

CLI — Die zweite Haut des Amiga (1)

Der Amiga ist nicht nur über die Benutzeroberfläche der Workbench bedienbar. Es gibt noch eine zweite Ebene, mit der sich viel mehr anfangen läßt, als dies mit der Workbench möglich ist: das CLI. Wir starten in dieser Ausgabe einen Kurs, der Sie in den Umgang mit dem CLI einführt.

Der Amiga bietet dem Benutzer mit der Workbench eine komfortable Benutzeroberfläche. Programme können durch einfaches »anklicken« mit der Maus gestartet werden. Auch lassen sich Operationen wie beispielsweise Löschen oder Umbenennen leicht durchführen. Über die Workbench-Ebene können jedoch keine tiefgreifenden Manipulationen auf der Diskette vorgenommen werden. So läßt sich über die Workbench beispielsweise nicht feststellen, wie viele Dateien tatsächlich in welchen Schubladen (Subdirectories) auf der Diskette vorhanden sind. Oder wie wollen Sie ein Programm starten, das zwar auf der Diskette vorhanden ist, aber über kein Icon (Bildsymbol) verfügt? Doch es geht auch anders: Mit dem CLI (Command-Line-Interface, zu deutsch etwa: Kommandoübersetzer).

Doch was ist das CLI, wo ist es zu finden? Im Amiga-Handbuch stehen nicht viele Informationen zu diesem Programm, das sich (normalerweise unsichtbar) mit auf jeder Workbench-Diskette befindet. Bevor wir aber das CLI erklären, machen wir das Command Line Interface für den Anwender sichtbar!

Dazu fertigen Sie sich zuerst eine Arbeitskopie der Workbench-Diskette an und legen dann diese Kopie in das interne Laufwerk. Starten Sie den Amiga erneut und rufen dann das Programm »Preferences« auf. Im linken unteren Viertel des Preferences-Menüs sehen Sie ein kleines Feld, in dem »CLI ON/OFF« steht. Bei einer nicht modifizierten Workbench steht das Feld (der »Schalter«) auf »CLI OFF«. Klicken Sie nun das Wort »ON« an, speichern die Änderung mit SAVE wieder auf Diskette, bringen den Schreibschutz wieder an. Der Amiga

ist anschließend neu zu starten. Klicken Sie das Symbol der Workbench-Diskette an und, nachdem das erste Fenster sichtbar ist, die Schublade mit dem Namen »System«. Durch die Änderung der Preferences findet sich nun ein neues Symbol in diesem Ordner, das den Namen »CLI« trägt. Bevor wir uns aber näher mit dem CLI befassen, soll zuvor geklärt werden, was dieses Programm eigentlich macht und wie es einzusetzen ist!

Neue Wege mit dem CLI

Von der Workbench aus kann der Benutzer nur mit Dateien umgehen, die über ein Icon verfügen. Andere Dateien können von der Workbench aus nicht beeinflußt oder aufgerufen werden. Hier greift das CLI ein: Es ermöglicht die direkte Beeinflussung aller auf einer Diskette vorhandenen Dateien. (Für alle Leser, die schon einmal mit Personal Computern gearbeitet haben: Das CLI ist in etwa vergleichbar mit MS-DOS.)

Die Befehle werden nicht (wie bei der Workbench) durch Anklicken eines Icons ausgeführt, sondern durch Kommandos, die der Anwender über die Tastatur eingibt. Versuchen wir doch gleich, anhand von praktischen Beispielen den Umgang und die Möglichkeiten, die uns das CLI in die Hand gibt, kennenzulernen. Rufen Sie dazu das Programm durch zweimaliges Anklicken des CLI-Icons in der Schublade »System« auf. Nach kurzer Ladezeit erscheint auf dem Bildschirm ein neues Fenster. Dies ist das Arbeitsfenster des CLI. Hier erfolgen alle Ein- und Ausgaben, die das Command-Line-Interface betreffen. Es empfiehlt sich, das Fenster zu vergrößern, so daß es fast den ganzen Bildschirm ausfüllt.

Anhand eines einfachen Befehls soll Ihnen nun der Unterschied zur Workbench erklärt werden. Lassen Sie dazu die Kopie der Workbench-Diskette im internen Laufwerk eingelegt und geben Sie das Kommando »dir« ein, das mit der RETURN-Taste abzuschließen ist. Nach kurzer Zeit erscheinen viele einzelne Namen in zwei Spalten auf dem Bildschirm. Dieser erste Befehl, den wir kennenlernen, hat die Aufgabe, das Directory (Inhaltsverzeichnis) der eingelegten Diskette anzuzeigen. Sie werden feststellen, daß nach Eingabe des Dir-Befehls plötzlich viel mehr Dateinamen auf dem Bildschirm erscheinen, als dies im Symbolfenster auf der Workbench der Fall wäre. Dort sehen Sie nur Dateien, die über ein Icon verfügen. Im CLI-Fenster wären das die Programme, die die Endung »info« aufweisen (auf der Workbench-Diskette beispielsweise die Files »Preferences.info« oder »System.info«). Zu jedem dieser Icon-Files gehört auch das eigentliche Programm, welches durch das Icon aufgerufen wird (bei unserem Beispiel »Preferences« oder »System«).

Falls Sie über weitere Diskettenlaufwerke verfügen, können Sie sich über einen Zusatzparameter auch die Directories dieser Floppy-Stationen anzeigen lassen. Dazu ist hinter dem Dir-Befehl der Name des entsprechenden Laufwerks anzugeben. (Die vier möglichen Diskettenlaufwerke tragen die Bezeichnungen »df0« bis »df3«, die beiden anschließbaren Festplatten »dh0« und »dh1«.) Dabei ist nur zu beachten, daß der Laufwerksname durch ein Leerzeichen vom Befehl getrennt ist und daß am Ende ein Doppelpunkt steht. Das Directory des ersten externen Laufwerks würden Sie also mit folgendem Befehl erhalten:

Wenn kein externes Laufwerk angeschlossen ist, erfolgt eine entsprechende Fehlermeldung.

Da es manchmal vorkommen kann, daß mehr Dateinamen im Directory stehen, als auf dem Bildschirm angezeigt werden können, läßt sich die Ausgabe mit der Leertaste anhalten. Durch Druck auf <RETURN> oder die Backspace-Taste wird die Ausgabe fortgesetzt.

Das Directory einer Diskette besteht im allgemeinen aber nicht nur aus Dateinamen, sondern auch noch aus Subdirectories (Unterverzeichnissen). Ein Subdirectory kann man sich als Directory im Directory vorstellen. Im Hauptverzeichnis, in dem Sie sich gerade befinden, sehen Sie beispielsweise die Namen »c«, »s«, »devs« oder »Trashcan«. Hinter diesen Namen verbergen sich Subdirectories, was durch die angeschlossene Kennung »(dir)« ersichtlich ist.

Zusätze zum Dir-Befehl

Um den Inhalt eines Subdirectories anzusehen, ist ein weiterer Zusatzparameter beim Dir-Befehl nötig, nämlich der Name des Unterverzeichnisses. Möchten Sie sich etwa den Inhalt des Subdirectories »c« ansehen, ist folgendes einzugeben:

```
dir df0:c
```

Die Angabe des Laufwerknamens ist, da sowieso gerade diese Floppy-Station ausgewählt ist, nicht unbedingt nötig. Es ist aber empfehlenswert, immer das Laufwerk mitanzugeben, da man bei mehreren angeschlossenen Laufwerken leicht den Überblick verlieren kann.

Der Dir-Befehl bietet aber noch mehr Möglichkeiten, als

nur das komplette Directory anzuzeigen. Dazu existiert der Zusatz »opt«, der in Zusammenhang mit ihm folgenden Buchstaben den Dir-Befehl modifiziert. Durch Angabe des Buchstabens »a« hinter dem Zusatz »opt« zeigt der Amiga sämtliche Dateien auf der Diskette an, einschließlich der Namen in allen Unterverzeichnissen. Geben Sie dazu als Beispiel folgendes ein:

```
dir df0: opt a
```

Es werden sämtliche auf der Diskette enthaltene Dateinamen angezeigt. Beachten Sie bitte auch bei diesem Befehl die Schreibweise: Die einzelnen Leerzeichen dürfen nicht vergessen werden!

Gibt man anstelle des Buchstabens »a« ein »d« ein, erfolgt die Ausgabe aller Dateiverzeichnisse, also aller Namen, hinter denen die Kennung »(dir)« steht.

Ein besonderer Befehl ist der folgende:

```
dir df0: opt i
```

Tippt man diese Zeichenfolge ein, gibt der Amiga jeweils nur einen Dateinamen aus, auf

den ein Fragezeichen folgt. Bei Druck auf die RETURN-Taste übergeht das CLI die entsprechende Datei und gibt den Namen der folgenden Datei aus. Stellt der angezeigte Name ein Subdirectory dar, kann durch die Eingabe von »e« in dieses gewechselt werden. <RETURN> überspringt auch hier den Dateinamen. Wurde jedoch in das Subdirectory übergewechselt und man möchte wieder in das Hauptdirectory zurück, ist »b« einzugeben. Das CLI wechselt auf die nächsthöhere Ebene zurück und macht dort weiter, wo der Einsprung in das Unterverzeichnis stattfand. Der Rest des Subdirectories wird ignoriert.

Falls hinter dem Fragezeichen die Zeichenkette »del« (»delete« = löschen) eingetippt wird, löscht dieser Befehl die betreffende Datei. Auch Dateiverzeichnisse lassen sich auf diese Weise entfernen. Dabei ist nur zu beachten, daß zuerst aus dem betreffenden Subdirectory sämtliche Dateien entfernt sein müssen!

Der Buchstabe »t« hinter »opt« schließlich erlaubt die

Ausgabe des Inhalts der betreffenden Datei. Dies hat allerdings nur Sinn, wenn es sich dabei um eine Textdatei handelt. Ein Beispiel:

Sie möchten den Inhalt einer bestimmten Datei im Klartext auf dem Bildschirm ausgeben.

Viele Möglichkeiten

Nehmen wir dazu die Datei »startup-sequence«, die sich im Subdirectory »s« der Workbench-Diskette befindet. Diese Datei ist das erste File, das nach dem Einlegen der Workbenchdiskette geladen und gestartet wird (die Datei ist vergleichbar mit einer Autostart- oder Boot-Datei oder einer Batch-Datei unter MS-DOS). In dieser Datei stehen alle Befehle und Parameter, die der Amiga abarbeitet, bis nach einiger Wartezeit endlich die Workbench erscheint. Diese Startdatei besteht aus ASCII-Zeichen, ist also eine Textdatei.

Vergewissern Sie sich, daß Ihre Workbench-Kopie eingelegt ist und geben Sie folgendes ein:

```
dir df0: opt i
```

Drücken Sie nun so lange <RETURN>, bis »s (dir)« erscheint. Mit der Eingabe von »e« gelangen Sie nun in dieses Unterverzeichnis. Auf dem Bildschirm sollte jetzt »startup-sequence ?« sichtbar sein. Drücken Sie »t« und schließen die Eingabe mit <RETURN> ab. Blitzschnell erscheinen viele Zeilen, die wie ein Programm aussehen. Wenn Sie sich die einzelnen Zeilen ansehen, erkennen Sie einige Befehle (beispielsweise »echo«, »setmap« oder »loadwb«), die Sie ebenfalls vom CLI aus benutzen können. Auf diese Befehle soll aber erst später eingegangen werden.

Die Eingabe von »q« beendet jederzeit die Ausführung des Befehls »dir opt«. Durch die Tastenkombination <CTRL C> läßt sich ebenfalls ein Abbruch erzwingen.

Sie haben jetzt einen wichtigen Befehl kennengelernt, mit dem sich schon eine Menge anfangen läßt. Der DIR-Befehl zeigt uns aber nicht alles an. Es gibt noch weitere Informationen zu Dateien, die ein anderer Befehl viel besser anzei-

AMIGA

TOP HIT

VizaWrite Desktop

- komfortable Textverarbeitung mit umfangreicher Bausteinverwaltung
- 2farbige Bilder an jeder Stelle des Textes einfügbar
- Schriftsätze und Grafiken auf Screen und Drucker (What you see is what you get)
- unterstützt Laser- und Matrixdrucker
- arbeitet problemlos mit EASYL-Grafiktablett und Pagesetter
- arbeitet wahlweise auch im Interlace-Modus



HITLISTE des Monats

- | | |
|--|------------|
| (1) VIZAWRITE DESKTOP | 498,- DM* |
| deutsche Textverarbeitung der Superlative | |
| (2) EASYL | 998,- DM* |
| prof. Zeichen- und Grafiktablett (DIN A4) | |
| (3) METACOMCO Shell | 149,- DM* |
| erweiterte CLI-Benutzerumgebung | |
| (4) AEGIS Draw plus (prof. CAD-System/Hi-Res) | 589,- DM* |
| (5) COMSPEC 2-MByte-Speicherkarte (durchgeschleift) | 1998,- DM* |
| (6) TXED (Supereditor) | 128,- DM* |
| (7) CHESSMATE (3-D-Schachprogramm m. Sprachausg.) | 98,- DM* |
| (8) DELUXE Paint II (prof. Zeichenpgm., auch f. EasyL) | 298,- DM* |
| (9) METACOMCO Makro-Assembler | 248,- DM* |
| (10) JITTER-RID (unentbehr. im Interlace-Modus) | 79,- DM* |
| (11) PAGESETTER (DIP Seitenlayoutprogramm) | 398,- DM* |

Neuvorstellungen

- | | |
|--|-----------|
| (-) DISCOVERY (unentbehrlicher Diskettenhelfer) | 198,- DM* |
| (-) LATTICE-C-Compiler (inkl. Text-Utilities) | 448,- DM* |
| (-) AEGIS ANIMATOR & IMAGES (Trickfilmanimation) | 298,- DM* |

Schweiz:



MICROTRON

Bahnhofstrasse 2
CH-2542 Pieterlen
Telefon 032/87 24 29

Deutschland:



Werbung & EDV GmbH

Bornhofenweg 5
D-6200 Wiesbaden
Telefon 061 21/40 79 89
4078 76
Telefax 061 21/40 73 21

* Unverbindliche Preisempfehlung
Fachhändleranfragen erwünscht.

Erhältlich auch beim Fachhändler. Händlernachweis, Info, Softwareliste Atari/IBM & Kompatible anfordern.
Bitte senden Sie mir ausführliche Informationen
☐ IBM & KOM. ☐ AMIGA ☐ C128 ☐ C64 ☐ ATARI/ST
Absender:

Amiga 6/7/87

gen kann: Das Kommando »list«.

Der DIR-Befehl zeigte uns zwar alle Dateien auf der Diskette an, konnte uns jedoch keine Auskunft über deren Länge (Anzahl der Bytes) oder das Datum des letzten Updates (der letzten Speicherung) geben. (Der Amiga speichert zu jeder Datei auch die aktuelle Uhrzeit und das Datum. So läßt sich erkennen, wann die Datei das letztemal geändert wurde.) Um an diese »versteckten« Informationen zu gelangen, setzt man den Befehl »list« ein.

Versteckte Informationen

Geben Sie versuchsweise »list« gefolgt von <RETURN> ein. Zu jeder Datei des aktuellen Directories gibt das CLI jetzt zusätzliche Informationen aus. Zuerst erfährt man, wie viele Bytes in einer Datei gespeichert sind. Findet sich hier keine Angabe von Zahlen, sondern die Zeichenfolge »dir«, ist dies der Name eines Dateiverzeichnisses. Die folgenden Buchstaben »rwed« geben an, welche Operationen mit dieser Datei durchführbar sind. Dabei gilt:

- »r« — die Datei kann gelesen werden
- »w« — das File läßt sich überschreiben
- »e« — die Datei kann als Kommando aufgerufen werden
- »d« — die Daten lassen sich löschen

Im Regelfall sind alle vier Schalter auf »an«.

Die letzten beiden Zeichenblöcke geben Auskunft darüber, wann die Datei zum letztenmal gespeichert wurde (Datum und Uhrzeit). Allerdings sollten die letzten Angaben nicht zu genau genommen werden, da für eine genaue Angabe eine im Amiga eingebaute Echtzeituhr erforderlich ist, die batteriegepuffert arbeitet. Da dies aber nicht zur serienmäßigen Ausstattung des Amiga gehört, müssen diese Daten nicht unbedingt mit den Zeiten übereinstimmen, während denen die Datei gespeichert wurde.

Der list-Befehl erlaubt wie der dir-Befehl die Angabe zusätzlicher Parameter. Geben Sie versuchsweise folgendes ein:

```
list df0: p s#?
```

Das CLI zeigt Ihnen nur Dateien an, die als ersten Buch-

staben ein »s« besitzen. Mit dem Parameter »p« läßt sich also eine Selektion der Dateien erzielen. Es werden nur Dateien angezeigt, die in die Maske nach dem Parameter »p« passen. Die Zeichen »#?« ersetzen den Rest des Namens, ein »?« steht für einen einzelnen Buchstaben. Einige Beispiele für zulässige Muster:

```
list df0: p sy#?
list df0: p sy??em
list df0: p #?.info
```

Das erste Beispiel würde alle Dateien anzeigen, deren Anfangsbuchstaben »sy« sind, also etwa »system« und »system.info«. Beim zweiten Befehl gibt das CLI nur Dateien auf dem Bildschirm aus, deren erste und letzten Buchstaben »sy« und »em« wären. Die beiden mittleren Zeichen unterliegen keinen Beschränkungen, könnten also Buchstaben, Zahlen oder Zeichen sein.

Der dritte Befehl zeigt alle Dateien an, deren letzte fünf Zeichen »info« lauten (beispielsweise »system.info«, »trashcan.info« oder »preferences.info«). Diese Maskierung läßt sich auch auf den dir-Befehl anwenden. Zwei Beispiele:

```
dir df0: #?.info
dir df0: ex#?
```

Ein weiterer Parameter beim list-Befehl ist der Buchstabe »s«. Es werden nur Dateinamen ausgegeben, deren Namen den hinter »s« angegebenen Text enthalten.

```
list df0: s refer
```

würde bei eingelegter Workbench-Diskette die Dateinamen »preferences« und »preferences.info« auf dem Bildschirm ausgeben.

Sollen nur Dateien angezeigt werden, die man nach einem bestimmten Datum gespeichert hat, ist folgende Schreibweise anzuwenden:

```
list df0: since Tag-
Monat-Jahr
```

Dabei müssen die jeweiligen Daten in der Form eingegeben werden, wie sie auch bei der list-Ausgabe auf dem Bildschirm erscheinen. Ein Beispiel für eine korrekte Schreibweise:

```
list df0: since
11-jan-87
```

zeigt alle Dateien an, die nach dem 11. Januar 1987 gespeichert wurden. Das Gegenteil des Parameters »since« ist die Angabe von »upto«. In diesem Fall zeigt das CLI nur Dateien

an, die bis zu einem bestimmten Datum angelegt wurden.

```
list df0: upto 11-jan-87
```

gibt alle Dateien auf dem Bildschirm aus, die vor dem 11. Januar 1987 gespeichert wurden.

Der list-Befehl kennt noch drei weitere Optionen, die man jederzeit angeben kann.

Der Zusatz »keys« zeigt zusätzlich noch die Nummer des Datei-Startblocks (File-Header) der jeweiligen Datei auf der Diskette an. Dies ist aber eine Option, die nur dann sinnvoll einzusetzen ist, wenn man über einen Diskettenmonitor und genügend Kenntnis über den Dateiaufbau auf der Diskette verfügt.

Der zweite Zusatz »nodates« unterdrückt die Ausgabe des Datums und der Uhrzeit hinter jeder Datei. Es werden also nur die Länge der Datei und die Stellung der Schalter »rwed« ausgegeben.

Der Parameter »quick« schließlich unterbindet die Ausgabe aller Zusätze. Das CLI gibt nur die Dateinamen aus, entspricht mit dem Zusatz »quick« also weitgehend dem dir-Befehl.

Der list-Befehl ist durch die Vielzahl seiner möglichen Parameter ein komplexer Befehl. Es ist also auch folgende verwirrende Kombination möglich:

```
list df0: p e#? s ref
keys since 11-mar-1986
upto 17-jan-87 nodates
quiet
```

Sie sehen, der list-Befehl erfordert etwas Training, bis man ihn beherrscht.

Die »Ebenen« der Diskette

Wie wir am Anfang dieses Artikels gelesen haben, ist der Inhalt einer Diskette meist auf verschiedene Subdirectories verteilt. Sie können inzwischen das Directory von Haupt- und Unterverzeichnissen lesen. Aber können Sie sich auch die Struktur eines Aufbaus mit Haupt- und Dateiverzeichnissen vorstellen? Deshalb zuerst eine kleine Erläuterung (betrachten Sie dazu bitte auch Bild 1):

Dieses Bild stellt einen kleinen Ausschnitt der Workbench-Diskette dar. Im Bild oben sehen Sie das Hauptverzeichnis, von dem aus wir uns bis jetzt die Unterverzeichnisse mit dem dir- und list-Befehl angesehen haben. Wir hielten uns aber immer im Hauptverzeich-

nis auf. Zuweilen ist es aber nötig, daß wir uns in ein Subdirectory begeben, um dort zu arbeiten. Das CLI stellt für diese Aufgabe den Befehl »cd« (change directory = wechsele das Verzeichnis) zur Verfügung. Gibt man den Befehl »cd« ohne Parameter ein, so zeigt das CLI den Namen des aktuellen Dateiverzeichnisses an. Probieren wir dies gleich aus. Geben Sie bitte ein:

```
cd
```

In Ihrem CLI-Fenster müßte nun der Name der Diskette, der auch unter dem Diskettensymbol auf der Workbench zu sehen ist, erscheinen (Workbench, Workbench 1.x oder ähnlich). Wechseln Sie nun versuchsweise das aktuelle Directory. Nehmen wir an, Sie möchten in das Subdirectory »devs (dir)« überwechseln. Der Befehl dazu lautet:

```
cd devs
```

Überprüfen Sie mit

```
cd
```

ob Sie sich auch im richtigen Subdirectory befinden: In Ihrem CLI-Fenster müßte nun der Name der Diskette und, durch einen Doppelpunkt abgetrennt, der Name des Dateiverzeichnisses erscheinen, in diesem Fall »xxx:devs«. Wenn Sie Bild 1 betrachten, befinden wir uns jetzt im Kreis in der mittleren Reihe, in welchem »devs« steht. Versuchen wir den Befehl »dir«. Wir stellen fest, daß nun andere Dateien erscheinen als im Hauptdirectory. Hier finden sich drei Unterverzeichnisse mit den Namen »keymaps«, »printers« und »clipboards« (die untere Reihe in Bild 1). Mit dem Kommando

```
cd keymaps
```

würden Sie jetzt das Subdirectory »keymaps« zum aktuellen Verzeichnis erklären und befänden sich in der unteren Reihe von Bild 1. Soweit wollen wir aber nicht gehen. Wie gelangt man aber jetzt wieder nach »oben« ins Hauptdirectory? Dazu dient der Befehl

```
cd /
```

der Sie immer in das nächsthöhere (!) Directory zurückbringt. Wenn Sie sich beispielsweise in einem Subdirectory aufhielten, das vielleicht in der vierten Ebene angelegt ist, kämen Sie mit dem Befehl »cd/« nur zurück in die dritte Ebene. Um in das Hauptdirectory zu gelangen, wäre folgender Befehl nötig:

```
cd ////
```

Doch das geht auch einfacher: Mit

```
cd df0:
```

erreichen Sie immer das Hauptdirectory, egal, in welcher Tiefe Sie sich gerade aufhalten. (Sollten Sie auf einem anderen Laufwerk als »df0« arbeiten, ist natürlich immer das aktuelle Laufwerk anzugeben. »cd df0:« bringt Sie nicht ins Hauptdirectory von »df1«, wenn gerade auf »df1« gearbeitet wird!)

Selbstverständlich kann auch zwischen einzelnen Directory-Zweigen gewechselt werden. Dabei muß allerdings immer der Umweg über das Hauptdirectory genommen werden. Wenn Sie sich gerade im Dateiverzeichnis »devs« aufhalten, gelangen Sie nicht mit »cd trashcan« in dieses Nebendirectory. Die korrekte Schreibweise wäre in diesem Fall

```
cd /trashcan
```

oder

```
cd df0:trashcan
```

Befindet man sich sogar eine Ebene tiefer (etwa im Subdirectory »printers«, Bild 1), wäre folgender Befehl richtig:

```
cd ../trashcan
```

oder

```
cd df0:trashcan
```

Nehmen wir an, im Dateiverzeichnis »trashcan« wäre ein weiteres Subdirectory mit dem Namen »XXX« und wir wollen vom Unterverzeichnis »printers« nach »XXX« überwechseln. Dies ist etwas schwerer nachvollziehbar:

```
cd ../trashcan/XXX
```

oder, etwas einfacher

```
cd df0:trashcan/XXX
```

Der cd-Befehl dient aber nicht nur zum Wechseln der Directory-Ebenen, sondern auch dem Anwählen des gewünschten Laufwerks. Nehmen wir an, das interne Laufwerk »df0« ist derzeit das aktuelle Laufwerk. Sie möchten aber mit einer Diskette arbeiten, die sich in der externen Floppy-Station befindet, also in »df1«. Das Überwechseln geschieht mit folgendem Befehl:

```
cd df1:
```

Schon arbeiten Sie mit der Diskette in »df1«. Natürlich ist es auch möglich, direkt in bestimmte Verzeichnisse zu springen:

```
cd df1:XXX
```

wobei XXX der Verzeichnisna-

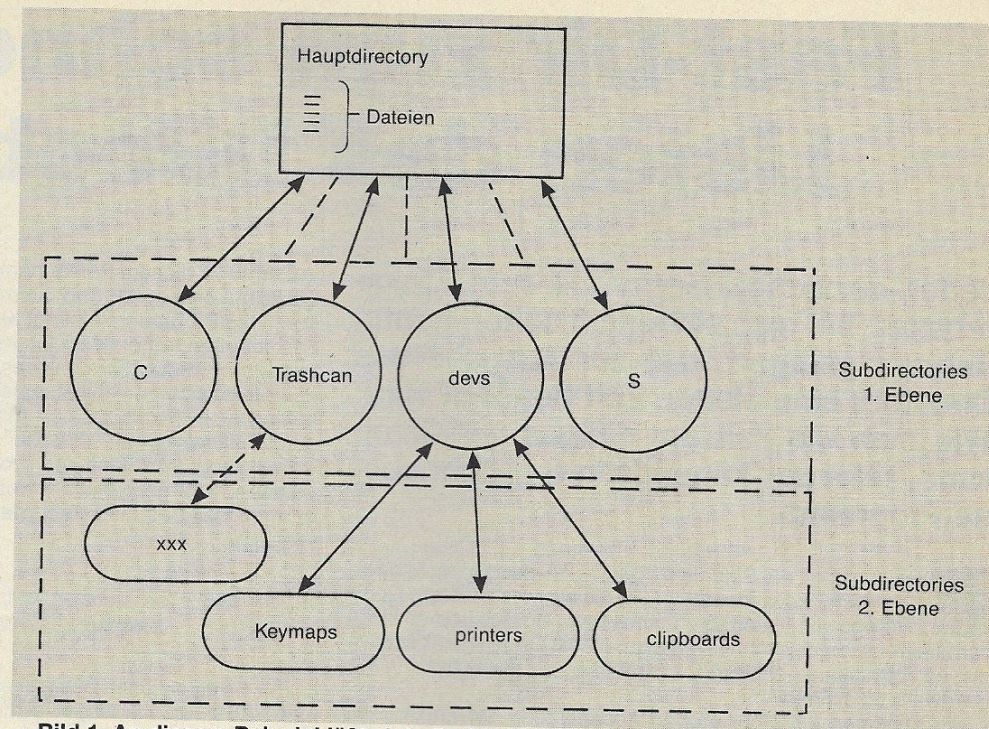


Bild 1. An diesem Beispiel läßt sich die Struktur der Subdirectories besser verstehen

me oder eine Kombination aus Verzeichniswegen sein darf. Ein Beispiel: Das aktuelle Laufwerk ist »df1«. In »df0« befindet sich die Workbench-Diskette, von der in das gedachte Verzeichnis »XXX« im Verzeichnis »devs/printers« gewechselt werden soll. Der Aufruf sieht dann folgendermaßen aus:

```
cd df0:devs/printers/XXX
```

Schon befindet man sich im Subdirectory »XXX« im Verzeichnis »devs/printers«.

Wie bereits weiter oben erwähnt, besitzt der Amiga eine Uhr, bei der auch das Datum mit gespeichert wird. Da der Amiga normalerweise nicht über eine (nötige!) batteriegepufferte Echtzeituhr verfügt, holt sich das Betriebssystem beim Laden der Workbench das Datum der letzten Modifikation von der Diskette und verwendet diese Daten als aktuelle Zeit. Diese Uhr kann aber auch von Ihnen gestellt werden. Dazu findet sich im CLI der Befehl »date«. Ohne Parameter einzugeben erfährt man die »aktuelle« Zeit und das Datum. Sollte es nötig sein, diese Daten zu aktualisieren, muß der date-Befehl folgendermaßen eingesetzt werden:

```
date 15-apr-87
```

um das Datum zu stellen. Die Uhrzeit läßt sich genauso leicht einstellen:

```
date 01:42:13
```

Diese Schreibweise ist auch kombinierbar. Mögliche Formen:

```
date 15-apr-87 12:32:00
date 14:12 13-mar-99
```

Die Reihenfolge spielt keine Rolle. Das CLI unterscheidet Zeit und Datum anhand des Doppelpunktes und Bindestriches.

Beim nächsten Schreibzugriff auf Diskette wird die nun aktualisierte Zeit gespeichert und ist (besitzt man keine gepufferte Uhr) die Basis beim nächsten Neustart.

Es gibt aber noch einen kleinen Trick beim Setzen des Datums: Sie können das aktuelle Datum relativ leicht um einen Wochentag erhöhen oder erniedrigen. Wird beispielsweise regelmäßig jeden Tag mit einer bestimmten Diskette (idealerweise der Workbench-Kopie) gearbeitet, erhöht folgendes Kommando automatisch das aktuelle Datum um einen Tag:

```
date tomorrow
```

Schon wurde aus einem »11-06-87« das Datum »12-06-87«. Ebenso leicht kann ein Tag abgezogen werden:

```
date yesterday
```

Bindet man diesen Befehl in die »startup-sequence« ein (wie das gemacht wird, erfahren Sie in einem späteren Teil dieses Kurses), aktualisiert der Amiga bei jedem Start das Datum.

Damit wollen wir den ersten Teil des CLI-Kurses abschließen. Um die Arbeit mit dem CLI zu beenden, geben Sie abschließend den Befehl »endcli« ein. Danach befinden Sie sich wieder auf der Workbench. Sie haben bereits einige Befehle kennengelernt, die Ihnen einen Eindruck von der Leistungsfähigkeit des CLI vermitteln. Sollten Sie noch Schwierigkeiten bei der Schreibweise haben, hilft die Eingabe eines durch ein Leerzeichen abgetrennten Fragezeichens hinter jedem Befehl weiter. Das CLI zeigt danach die möglichen Parameter und deren Schreibweise an.

In Teil 2 werden wir Befehle vorstellen, mit denen kopiert, gelöscht oder eine Datei umbenannt werden kann. Bis dahin, gutes Gelingen und viel Spaß mit dem Amiga. (dm)

In welchem Kursteil finde ich folgende Befehle?

Teil 1: dir, list, cd, date, endcli

Teil 2: filenote, delete, relabel, rename, type, disk-copy, info, format, install

Teil 3: makedir, ed, prompt, wait, loadwb

Teil 4: assign, echo, failat, if, say, stack