

Einstieg in die Shell (Folge 1)

Die Hintertür der Workbench

von Christian Spanik
und Hannes Rügheimer

Gehören Sie auch zu den Amiga-Fans, die sich noch nie mit der »Shell« befaßt haben, die mit dem Eingabefenster für DOS-Befehle nichts zu tun haben wollen und ihren Computer am liebsten über die Workbench bedienen? In der Tat gibt es seit der Version 2.x des Betriebssystems auf den ersten Blick kaum noch etwas, was man über die Piktogramm- und menügesteuerte Workbench nicht auch erreichen könnte. Doch der Schein trügt. Mit drei Artikeln beweisen wir Ihnen, welche Möglichkeiten in der textorientierten Bedienoberfläche stecken.

Schon seit dem ersten Amiga gibt es einige Dateien, die nie bzw. bei OS 2.x nicht ohne weiteres auf der Workbench erscheinen, weil sie kein Piktogramm besitzen. Erst ab der Zwei-Nuller können Sie solche Dateien mit den Workbench-Funktionen bearbeiten, also kopieren, umbenennen, aufrufen oder löschen. Dafür gibt es im Menü »Window« den Eintrag »Show/All Files«. Wenn Sie bereits unter OS 2.x auf deutsche Menütexte umgeschaltet haben, lautet er »Inhalt anzeigen aller Dateien«. Ist die Option aktiv, ordnet das System allen Dateien ohne Piktogramm ein Stan-

»Die Shell? Unter Workbench 2.x braucht man keine Shell.« Falsch. Auch mit der neuen Systemsoftware haben Sie über die Workbench keinen Zugriff auf wichtige Funktionen. Wer die Shell kennt, nutzt den Amiga ganz.

dardpiktogramm zu, und danach können sie wie jede andere Datei behandelt werden.

Dazu gehört z.B. der Programmstart per Doppelklick. Das wollen wir gleich einmal ausprobieren. Öffnen Sie die Schublade »System« der Workbench-Diskette bzw. der entsprechenden Partition auf der Festplatte und wählen Sie »Show/All Files«. In der Schublade »c:« befindet sich das Programm »Avail«. Starten Sie es mit einem Doppelklick. Anstatt nun wie gewohnt das Programm zu laden und abzuarbeiten, reagiert der Amiga anders: Auf dem Bildschirm erscheint das Fenster »Execute a File« (Befehl ausführen). Die Eingabezeile »Command« (Befehl) enthält den Namen Avail.

Erst nach dem Betätigen der Taste <Return> oder einem Maus-klick auf »OK« startet die Ausführung. Ein neues Fenster mit dem Titel »Output Window« (Workbench-Ausgabefenster) taucht auf. Bei uns sah das aus wie in Bild 1 gezeigt. Aufgeschlüsselt nach Chip Memory, Fast Memory und dem insgesamt verfügbaren Spei-

cher (Total) sehen Sie, wie viele Bytes frei (available), wie viele zur Zeit belegt (in use) und wie viele insgesamt im Computer vorhanden sind (maximum) bzw. wie groß der jeweils größte, zusammenhängende freie Block dieser Speicherbereiche ist. Die Werte bei Ihnen werden sich wohl von denen unserer Abbildung unterscheiden.

Ausführliche Anzeige der Speicherauslastung

Das System informiert mit AVAIL also genauer über die gegenwärtige Auslastung des Arbeitsspeichers als die spärlichen Angaben der Workbench-Kopfzeile. Das »Output Window« werden Sie übrigens über das Schließsymbol los.

Wichtig ist dabei nicht nur, daß Sie so Informationen erhalten, die Ihnen die Workbench vorenthält, sondern auch, wie Sie diese Daten abgerufen haben: Für die Eingabe des Programmnamens – oder des Befehls, was im Prinzip dasselbe ist – erschien das Execute-Fenster. Für die Ausgaben von AVAIL öffnete das System ein Ausgabefenster.

Das war bereits ein erster Schritt in Richtung Shell. Wir machen noch ein Experiment: Wählen Sie im Pull-down-Menü »Workbench« der Workbench die Funktion »Execute Command« (Befehl ausführen). Wieder erscheint das Execute-Fenster – allerdings ist diesmal die Eingabezeile für den Befehl leer. Geben Sie den Befehlsnamen INFO ein und danach <Return>. Ein zweites Mal öffnet sich ein Ausgabefenster und das enthält diesmal eine Liste aller im System angemeldeten Speichermedien. Wenn die Angaben aus dem Fenster rollen, ziehen Sie es am besten ein wenig größer.

Vielleicht erscheint auch eine der folgenden Meldungen:

info: Unbekannter Befehl
info failed returncode 10

In diesem Fall sollten Sie Ihre Startdiskette (in der Regel die Workbench) ins Laufwerk legen und »Execute Command« erneut aufrufen. Versuchen Sie es mit dem Befehl C:INFO. Damit weisen Sie den Amiga an, die Programmdatei INFO im Systemverzeichnis »c:« zu suchen. Aber dazu kommen wir später noch. Übrigens befindet sich beim erneuten Aufruf des Execute-Fensters noch die vorherige Eingabe in der Kommandozeile. So lassen sich eventuelle Fehleingaben korrigieren und damit leichter wiederholen.

Das zuletzt aufgerufene Programm INFO befindet sich wie AVAIL in der Schublade »c:«. Es gehört zu einer Reihe von Befehlsdateien, die speziell für den Aufruf aus dem Execute- oder einem Shell-Fenster vorgesehen sind. Sie können ja spaßeshalber nochmal »Execute Command« aufrufen und AVAIL in der Kommandozeile

Kursübersicht

Die Dokumentation der Modelle Amiga 500, 600 und 2000 mit OS 2.0 enthält nur eine knappe Beschreibung der Shell, der textorientierten Bedienerschnittstelle. Viele Systemfunktionen sind aber nur über die Shell erreichbar. Der sichere Umgang damit erleichtert jedem Anwender die Installation neuer Hard- und Software. Dieser dreiteilige Kurs informiert Sie über wichtige Konzepte sowohl der Shell als auch des Betriebssystems.

Folge 1: Über »Execute Command« zur Shell; grundlegende Bedienung der Shell; Programme starten; Übergabeparameter nutzen; Zeichensatz ändern mit SETFONT; die Befehle CD, DIR, LIST, DELETE; Mülleimer löschen; logische Datenträger; der Befehl ASSIGN.

Folge 2: Verzeichnishierarchien; Suchpfade einrichten mit PATH; Skript- bzw. Batchprogrammierung; Modifikation der Startup-Sequence, User-Startup und Shell-Startup; die Skriptdateien PCD, SPAT und DPAT.

Folge 3: Goddies der Shell; die nichtflüchtige RAM-Disk RAD; andere Geräte (PIPE:, SPEAK:, PAR:, PRT:); der Befehl MOUNT; Schutzbits; Hard- und Softlinks mit MAKE-LINK; Tipparbeit sparen mit CON-CLIP; die Befehle STATUS, ALIAS und RESIDENT.

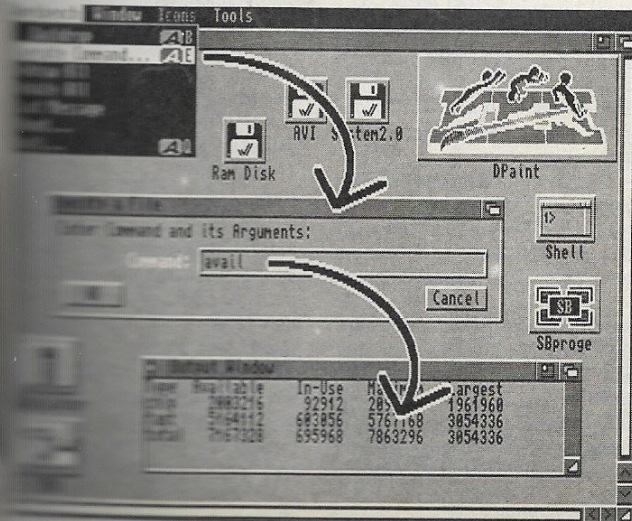
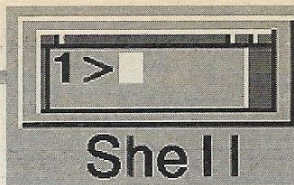


Bild 1: »Execute Command« und Eingabefenster der Workbench-Funktion »Execute Command« für die Programmausführung



eingeben. Das Ergebnis ist identisch mit dem unseres ersten Experiments. Andere Befehle zum Ausprobieren:

- ☐ CPU gibt Informationen über den bzw. die Prozessoren im Amiga aus,
- ☐ DATE zeigt Uhrzeit und Datum an,
- ☐ DIR und LIST zeigen in Textform den Inhalt des aktuellen Verzeichnisses,
- ☐ STATUS zeigt eine Übersicht der Programme, die aus Shell-Fenstern oder im Verlauf des Systemstarts aufgerufen wurden, und
- ☐ VERSION gibt die Versionsnummern von Kickstart und Workbench aus.

Weil für jeden aufgerufenen Befehl ein eigenes Ausgabefenster geöffnet wird, sollten Sie jedes nach der Ausführung wieder schließen, sonst wird die Workbench bald überfüllt sein. Außerdem belegt jedes Fenster etwa 30 bis 40 KByte RAM.

Vermutlich wird Ihnen der ständige Aufruf von »Execute Command« und das regelmäßige Schließen der Ausgabe-Fenster schon bald lästig. Es geht auch einfacher:

Öffnen Sie die Workbench oder die Partition »System...« Ihrer Festplatte. Direkt darin oder in der Schublade »System« befindet sich ein Piktogramm mit dem Namen »Shell«. Starten Sie das Programm mit einem Doppelklick.

Das System begrüßt Sie im darauf erscheinenden Fenster mit der Meldung »New shell process« (Neuer Shell-Prozeß) und einer Nummer. Unser »Prozeß« hat nichts mit Juristik zu tun – der Begriff steht vielmehr für eine weitere Task, ein weiteres Programm im Rahmen des Multitasking. Und diese neue Task ist diesmal eben eine Shell.

Das Gebilde in der nächsten Zeile, das »4.Workbench2.0:>«, »3.System2.x:>« oder ähnlich lautet, ist ein »Prompt«, eine Eingabeaufforderung, die immer dann erscheint, wenn die Shell auf eine neue Eingabe wartet. Die Aufforderung, deren Aufbau übrigens auch verändert werden kann, enthält zwei Informationen:

- ☐ Über die Shell-Nummer (in den Beispielen 4 und 3) läßt sich jeder Shell-Prozeß – Sie können ja mehrere öffnen – eindeutig identifizieren.
- ☐ Das aktuelle Verzeichnis (bei uns Workbench2.0: und System2.x:) bzw. die aktuelle Schublade.

Bevor wir diese Angelegenheit näher untersuchen, beweisen wir

erst mal, daß sich außer diesen Äußerlichkeiten kaum etwas ändert. Geben Sie hinter den Prompt

avail

ein und drücken Sie <Return>. Wenn der Computer nicht auf die Eingabe reagiert, aktivieren Sie das Shell-Fenster mit einem Mausklick und wiederholen die Eingabe.

Es erscheint die inzwischen bekannte Liste von AVAIL. Diesmal gibt es keine Unterscheidung zwischen Ein- und Ausgabefenster. Shell-Fenster dienen für beides zugleich. Sie brauchen jetzt kein Ausgabefenster mehr vergrößern, verkleinern oder schließen. Alle Aktivitäten finden innerhalb des Shell-Fensters statt, das die Profis in der Regel auch offen auf der Workbench liegenlassen.

Jedes Shell-Fenster merkt sich nämlich in einem 2024 Zeichen großen Pufferspeicher die letzten Eingaben. Mit den Cursor-Tasten nach oben und nach unten bewegen Sie sich durch diese Liste. Für eine leere Eingabezeile blättern Sie entweder bis ans Ende der Liste oder drücken <Ctrl x>. ■ Langsam müßten Sie sich heimisch in der Shell des Amiga fühlen. Schauen wir uns ein wenig um. Dazu benutzen wir die grundlegenden Befehle der Shell, die Sie auch später bei der alltäglichen Arbeit immer wieder brauchen werden.

Die zwei Befehlsdateien DIR und LIST haben wir schon erwähnt. Probieren Sie mal

dir
aus (<Return> nicht vergessen).

Schublade. Nur Dateien oder Verzeichnisse, für die es eine Datei mit gleichem Namen und dem Zusatz .info gibt, sind im Normalfall (also ohne »Show/All Files«) auf der Workbench sichtbar. Im vorliegenden Fall sehen wir die Piktogramm-Dateien zu den auf der Workbench-Diskette enthaltenen Schublade Expansion, Prefs und System. Die Datei »Disk.info« enthält die Bilddaten für das Datenträger- bzw. Disketten-Piktogramm. »Devs.info« gibt es übrigens erst auf der Workbench 2.1.

Da die gesamte Arbeit in der Shell textorientiert ist, im Gegensatz zur symbol- und menüorientierten Workbench, wird eben auch der Inhalt von Speichermedien und Schublade in Textform angezeigt. Die Informationen, die nach Eingabe von DIR erscheinen, sollen eher eine allgemeine Übersicht vom Inhalt des Verzeichnisses bieten. Anders sieht es aus, wenn Sie

list

verwenden. Dann enthält das Inhaltsverzeichnis im Shell-Fenster Zusatzinformationen, die Sie vielleicht von der Option »View/by Name« (Inhalt auflisten/nach Namen) aus dem Workbench-Menü »Window« (Fenster) kennen.

Vergleichen wir einmal unsere bisherigen Kenntnisse von der Shell mit denen über die Workbench. Auf der Workbench bekommt jede Diskette, jede Schublade ein eigenes Fenster. Um den Inhalt einer Schublade zu sehen, braucht man sie nur doppelt anzuklicken. Die Shell hat nur ein Fenster. Wie öffnet man ein Verzeichnis über die Shell?

Dafür gibt es den Befehl CD (change directory -> wechsele das Verzeichnis). Lassen Sie sich mit DIR nochmal den Inhalt des aktuellen Verzeichnisses ausgeben. Suchen Sie sich eines der mit (DIR) gekennzeichneten Namen aus und machen es mit CD zum aktuellen Verzeichnis. Beispiel:

ed c

Wenn Sie nun DIR oder LIST aufrufen, sehen Sie ein anderes Inhaltsverzeichnis – nämlich das der gerade betretenen Schublade. Sie entdecken dabei übrigens neben den bisher bekannten Befehlsdateien viele weitere. Widerstehen Sie aber der Versuchung, mit diesen Befehlen zu experimentieren, denn es sind einige dabei, die ganz schön Verwirrung stiften oder gar zum Datenverlust führen.

Ein Blick auf die Eingabeaufforderung, die nun z.B. »4.Workbench 2.0:c>« lauten könnte, zeigt, daß Sie sich nun im Unterverzeichnis

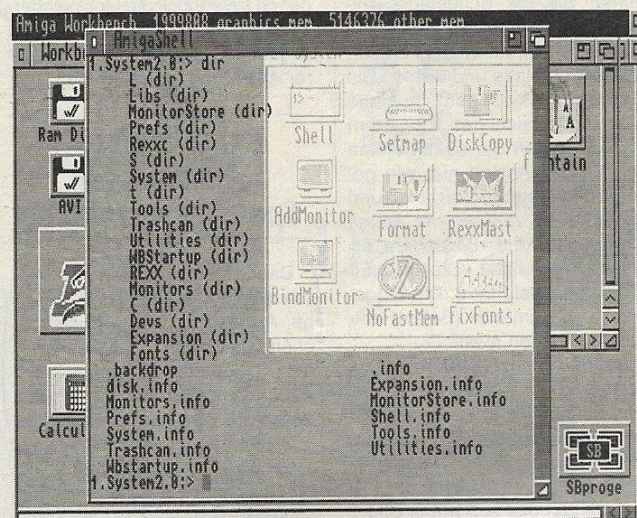


Bild 2: So gibt der DOS-Befehl DIR den Inhalt von Verzeichnissen bzw. Schublade aus

Hinter den Daten von AVAIL ist gleich wieder eine neue Eingabeaufforderung erschienen – die Shell ist bereit für eine neue Arbeit. Probieren Sie das gleich einmal aus, indem Sie INFO oder eines der anderen vorhin genannten Programme aufrufen. Das Shell-Fenster kann übrigens bei Bedarf vergrößert werden. Da wir uns bei den nächsten Experimenten nur mit der Shell beschäftigen, bringen Sie es am besten auf volle Bildschirmgröße.

Wenn Sie dasselbe Programm mehrmals aufrufen wollen, oder bei der Eingabe des Namens einen Fehler gemacht haben, können Sie ähnlich wie im Execute-Fenster die letzte Eingabe wiederholen. Dazu brauchen Sie nur die Taste <Cursor oben> drücken. Tun Sie es mehrmals, erscheinen in der aktuellen Eingabezeile die jeweils vorher eingegebenen An-

Sie bekommen eine mehr oder weniger umfangreiche Inhaltsübersicht des aktuellen Verzeichnisses (engl. directory), also von der Schublade, deren Name Teil der Eingabeaufforderung ist. Bild 2 zeigt, wie so etwas prinzipiell aussieht.

Gibt es innerhalb der aktuellen Schublade weitere Unterverzeichnisse bzw. Schublade, zeigt DIR dessen Namen zuerst an, zusammen mit der Kennzeichnung »(dir)«. Danach erscheinen die Namen der im aktuellen Verzeichnis enthaltenen Dateien. Zufälligerweise gibt es im gerade gezeigten Beispiel (auf der Workbench-Diskette) ausschließlich Dateien mit dem Zusatz ».info«. Davon haben Sie bisher auf der Workbench-Oberfläche nichts lesen können. Diese Dateien enthalten nämlich die Bildinformation für das Piktogramm einer Datei oder einer

»C« auf der Diskette »Workbench 2.0« bzw. Ihrer Systemdiskette oder -partition aufhalten.

Wenn Sie aus dieser Schublade wieder zurück ins übergeordnete Verzeichnis wollen, geben Sie statt dem Namen hinter CD das Zeichen »/« an:

cd /

Ein kurzes LIST oder DIR bzw. ein Blick auf den Prompt zeigt, daß Sie tatsächlich wieder an den Ausgangspunkt zurückgekehrt sind. Üben Sie vor dem Weiterlesen den Auf- und Abstieg innerhalb der Verzeichnishierarchie des Amiga.

■ Wir wollen unser Versprechen einlösen und Ihnen einige Möglichkeiten zeigen, für die es kein Pendant auf der Workbench gibt:

Wie bereits vorgeführt werden Programme von der Shell aus durch Eingabe ihres Namens und anschließendes <Return> gestartet. In vielen Fällen können Sie dem Befehl dabei einige Daten übergeben, die sein Verhalten beeinflussen (Optionen, Übergabeparameter). Beispiel:

dir all

DIR gibt daraufhin nicht nur die Namen der Unterverzeichnisse aus, sondern auch dessen Inhalt. Um ein Programm zu besonderem Verhalten zu motivieren, müssen Sie wissen, welche Optionen es besitzt. Die sind meist in der dazugehörigen Dokumentation oder in ReadMe- oder LiesMich-Dateien aufgeführt. Leider hat Commodore bei den Modellen 500 Plus und 600/600 HD am DOS-Handbuch gespart. Da bleibt Ihnen nur übrig, ein entsprechendes Fachbuch zu kaufen.

Einige Programme – insbesondere aus der Public-Domain- und Shareware-Szene – können ohne hin nur über die Shell bzw. die Eingabe des Namens gestartet werden, weil sie kein Piktogramm haben und meist für den Aufruf von der Workbench auch nicht vorbereitet sind. Oder aber sie laufen zu ungeahnten Leistungen auf, weil Sie beim Shell-Aufruf wesentlich mehr Optionen angeben können.

Ein kleiner Trick ist immer einen Versuch wert: Viele Programme verraten verfügbare Optionen, wenn Sie hinter dem Programmnamen in der Shell ein Fragezeichen eingeben. Leider ist der Gehalt der daraufhin erscheinenden Informationen von Programm zu Programm unterschiedlich. Andere Programme kennen diese Funktion nicht, ignorieren das Fragezeichen oder versuchen, es als Option zu interpretieren.

Ein Schmäckerl am Rande: Gefällt Ihnen die Schrift der Shell?

Sie können den Zeichensatz mit einem DOS-Befehl ändern:

setfont Topaz 11

oder

setfont Courier 15

Wenn SETFONT die Meldung »object not found« ausgibt, hat es den angegebenen Zeichensatz nicht gefunden. Überprüfen Sie mit

cd fonts:

dir

(Doppelpunkt nicht vergessen) den Vorrat an Systemzeichensätzen. Die Namen der Unterverzeichnisse sind gleichzeitig die der Zeichensätze. In jedem befindet sich je eine Datei pro vorhandene Größe. Mit

setfont topaz 8

stellen Sie den Standardzeichensatz wieder ein und mit

cd sys:

kommen Sie übrigens wieder zu dem Verzeichnis zurück, das beim Start der Shell zum aktuellen Verzeichnis gemacht wird. Wir erklären noch in diesem Artikel, warum das so ist.

■ Langsam aber sicher beherrschen Sie die grundlegenden Befehle der Shell. Vielleicht gehen Sie ja sogar dazu über, dort hin und wieder Tätigkeiten auszuführen, die Sie auf der Workbench auch erledigen könnten. Für den Umgang mit Dateien etwa bietet die Shell eine Reihe leistungsfähiger Befehle wie COPY, RENAME oder DELETE.

Da wäre DELETE (löschen). Uns hat es schon immer geärgert, daß die Workbench-Funktion »Format« grundsätzlich einen Mülleimer (Trashcan) auf den frisch formatierten Datenträger plazierte, obwohl wir den auf unseren Datendisketten nie benötigen. Über die Workbench läßt sich der Mülleimer nicht löschen. Aber in der Shell.

Das probieren wir aus. Formatieren Sie eine leere Diskette über die Workbench. Öffnen Sie danach den Datenträger und klicken Sie den Mülleimer an, so daß er sich öffnet. Die Menüfunktion »Icons/Delete« ist nicht anwählbar und damit läßt sich der Eimer auch nicht entfernen. Ab in die Shell...

Amiga-Fans mit nur einem Diskettenlaufwerk haben jetzt allerdings ein Problem. Um den Mülleimer zu löschen, muß die formatierte Diskette im Laufwerk liegen. Aber: Um DELETE aufzurufen, muß sich die Workbench-Diskette darin befinden, weil sich das Programm DELETE darauf befindet. Die eleganteste Lösung dafür ist

resident c:delete

RESIDENT lädt das Programm DELETE in den Speicher und rich-

tet es dort häuslich ein. Fortan brauchen Sie die Workbench-Diskette nicht mehr beim Löschen über die Shell.

Das interne Laufwerk hat den Namen »df0:«. Mit

cd df0:

machen Sie das Hauptverzeichnis der Diskette, also dasjenige, dessen Inhalt beim Öffnen des Diskettenpiktogramms im Fenster erscheint, zum aktuellen Verzeichnis. Mit

dir

können Sie sich den Mülleimer noch einmal anschauen, bevor er mit

delete Trashcan

delete Trashcan.info

entsorgt wird.

Wenn Sie den gefüllten Mülleimer einer bereits gebrauchten Diskette löschen wollen, müssen Sie vorher die Workbench-Funktion »Icons/Empty Trash« ausführen. Sonst würde DELETE die Fehlermeldung

Trashcan Not Deleted: directory not empty

liefern und das Verzeichnis »Trashcan« nicht entfernen.

Daß Sie solche Experimente nicht auf Ihrer Systemdiskette oder -partition, geschweige denn auf den Originaldisketten durchführen, versteht sich wohl von selbst. Wenn der Mülleimer erst mal weg ist, ist er weg – und kann nicht mehr zurückgeholt werden.

Eine Nachbemerkung zu CD. Natürlich ist es lästig, für eine einzelne Operation in einem bestimmten Verzeichnis dieses mit CD zu »öffnen«. Sie brauchen es auch nicht. Die Anweisungsfolge

cd df0:

delete trashcan

läßt sich durch

delete df0:trashcan

ersetzen. Das bedeutet: Befindet sich die Datei, mit der ein Shell-Befehl irgend etwas machen soll, nicht im aktuellen Verzeichnis, so ist der Name des entsprechenden Verzeichnisses vor den Dateinamen zu setzen. Das gilt nicht nur für Dateien:

cd devs

dir

ist gleichbedeutend mit

dir devs

oder

cd devs

cd printers

dir

mit

dir devs/printers

Durch die Aneinanderreihung der Verzeichnisnamen teilen Sie der Shell also mit, welchen Pfad sie »gehen« muß, um die gewünschten Daten zu finden. Das

aktuelle Verzeichnis ist lediglich eine Erleichterung für den Fall, daß Sie in einem »weit entfernten« Verzeichnis mehrere Operationen durchzuführen haben. Dann nämlich wäre die Angabe der Verzeichnisreihe bei jedem Befehl mehr als lästig.

Der Pfad, den die Shell gehen soll

Befinden sich die Dateien auf einem anderen Datenträger wie das aktuelle Verzeichnis, so beginnt die Reihe der Verzeichnisnamen, also der Pfad durch die Verzeichnishierarchie mit dem Laufwerksnamen. Beispiel:

dir df0:devs/printers

Und noch ein Tip: Sie können statt des Laufwerksnamens »df0« auch den Namen der Diskette verwenden. Beispiel:

dir empty:

Laufwerks- bzw. Datenträgernamen sind immer mit einem Doppelpunkt abzuschließen.

■ Hat es Sie eigentlich überrascht, daß hinter dem Mülleimer der Workbench ein Verzeichnis mit Sonderaufgaben steckt? Das wundert uns nicht. In der Tat ist die Shell der Schlüssel zum Verständnis des Amiga-Betriebssystems. Hier werden viele Abläufe und Zusammenhänge schneller sichtbar, als auf der abgeschirmten, für den mit dem Computer weniger vertrauten Anwender gedachten Workbench. Ein weiteres, fast klassisches Beispiel dafür ist die Zuweisung von Datenträgernamen an Verzeichnisse. Obwohl sich dieses Konzept auf sehr viele Vorgänge im Amiga auswirkt, kennen es viele Workbench-Anwender nicht.

Worum geht's? Nun, für seine Arbeit benötigt das Amiga-Betriebssystem ständig Zugriff auf bestimmte Dateien. Die Zeichensätze, nach denen SETFONT sucht, sind nur ein Beispiel. Wo sich diese Dateien konkret befinden, ist – je nach Ausstattung und Einrichtung – von Amiga zu Amiga verschieden. Bei Computern mit Festplatte befinden sie sich an anderer Stelle als bei denen mit einem Diskettenlaufwerk, und dort möglicherweise woanders als bei einem Amiga mit zwei oder drei Laufwerken.

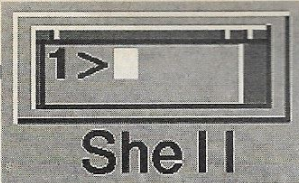
Nehmen wir die Zeichensätze (fonts) noch einmal als Beispiel. Für das Betriebssystem wäre es am einfachsten, wenn Sie auf einer Diskette namens »fonts« vorlägen.

SETFONT stemrol
sätzen:
automa
ten Zei
»fonts:«
und wü
spreche

Die
mal im
Workbe
men »V
Festpla
oder so
Entwick
nen Tri
triebssy
konzipi
Zeiche
»fonts:«
eine ec
ner Lis
zufinde
sich di
findet.
unsere
eben n
werksn
bench2
So el
Datent

Alle

Hi Am
Eurem
wegen
COMP
große
Greifha
einen P
zum CD



diglich
all, daß
ernten«
ationen
n näm-
rzeich-
mehr als

SETFONT sowie alle anderen Systemroutinen, die mit den Zeichensätzen zu tun haben, würden dann automatisch vor den angeforderten Zeichensatznamen die Folge »fonts:« (z.B. fonts:courier) setzen und wüßten so, wo sich die entsprechenden Dateien befinden.

Die Zeichensätze lagern aber mal im Verzeichnis fonts einer Workbench-Diskette mit dem Namen »Workbench2.0«, mal auf der Festplattenpartition »System2.1« oder sonstwo. Also haben sich die Entwickler des Betriebssystems einen Trick einfallen lassen: Das Betriebssystem wurde tatsächlich so konzipiert als befänden sich die Zeichensätze auf dem Datenträger »fonts:«. Für den Datenzugriff auf eine echte Diskette muß es eh in einer Liste nachsehen, um herauszufinden, in welchem Laufwerk sich die gewünschte Diskette befindet. Und in dieser Liste steht bei unserem imaginären Datenträger eben nicht df0 oder sonst ein Laufwerksname, sondern z.B. »Workbench2.0/fonts«.

So ein logischer bzw. imaginärer Datenträger ist wie ein Wegweiser

auf die tatsächliche Fundstelle. Auch die Programmdateien der Befehle DIR, LIST, AVAIL, INFO sucht das System in einem logischen Datenträger. Er heißt c. Probieren Sie's aus:

dir c:

Egal, in welchem aktuellen Verzeichnis Sie sich befinden – Sie sehen auf jeden Fall den Inhalt der Systemschublade c. Ähnliches können Sie mit Befehlseingaben wie

dir s:

dir devs:

dir libs:

erreichen. Die logischen Datenträgernamen s, devs und libs wie auch c sind in diesem Fall identisch mit den Namen der entsprechenden Schubladen auf der Workbench- bzw. Systemdiskette.

Für die Verwaltung dieser Wegweiser ist das Programm ASSIGN zuständig. Wenn Sie es ohne weitere Angaben in der Shell aufrufen, erscheint eine Liste gültiger Zuweisungen. ASSIGN kann aber auch neue Zuweisungen herstellen bzw. alte durch neue Zuweisungen ersetzen und damit den Datenzugriff

auf logische Datenträger umleiten. Die Anweisung

assign c: ram:c

würde das System beispielsweise veranlassen, die Schublade mit den DOS-Programmen ab sofort auf der RAM-Disk zu suchen. Das sollten Sie allerdings nur tun, wenn Sie die C-Schublade vorher dorthin kopiert haben – sonst sägen Sie im wörtlichen Sinne den Ast ab, auf dem Sie sitzen. Wem das jetzt doch passiert sein sollte, der kann es mit

sys:c/assign c: sys:c

wieder rückgängig machen.

Schauen wir nochmal auf die vorhin erschienene Zuweisungsliste. Auf logische Datenträger, die in dieser Liste nicht erscheinen, kann der Amiga auch nicht zugreifen. Wenn Sie beispielsweise

dir Hurra:

eingeben, wird der Amiga Sie auffordern, eine Diskette namens »Hurra« einzulegen.

Würden Sie eine Diskette mit dem Namen »c« einlegen, würde der Amiga dort seine Befehle suchen. Gleiches passiert Diskettenanwendern bei der Systemsoft-

ware ab Version 2.0 übrigens mit den Zeichensätzen. Ihr Inhalt befindet sich bei Hard-disk-Amigas in einem Verzeichnis namens »fonts«. Bei Non-HD-Computern sind sämtliche Schriftinformationen auf der gleichnamigen Diskette untergebracht. Möchte ein Programm auf das Schriftenverzeichnis zugreifen, fordert Amiga-DOS den logischen Datenträger »Fonts:« an.

Natürlich können Sie das alles auch für eigene Zwecke nutzen: Wenn etwa Ihr Amiga immer wieder eine bestimmte Diskette anfordert, können Sie den Inhalt dieser Diskette auf Festplatte oder (wenn dort Platz genug ist) in die RAM-Disk kopieren, und per ASSIGN die entsprechende Zuweisung vornehmen. Beispiel:

assign Daten: Work:Daten

Mit einer Reihe gezielter ASSIGNS lassen sich Probleme leicht lösen, die bei der Installation von Programmen auf Festplatte oder bei ständigen Diskettenwechseln auftreten. Wir denken, schon aus diesem Grund lohnt es sich, ständig ein Shell-Fenster auf der Workbench offen zu haben. pa

n auf ei-
wie das
beginnt
namen,
erzeich-
ufwerks-

können
ns »df0«
ette ver-

enträger-
dem Dop-

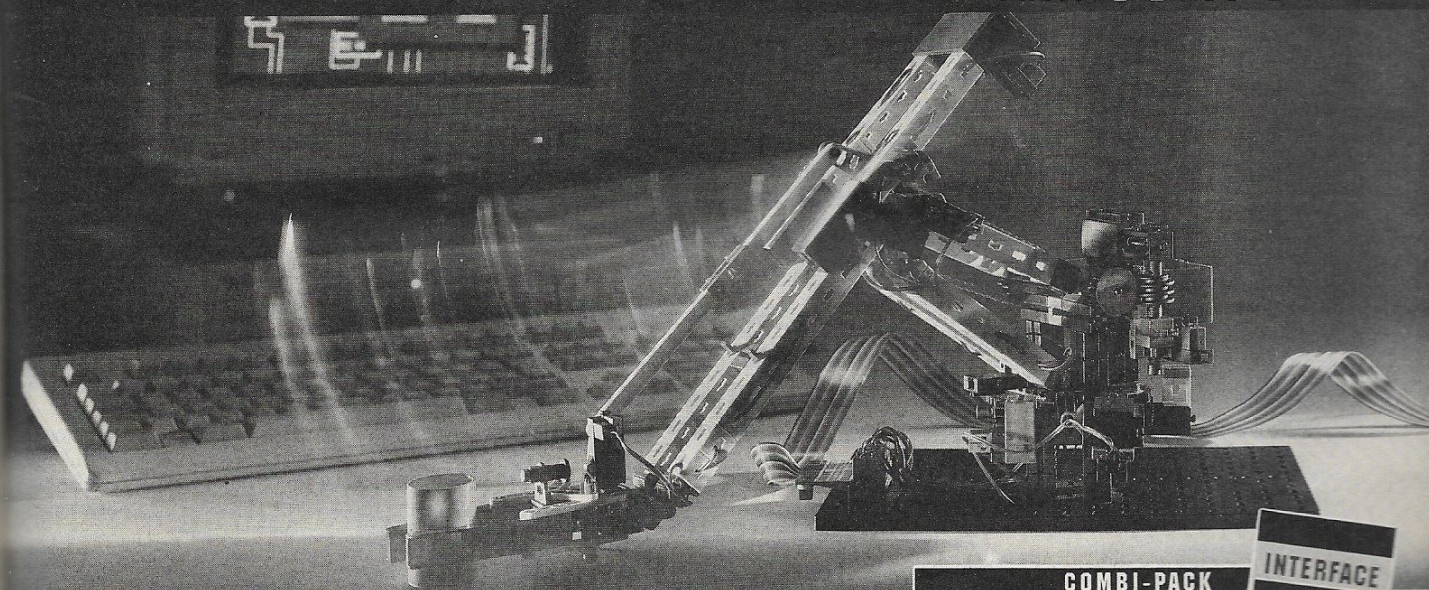
ch über-
fülleimer
chnis mit
Das wun-
at ist die
Verständ-
systems-
e und Zu-
sichtbar-
n, für den
niger ver-
edachten
fast klas-
st die Zu-
gernamen
hl sich die
e Vorgän-
ennen es
der nicht
für seine

Amiga-
Zugriff auf
Zeichen-
SETFONT
ispiel. Wo
kret befin-
attung und
a zu Amiga
putern mit
sich an an-
nen mit ei-
t, und dor-
ers als bei
oder drei

ichensätze
s Beispiel
im wäre es
Sie auf eine
« vorlägen

GAZIN 12/1982

PROFI COMPUTING



Alles was ein Profi braucht.

Amiga-Fans, endlich könnt Ihr mit Eurem Computer mal wirklich was bewegen. Denn der fischertechnik PROFI COMPUTING-Baukasten bietet eine große Modell-Auswahl: vom Roboter mit Greifhand, über einen Scheckkartenleser, einen Plotter und Geldautomaten bis hin zum CD-Player. Alles mit der Super-Soft-

ware LUCKY LOGIC, mit mausgesteuerter interaktiver Benutzeroberfläche. Dazu das INTERFACE mit 8 Digitaleingängen, 2 Analogeingängen und 4 Motorausgängen. Für alles gibt es ausführliche Experimentier- und Softwarehandbücher sowie eine Softwareanleitung. Wenn Ihr jetzt mehr Infos haben



wollt, dann ab zum Telefon und Prospekt anfordern: 074 43/12 - 3 69.

fischertechnik
Alles andere ist nur Spielzeug