

CLI — Die zweite Haut des Amiga

Teil 2

Die zweite Anwenderschnittstelle des Amiga, das CLI (Command-Line-Interface), ist ein mächtiges Instrument im Umgang mit diesem Computer. Dieser Kurs deckt die Möglichkeiten auf, die das CLI dem Amiga-Anwender bietet.

Eine Möglichkeit, den Amiga zu bedienen, ist das Arbeiten mit der Workbench. Hier werden alle Operationen mit der Maus erledigt, die Tastatur muß nur noch in seltenen Fällen benutzt werden. Leider lassen sich über die Workbench-Ebene nicht alle Amiga-DOS-Funktionen benutzen. Hierfür ist das auf der Workbench-Diskette zu findende CLI (Command-Line-Interface) das geeignete Werkzeug. Dieser Kurs wird Sie mit allen Finessen des CLI vertraut machen. Nachstehend finden Sie eine Übersicht über die Befehle, die bereits besprochen wurden und in den nächsten Ausgaben behandelt werden.

Kursübersicht:

Teil 1: dir, list, cd, date, endcli
 Teil 2: filenote, delete, relabel, rename, type, copy, diskcopy, info, format, install
 Teil 3: makedir, ed, prompt, wait, loadwb, setclock
 Teil 4: assign, echo, failat, if, say, stack, protect
 Teil 5: run, edit, execute, join, lab, quit, skip

Nachdem in der letzten Folge Befehle behandelt wurden, die dem Betrachten des Directories (Inhaltsverzeichnis) der Diskette dienen, setzen wir die Reihe mit Befehlen fort, die gezielt Dateien beeinflussen können. Dieser Kurs läßt sich natürlich dann am besten nachvollziehen, wenn das Gelernte gleich in die Praxis umgesetzt wird. Schalten Sie bitte Ihren Amiga an und legen Sie eine Kopie der Workbench-Diskette ein. Aktivieren Sie anschließend das CLI (im »System«-Directory) durch einen Doppelpunkt mit der Maus. Sie können nun die Maus aus der Hand legen, da das CLI nur über die Tastatur bedient wird.

Da wir natürlich keine wichtigen Dateien beschädigen wollen, erstellen wir uns eine kleine Testdatei. Geben Sie dazu

bitte folgende Befehle ein (die Erklärung folgt später):

```
COPY * TO df0:Test
```

Der Cursor springt nun in eine neue Zeile, liefert aber kein Prompt (>1) mehr. Drücken Sie jetzt zwei- bis dreimal <RETURN> und anschließend <CTRL \> (dies ist der Schrägstrich, der auf der Tastatur in der obersten Reihe rechts vor dem Linkspfeil steht).

Die Floppystation läuft kurz an und erzeugt eine leere Datei mit dem Namen »Test«. Im folgenden werden wir uns bei Diskettenoperationen auf diese Datei beschränken, um keine falschen Dateien zu beschädigen!

Das DOS (Diskettenbetriebssystem) des Amiga erlaubt es, an jede Datei auf der Diskette einen Kommentar von 80 Zeichen anzufügen. Dieser Kommentar kann dazu dienen, bestimmte Informationen zu einer Datei festzuhalten. Beispielsweise, um welche Version des Programms es sich dabei handelt oder welche Dateien noch zu dem Programm gehören. Der Kommentar läßt sich mit dem List-Befehl (siehe Folge 1) betrachten. Dabei wird in einer zusätzlichen Zeile der Kommentar, falls vorhanden, auf dem Bildschirm ausgegeben. Der Text kann auch von der Workbench-Ebene aus betrachtet werden, und zwar mit dem Kommando »Info« aus dem ersten Pull-Down-Menü.

Der Kommentar verfügt über einige Eigenschaften, die möglicherweise nicht sofort verständlich sind. So wird er beispielsweise beim Kopieren einer Datei nicht mitübertragen. Die kopierte Datei, die eigentlich ein identisches Abbild der ersten Datei sein sollte, enthält also keinen Kommentar!

Ein weiterer Punkt ist folgender: Wird eine Datei mit einem neuen Inhalt überschrieben, bleibt der Kommentar der alten Datei erhalten (auch wenn dieser nun nicht mehr zutrifft).

Doch nun zum eigentlichen Befehl, der das Anlegen eines Kommentars ermöglicht: dem Befehl »Filenote«. Mit diesem Kommando kann ein Text zu einer Datei eingegeben werden. Der Text kann aber mit diesem Befehl nicht angesehen werden. Dies ist nur mit dem List-Befehl machbar.

Versuchen wir gleich einmal, eine Datei mit einem Kommentar auszustatten. Vergewissern Sie sich zuerst, daß die eingelegte Diskette nicht schreibgeschützt ist!

Als Beispiel wollen wir der Datei »Test« einen Kommentar anfügen. Der Befehl dazu lautet folgendermaßen:

```
Filenote Test "Kommentar"
```

Dabei sollte der Kommentar immer in Anführungszeichen gesetzt werden, damit auch Leerzeichen oder Sonderzeichen korrekt übernommen werden. Kontrollieren wir gleich, ob der Befehl korrekt ausgeführt wurde. Geben Sie nun ein:

```
List Test
```

Auf dem Bildschirm sollten jetzt die vom List-Befehl her bekannten Ausgaben erscheinen. Zusätzlich findet sich in einer neuen Zeile, durch einen Doppelpunkt eingeleitet, der Kommentar, den wir eben angefügt haben.

Der Text kann auch wieder gelöscht werden. Dazu ist nur anstelle des Kommentars ein leerer String zu übergeben.

Nach Eingabe von
 Filenote Test ""

ist die Datei wieder vom Kommentar befreit. Prüfen Sie dies mit

```
List Test
```

nach. Die Kommentarzeile, die vorher durch einen Doppelpunkt eingeleitet wurde, ist nun verschwunden.

Ein weiterer, wichtiger Befehl des CLI ist das Kommando »Info«. Der Befehl gibt Auskunft über die angeschlossenen Disketten- und Festplattenlaufwerke. Nach Eingabe des Befehls

```
Info
```

wird ein Statusreport über alle angeschlossenen Laufwerke und die RAM-Floppy ausgegeben (Bild 1).

Dieser Befehl dient hauptsächlich dazu, einen schnellen Überblick über die momentan verfügbaren Disketten- und Festplattenspeicher sowie deren Kapazitäten zu geben. Er eignet sich auch zum Einbinden in die »Startup-Sequenz«. In diese Datei eingebaut, erhält der Anwender gleich beim Start des Amiga die Information über die angeschlossenen Laufwerke angezeigt (wie der Befehl in die »Startup-Sequenz« implementiert werden kann, erfahren Sie in einem der folgenden Kursteile).

Die nächsten Befehle, mit denen wir uns befassen, nehmen direkten Einfluß auf Dateien oder die gesamte Diskette. Diese Befehle bewirken in den meisten Fällen eine direkte Veränderung einzelner Bytes, Dateien oder der kompletten Diskette. Sie sind deshalb mit Vorsicht zu genießen, da durch falsche Handhabung leicht eine Datei oder Diskette »zerstört« werden kann.

Der erste Befehl dieser Art, der besprochen wird, dient dem Neu- beziehungsweise Umbenennen einer Datei oder eines Dateiverzeichnisses

WB A2000-1.6.87:									
2) info									
Mounted disks:									
Unit	Size	Used	Free	Full	Erps	Status	Name		
JH1	340K	7	671	1%	0	Read/Write	JH1		
JH0	20M	14349	26449	35%	0	Read/Write	JH0		
DF2	880K	1670	88	94%	0	Read/Write	ExtrasD		
DF1	880K	1744	14	99%	0	Read Only	WB A2000-1.6.87		
DF0	880K	1742	16	99%	0	Read/Write	Workdisk DM		
RAM	1K	1	0	100%	0	Read/Write			
Volumes available:									
ExtrasD [Mounted]									
WB A2000-1.6.87 [Mounted]									
Workdisk DM [Mounted]									
RAM Disk [Mounted]									
JH0 [Mounted]									
JH1 [Mounted]									
2) #									

Bild 1. Ein mit »info« erstellter Report

(Subdirectory). Es ist dies der Befehl »Rename«.

Die Schreibweise des Befehls sieht folgendermaßen aus:

```
RENAME alter Name TO
neuer Name
```

Benennen wir doch gleich einmal eine Datei um. Nehmen wir als Beispiel das Programm »Test«, das den neuen Namen »Abfall« erhalten soll. Der Befehl dazu sieht wie folgt aus:

```
RENAME Test TO Abfall
```

Eine Kontrolle mit dem Dir-Befehl ergibt, daß die ehemalige Datei »Test« nun den Namen »Abfall« trägt. Da wir aber den ursprünglichen Namen behalten wollen, benennen wir das Unterverzeichnis wieder um:

```
RENAME Abfall TO Test
```

(Die Angabe von »TO« wäre nicht unbedingt nötig, erleichtert aber die Lesbarkeit des Befehls.) Sollen in einem der beiden Namen Sonderzeichen oder Leerzeichen verwendet werden, so sind die Namen in Anführungszeichen zu setzen. Der Rename-Befehl kann aber noch mehr, als nur eine Datei umbenennen! Geben Sie probierhalber diesen Befehl ein:

```
RENAME Test TO Trashcan/
Test1
```

Der Befehl mag Ihnen vielleicht etwas merkwürdig erscheinen, doch in dieser Weise wirkt der Rename-Befehl wie eine Move-Anweisung. Wenn Sie sich mit »Dir« das Directory anzeigen lassen, werden Sie feststellen, daß die Datei »Test« nicht mehr zu sehen ist. Sie ist aber nur aus dem Hauptdirectory verschwunden. »Test« befindet sich jetzt mit dem neuen Namen »Test1« im Subdirectory »Trashcan«, wurde also unter gleichzeitiger Änderung des Namens in ein anderes Directory verlagert.

Auch diesen Vorgang wollen wir wieder rückgängig machen. Die Kommandofolge

```
CD df0:
RENAME Trashcan/Test1 TO
Test
```

bringt die Datei wieder ins Hauptdirectory zurück und gibt ihr den alten Namen, wie eine Kontrolle mit »Dir« beweist.

Ein weiteres Kommando zum Ändern eines Namens ist der Befehl »Relabel«. Allerdings können damit keine Dateien oder Subdirectories umbenannt werden. Dieser Befehl wirkt nur auf den Namen einer Diskette. Möchten Sie also den Namen Ihrer Diskette (der

auch unter dem Diskettensymbol in der Workbench erscheint) ändern, ist die nachstehende Form zu verwenden (es sind bis zu 30 Zeichen erlaubt):

```
RELABEL Laufwerk: "neuer
Name"
```

Die Laufwerksangabe ist nicht zwingend erforderlich, empfiehlt sich aber der Übersichtlichkeit halber. Auf die Praxis angewendet würde die Befehlsfolge für Ihre im internen Laufwerk eingelegte Workbench-Kopie so aussehen:

```
RELABEL df0: "Meine
Workbench"
```

Zu beachten ist hierbei, daß die Leerzeichen nicht vergessen werden dürfen. Des weiteren muß ein Ausdruck, der Leerzeichen oder Sonderzeichen enthält, in Anführungszeichen gestellt werden. Wenn Sie das obige Beispiel eingegeben haben, trägt die eingelegte Diskette nun den Namen »Meine Workbench«. Dies können Sie leicht überprüfen, wenn Sie den Befehl »Info« eingeben.

Löschen und Kopieren

Von Zeit zu Zeit kommt es vor, daß man einige Dateien oder Unterverzeichnisse auf seiner Diskette löschen möchte, um beispielsweise Platz für neue Programme zu schaffen. Dazu dient der Befehl »Delete«, der aber sehr mit Vorsicht zu genießen ist, da die betreffenden Dateien endgültig verloren sind (es existieren zwar schon einige Möglichkeiten, die Dateien wieder zu retten, doch ist dafür ein genaues Wissen über den Diskettenaufbau sowie der Besitz eines Diskettenmonitors nötig). Die folgenden Beispiele sollten also nicht ausprobiert werden, wenn Sie nicht sicher sind, daß Sie die betreffenden Dateien noch auf anderen Disketten besitzen. Da wir aber mit einer Kopie (!) der Workbench-Diskette arbeiten, sollte uns das nicht weiter stören, da ja noch ein Original vorhanden ist.

Schreiten wir auch hier gleich wieder zur Praxis und versuchen, eine Datei zu löschen. Hier bietet sich das File »Test« an, welches keine »lebensnotwendigen« Daten enthält. Gelöscht wird die Datei mit folgendem Befehl:

```
DELETE Test
```

oder, falls Sie sich in einem Un-

terverzeichnis oder einer anderen Diskette befinden, mit

```
DELETE df0: Test
```

Vergessen Sie nicht die Leerzeichen zwischen den einzelnen Parametern!

Es können auch mehrere Dateien (maximal 10) mit einem einzigen Befehl gelöscht werden. Dazu sind nur die einzelnen Namen durch ein Leerzeichen getrennt in die gleiche Befehlszeile zu schreiben. Das nächste Beispiel würde fünf verschiedene Dateien löschen:

```
DELETE df0: Test1 Test2
Test3 Test4 Test5
```

Selbstverständlich können auch Pfadnamen oder Masken verwendet werden. Drei weitere Beispiele:

```
DELETE df0: P#?
DELETE df0: Devs/
Keymaps/USA2
DELETE df0: System/
Er#?t
```

Im ersten Beispiel würden alle Dateien gelöscht, die im Hauptverzeichnis von Laufwerk 0 stehen und mit »P« beginnen. Der zweite Fall löscht das Programm »USA2«, das sich im Subdirectory »Devs/Keymaps« befindet. Der dritte Befehl schließlich löscht alle Dateien, die sich im Unterverzeichnis »System« befinden, mit »Er« beginnen und als letzten Buchstaben ein »t« besitzen.

Der Delete-Befehl gibt bei mehr als einer gelöschten Datei eine Meldung aus. Wer diese Meldung unterbinden möchte, kann am Schluß der Befehlszeile »QUIET« oder abgekürzt »Q« eingeben. Dieser Zusatz unterdrückt die Ausgabe der Meldung. Ein Beispiel:

```
DELETE Test1 Test2 Test3
QUIET
```

oder abgekürzt

```
DELETE Test1 Test2 Test3
Q
```

Auch beim Delete-Befehl kann mit Jokern und Maskierung gearbeitet werden (#?, ??).

»Delete« läßt außer »Quiet« noch den Parameter »All« zu, der, am Schluß der Befehlszeile angehängt, folgende Wirkung hat: Wenn Sie beispielsweise die Kommandozeile

```
DELETE df0: De#? ALL
```

eingeben würden, so wären anschließend alle Dateien und auch Subdirectories verschwunden, deren erste beiden Buchstaben mit »De« begonnen hätten. Man kann die

Maskierung auch noch weiter treiben. Der Befehl

```
DELETE #?/#?E#?
```

würde sämtliche Dateien auf der gesamten (!) Diskette löschen, die irgendwo in ihrem Namen ein »e« enthalten (auch alle Subdirectories!). Mit diesem Befehl sollte also sehr vorsichtig umgegangen werden, da man leicht Dateien unrettbar löschen kann, die noch gebraucht werden. Außerdem erfolgt vor der Befehlsausführung keine (!) Sicherheitsabfrage, ob die Dateien wirklich entfernt werden sollen.

Eines ist beim Delete-Befehl noch anzumerken: Es kann kein Subdirectory gelöscht werden, wenn in ihm noch Dateien enthalten sind. Bevor das Unterverzeichnis gelöscht werden kann, müssen also zuerst alle (!) Dateien aus diesem entfernt werden.

Des weiteren können sich Dateien auch Löschversuchen widersetzen. Jede Datei enthält vier Flags (Schalter), die bestimmte Manipulationen mit der Datei zulassen oder verbieten (siehe Amiga 6/7, Seite 60). Den Zustand dieser vier Flags (»r-w-e-d«) können Sie sich auch mit dem List-Befehl ansehen. Nur wenn das »d«-Flag gesetzt ist, läßt sich die Datei entfernen. Ändern lassen sich diese Flags mit dem »Protect«-Befehl, der im vierten Teil des CLI-Kurses besprochen wird.

Vorbereiten einer Diskette

Um Daten auf eine Diskette speichern zu können, ist nötig, daß die betreffende Diskette formatiert ist. Formatieren bedeutet in diesem Fall, daß die Magnetscheibe(n) in konzentrische Kreise und die Kreise in einzelne Sektoren eingeteilt werden. Es soll hier nicht näher auf die Aufteilung einer Diskette eingegangen werden. Wichtig ist hier nur zu wissen, daß Disketten auf jeden Fall formatiert sein müssen, bevor sie genutzt, das heißt Daten auf ihnen gespeichert werden können. Beim Amiga dient dazu der Befehl »Format«. Dieser Befehl ermöglicht es allerdings nur, Disketten zu formatieren. Festplatten oder 5¼-Zoll-Laufwerke können mit diesem Befehl nicht vorbereitet werden (auf diesen speziellen Befehl wird in einer späteren Folge eingegangen). Der Befehl benötigt folgende Schreibweise:

```
FORMAT DRIVE Laufwerk:
NAME Name NOICONS
```

Es ist auf die korrekte Schreibweise zu achten. Leerzeichen dürfen nicht vergessen werden! Außerdem sind die Formalparameter »DRIVE« und »NAME« immer mit anzugeben. Der Parameter »NO-ICONS« kann, muß aber nicht eingetippt werden. Läßt man ihn weg, installiert das DOS auf der Diskette das Subdirectory »Trashcan« und das zugehörige Icon (Symbol). Gibt man dagegen »NOICONS« an, werden die »Trashcan«-Dateien nicht erzeugt. Um beispielsweise eine Diskette im externen Laufwerk (df1) zu formatieren, müßte folgende Befehlszeile eingegeben werden (versuchen Sie nicht, diesen Befehl nachzuvollziehen, wenn Sie sich nicht sicher sind, ob Sie die Daten in dem betreffenden Laufwerk noch benötigen. Der Amiga verlangt zwar zuerst eine Sicherheitsbestätigung, doch wie leicht kann man sich vertippen... Nach der Formatierung sind diese Daten unrettbar (!) verloren!):

FORMAT DRIVE df1: NAME
"Leere Diskette"

Auch hier muß der letzte Ausdruck wieder in Anführungszeichen gesetzt werden,

wenn Leerzeichen oder Sonderzeichen enthalten sind (zulässig sind insgesamt 30 Zeichen). Wenn die Diskette nicht schreibgeschützt war, besitzt sie nun den Diskettenamen »Leere Diskette« und kann beschrieben werden.

Eine so vorbereitete Diskette kann in diesem Stadium allerdings nur zur reinen Datenspeicherung verwendet werden. Wie Sie sicher wissen, verlangt der Amiga nach dem Einschalten beziehungsweise Laden des Kickstarts eine Workbench-Diskette. Der Amiga würde die eben formatierte Diskette aber nicht als Startdiskette anerkennen, sondern solange nach einer Workbench-Diskette verlangen, bis eine mit einem Startsektor versehene Diskette eingelegt wird. Dies erledigt der nächste Befehl: »Install« (nicht bei Festplatten oder 5 1/4-Zoll-Laufwerken).

Um beispielsweise eine Diskette in Laufwerk #1 (df1) startfähig zu machen, reicht die Eingabe der nächsten Zeile:

INSTALL df1:

Das Laufwerk läuft kurz an

und installiert den Startsektor. Nach diesem Vorgang akzeptiert der Amiga diese Diskette als bootfähige (Workbench-) Diskette.

Kopieren von Dateien

Jeder Anwender muß von Zeit zu Zeit Dateien von einem Gerät auf ein anderes kopieren. Sei es, um seinen Druckertreiber oder ein anderes Programm auf eine andere Diskette zu übertragen oder, im Fall des Amiga, eine Datei auszudrucken.

Zum Kopieren einer einzelnen Datei dient der Befehl »Copy«. Dieses Kommando dupliziert den Inhalt einer angegebenen Datei auf ein anderes Gerät. Diese können Diskettenlaufwerke, Festplatten, das RAM, Drucker oder im einfachsten Fall der Bildschirm sein.

Besprechen wir also der Reihe nach die verschiedenen Möglichkeiten, wie man mit dem Copy-Befehl Dateien vervielfältigen kann:

Die erste Art bezieht sich natürlich auf diejenigen Geräte, wo dieser Befehl bevorzugt eingesetzt wird: den Disketten-

oder Festplatten-Laufwerken und das RAM.

Die Schreibweise dafür lautet allgemein:

COPY Quell-Datei TO
Ziel-Datei/-Gerät

Erklären wir dies gleich an einem Beispiel: Sie möchten die Datei »Clock« von Ihrer Workbench-Diskette im internen Laufwerk (df0) auf eine zweite Diskette im ersten angeschlossenen Laufwerk (df1) kopieren. Die Datei befindet sich im Hauptdirectory und soll auch auf der zweiten Diskette im Hauptdirectory stehen. Der Befehl dazu würde lauten:

COPY Clock TO df1:

oder, noch exakter

COPY df0:Clock TO df1:

(Das »TO« ist nicht zwingend erforderlich, erhöht aber die Lesbarkeit.) Wenn wir davon ausgehen, daß wir uns von Anfang an im Hauptdirectory der im internen Laufwerk (df0) eingelegten befinden haben, sind beide Befehle richtig. Ist aber eine andere Diskettenstation das aktuelle Laufwerk oder befindet man sich in einem Subdirectory, wäre nur

AMIGA-LAUFWERK 500/1000/2000

- NEC 1036A
- 100% kompatibel / 880 KB
- 3,5-Zoll-Slimline
- Metallgehäuse (AMIGA-Farbe)
- Anschlußfertig

NEC 1036A

349,- DM

259,- DM

AMIGA 1000 SPEICHERERWEITERUNG (INTERN)

- 4 MB Speichererweiterung
- variable Bestückung (512 KB, 1, 2, 4 MB)
- batteriegepufferte Echtzeituhr
- Fast-RAM (no Waitstates)
- 1 MB bestückt

Leerplatte m. Bestückungsplan

749,- DM

170,- DM

MICHAEL KRÖNING Computersysteme

Deichsberg 2 · 4790 Paderborn · Telefon 052 54/693 69, ab 16.00 Uhr

Versand per Nachnahme!



COMPUTERSOFT JONIGK

AMIGA			
Archon II	89,-	Surgeon	169,-
Typoon	86,-	Starcross	96,-
Balance of Power	139,-	Starfighter	86,-
Deep Space	96,-	Strip Poker	49,-
Defender of the Crown	96,-	The Pawn	96,-
Deja Vu	96,-	The Guild of Thieves	96,-
Fight Simulator II	149,-	The Seven Cities of Gold	96,-
Golf Games	119,-	Trinity	109,-
Hacker II	89,-	Ultima III	86,-
Leaderboard	86,-	Univert	96,-
Leaderboard Tournament	49,-	Winter Games	79,-
Marble Madness	89,-	World Games	79,-
Mean 18	96,-	***** ANWENDER *****	
Portal	119,-	Aegis Images (Malprogramm)	96,-
Quiver	69,-	Aegis Draw (CAD-Programm)	296,-
SDI (1 MByte RAM)	129,-	Aegis Draw Plus (CAD-Programm)	599,-
Shanghai	86,-	Aegis Impact (Business Graph.)	439,-
Sinbad	129,-	Delux Music	296,-
		Delux Paint II	296,-
		Delux Print	249,-
		Deluxe Video (V1.2)	296,-
		Diskwerk (Disk. Utility)	129,-
		Lattice C Compiler	449,-
		Marauder II (Copierprogramm)	128,-
		Metacomco Lisp	458,-
		Metacomco Macro Assembler	249,-
		Metacomco Pascal	238,-
		Metacomco Toolkit	128,-
		Modula II (Diy)	498,-
		Page Setter (Zeitungslayout)	359,-
		Power Window	239,-
		Superbase (Datenwerk)	249,-
		Zing V1.2	249,-
		Digi View (Digitizer)	658,-
		Compucon (2-Megabyte-Erweiterung)	1998,-
		Jitter Rid (Bildschirmfilter)	79,-
		Discovery IV	198,-
		Discovery Demodiskette	50,-
		Vizawrite Amiga IV	498,-
		Vizawrite Amiga Demodiskette	20,-



CSJ COMPUTERSOFT JONIGK

An der Tiefenriede 27 · 3000 Hannover 1 · Tel. Bestellservice (0511) 8663 83
Wir erwarten täglich neue Programme für den Amiga
Versand Inland: Inland Vorkasse + 2,50 DM (Euroscheck in DM); per Nachnahme + 7,- DM

* HÄNDLERANFRAGEN ERWÜNSCHT *

*** CSJ AMIGA NEWS anfordern ***

Wenn mal was nicht funktioniert ...

Computer-Service

commodore

Schneider

ATARI

Installation
Wartung
Reparatur



Technischer Kundendienst

Im gesamten Bundesgebiet vertreten
70 Niederlassungen

Hotline Bereich Nord (040) 2201913

Hotline Bereich Mitte (0201) 35923

Hotline Bereich Süd (08165) 74220

der zweite Befehl richtig (näheres dazu im 1. Teil, Ausgabe 6/7).

Im nächsten Fall soll die gleiche Datei vom selben Laufwerk auf df1 kopiert werden. Die Kopie erhält aber einen anderen Namen. Die Schreibweise:

```
COPY df0:Clock TO df1:
"Neuer Name"
```

Sie können also gleich bei der Befehlseingabe den neuen Namen der Zieldatei festlegen und müssen sie nicht erst später umbenennen. Dies funktioniert auch mit Pfadangaben, um in Unterverzeichnisse zu gelangen. Beispiel: Die Datei »List«, die sich im »C«-Directory von df0 befindet, soll auf df1 in das Unterverzeichnis »XXX« übertragen werden. Gleichzeitig erhält die Datei den neuen Namen »Test«. Die Kommandozeile

```
COPY df0:C/List TO df1:
xxx/Test
```

erledigt diese Aufgabe. Natürlich kann der Copy-Befehl auch maskiert werden (maskieren = Platzhalter einsetzen).

```
COPY df0:C/A#? TO df1:xxx
```

Dieser mit Jokern arbeitende Befehl überträgt beispielsweise alle Dateien, die sich im C-Directory von Laufwerk #0 befinden und mit »A« beginnen, in den Ordner »XX« von Laufwerk #1.

Auch beim Copy-Befehl können Sie, wie beim Delete-Befehl, zwei Parameter am Schluß der Befehlszeile eingeben. Der erste ist »QUIET« oder »Q«, der die schon vom Delete-Kommando her bekannte Funktion hat, nämlich die Unterdrückung von Textausgaben auf dem Bildschirm. Beispiel:

```
COPY df0:C TO df1: QUIET
```

Das C-Directory (df0) wird in das Hauptdirectory von df1 kopiert. Dabei werden Bestätigungsmeldungen des Copy-Befehls unterbunden.

Der zweite Formalparameter ist das Wort »ALL«. Damit sagen Sie dem Copy-Befehl, daß, falls Sie ein Directory kopieren wollen, auch alle Unterverzeichnisse dieses Ordners mitkopiert werden sollen (normalerweise werden nur die Dateien, nicht (!) die Subdirectories kopiert). Um dies zu verdeutlichen, folgendes Beispiel:

```
COPY df0:Devs TO df1:
```

Es würden alle Dateien übertragen, die sich im Verzeichnis

»Devs« befinden, nicht aber die Unterverzeichnisse.

Folgende Befehlszeile würde auch noch die Unterverzeichnisse mitkopieren und gleichzeitig auf der Zieldiskette die entsprechenden Ordner erzeugen:

```
COPY df0:Devs TO df1: ALL
```

Dies würde auch mit Festplattenlaufwerken (dh0/dh1 beziehungsweise beim Amiga 2000 jh0 etc.) und dem RAM funktionieren. Beispiele:

```
COPY df0:c TO dh0: ALL
```

```
COPY df1:c TO RAM:
```

Wie weiter oben erwähnt, läßt sich die Ausgabe einer Datei auch auf den Drucker umlenken. Der Drucker würde also eine schriftliche Kopie einer Datei ergeben. Unter AmigaDOS wird ein angeschlossener Drucker durch »prt:« (für Printer) angesprochen. Legen wir als Beispiel die Ausgabe einer Datei um, beziehungsweise kopieren die Datei auf ein Blatt Papier. Dafür benutzen wir die Startdatei des Amiga (Startup-Sequence), die sich im S-Verzeichnis befindet und von der wir wissen, daß sie lesbaren ASCII-Text enthält. Die Befehlszeile sieht so aus:

```
COPY df0:S/startup-
sequence TO prt:
```

Schon wird ein Ausdruck dieser Datei erzeugt.

Auf diese Weise können beispielsweise auch Daten über die serielle »SER:« oder parallele »PAR:« Schnittstelle gesendet werden.

Ein weiteres Gerät ist der Bildschirm »CON:«.

Wenn Sie, wie auch im AmigaDOS-Handbuch beschrieben, nachstehende Zeile abtippen, werden alle Eingaben von Tastatur (die Tastatur wird durch den Stern »*« gekennzeichnet) auf ein neu eröffnetes Bildschirmfenster umgeleitet. Sie tippen also etwas ein, das anschließend direkt in einem anderen Fenster, das den Titel »Ausgabefenster« trägt, ausgegeben wird (Abbruch mit <CTRL C> oder <CTRL \>, eingegeben im CLI-Fenster).

```
COPY * TO CON:10/10/200/
100/Ausgabefenster
```

Vielleicht verstehen Sie jetzt auch die Anweisung am Anfang des Artikels, mit der wir die Test-Datei erzeugt haben. Diese Anweisung macht nichts anderes, als Tastatureingaben in eine Datei mit dem Namen »Test« umzuleiten und nach

Druck auf <CTRL \> auf Diskette zu speichern.

Ein Befehl, der (wenn keine Kopierprogramme vorhanden sind) das Kopieren kompletter Disketten erlaubt, ist das Kommando »Diskcopy«. Damit lassen sich originalgetreue Kopien von 3½-Zoll-Disketten anfertigen. Kaputte Sektoren oder Tracks werden damit aber nicht übertragen. Diskcopy führt dieselbe Funktion aus, als wenn Sie auf der Workbench-Ebene zwei Diskettensymbole übereinander legen. Dabei ist es nicht nötig, die Zieldiskette vorher zu formatieren. Dies wird von »Diskcopy« erledigt. (Es empfiehlt sich, die Quelldiskette mit Schreibschutz zu versehen.) Das Format des Befehls sieht so aus:

```
DISKCOPY Quell-Laufwerk
Ziel-Laufwerk
```

Ein Beispiel: Es soll von der Diskette, die sich im internen Laufwerk (df0) befindet, eine Sicherungskopie auf eine Diskette erstellt werden, die sich in df1 befindet:

```
DISKCOPY df0: TO df1:
```

Nach Eingabe dieses Befehls wird nachgefragt, ob die Diskette wirklich kopiert werden soll. Nach Bestätigung befindet sich nach Abschluß des Vorgangs ein identisches Abbild der Diskette im Ziel-Laufwerk. Die Zieldiskette trägt zur Unterscheidung den Namen »COPY OF xxx«, wobei »xxx« der Name der Ursprungsdiskette ist. Soll die Zieldiskette einen anderen Namen erhalten, kann auch noch der Formalparameter »NAME« am Schluß der Befehlszeile angehängt werden. Dies sieht dann so aus:

```
DISKCOPY df0: TO df1:
NAME "Neuer Name"
```

Die Diskette erhält so gleich beim Kopieren einen neuen Namen.

Es kann auch mit nur einem Laufwerk dupliziert werden. Wenn Sie

```
DISKCOPY df0: TO df0:
```

eingeben, werden Sie (wenn es an der Zeit ist) vom Programm aufgefordert, die jeweils benötigte Diskette (Quelle oder Ziel) einzulegen.

Der letzte Befehl, den wir in diesem Teil besprechen, dient zur Abwechslung nicht der Datei- oder Diskettenmanipulation, sondern »nur« zum Betrachten von Dateien, geeigneterweise ASCII-Dateien, also beispielsweise Textfiles. Es ist dies das Kommando »Type«,

das die Inhalte angegebener Dateien auf dem Bildschirm ausgibt. »Type« kann auch wieder in mehreren Variationen verwendet werden. Die einfachste Form ist die Ausgabe einer Datei auf den Bildschirm. Nehmen wir wieder die Datei »Startup-Sequence« im Ordner »S«, von der wir ja schon wissen, daß sie lesbaren Text im ASCII-Format enthält:

```
TYPE S/startup-sequence
```

Der Inhalt des Files wird auf dem Bildschirm angezeigt. Sollte der Text aus dem CLI-Fenster scrollen, hält ein Druck auf die Leertaste die Ausgabe an. <RETURN> oder <BACKSPACE> (der Linkspfeil rechts oben auf der Tastatur) setzt die Anzeige fort. <CTRL C> bricht die ganze Ausgabe ab.

Sie können aber auch die Ausgabe einer Datei auf ein anderes Gerät lenken. Der nächste Befehl gibt beispielsweise die Datei auf einen angeschlossenen Drucker aus:

```
TYPE S/startup-sequence
TO prt:
```

Auch beim Type-Kommando lassen sich Zusatzparameter mitangeben. Der Parameter »OPT N«, am Schluß der Befehlszeile eingegeben, bewirkt, daß vor jede ausgegebene Zeile eine Zeilennummer gesetzt wird (nur für die Ausgabe gültig). Das obige Beispiel, diesmal mit Zeilennummern:

```
TYPE S/startup-Sequence
TO prt: OPT N
```

Der Parameter »OPT H« hat zur Folge, daß die betreffende Datei nicht im Klartext, sondern als Hex-Dump (Ausgabe von Hexadezimal-Zahlen) angezeigt wird. Dies dürfte aber nur für Programmierer von Interesse sein, deshalb soll es hier nur erwähnt werden.

Mit diesem letzten, leicht verständlichen Befehl schließen wir diesen Teil des Kurses wieder ab. Bedenken Sie, wenn Sie die Befehle ausprobieren, daß Sie nur mit Kopien arbeiten. Schnell kann eine Datei oder Diskette unrettbar zerstört werden!

Übrigens: Auch bei diesen Befehlen hilft ein Fragezeichen, das nach dem Befehl und einem Leerzeichen eingegeben wird, meist weiter. Die CLI-Kommandos zeigen dann kurz die Schreibweise an, die sie erwarten.

Gutes Gelingen bis zum nächstenmal. (dm)

Unterstützende Literatur: Das Amiga-Handbuch, Markus Breuer, Markt&Technik-Verlag, 49 Mark