellt wird. Das Programm benötigt nach dem Start die Angabe des Namens und des Pfades von Quell- und Zieldateien. ag

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Albert Absmeier – verantwortlich für den redaktionellen Teil
Textchef: Jens Maasberg

Leitender Redakteur: Andreas Greil (ag)

Redaktion: Albert Petryszyn (pe)

Mitarbeiter dieser Ausgabe: Thomas Hauser, Hartwig Taubenberger, Martin Jobst, Paul Schokal, Maurizio Lotauro, Thomas Taubenberger, Arnd Goebel

Redaktionsassistentz: Catharina Winter, Petra Kessner (414)

Telefax: 089/46 13-433, Hotline: Donnerstag 14 bis 17 Uhr

Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

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. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Verlagsleitung: Wolfram Höfler

Operation Manager: Michael Koeppel

Art-director: Friedemann Porscha

Titelgestaltung: Wolfgang Berns

Layout: Alexander Kowarzyk (Cheflayouter), Neclé Dizdar, Marian Schwarz

Bildredaktion: Janos Feitser (Ltg.), Sabine Tennstaedt; Roland Müller (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)

Anzeigendirektor: Ralph Peter Rauchfuss

Anzeigenleitung: Philipp Schiede (399) – verantwortlich für die Anzeigen

Telefax: 089/46 13-775

Anzeigenverwaltung und Disposition: Monika Burseg (147)

Auslandsrepräsentation:

Auslandsniederlassungen:

Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Fax 042-4 15 770, 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, Große Neugasse 28, A 1040-Wien, Telefon: 0222/5 87 13 93, Telex: 047-132532

Anzeigen-Auslandsvertretung:

England: F. A. Smyth & Associates Limited, 23a, Aylmer Parade, London, N2 OPQ. Telefon: 0044/1/3405058, Telefax: 0044/1/3419602

Israel: Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562256

Taiwan: AIM International Inc., 4F-1, No. 200, Sec. 3, Hsin-I Rd.; Taipei, Taiwan, R.O.C., Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710

Korea: Young Media Inc., C.P.O. Box: 6113, Seoul/Korea, Tel. 0082-2-7564819, /-7742759, Fax 0082-2-7575789

Vertriebsdirektor: Uwe W. Hagen

Vertriebsmarketing: Robert Riesinger (364)

Vertrieb Handelsaufgabe: Inland (Groß-, Einzel- und Buchhandelsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Ludwigstraße 26, 7000 Stuttgart 1, Tel. 0711/6483-110

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 46 13-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Verkaufspreis: Das Einzelheft kostet DM 16,-

Produktion: Technik: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887); Herstellung: Otto Albrecht (Ltg./917)

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. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind.

Haftung: Für den Fall, daß in diesem Heft unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/46 13-185, Fax 46 13-774.

© 1990 Markt & Technik Verlag Aktiengesellschaft

Vorstand: Otmar Weber (Vors.), Bernd Balzer

Direktor Zeitschriften: Michael M. Pauly

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 522052, Telefax 089/46 13-100

Telefon-Durchwahl im Verlag: Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/46 13 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

```

128 wG IF title2$ <> "" THEN
129 8B4 IF LEN(title$) <> (LEN(title2$)+SGN(spos%)) THEN MENUER
ROR "SMENUTEXT", "", 220
130 Oi title2$=title2$+CHR$(0)
131 kw title&=SADD(title2$)
132 HJ0 ' Erzeugt die IntuiText-Struktur fuer den Alternativtext
133 bu4 Intuit&=AllocRemember&(VARPTR(RememberKey&), 20, 65537&)
' Alloca la memoria per la struttura
134 h0 IF Intuit&=0 THEN MENUERROR "SMENUTEXT", "", 7
135 Lz POKE Intuit&, fpen2% ' Foreground col
or
136 nw POKE Intuit&+1, bpen2% ' Background col
or
137 dw POKE Intuit&+2, 1 ' Setzt die Schr
eibart auf JAM2
138 P1 POKE Intuit&+12, title& ' Setzt die Adre
sse des Alternativtextes
139 1w POKE Intuit&+22, Intuit& ' Setzt die Adre
sse der Struktur
140 C53 END IF
141 JL1 END SUB
142 g40 ' Routine des Bild-Managements
143 jz SUB SMENUIMAGE(xpos%, ypos%, spos%, flags%, mexlude&, key$, Image
%(1), Image2%(1), xset%, yset%) STATIC
144 iF2 SHARED RememberKey&
145 6e IF xpos% < 1 OR xpos% > 10 THEN MENUERROR "SMENUIMAGE", "xpo
s%", 210
146 nS IF ypos% < 1 OR ypos% > 19 THEN MENUERROR "SMENUIMAGE", "ypo
s%", 210
147 w1 IF spos% < 0 OR spos% > 31 THEN MENUERROR "SMENUIMAGE", "spo
s%", 210
148 3b ImageB%=LBOUND(Image%)
149 zv ImageDataSize=Image%(ImageB%+1)*((Image%(ImageB%+0)+15)\
16)*Image%(ImageB%+2) ' Größe des Bildes in WORD
150 HY IF ImageDataSize=0 OR ImageDataSize > (UBOUND(Image%)-2-
ImageB%) THEN MENUERROR "SMENUIMAGE", "Image%", 210
151 NU0 ' Hier werden die Items oder Subitems gesetzt
152 j2 ' WINDOW(7) giebt die Anfangsadresse der Window-Struktur au
s
153 tp2 MenuAdd&=PEEK(WINDOW(7)+28) ' Adresse
der Menue-Struktur
154 hQ IF MenuAdd&=0 THEN MENUERROR "SMENUIMAGE", "", 200
155 WX Address&=MenuAdd&
156 Qd FOR i%=1 TO xpos%-1
157 Yt4 Address&=PEEK(Address&)
158 DS IF Address&=0 THEN MENUERROR "SMENUIMAGE", "", 200
159 Nx3 NEXT ' Adress be
inhaltet am Schleifenende die Adresse des Menues xpos%
160 yI2 Address&=Address&+18
161 Zn FOR i%=1 TO ypos%-1
162 dy4 Address&=PEEK(Address&)
163 IX IF Address&=0 THEN MENUERROR "SMENUIMAGE", "", 200
164 ie3 NEXT ' Adress beinhaltet am Sc
hleifenende die Adresse des Pointers zum Item ypos%
165 sI2 IF spos% > 0 THEN ' Setzt die Adresse des
Subitems
166 Vo4 Address&=PEEK(Address&)+28
167 HP FOR i%=1 TO spos%-1
168 j46 Address&=PEEK(Address&)
169 Od IF Address&=0 THEN MENUERROR "SMENUIMAGE", "", 200
170 GN5 NEXT ' Adress beinhaltet am Sc
hleifenende die Adresse des Pointers zum SubItem spos%
171 ha3 END IF
172 oY0 ' Erzeugt die Image-Struktur
173 V22 ImageAdd&=AllocRemember&(VARPTR(RememberKey&), 20, 65537&)
' Teilt der Struktur den nötigen Speicherplatz zu
174 Iv IF ImageAdd&=0 THEN MENUERROR "SMENUIMAGE", "", 7
175 Br ImageData&=AllocRemember&(VARPTR(RememberKey&), ImageDataS
ize&*2, 65539&) 'Man kann hier nicht direkt das Array
176 fo IF ImageData&=0 THEN MENUERROR "SMENUIMAGE", "", 7
' verwenden, da es in die evtl. vorhandene
177 JR k&=0
'FastMemory abgelegt wuerde
178 2J FOR i&=3+ImageB% TO ImageDataSize&+2+ImageB%
' Speichert die Bilddaten
179 YS4 POKEW ImageData&+k&, Image%(i&)
180 LP k&=k&+2
181 O53 NEXT
182 Np2 POKEW ImageAdd&+4, Image%(ImageB%+0) ' Breite
183 ZQ POKEW ImageAdd&+6, Image%(ImageB%+1) ' Höhe
184 zu POKEW ImageAdd&+8, Image%(ImageB%+2) ' Tiefe
185 7Q POKEW ImageAdd&+10, ImageData&
der Bilddaten
186 1K POKE ImageAdd&+14, 2*Image%(ImageB%+2)-1 ' PlanePi
ck
187 N70 ' Erzeugt die MenuItem-Struktur
188 Ws2 MenuItem&=AllocRemember&(VARPTR(RememberKey&), 34, 65537&)
' Teilt der Struktur den nötigen Speicherplatz zu
189 sy IF MenuItem&=0 THEN MENUERROR "SMENUIMAGE", "", 7
190 6L POKEW MenuItem&+8, Image%(ImageB%+0) ' Breite Schr
iftbox
191 ea POKEW MenuItem&+10, Image%(ImageB%+1) ' Höhe Schrif
tbox
192 rs POKEW MenuItem&+18, ImageAdd& ' Adresse der
Image-Struktur
193 JX CALL ClearMenuStrip&(WINDOW(7)) ' Löscht die
Menues
194 Sn POKEW Address&, MenuItem& ' Speichert die Adresse d
er letzten Struktur in die vorletzte
195 9D POKEW MenuItem&+4, xset% ' Seicher
t horizontales Offset
196 bq POKEW MenuItem&+6, yset% ' Seicher
t vertikales Offset
197 7R POKEW MenuItem&+12, flags% ' Seicher
t die Flags
198 oA POKEW MenuItem&+14, mexlude& ' Seicher
t die Flags gegenseitiger Ausschließung
199 ce IF key$ <> "" THEN POKE MenuItem&+26, ASC(key$) ' Seich
ert die Befehlstaste
200 Jx CALL SetMenuStrip&(WINDOW(7), MenuAdd&) ' Stellt
wieder die Menues auf
201 X70 ' Eventuelles Alternativbild
202 Hn2 IF (flags% AND 192)=0 THEN
203 GA4 ImageB2%=LBOUND(Image2%)
204 L8 IF Image%(ImageB%+0) <> Image2%(ImageB2%+0) OR Image%(I
mageB%+1) <> Image2%(ImageB2%+1) THEN MENUERROR "SMENUIM
AGE", "", 220
205 jt ImageDataSize&=Image2%(ImageB2%+1)*((Image2%(ImageB2%+0
)+15)\16)*Image2%(ImageB2%+2) ' Dimensione imagine in
WORD
206 lp IF ImageDataSize&=0 OR ImageDataSize& > (UBOUND(Image2%)
-2-ImageB2%) THEN MENUERROR "SMENUIMAGE", "Image2%", 210
207 N70 ' Erzeugt die Image-Struktur
208 4b4 ImageAdd&=AllocRemember&(VARPTR(RememberKey&), 20, 65537&
) ' Teilt der Struktur den nötigen Speicherplatz zu
209 rU IF ImageAdd&=0 THEN MENUERROR "SMENUIMAGE", "", 7
210 3d ImageData&=AllocRemember&(VARPTR(RememberKey&), ImageDat
aSize&*2, 65539&)
211 cg IF ImageData&=0 THEN MENUERROR "SMENUIMAGE", "", 7
212 IK k&=0
213 Ki FOR i&=3+ImageB2% TO ImageDataSize&+2+ImageB2% ' Sei
chert die Bilddaten
214 b26 POKEW ImageData&+k&, Image2%(i&)
215 uy k&=k&+2
216 Ze5 NEXT
217 s24 POKEW ImageAdd&+4, Image2%(ImageB2%+0) ' Bre
ite
218 N1 POKEW ImageAdd&+6, Image2%(ImageB2%+1) ' Höh
e
219 KK POKEW ImageAdd&+8, Image2%(ImageB2%+2) ' Tie
fe
220 Ix POKEW ImageAdd&+10, ImageData& ' Adr
esse der Bilddaten
221 PQ POKE ImageAdd&+14, 2*Image2%(ImageB2%+2)-1 ' Pla
nePick
222 Mx POKEW MenuItem&+22, ImageAdd&
223 XQ3 END IF
224 eg1 END SUB
225 Ge0 ' Routine zum Löschen der Menues und Räumen des besetzten S
peichers
226 HO SUB SMENUOFF STATIC
227 3a2 SHARED RememberKey&
228 4w MenuAddress&=PEEK(WINDOW(7)+28)
229 70 WHILE MenuAddress& <> 0
230 tJ4 POKEW MenuAddress&+18, 0 ' Setzt die Poin
ters zur ITEM-Struktur auf 0
231 ZM MenuAddress&=PEEK(MenuAddress&) ' Adresse des fo
lgenden Menues
232 Bz3 WEND
233 hk2 CALL FreeRemember&(VARPTR(RememberKey&), 1)
234 yM MENU RESET
235 pr1 END SUB

```

Listing 1. Fortsetzung

```

236 200 ' Routine zur Feststellung einer und eventuell welcher Item
      wahl
237 r0 SUB SUBMENU(SubNum%) STATIC
238 7y2 intuitmessage&=PEEK(WINDOW(7)+94) ' Adresse IntuitM
      essage-struktur
239 qk menucode&=intuitmessage&+24 ' Menuecode. Er be
      inhaltet hier Hinweise
240 H1f ' zum gewählten Me
      nue.
241 OG ' Die 5 bedeutungs
      vollsten Bits sind
      jene des Subitems
242 nx2 SubNum%=(PEEK(menucode&)/(2^11) AND 31)+1 ' Anzahl de
      r SubItems
243 e1 IF SubNum%=32 THEN SubNum%=0 ' Kein Subi
      tem wenn alle Bits gesetzt sind
244 y01 END SUB
245 C50 ' Routine zum Ein-Ausschalten des Menues, Items oder Subite
      ms
246 qW SUB SMENU (xpos%,ypos%,spos%,enab%) STATIC
247 Ew2 IF xpos%<1 OR xpos%>10 THEN MENUERROR "SMENU","xpos%",2
      10
248 pi IF ypos%<0 OR ypos%>19 THEN MENUERROR "SMENU","ypos%",2
      10
249 43 IF spos%<0 OR spos%>31 THEN MENUERROR "SMENU","spos%",2
      10
250 RG IF enab%<0 OR enab%>1 THEN MENUERROR "SMENU","enab%",2
      10
251 RE IF ypos%=0 THEN
252 wI4 MENU xpos%,ypos%,enab%
253 ES EXIT SUB
254 2v3 END IF
255 Xk2 Address&=PEEK(WINDOW(7)+28) ' Adresse
      Menue
256 RW IF Address&=0 THEN MENUERROR "SMENU","",200
257 3G FOR i%=1 TO xpos%-1
258 Bw4 Address&=PEEK(Address&)
259 UZ IF Address&=0 THEN MENUERROR "SMENU","",200
260 Hm3 NEXT
261 S82 Address&=PEEK(Address&+18) ' Adresse
      Item
262 Xc IF Address&=0 THEN MENUERROR "SMENU","",200
263 DR FOR i%=1 TO ypos%-1
264 Hc4 Address&=PEEK(Address&)
265 af IF Address&=0 THEN MENUERROR "SMENU","",200
266 N33 NEXT
267 x12 IF spos% > 0 THEN
268 Oz4 Address&=PEEK(Address&+28) ' Adresse
      SubItem
269 ej IF Address&=0 THEN MENUERROR "SMENU","",200
270 w4 FOR i%=1 TO spos%-1
271 Oj6 Address&=PEEK(Address&)
272 hm IF Address&=0 THEN MENUERROR "SMENU","",200
273 UZ5 NEXT
274 MF3 END IF
275 qC2 flags%=PEEK(Address&+12)
276 lr IF enab%=0 THEN
277 LU4 flags%=flags% AND -17
278 5o3 ELSE
279 YQ4 flags%=flags% OR 16
280 SL3 END IF
281 Tx2 POKEW Address&+12,flags%
282 ac1 END SUB
283 EVO SUB MENUERROR(RoutineName$,VariableName$,ErrorCode%) STATIC
284 wZ2 PRINT
285 L0 PRINT " *** Fehler im Aufrufen der Routine ";RoutineName$
286 sE IF ErrorCode%=7 THEN
287 504 PRINT " ungenügend Speicherplatz"
288 bz3 ELSEIF ErrorCode%=200 THEN
289 U14 PRINT " Der Item wird von keinem anderen Item voran
      gegangen"
290 j33 ELSEIF ErrorCode%=210 THEN
291 qG4 PRINT " Der Parameter ";VariableName$;" gilt nicht"
292 r73 ELSEIF ErrorCode%=220 THEN
293 E4 PRINT " Alternierende Items haben andere Größen"
294 gZ3 END IF
295 c42 SMENUOFF
296 L7 ERROR ErrorCode%
297 pr1 END SUB
(C) 1990 M&T

```

Listing 1. Schluß

Programmname: Image
Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache: Amiga-Basic
Bemerkung: siehe Text

Programmautor: Maurizio Lotauro

```

1 rI0 '*****
2 Op '* *
3 Jo '* Programm "Image" *
4 Qr '* *
5 Ft '* Utility zur Erzeugung kleiner Bilder die als *
6 xc '* DATAS abgespeichert werden *
7 EA '* um sie in Basicprogrammen eingliedern zu können. *
8 Uv '* *
9 zQ '*****
10 xj2 DEFINT a-z
11 MZ LIBRARY "intuition.library"
12 Ze DIM ShiftArea(504),Image(590)
13 NOO ' Menu
14 Lv2 MENU 1,0,1,"Project"
15 OI MENU 1,1,1,"New "
16 Oj MENU 1,2,1,"Open "
17 Uo MENU 1,3,1,"Save "
18 Ob MENU 1,4,1,"Save As"
19 gu MENU 1,5,1,"Size "
20 RB MENU 1,6,1,"Quit "
21 9E MENU 2,0,0,""
22 CI MENU 3,0,0,""
23 FM MENU 4,0,0,""
24 QC ON MOUSE GOSUB MouseHandler
29 41 ON MENU GOSUB MenuHandler
26 QH ON ERROR GOTO ErrorHandler
27 5r SCREEN 1,640,256,2,2

```

```

28 9N0 ' Maximale Nutzfläche im Window (0,0)-(631,251)
29 ZP ' Maximalbreite 105 pixel
30 Nm ' Maximalhöhe 42 pixel
31 zO2 GOSUB AskSize
32 wG PaintColor=0 : Change=0
33 e0 MaxX=(ImageWidth-1)*6+4 : MaxY=(ImageHeight-1)*6+4
34 3q0 ' Zoomwindow
35 DE2 WINDOW 2,,(0,0)-(631,241),16,1
36 SN GOSUB SmallWindow
37 WU GOSUB NewMenu
38 DQ WHILE 0<>1 : SLEEP : WEND
39 NI1 END
40 xD0 MouseHandler:
41 Mz2 WHILE MOUSE(0)<>0
42 Ft4 x=MOUSE(5) : y=MOUSE(6)
43 zB0 ' Zeichnung
44 Ao4 IF WINDOW(0)=2 AND x<=MaxX AND y<=MaxY THEN
45 X36 Change=-1
46 FX WINDOW OUTPUT 2
47 76 LINE (x-(x MOD 6),y-(y MOD 6))-STEP(4,4),PaintColor,b
      f
48 Nb WINDOW OUTPUT 3
49 ab PSET (x\6+1,y\6+1),PaintColor
50 kd5 END IF
51 vt4 IF WINDOW(0)=3 THEN
52 ik0 ' Farbe
53 Wg6 IF x>167 AND x<209 AND y>2 AND y<40 THEN
54 Th8 WINDOW OUTPUT 3
55 zM LINE (167,2)-(209,40),1,b
56 C5 PaintColor=y\22
57 NP LINE (167,2+PaintColor*19)-STEP(42,19),3,b
58 sl7 END IF
59 QN0 ' Shift
60 Z2 ' x
61 kU6 IF y>15 AND y<27 AND x>111 AND x<162 THEN
62 oK8 Change=-1
63 Wo WINDOW OUTPUT 2
64 rB IF x>136 THEN ' Rechts

```

```

65 u1A      GET (MaxX-4,0)-(MaxX,MaxY),ShiftArea
66 Ot      SCROLL (0,0)-(MaxX,MaxY),6,0
67 dh      LINE (5,0)-(5,MaxY),2
68 hc      PUT (0,0),ShiftArea
69 iw      WINDOW OUTPUT 3
70 Ix      GET (ImageWidth,1)-(ImageWidth,ImageHeight),Shift
          Area
71 RY      SCROLL (1,1)-(ImageWidth,ImageHeight),1,0
72 ro      PUT (1,1),ShiftArea
73 Pt9     ELSE                                     ' Links
74 OJA     GET (0,0)-(4,MaxY),ShiftArea
75 lT      SCROLL (0,0)-(MaxX,MaxY),-6,0
76 BF      LINE (MaxX-5,0)-(MaxX-5,MaxY),2
77 lG      PUT (MaxX-4,0),ShiftArea
78 r5      WINDOW OUTPUT 3
79 rT      GET (1,1)-(1,ImageHeight),ShiftArea
80 OJ      SCROLL (1,1)-(ImageWidth,ImageHeight),-1,0
81 q0      PUT (ImageWidth,1),ShiftArea
82 G99     END IF
83 HA7     END IF
84 S00     y
85 eg6     IF x>124 AND x<149 AND ((y>2 AND y<14) OR (y>28
          AND y<40)) THEN
86 C18     Change=-1
87 uC      WINDOW OUTPUT 2
88 uw      IF y<14 THEN                             ' Auf
89 EFA     GET (0,0)-(MaxX,4),ShiftArea
90 As      SCROLL (0,0)-(MaxX,MaxY),0,-6
91 MF      LINE (0,MaxY-5)-(MaxX,MaxY-5),2
92 Tg      PUT (0,MaxY-4),ShiftArea
93 6K      WINDOW OUTPUT 3
94 fV      GET (1,1)-(ImageWidth,1),ShiftArea
95 Tn      SCROLL (1,1)-(ImageWidth,ImageHeight),0,-1
96 Kg      PUT (1,ImageHeight),ShiftArea
97 a19     ELSE                                     ' Ab
98 eTA     GET (0,MaxY-4)-(MaxX,MaxY),ShiftArea
99 7c      SCROLL (0,0)-(MaxX,MaxY),0,6
100 lb     LINE (0,5)-(MaxX,5),2
101 E9     PUT (0,0),ShiftArea
102 FT     WINDOW OUTPUT 3
103 yc     GET (1,ImageHeight)-(ImageWidth,ImageHeight),Shif
          tArea
104 v2     SCROLL (1,1)-(ImageWidth,ImageHeight),0,1
105 OL     PUT (1,1),ShiftArea
106 eX9    END IF
107 fY7    END IF
108 gZ5    END IF
109 CO3    WEND
110 8k1    RETURN
111 J70    MenuHandler:
112 E42    ON MENU(1) GOTO NewMenu,OpenMenu,SaveMenu,SaveAsMenu,Size
          Menu,QuitMenu
113 sS0    NewMenu:
114 W62    IF Change THEN
115 4L4    WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
116 Er     PRINT
117 LS     PRINT " Die Zeichnung wurde abgeändert"
118 7J     PRINT " jedoch noch nicht abgespeichert."
119 Hu     PRINT
120 uG     INPUT " Willst Du wirklich von Neuem beginnen (J/N)? "
          ,Answer$
121 J1     WINDOW CLOSE 4
122 2o     IF UCASE$(Answer$)<>"J" THEN RETURN
123 vo3    END IF
124 dv2    Change=0
125 cy0    ' Rechteckige Abgrenzung des Bildes
126 dr2    WINDOW OUTPUT 3
127 LP     LINE (0,0)-(ImageWidth+1,ImageHeight+1),1,bf
128 kh     LINE (0,0)-(ImageWidth+1,ImageHeight+1),2,b
129 RIO    ' Raster
130 bt2    WINDOW OUTPUT 2
131 3b     LINE (0,0)-(MaxX,MaxY),1,bf
132 4c0    Grid:
133 Mu     ' Vertikale Linien
134 Po2    FOR i=5 TO MaxX-5 STEP 6
135 bL4    LINE (i,0)-(i,MaxY),2
136 dt3    NEXT i
137 bk0    ' Horizontale Linien
138 Zu2    FOR i=5 TO MaxY-5 STEP 6
139 CS4    LINE (0,i)-(MaxX,i),2
140 hx3    NEXT i
141 dF2    RETURN
142 M20    OpenMenu:
143 zZ2    IF Change THEN
144 Xo4    WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
145 hK     PRINT
146 ov     PRINT " Die Zeichnung wurde abgeändert"
147 am     PRINT " jedoch noch nicht abgespeichert."
148 kN     PRINT
149 RN     INPUT " Willst Du wirklich ein anderes Bild laden (J/N
          )? ",Answer$
150 mU     WINDOW CLOSE 4
151 VH     IF UCASE$(Answer$)<>"J" THEN RETURN
152 OH3    END IF
153 O10    InputName:
154 182    WINDOW 4,"Bildname",(50,100)-(550,150),0,1
155 rU     PRINT
156 sV     PRINT
157 2N     INPUT " Name des Bildes: ",ImageName$
158 uc     WINDOW CLOSE 4
159 82     IF ImageName$="" THEN RETURN
160 sm     MENU OFF
161 Ez     MOUSE OFF
162 K1     OPEN ImageName$ FOR INPUT AS #1
163 DH     INPUT #1,a$
164 Sq     IF a$<>"* FILETYPE=IMAGE" THEN
165 s94    WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
166 2f     PRINT
167 35     PRINT " Datei nicht artgerecht."
168 4h     PRINT
169 ya     PRINT " (Taste drücken um fortzufahren)"
170 2L     MOUSE OFF : MENU OFF
171 y9     WHILE INKEY$="" : SLEEP : WEND
172 1l     MOUSE ON : MENU ON
173 1o     CLOSE #1
174 tB     GOTO InputName
175 le3    END IF
176 QU2    INPUT #1,a$
177 sZ     INPUT #1,a$,Image(0),Image(1),Image(2)
178 eL     IF Image(0)<1 OR Image(0)>105 OR Image(1)<1 OR Image(1
          )>42 OR Image(2)<>1 THEN
179 6N4    WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
180 Gt     PRINT
181 sI     IF Image(2)<>1 THEN
182 wZ6    PRINT " Tiefe anders als 1"
183 YH5    ELSE
184 h16    PRINT " nicht erlaubte Abmessungen"
185 vo5    END IF
186 Gg4    PRINT " Ich kann die Zeichnung nicht einlesen"
187 NO     PRINT
188 Ht     PRINT " (Taste drücken um fortzufahren)"
189 Le     MOUSE OFF : MENU OFF
190 HS     WHILE INKEY$="" : SLEEP : WEND
191 K4     MOUSE ON : MENU ON
192 K7     CLOSE #1
193 CU     GOTO InputName
194 4x3    END IF
195 CB2    k=3
196 x0     FOR i=1 TO Image(1)
197 lp4    INPUT #1,a$
198 RP     Image(k)=VAL(MID$(a$,6))
199 PN     k=k+1
200 LN     FOR j=1 TO (Image(0)+15)\16-1
201 mi6    INPUT #1,Image(k)
202 SQ     k=k+1
203 2g3    NEXT j,i
204 WJ2    CLOSE #1
205 cz     ImageWidth=Image(0) : ImageHeight=Image(1)
206 p7     WINDOW OUTPUT 2
207 tz     CLS
208 Tp     MaxX=(ImageWidth-1)*6+4 : MaxY=(ImageHeight-1)*6+4
209 P1     GOSUB Grid
210 GB     GOSUB SmallWindow
211 ho     MOUSE OFF : MENU OFF
212 63     LINE (0,0)-(ImageWidth+1,ImageHeight+1),2,b
213 SC     PUT (1,1),Image,PSET
214 fy     FOR j=1 TO ImageHeight
215 M14    FOR i=1 TO ImageWidth
216 5J6    WINDOW OUTPUT 3
217 Za     k=POINT(1,j)
218 -1J    WINDOW OUTPUT 2
219 CU     LINE ((i-1)*6,(j-1)*6)-STEP(4,4),k,bf

```

Listing 2. Fortsetzung

```

220 Lz3 NEXT i,j
221 y52 MENU ON : MOUSE ON
222 DV Change=0
223 xZ1 RETURN
224 Nz0 SaveAsMenu:
225 uH2 WINDOW 4,"Bildname",(50,100)-(550,150),0,1
226 Od PRINT
227 1e PRINT
228 qx INPUT " Name des abzuspichernden Bildes: ",ImageName$
229 3l WINDOW CLOSE 4
230 YB0 SaveMenu:
231 mE2 IF ImageName$="" THEN GOTO SaveAsMenu
232 wj MENU OFF : MOUSE OFF
233 Ma WINDOW OUTPUT 3
234 JJ GET (1,1)-(ImageWidth,ImageHeight),Image
235 60 OPEN ImageName$ FOR OUTPUT AS #1
236 MA PRINT #1,"!* FILETYPE=IMAGE"
237 yw ClearPath ImageName$,LabelName$
238 68 PRINT #1,LabelName$+": "
239 Om PRINT #1," DATA ";
240 Ku WRITE #1,3+ImageHeight*((ImageWidth+15)\16),Image(0),Image(1),1
241 wv k=3
242 10 FOR i=1 TO ImageHeight
243 4q4 PRINT #1," DATA ";
244 gC FOR j=1 TO (ImageWidth+15)\16-1
245 KN6 PRINT #1,Image(k);",";
246 A8 k=k+1
247 Sj5 NEXT j
248 wp4 PRINT #1,Image(k)
249 DB k=k+1
250 Tj3 NEXT i
251 H42 CLOSE #1
252 hz Change=0
253 Ub MENU ON : MOUSE ON
254 S41 RETURN
255 bQ0 SizeMenu:
256 jX2 OldImageHeight=ImageHeight : OldImageWidth=ImageWidth
257 rg ImageWidth=0 : ImageHeight=0
258 e3 GOSUB AskSize
259 4M Answer$="J"
260 w9 IF ImageWidth<OldImageWidth OR ImageHeight<OldImageHeight THEN
261 Qh4 WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
262 aD PRINT
263 R9 PRINT " Die neuen Abmessungen sind kleiner als die vorhergehenden"
264 SW PRINT " somit könnte ein Teil des Bildes verlorengehen"
265 dG PRINT
266 pW INPUT " Fortfahren (J/N)? ",Answer$
267 fN WINDOW CLOSE 4
268 G93 END IF
269 052 IF UCASE$(Answer$)<>"J" OR (ImageWidth=OldImageWidth AND ImageHeight=OldImageHeight) THEN
270 nX4 ImageWidth=OldImageWidth : ImageHeight=OldImageHeight
271 jL RETURN
272 KD3 END IF
273 LE2 NewMaxX=(ImageWidth-1)*6+4 : NewMaxY=(ImageHeight-1)*6+4
274 HJ IF MaxX<>NewMaxX THEN
275 wE4 WINDOW OUTPUT 2
276 5o IF NewMaxX>MaxX THEN
277 2R6 LINE (MaxX+1,0)-(NewMaxX,NewMaxY),1,bf
278 5J WINDOW OUTPUT 3
279 WL LINE (OldImageWidth+1,1)-(ImageWidth,ImageHeight),1,bf
280 7q5 ELSE
281 zy6 LINE (NewMaxX+1,0)-(MaxX,MaxY),0,bf
282 9N WINDOW OUTPUT 3
283 PQ LINE (ImageWidth+1,0)-(OldImageWidth+1,OldImageHeight+1),0,bf
284 WP5 END IF
285 XQ3 END IF
286 Y22 IF MaxY<>NewMaxY THEN
287 8Q4 WINDOW OUTPUT 2
288 T4 IF NewMaxY>MaxY THEN
289 Fw6 LINE (0,MaxY+1)-(NewMaxX,NewMaxY),1,bf
290 HV WINDOW OUTPUT 3
291 OE LINE (1,OldImageHeight+1)-(ImageWidth,ImageHeight),1,bf
292 J25 ELSE
293 ND6 LINE (0,NewMaxY+1)-(MaxX,MaxY),0,bf
294 LZ WINDOW OUTPUT 3
295 3F LINE (0,ImageHeight+1)-(OldImageWidth+1,OldImageHeight+1),0,bf
296 ib5 END IF
297 jc3 END IF
298 c82 Change=-1
299 3B MaxX=NewMaxX : MaxY=NewMaxY
300 NN GET (1,1)-(ImageWidth,ImageHeight),Image
301 Me WINDOW OUTPUT 2
302 uW GOSUB Grid
303 lg GOSUB SmallWindow
304 uc PUT (1,1),Image
305 bY LINE (0,0)-(ImageWidth+1,ImageHeight+1),2,b
306 Iu1 RETURN
307 410 QuitMenu:
308 eE2 IF Change THEN
309 CT4 WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
310 Mz PRINT
311 LU PRINT " Die abgeänderte Zeichnung wurde noch nicht abgelegt."
312 O1 PRINT
313 KT INPUT " Soll sie vor dem Ausstieg abgelegt werden (J/N)? ",Answer$
314 iJ IF UCASE$(Answer$)<>"N" THEN GOSUB SaveAsMenu
315 Iu3 END IF
316 Ja2 WINDOW 4,"Achtung !!!",(50,100)-(550,150),0,1
317 T6 PRINT
318 U7 PRINT
319 MD INPUT " Willst Du sicher aussteigen (J/N)? ",Answer$
320 WE WINDOW CLOSE 4
321 F1 IF UCASE$(Answer$)<>"J" THEN RETURN
322 Om MENU RESET
323 P5 WINDOW CLOSE 2
324 VC WINDOW CLOSE 3
325 16 SCREEN CLOSE 1
326 KW STOP
327 Oe0 ' Größeneingabe
328 JU AskSize:
329 k32 WINDOW 4,"Größeneingabe",(50,100)-(550,150),0,1
330 gJ PRINT
331 rJ0 InputWidth:
332 ak2 INPUT " Breite der Zeichnung in Pixel (max 105): ",ImageWidth
333 gN IF ImageWidth<1 OR ImageWidth>105 THEN
334 BX4 PRINT "*** FEHLER: Maß außerhalb des zugelassenen Bereiches"
335 lm GOTO InputWidth
336 MF3 END IF
337 nQ2 PRINT
338 h40 InputHeight:
339 vu2 INPUT " Höhe der Zeichnung in Pixel (max 42): ",ImageHeight
340 7o IF ImageHeight<1 OR ImageHeight>42 THEN
341 Ie4 PRINT "*** FEHLER: Maß außerhalb des zugelassenen Bereiches"
342 E7 GOTO InputHeight
343 TM3 END IF
344 uc2 WINDOW CLOSE 4
345 vX1 RETURN
346 wK0 ErrorHandler:
347 Ca2 WINDOW 4,"FEHLER "+STR$(ERR),(50,100)-(550,150),0,1
348 yb PRINT
349 zc PRINT
350 f8 IF ERR=49 THEN
351 6S4 PRINT " Disk nicht vorhanden"
352 zb3 ELSEIF ERR=53 THEN
353 sq4 PRINT " Datei nicht gefunden !"
354 Hx3 ELSEIF ERR=57 THEN
355 Py4 PRINT " Device I/O Error"
356 yZ3 ELSEIF ERR=61 THEN
357 YN4 PRINT " Disk ist voll"
358 Cq3 ELSEIF ERR=64 THEN
359 6H4 PRINT " Falscher Dateiname"
360 UC3 ELSEIF ERR=68 THEN
361 OJ4 PRINT " Das angesprochene Gerät ist nicht verfügbar"
362 3e3 ELSEIF ERR=70 THEN
363 A14 PRINT " Disk ist schreibgeschützt"
364 LO3 ELSEIF ERR=74 THEN
365 y94 PRINT " Unbekannter Diskname"
366 qJ3 END IF
367 Hu2 PRINT
368 81 PRINT " (Taste drücken um fortzufahren)"

```

```

369 AL WHILE INKEY$="" : SLEEP : WEND
370 K2 WINDOW CLOSE 4
371 vD RESUME ResumeError
372 aA0 SmallWindow:
373 pb2 WINDOW 3,"SIZE "+STR$(ImageWidth)+" X"+STR$(ImageHeight),
(420,198)-(631,241),18,1
374 Um0 ' Farbpalette
375 5M2 LINE (167,2)-(209,21),1,b
376 00 LINE (167,21)-(209,40),1,bf
377 XZ LINE (167,2+PaintColor*19)-STEP(42,19),3,b
378 Ub0 ' Cursortasten zur Verschiebung des Bildes
379 dU ' Cursor links
380 Nx2 LINE (111,15)-(136,27),1,b
381 nw LINE (114,21)-(133,21),1
382 oH AREA (114,21) : AREA STEP(6,-2) : AREA STEP(0,4)
383 gm AREAFILL
384 lv0 ' Cursor rechts
385 xe2 LINE (137,15)-(162,27),1,b
386 EU LINE (140,21)-(159,21),1
387 mb AREA (159,21) : AREA STEP(-6,2) : AREA STEP(0,-4)
388 lr AREAFILL
389 qg0 ' Cursor auf
390 492 LINE (124,2)-(149,14),1,b
391 3I LINE (136,4)-(137,12),1,b
392 am AREA (136,4) : AREA STEP(-4,3) : AREA STEP(9,0) : AREA ST

```

```

EP(-4,-3)
393 qw AREAFILL
394 cf0 ' Cursor ab
395 Jq2 LINE (124,28)-(149,40),1,b
396 Bn LINE (136,30)-(137,38),1,b
397 EH AREA (136,38) : AREA STEP(-4,-3) : AREA STEP(9,0) : AREA
STEP(-4,3)
398 v1 AREAFILL
399 Wb0 ResumeError:
400 Za2 WINDOW 2
401 4I WINDOW OUTPUT 3
402 FY CALL WindowToFront#(WINDOW(7))
403 u1 MENU ON : MOUSE ON
404 sU1 RETURN
405 Da0 SUB ClearPath(PathName$,NoPathName$) STATIC
406 kg2 NoPathName$=MID$(PathName$,INSTR(PathName$,"")+1)
407 PP i%=1
408 b9 WHILE i%<>0
409 9o4 i%=INSTR(NoPathName$,"/")
410 Bk NoPathName$=MID$(NoPathName$,i%+1)
411 4s3 WEND
412 gi1 END SUB
(C) 1990 M&T

```

Listing 2. Schluß

Programmname: Brush2BasicImage
Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache: Amiga-Basic
Bemerkung: siehe Text

Programmautor: Maurizio Lotauro

```

-----
1 d40 '*****
2 Op '* *
3 02 '* Programm "Brush2BasicImage" *
4 Qr '* *
5 Nr '* Konvertiert eine BRUSH-Datei in DATAS die man *
6 59 '* in Basicprogrammen einbinden kann. *
7 Tu '* *
8 8X '* Verwendet intuition.library *
9 Vw '* *
10 mD '*****
11 H3 Init:
12 LW2 CLEAR
13 0b LIBRARY "intuition.library"
14 In DEFINT a-z
15 ya DECLARE FUNCTION AllocRemember&() LIBRARY
16 G7 ON ERROR GOTO ErrorHandler
17 Rk RememberKey&=0
18 oJ0 InputInFile:
19 rx2 CLS
20 Zk PRINT : INPUT " Name der Inputdatei: ",FileName$
21 5d OPEN FileName$ FOR INPUT AS #1
22 ru a$=INPUT$(4,1) ' liest FORM
23 G4 IF a$<>"FORM" THEN
24 8P4 PRINT " *** FEHLER : keine IFF-datei"
25 dQ CLOSE #1
26 H5 GOTO EndPrGm
27 NG3 END IF
28 wC2 dummy$=INPUT$(4,1)
29 Cx a$=INPUT$(4,1) ' liest ILBM
30 ob IF a$<>"ILBM" THEN
31 bX4 PRINT " *** FEHLER : keine Bilddatei"
32 kX CLOSE #1
33 OC GOTO EndPrGm
34 UN3 END IF
35 S10 InputOutFile:
36 ya2 PRINT : INPUT " Name der Outputdatei: ",ImageName$
37 xs OPEN ImageName$ FOR OUTPUT AS #2
38 2G PRINT : PRINT " W A I T"
39 VV dummy$=INPUT$(8,1)
40 kA BrushWidth=CVI(INPUT$(2,1)) ' Brushbreite
41 1S BrushHeight=CVI(INPUT$(2,1)) ' Brushhöhe
42 AQ dummy$=INPUT$(4,1)
43 aj BrushDepth=ASC(INPUT$(1,1)) ' Anzahl der Bild

```

```

ebenen (BITPLANES)
44 uM dummy$=INPUT$(1,1)
45 Ll CompFlag=ASC(INPUT$(1,1)) ' Kompressionsfla
g
46 Wy IF CompFlag>1 THEN
47 7k4 PRINT
48 tR PRINT " Sorry, diese Art von Kompressionsalgorithmus h
abe ich nicht"
49 PQ PRINT " implementiert"
50 fT GOTO EndPrGm
51 le3 END IF
52 ok2 dummy$=INPUT$(9,1)
53 Oc a$=INPUT$(4,1)
54 db WHILE a$<>"BODY" ' Sucht die Ch
unk ID BODY
55 CZ4 ChunkSize&=CVI(INPUT$(4,1))
56 IA dummy$=INPUT$(ChunkSize&,1)
57 Sg a$=INPUT$(4,1)
58 NB3 WEND
59 2X2 NData&=CVI(INPUT$(4,1)) ' Byteanzahl in
BODY
60 VB BrushBody&=AllocRemember&(VARPTR(RememberKey&),NData&+(ND
ata& AND 1),65537&)
61 c0 IF BrushBody&=0 THEN ERROR 7
62 M4 PRINT : PRINT " Reading brush bitplanes..."
63 5W FOR i&=0 TO NData&-1
64 Vh4 b$=INPUT$(1,1)
65 Cu POKE BrushBody&+i&,ASC(b$)
66 n93 NEXT i&
67 J62 CLOSE #1
68 3y nwords=(BrushWidth+7)\8
69 tP nwords=nwords+(nwords AND 1) ' Byteanzahl pro Ze
ile
70 hf nwperplane=nwords*BrushDepth ' Byteanzahl pro Ze
ile pro Bitplane
71 XO totalnum&=nwperplane*BrushHeight ' Byteanzahl total
72 Op0 ' Wenn CompFlag=0 dann liegt keine Kompression vor
73 o0 ' Wenn CompFlag=1 dann wird dekomprimiert
74 vM2 IF CompFlag=1 THEN
75 KW4 LOCATE CSRLIN-1
76 kk PRINT " Decompressing data..."
77 LD NewArray&=AllocRemember&(VARPTR(RememberKey&),totalnum&
+(totalnum& AND 1),65537&)
78 1A IF NewArray&=0 THEN ERROR 7
79 ka index1&=0 : index2&=0
80 oK WHILE index1&<NData&
81 wB6 value=PEEK(BrushBody&+index1&)
82 rf IF value<128 THEN
83 Cx8 FOR i=1 TO value+1
84 RMA index1&=index1&+1
85 U4 POKE NewArray&+index2&,PEEK(BrushBody&+index1&)
86 aX index2&=index2&+1

```

Listing 3. Fortsetzung

```

87 q69     NEXT i
88 uR7     ELSEIF value>128 THEN
89 WR8     index1&=index1&+1
90 dM     anothervalue=PEEK(BrushBody&+index1&)
91 1x     FOR i=1 TO (257-value)
92 KbA     POKE NewArray&+index2&,anothervalue
93 he     index2&=index2&+1
94 xd9     NEXT i
95 TM7     END IF
96 dY6     index1&=index1&+1
97 Oo5     WEND
98 FU4     BrushBody&=NewArray&
99 XQ3     END IF
100 E32    PRINT #2,"'* FILETYPE=IMAGE"
101 mk     ClearPath ImageName$,LabelName$
102 y1     PRINT #2,LabelName$+" ":
103 sf     PRINT #2," DATA ";
104 4y     WRITE #2,3+BrushDepth*BrushHeight*((BrushWidth+15)\16),B
rushWidth,BrushHeight,BrushDepth
105 WTO ' Transformation der Daten in einzelne Bitebenen
106 p12    LOCATE CSRLIN-1
107 Q1     PRINT " Decoding bitplanes... "
108 9H     FOR k=0 TO BrushDepth-1
109 2w4     uplim&=(totalnum&-nwperplane+k*nwords)
110 Ww     FOR i&=(k*nwords) TO uplim& STEP nwperplane
111 On6     PRINT #2," DATA ";
112 QH     FOR j=0 TO nwords-3 STEP 2
113 GG8     PRINT #2,PEEKW(BrushBody&+j+i&);",";
114 Ja7     NEXT j
115 rh6     PRINT #2,PEEKW(BrushBody&+j+i&)
116 bx5     NEXT i&
117 Og3     NEXT k
118 CO2    CLOSE #2
119 EI     LOCATE CSRLIN-3
120 j1     PRINT " D O N E" : PRINT : PRINT SPACES(26)
121 6c     BEEP
122 p40    EndPrgm:
123 vy2    CALL FreeRemember&(VARPTR(RememberKey&),1)
124 Mz     PRINT
125 SY     INPUT " Sollen weitere Brushes konvertiert werden (J/N)?
",Answer$
126 ZY     IF UCASE$(Answer$)="J" THEN GOTO Init
127 hv     LIBRARY CLOSE
128 oj1    END
129 iF0    ErrorHandler:
130 252    CALL FreeRemember&(VARPTR(RememberKey&),1)
131 JX     PRINT : PRINT " *** FEHLER ";ERR;" :";
132 KR     IF ERR=7 THEN
133 Xs4     PRINT " Ungenügend Speicherplatz vorhanden"
134 ov3     ELSEIF ERR=49 THEN
135 cy4     PRINT " Disk nicht vorhanden"
136 V73     ELSEIF ERR=53 THEN
137 OM4     PRINT " Datei nicht gefunden !"
138 fJ3     ELSEIF ERR=55 THEN
139 Ja4     PRINT " Datei bereits offen !"
140 pV3     ELSEIF ERR=57 THEN
141 xW4     PRINT " Device I/O Error"
142 W73     ELSEIF ERR=61 THEN
143 4L4     PRINT " Disk voll"
144 kO3     ELSEIF ERR=64 THEN
145 ep4     PRINT " Falscher Dateiname"
146 2k3     ELSEIF ERR=68 THEN
147 wr4     PRINT " Das angesprochene Gerät ist nicht verfügbar"
148 bC3     ELSEIF ERR=70 THEN
149 iq4     PRINT " Disk ist schreibgeschützt"
150 tY3     ELSEIF ERR=74 THEN
151 Wh4     PRINT " Unbekannter Diskname"
152 OH3     END IF
153 pS2    PRINT
154 gO     PRINT " (Taste drücken um fortzufahren)"
155 it     WHILE INKEY$="" : SLEEP : WEND
156 4A     CLS
157 16     RESUME Init
158 lL0 ' Routine zur Entnahme des Dateinamens
159 Fc     SUB ClearPath(PathName$,NoPathName$) STATIC
160 mI2    NoPathName$=MID$(PathName$,INSTR(PathName$,"")+1)
161 RR     i%=1
162 dB     WHILE i%<>0
163 Bq4     i%=INSTR(NoPathName$,"/")
164 Dm     NoPathName$=MID$(NoPathName$,i%+1)
165 6u3    WEND
166 ik1    END SUB
(C) 1990 M&T

```

Listing 3. Schluß

Programmname: ShowBasicImage

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: Amiga-Basic

Bemerkung: siehe Text

Programmautor: Maurizio Lotauro

```

1 zGO '*****
2 Ef '* *
3 Nd '* Programm "ShowBasicImage" *
4 Gh '* *
5 70 '* Liest Dateien ein, die mit "Brush2BasicImage" *
6 x9 '* erzeugt wurden. *
7 Jk '* *
8 yN '* Verwendet intuition.library *
9 Lm '* *
10 8P '*****
11 yk2 DEFINT a-z
12 Na LIBRARY "intuition.library"
13 wY DECLARE FUNCTION AllocRemember&() LIBRARY
14 E5 ON ERROR GOTO ErrorHandler
15 An0 InputName:
16 QJ2 RememberKey&=0
17 uf PRINT : PRINT " Name des Bildes : ",ImageName$
18 Oh OPEN ImageName$ FOR INPUT AS #1
19 tx INPUT #1,a$
20 8W IF a$<>"'* FILETYPE=IMAGE" THEN
21 hK4 PRINT
22 zb PRINT " *** FEHLER : Datei nicht artgerecht"
23 JM PRINT
24 dP PRINT " (Taste drücken um fortzufahren)"
25 cn WHILE INKEY$="" : SLEEP : WEND
26 eR CLOSE #1

```

```

27 z5 CLS
28 Xp GOTO InputName
29 P13 END IF
30 482 INPUT #1,a$
31 Uh INPUT #1,a$,ImageWidth,ImageHeight,ImageDepth
32 ge PRINT : PRINT : PRINT : PRINT " Bildschirmauflösung : "
: PRINT
33 m2 PRINT : PRINT SPC(20) "1 - Low resolution, non interlace
d"
34 4X PRINT : PRINT SPC(20) "2 - High resolution, non interlace
d"
35 OX PRINT : PRINT SPC(20) "3 - Low resolution, interlaced"
36 g2 PRINT : PRINT SPC(20) "4 - High resolution, interlaced"
37 3Q0 InputScreenMode:
38 iG2 a$=INKEY$
39 BH WHILE a$<"1" OR a$>"4" : a$=INKEY$ : WEND
40 tO ScreenMode=VAL(a$)
41 J6 ScreenWidth=320*(2-(ScreenMode MOD 2))
42 nG ScreenHeight=256*(1+(ScreenMode\3))
43 PA IF ImageWidth>ScreenWidth OR ImageHeight>ScreenHeight T
HEN
44 4h4 PRINT
45 wX PRINT " *** FEHLER : die Auflösung erlaubt nur Zeichnun
gen kleinerer Maße"
46 6j PRINT
47 Oc PRINT " (Taste drücken um fortzufahren)"
48 zA WHILE INKEY$="" : SLEEP : WEND
49 6A LOCATE CSRLIN-3
50 Dn PRINT SPACES(75) : PRINT : PRINT SPACES(75)
51 EE LOCATE CSRLIN-4
52 84 GOTO InputScreenMode
53 ng3 END IF
54 xH2 IF ImageDepth=5 AND (ScreenMode MOD 2)=0 THEN
55 Fs4 PRINT
56 SK PRINT " ***FEHLER : die Auflösung erlaubt nur weniger B
itebenen"
57 Hu PRINT

```

```

58 Bn PRINT " (Taste drücken um fortzufahren)"
59 AL WHILE INKEY$="" : SLEEP : WEND
60 HL LOCATE CSRLIN-3
61 Oy PRINT SPACE$(75) : PRINT : PRINT SPACE$(75)
62 PP LOCATE CSRLIN-4
63 JF GOTO InputScreenMode
64 yr3 END IF
65 xd2 PRINT : PRINT SPC(30) "LOADING DATA..."
66 Tz0 ' Erzeugt die IMAGE-Struktur und die Bilddaten
67 UY2 ImageAdd=&AllocRemember&(VARPTR(RememberKey&),20,65537&)
68 BS IF ImageAdd=&0 THEN ERROR 7
69 B3 ImageDataSize=&ImageHeight*((ImageWidth+15)\16)*ImageDepth
70 nN ImageData=&AllocRemember&(VARPTR(RememberKey&),ImageDataSize&*2,65539&)
71 bf IF ImageData=&0 THEN ERROR 7
72 24 k&=0
73 FD FOR m=1 TO ImageDepth
74 Jg4 FOR i=1 TO ImageHeight
75 nr6 INPUT #1,a$
76 nu POKEW ImageData&+k&,VAL(MID$(a$,6))
77 gk k&=k&+2
78 OW FOR j=1 TO (ImageWidth+15)\16-1
79 1p8 INPUT #1,d
80 tL POKEW ImageData&+k&,d
81 ko k&=k&+2
82 VN3 NEXT j,i,m
83 OZ2 POKEW ImageAdd&,(ScreenWidth-ImageWidth)\2 ' x offset
84 vb POKEW ImageAdd&+2,(ScreenHeight-ImageHeight)\2 ' y offset
85 Bj POKEW ImageAdd&+4,ImageWidth ' Breite
86 uB POKEW ImageAdd&+6,ImageHeight ' Höhe
87 6o POKEW ImageAdd&+8,ImageDepth ' Tiefe
88 Oc POKEW ImageAdd&+10,ImageData& ' Adresse der Bilddatei
89 3P POKE ImageAdd&+14,2^ImageDepth-1 ' PlanePick
90 gT CLOSE #1

```

```

91 8g SCREEN 1,ScreenWidth,ScreenHeight,ImageDepth,ScreenMode
92 3d0 ' Das Window wird zum Auffinden der Screenadresse
misbraucht
93 3P2 WINDOW 2,,0,1
94 1s ScreenAdd=&PEEK(LWINDOW(7)+46)
95 gC BEEP
96 nS CALL DrawImage&(ScreenAdd&+84,ImageAdd&,0,0)
97 OT0 ' Löscht den Mouse- und Tastaturbuffer
98 vk2 WHILE INKEY$<>" " : WEND : WHILE MOUSE(0)<>0 : WEND
99 c0 WHILE INKEY$="" AND MOUSE(0)=0 : SLEEP : WEND
100 oU WINDOW CLOSE 2
101 PU SCREEN CLOSE 1
102 VJ GOTO EndPrm
103 bp0 ErrorHandler:
104 s62 PRINT : PRINT " *** FEHLER ";ERR;" :";
105 t0 IF ERR=7 THEN
106 Mw4 PRINT " ungenügend Speicherplatz"
107 N43 ELSEIF ERR=49 THEN
108 BX4 PRINT " Disk nicht vorhanden"
109 4g3 ELSEIF ERR=53 THEN
110 xv4 PRINT " Datei nicht gefunden !"
111 Es3 ELSEIF ERR=55 THEN
112 I94 PRINT " Datei bereits offen !"
113 O43 ELSEIF ERR=57 THEN
114 W54 PRINT " Device I/O Error"
115 Hv3 ELSEIF ERR=64 THEN
116 BM4 PRINT " Falscher Dateiname"
117 ZH3 ELSEIF ERR=68 THEN
118 UL4 PRINT " Das angesprochene Gerät ist nicht vorhanden"
119 O33 ELSEIF ERR=74 THEN
120 104 PRINT " Unbekannter Diskname"
121 tm3 END IF
122 p40 EndPrm:
123 vy2 CALL FreeRemember&(VARPTR(RememberKey&),1)
124 es LIBRARY CLOSE
125 lg1 END
(C) 1990 M&T

```

Listing 4. Schluß

Programmname:	SuperMenuII Demo
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic
Bemerkung:	siehe Text
Programmautor: Maurizio Lotauro	

```

1 DU0 '*****
2 Ef '*
3 mX '* Demoprogramm zum Gebrauch der SUPERMENUES-Routinen *
4 Gh '*
5 HY '*****
6 062 LIBRARY "graphics.library"
7 IV LIBRARY "intuition.library"
8 rT DECLARE FUNCTION AllocRemember&() LIBRARY
9 Of DECLARE FUNCTION TextLength%() LIBRARY
10 3I ON ERROR GOTO HandleErrors
11 dL INITIALIZE
12 YB PRINT
13 JJ PRINT " Die SUPERMENUES werden eingelesen, BEEP abwarten ."
14 OA MENU 1,0,1,"Erstes Menue"
15 RL MENU 2,0,1,"Zweites Menue"
16 ML MENU 3,0,1," " ' Menue 3 und 4 werden hier nicht verwendet
17 FI MENU 4,0,1," "
18 oJ Mexclude&=0
19 pc flags%=enabled%+comp%
20 dJ SMENUTEXT 1,1,0,flags%,Mexclude&,"", "ItemA", "0,1,"
21 mx SMENUTEXT 1,2,0,flags%,Mexclude&,"", "ItemB", "3,1,"
22 JJ flags%=enabled%+box%+command%
23 tG SMENUTEXT 1,3,0,flags%,Mexclude&,"Q", "Quit", "0,1,"
24 Nv Mexclude&=Item2&

```

```

25 Gc flags%=enabled%+check%+checked%+comp%
26 Pm SMENUTEXT 1,1,1,flags%,Mexclude&,"", " SubItem1",2,1,"", 0,1,50,0
27 MU Mexclude&=item1&
28 QY flags%=enabled%+comp%+check%
29 K6 SMENUTEXT 1,1,2,flags%,Mexclude&,"", " SubItem2",0,1,"", 0,1,50,8
30 OV Mexclude&=0
31 Ej flags%=enabled%+selectimage%
32 rV SMENUTEXT 2,1,0,flags%,Mexclude&,"", " SubMenu a", "0,1," rechts "3,1,0,0
33 RD SMENUTEXT 2,2,0,flags%,Mexclude&,"", " SubMenu a", "0,1," links "0,1,0,9
34 DU SMENUTEXT 2,3,0,flags%,Mexclude&,"", " SubMenu", "0,1," horizontal "0,1,0,18
35 7o SMENUTEXT 2,4,0,flags%,Mexclude&,"", " SubMenu a", "0,1," Tabelle "0,1,0,27
36 la flags%=enabled%+comp%+checked%+toggle%+check%
37 hU SMENUTEXT 2,1,1,flags%,Mexclude&,"", " Checked", "0,1," "0,1,100,0
38 d4 flags%=enabled%+comp%+checked%+check%
39 iC Mexclude&=Item3&
40 Re SMENUTEXT 2,1,2,flags%,Mexclude&,"", " nur oben", "0,1," "0,1,100,8
41 dl flags%=enabled%+comp%+check%
42 fD Mexclude&=Item2&
43 18 SMENUTEXT 2,1,3,flags%,Mexclude&,"", " nur unten", "0,1," "0,1,100,16
44 z1 flags%=enabled%+check%+toggle%+comp%
45 Fk Mexclude&=0
46 7E SMENUTEXT 2,1,4,flags%,Mexclude&,"", " Toggle", "0,1," "0,1,100,24
47 Rg flags%=enabled%+command%+comp%
48 L9 SMENUTEXT 2,1,5,flags%,Mexclude&,"P", " Taste", "0,1," "0,1,100,32
49 W1 flags%=enabled%+selectimage%
50 X7 SMENUTEXT 2,2,1,flags%,Mexclude&,"", " abwechselnd", "0,1," Text "0,1,-100,-8

```

Listing 5. Fortsetzung

```

51 l8 flags%=enabled%+comp%
52 nM SMENUTEXT 2,2,2,flags%,Mexclude&,"",Komplement ",0,1,"
",0,1,-100,0
53 nV flags%=enabled%+box%
54 q0 SMENUTEXT 2,2,3,flags%,Mexclude&,"",Box ",0,1,"
",0,1,-100,8
55 xe flags%=enabled%+none%
56 Rq SMENUTEXT 2,2,4,flags%,Mexclude&,"",nichts ",0,1,"
",0,1,-100,16
57 Lq flags%=comp%
58 GE SMENUTEXT 2,2,5,flags%,Mexclude&,"",abgeschaltet ",0,1,"
",0,1,-100,24
59 TG flags%=enabled%+comp%
60 aH SMENUTEXT 2,3,1,flags%,Mexclude&,"",links ",0,1","",0,1,1
00,0
61 Cd SMENUTEXT 2,3,2,flags%,Mexclude&,"",Zentrum ",2,1","",0,1,
148,0
62 hL SMENUTEXT 2,3,3,flags%,Mexclude&,"",rechts ",3,1","",0,1,
212,0
63 XC SMENUTEXT 2,4,1,flags%,Mexclude&,"",1 ",1,0","",0,1,100,
0
64 Ba SMENUTEXT 2,4,2,flags%,Mexclude&,"",2 ",0,2","",0,1,124,
0
65 n1 SMENUTEXT 2,4,3,flags%,Mexclude&,"",3 ",0,3","",0,1,148,
0
66 pL SMENUTEXT 2,4,4,flags%,Mexclude&,"",4 ",1,2","",0,1,100,
8
67 Rm SMENUTEXT 2,4,5,flags%,Mexclude&,"",5 ",1,3","",0,1,124,
8
68 jA SMENUTEXT 2,4,6,flags%,Mexclude&,"",6 ",2,0","",0,1,148,
8
69 tS SMENUTEXT 2,4,7,flags%,Mexclude&,"",7 ",2,3","",0,1,100,
16
70 Bq SMENUTEXT 2,4,8,flags%,Mexclude&,"",8 ",3,0","",0,1,124,
16
71 tJ SMENUTEXT 2,4,9,flags%,Mexclude&,"",9 ",3,2","",0,1,148,
16
72 Gr RESTORE Alogo
73 Dz READ i%
74 gD DIM MenuImage%(i%-1)
75 Rg FOR j%=0 TO i%-1 : READ MenuImage%(j%) : NEXT
76 qw RESTORE VAlogo
77 H3 READ i%
78 vL DIM MenuImage2%(i%-1)
79 9h FOR j%=0 TO i%-1 : READ MenuImage2%(j%) : NEXT
80 lW flags%=enabled%+selectimage%
81 pK Mexclude&=0
82 zp SMENUIMAGE 2,5,0,flags%,Mexclude&,"",MenuImage%(),MenuIa
ge2%(),0,40
83 re flags%=enabled%+comp%
84 zG SMENUTEXT 2,5,1,flags%,Mexclude&,"",M I G A ",0,1","",0
,1,100,19
85 W2 BEEP
86 kN PRINT
87 UB PRINT " SuperMenues bereit!"
88 GV WHILE NOT(MenuNum%=1 AND ItemNum%=3)
89 N24 MenuNum%=MENU(0)
90 NK WHILE MenuNum%=0
91 Pb6 MenuNum%=MENU(0)
92 vj5 WEND
93 uw4 ItemNum%=MENU(1)
94 U1 SUBMENU SubNum%
95 ek PRINT "MenuNum=";MenuNum%,"ItemNum=";ItemNum%,"SubNum="
;SubNum%
96 zn3 WEND
97 nv2 SMENU 1,0,0,0
98 yZ PRINT "Erstes Menue abgeschaltet. Taste druecken um fortzu
fahren"
99 oz WHILE INKEY$="" : SLEEP : WEND
100 u3 SMENU 1,0,0,1
101 w5 SMENU 1,1,0,0
102 s1 PRINT "Erstes Item abgeschaltet. Taste druecken um fortzuf
ahren"
103 s3 WHILE INKEY$="" : SLEEP : WEND
104 3D SMENU 1,1,0,1
105 2C SMENU 1,1,1,0
106 PQ PRINT "Erstes Subitem abgeschaltet. Taste druecken um fort
zufahren"
107 w7 WHILE INKEY$="" : SLEEP : WEND
108 9K SMENU 1,1,1,1

```

```

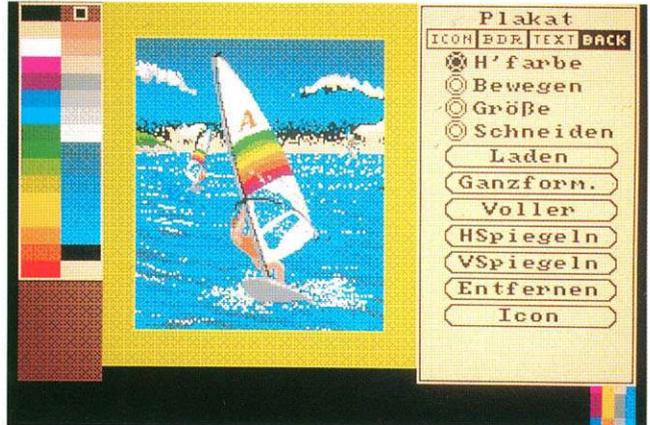
109 jh CALL SMENUOFF
110 2e LIBRARY CLOSE
111 XS1 END
112 lV0 HandleErrors:
113 ya2 PRINT " *** Fehler ";ERR; " : ";
114 Rp IF ERR=53 THEN
115 Dq4 PRINT
116 lV PRINT "Ihr mußt die Dateien 'intuition.bmap' und
'graphics.bmap'"
117 P3 PRINT "in der aktuellen oder in der 'Libs:'
Directory haben."
118 Jy PRINT "Diese Dateien findet ihr in der Basicdemos-
Schublade auf dem Disk EXTRAS 1.2."
119 rk3 END IF
120 us2 CALL SMENUOFF
121 bp LIBRARY CLOSE
122 id1 END
123 eP0 '* FILETYPE=IMAGE
124 3Y Alogo:
125 Ho2 DATA 192,100,27,1
126 Ny DATA -1,-1,-1,-1,-1,-1,-1,-1
127 Oz DATA -1,-1,-1,-1,-1,-1,-1,-1
128 E3 DATA -1,-1,-1,-2,16383,-1,-1
129 PG DATA -1,-1,-1,-4,16383,-1,-1
130 Kq DATA -1,-1,-1,-16,16383,-1,-4081
131 MD DATA -1,-1,-1,-32,16383,-1,-2
132 Lz DATA -1,-1,-1,-128,16383,-1,-1
133 ln DATA -1,-1,-1,-256,16383,-1,-15
134 xX DATA -1,-1,-1,-992,16383,-1,-2033
135 Vu DATA -1,-1,-1,-1952,16383,-1,-3585
136 rG DATA -1,-1,-1,-7712,16383,-1,-1
137 7T DATA -1,-1,-1,-14368,16383,-1,-8
138 CG DATA -1,-1,-1,4064,16383,-1,-2033
139 bK DATA -1,-1,-2,16352,16383,-1,-1
140 Gz DATA -1,-1,-8,32736,16383,-1,-1
141 6e DATA -1,-1,-15,-32,16383,-1,-1
142 tV DATA -1,-1,-57,-32,16383,-1,-4096
143 lv DATA -1,-1,-128,0,16383,-1,-4096
144 Xc DATA -1,-1,-449,-32,16383,-1,-4096
145 Os DATA -1,-1,-897,-32,16383,-1,-4096
146 SO DATA -1,-1,-3585,-32,16383,-1,-4096
147 Rf DATA -1,-1,-6145,-32,16383,-1,-4096
148 l4 DATA -1,-1,-28673,-32,16383,-1,-4096
149 sr DATA -1,-1,8191,-64,16383,-1,-4096
150 2q DATA -1,-64,1023,-1024,2047,-1,-4096
151 2h DATA -1,-1,-1,-1,-1,-1,-1,-4096
152 3i DATA -1,-1,-1,-1,-1,-1,-1,-4096
153 8t0 '* FILETYPE=IMAGE
154 mn VAlogo:
155 lI2 DATA 192,100,27,1
156 rS DATA -1,-1,-1,-1,-1,-1,-1,-1
157 sT DATA -1,-1,-1,-1,-1,-1,-1,-1
158 hA DATA -1,-1,-128,28687,-1,-14337,-1
159 IY DATA -1,-1,-130,-4129,-1,-30721,-4033
160 rB DATA -1,-1,-259,-8257,-2,2047,-2
161 Kq DATA -1,-1,-517,-16513,-4,2047,-1
162 jC DATA -1,-1,-1033,32511,-16,2047,-1
163 tB DATA -1,-1,-2066,-513,-32,2047,-1
164 pV DATA -1,-1,-4131,-1025,-124,2047,-137
165 Cg DATA -1,-1,-8261,-2049,-244,2047,-1
166 GA DATA -1,-1,-16521,-4097,-964,2047,-3856
167 Sr DATA -1,-1,32495,-8193,-1796,2047,-1
168 Oa DATA -1,-2,-545,-16385,-7684,2047,-1021
169 9k DATA -256,-8163,-1089,32767,-14340,2047,-81
170 kW DATA -65,30699,-2178,-1,4092,2047,-1033
171 pJ DATA -33,-17417,-4355,-2,16380,2047,-29
172 2Q DATA -17,-8721,-8709,-8,-4,2047,-4096
173 OL DATA -9,-4385,-17417,-16,0,2047,-1
174 tn DATA -5,-2241,30703,-57,-4,2047,-5
175 56 DATA -3,-1154,28639,-113,-4,2047,-2066
176 y7 DATA -2,-771,-24641,-449,-4,2047,-1025
177 be DATA -1,32251,-16513,-769,-4,2047,-1
178 lL DATA -1,-17417,32511,-3585,-4,2047,-512
179 6H DATA -1,-10262,-513,-7169,-8,2047,-1
180 Pz DATA -1,-8164,1016,127,-128,255,-1
181 Gr DATA -1,-1,-1,-1,-1,-1,-1,-1
182 XC DATA -1,-1,-1,-1,-1,-1,-1,-4096
(C) 1990 M&T

```

Listing 5. Schluß

Die glorreichen Drei

Animation · Video · Grafik

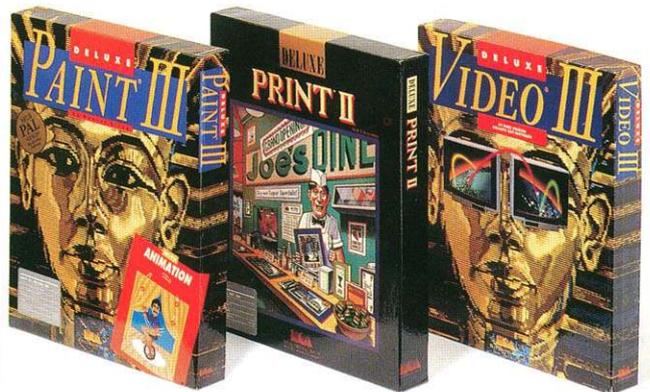


Create Interactive Multimedia presentations. . .

- Business presentations
- Entertainment
- Education
- Training



Click Buttons to Change Screen



Jetzt kommt Leben ins Bild. Mit der Multimedia-Software, die's ganz schön bunt mit Ihrem Amiga treibt – und ganz neue Perspektiven eröffnet, weil die Programme sich untereinander so gut verstehen.

Deluxe Paint III

Computergrafik mit Animationsfunktion in trickfilmartiger Darstellung. Mit wenigen Anweisungen geraten Ihre Bilder in Bewegung und verwandeln Ihren Amiga in ein Trickfilm-Studio.

Systemanforderungen:
Amiga 500, 1000, 2000 mit mindestens 1 Mbyte Arbeitsspeicher.

Bestell-Nr. 54138

DM 249,-*

Update von Deluxe Paint II auf Deluxe Paint III.

Bestell-Nr. 54138U

DM 99,-*

Deluxe Video III

Ihre Video-Produktion am Bildschirm. Es verarbeitet die Bilder aus Deluxe Paint III. Mit Umwandlung über Timing-Kontrolle

und Anim-Dateien. Die vier Einzelprogramme verknüpfen Musik, Ton, Bilder und Animation zu tollen Multimedia-Shows, die wieder auf Disketten oder Videokassetten überspielt werden können.

Systemanforderungen:
Amiga 500, 1000, 2000, 2500 mit mindestens 1 Mbyte RAM, 2 Diskettenlaufwerke (Festplatte empfohlen).

Bestell-Nr. 52586

DM 299,-*

Update von Deluxe Video 1.2 auf Deluxe Video III.

Bestell-Nr. 52586U

DM 149,-*

Deluxe Print II

Gestalten Sie Visitenkarten, Briefköpfe, Einladungen oder Poster nach professioneller Art: ein Programm für kreatives Design von Texten, Signets, Logos. Mit einer Palette leuchtender Farben.

Bestell-Nr. 52582

DM 199,-*

Updates erhalten Sie gegen Einsendung Ihrer Originaldiskette und eines Verrechnungsschecks an Markt & Technik Verlag AG, Kundenbetreuung, Update-Service, Hans-Pinsel-Straße 2, 8013 Haar

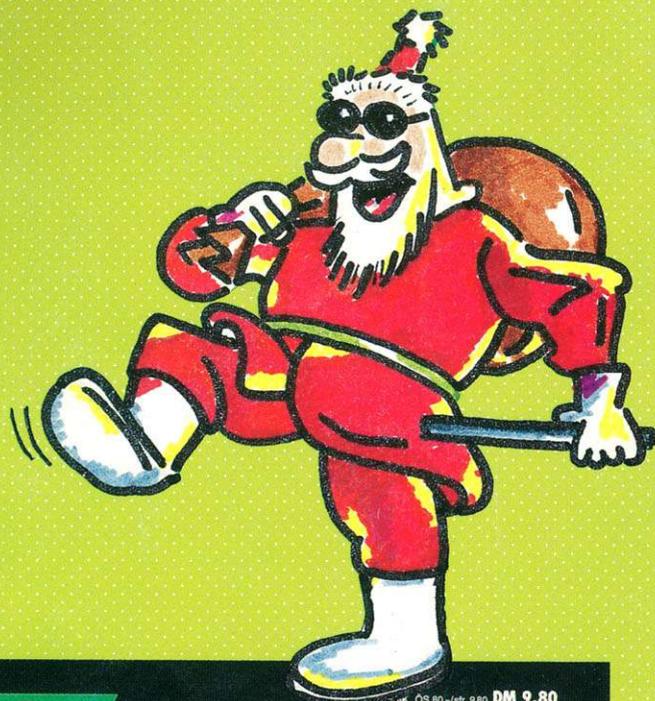
Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften und in den Fachabteilungen der Warenhäuser.




Markt & Technik
Zeitschriften · Bücher
Software · Schulung

* Unverbindliche Preisempfehlung

DIE 100 BESTEN SPIELE 1990

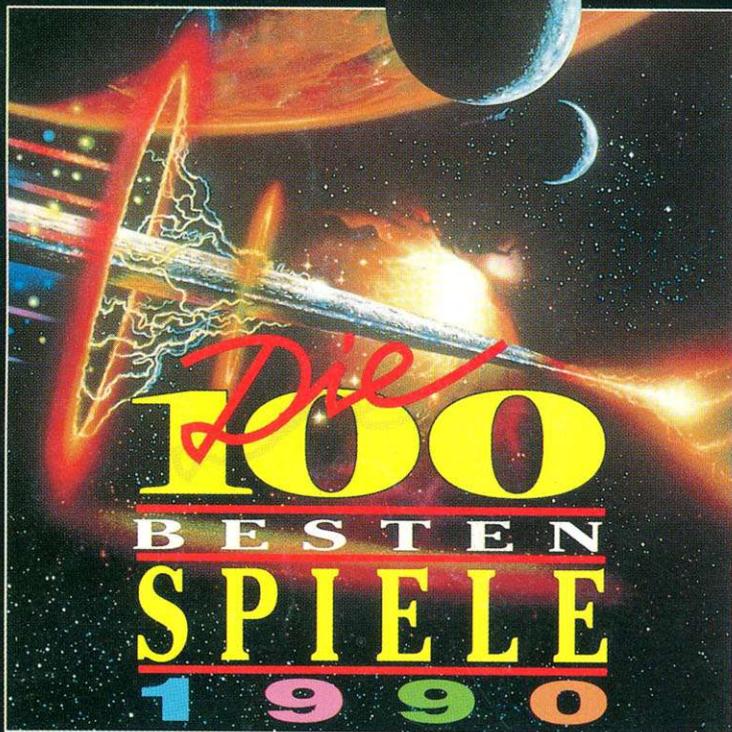


★
Im neuen
**POWER PLAY
SPECIAL 2** ab
28. November bei
Eurem
Zeitschriften-Händler!

Auf über 100 Seiten testet **POWER PLAY**
in 100 ausführlichen Spiele-Tests
Versionen für alle wichtigen
Computer- und Videospiele mit
entsprechenden Farabbildungen: die
besten Spiele für Amiga, C64, ST und PC.
Videospiele: Top-Module für Sega,
Nintendo, Mega Drive und PC-Engine.
Alle Spiele sind alphabetisch geordnet
mit praktischen Tips zu jedem Spiel!
Außerdem: die Übersicht wann welcher
Test und Tip in welchem **POWER PLAY**
stand - mit **POWER PLAY-WERTUNG!** ★

**POWER
PLAY**

OS 80.-/Jahr: 9,80 DM 9,80



AUF EINEN BLICK: Alle Tests & Tips ■ **POWER-TIPS:** Nützliche Hilfen für jedes Programm ■ **VIDEOSPIELE:** Top-Module für Sega, Nintendo und PC-Engine ■ **COMPUTERSPIELE:** Die Besten für Amiga, Atari ST, C64 und MS-DOS-PCs ■ **ENTSCHEIDUNGSHILFE:** Ausführliche Spieletests

POWER
PLAY

SPIELE-SONDERHEFT 2