

CLI – die zweite Haut des Amiga

Die letzte Folge behandelte Befehle, die sich zum Einbau in eine Batch-Datei (Kommandodatei, Stapeldatei) eignen. Batch-Dateien sind kleine Programme, die mehrere CLI-Befehle enthalten können. Beim Aufruf dieser Datei werden die einzelnen Befehle der Reihe nach abgearbeitet. Dieses Thema wird nun weitergeführt mit einem Beispiel, wie eine Stapeldatei aussehen könnte.

Um eine flexible Ausführungsdatei programmieren zu können, stellt das CLI verschiedene Befehle zur Verfügung wie etwa IF.THEN..ELSE..ENDIF-Konstruktionen, die im letzten Teil besprochen wurden. Es existiert noch eine zweite Befehlskombination, die in etwa mit dem von Basic her bekannten Befehl GOTO gleichzusetzen ist. Es sind dies die Befehle SKIP und LAB. LAB dient dazu, eine Sprungmarke zu fixieren, zu der mit SKIP gesprungen werden kann. Ein Beispiel:

```
BINDDRIVERS
LOADWB
...
...
SKIP Marke1
...
...
LAB Marke1
...
```

Alle Befehle, die auf die LAB-Zeile folgen, werden nach dem Sprung zu der definierten Marke weiterbearbeitet. Es müssen nur noch die Bedingungen definiert werden, bei denen ein Sprung erfolgen soll (siehe Schema in Bild 1). Eine Möglichkeit dafür ist eine IF-Abfrage:

```
...
...
IF EXISTS Preferences
THEN setze Abarbeitung
fort
ELSE SKIP Marke1
ENDIF
...
...
ENDCLI > NIL:
LAB Marke1
ECHO "Datei nicht vorhanden"
ENDCLI > NIL:
```

Batch-Dateien erlauben dem Anwender, durch einen einzigen Aufruf viele Befehle der Reihe nach abzuarbeiten. In diesem Kursteil erfahren Sie, was alles mit einer Batch-Datei gemacht werden kann.

Bei unserem Beispiel wird zuerst geprüft, ob die Datei »Preferences« auf dem aktuellen Laufwerk vorhanden ist. Wenn ja, arbeitet der Amiga die Batch-Datei weiter ab. Ist die Datei aber nicht vorhanden, springt das Betriebssystem mit SKIP zu der mit LAB definierten Marke und führt die Befehle aus, die der LAB-Zeile folgen. Anschließend wird das CLI-Fenster geschlossen. Diese Kombination kann also eingesetzt werden, um mehr Struktur in eine Stapeldatei zu bringen. Allerdings ist es wohl in den seltensten Fällen nötig, diese Kombination zu verwenden, da sich ein solches Problem auch anders lösen läßt.

Ein weiterer, allerdings selten gebrauchter Befehl ist QUIT. Dieser Befehl dient zum Beenden von Stapeldateien und Kommandofolgen und kann beispielsweise anstelle von ENDCLI stehen. Im Unterschied zu ENDCLI wird bei QUIT das CLI-Fenster nicht geschlossen. Verändern wir doch das obige Beispiel, indem wir QUIT einbauen:

```
...
...
IF EXISTS Preferences
THEN setze Abarbeitung
fort
ELSE SKIP Marke1
ENDIF
...
...
QUIT
LAB Marke1
ECHO "Datei nicht vorhanden"
QUIT
```

Die Datei tut nichts anderes als oben beschrieben. Einziger Unterschied ist, daß nach der Abarbeitung das CLI-Fenster nicht geschlossen wird, was bei ENDCLI der Fall wäre.

Quit kann aber noch mehr: Wenn hinter dem Befehl ein

Parameter angegeben wird, setzt Amiga-DOS diesen Wert nach Beendigung des Befehls automatisch als neues Fehler-niveau fest:

QUIT 40

Return code

Nach Beendigung des QUIT-Befehls würde das Fehler-niveau also den Wert 40 aufweisen. Der Befehl, der wie auch SKIP und LAB nur in Stapeldateien verwendet werden kann, ist sicher nicht dafür geeignet, dauernd eingesetzt zu werden. In speziellen Situationen kann er aber eine nützliche Anwendung finden.

Batchen mit Komfort

Mit Batch-Dateien läßt sich also eine Menge anfangen. Um aber ganz flexibel damit zu arbeiten, kann man beim Aufruf der Stapeldatei gleich einen Parameter übergeben, mit dem in der Befehlsdatei gearbeitet wird. Das könnte beispielsweise der Name eines Programms sein. Für das folgende Beispiel nehmen wir an, daß eine Befehlsfolge programmiert werden soll, die eine beliebige Datei mit einer bestimmten Folge von Befehlen »bearbeitet«. Wir möchten, daß im Beispiel die (beliebige) Datei zuerst einmal ins RAM kopiert wird, von dort auf Bildschirm und Drucker ausgegeben und auf eine in »df1« eingelegte Diskette kopiert wird. Dieser Vorgang ergibt zwar keinen rechten Sinn, doch dient er hervorragend zur Demonstration über den Aufbau einer flexiblen Stapeldatei.

Wir möchten erreichen, daß nur einmalig beim Aufruf der Batch-Datei der Name des Programms übergeben wird, mit dem wir den Vorgang durchführen. Da in der Befehlsdatei der Name des Programms sicher öfter verlangt wird, ist es das einfachste, einen Platzhalter dafür einzusetzen. Genau dies erreicht man mit der Definition »KEY Name«.

Bei Angabe dieser Definition am Beginn der Datei wird dem Amiga mitgeteilt, daß hinter dem Namen der Batch-Datei noch ein weiterer Parameter folgt. Hinter dem Namen kann, durch einen Schrägstrich ge-

Teil 5

KURSÜBERSICHT

Der CLI-Kurs erläutert Ihnen in mehreren Folgen den Umgang mit der Amiga-Benutzerschnittstelle CLI (Command-Line-Interface). Am Ende haben Sie alle wichtigen Befehle kennengelernt und können die Kommandos nach eigenen Wünschen einsetzen. Die folgende Kursübersicht zeigt Ihnen, in welchen Kursteilen welche CLI-Kommandos erklärt werden:

TEIL 1 behandelte Befehle zum Anzeigen des Directories und Stellen der Uhr (DIR, LIST, CD, DATE, ENDCLI)

TEIL 2 erklärte Ihnen, wie Sie Dateien kopieren, löschen, umbenennen und Disketten formatieren können (FILENOTE, DELETE, RELABEL, RE-NAME, TYPE, COPY, DISKCOPY, INFO, FORMAT, INSTALL)

TEIL 3 zeigte unter anderem auf, wie Texte und Startdateien erstellt und bearbeitet werden (MAKEDIR, ED, PROMPT, WAIT, LOADWB, SETCLOCK, EXECUTE)

TEIL 4 befaßte sich mit Befehlen, die sich zum Einbau in eine Batch-Datei anbieten (AS-SIGN, ECHO, FAILAT, IF, THEN, ELSE, ENDIF, STACK, PROTECT, BINDDRIVERS)

TEIL 5 setzt die Erklärung von Kommandos für eine Batch-Datei fort (RUN, EDIT, JOIN, LAB, QUIT, SKIP, ADD-MEM, VERSION)

TEIL 6 geht auf neue Spezialbefehle für Zusatzgeräte wie etwa Festplatten ein (SEARCH, NEWCLI, SORT, WHY, DJMOUNT, PREP, DPFORMAT, MOUNT, ADDBUFFERS, DISKCHANGE)

TEIL 7 schließt den CLI-Kurs mit den letzten Workbench-Befehlen ab (CHANGETASKPRI, DISKDOCTOR, FAULT, SETDATE, SETPREFS, STATUS)

KURSE

trennt, stehen, ob der Parameter nötig ist (/A). Ein Beispiel:

```
.key name/a
...
LOADWB
```

Nehmen wir an, der Name der Batch-Datei wäre »Test« und der des zu bearbeitenden Programms »xyz«. Über die Tastatur müssen Sie zum Aufruf dieser Stapeldatei

```
EXECUTE Test xyz
```

eingeben. Ab jetzt würde überall in der Batch-Datei, wo <Name> steht, der Platzhalter »xyz« eingesetzt. Für das obige Beispiel sieht die zu erstellende Stapeldatei so aus:

```
.key name/a
COPY <NAME> TO ram:
TYPE ram:<Name>
COPY ram:<Name> TO prt:
COPY ram:<Name> TO df1:
```

Anstelle von <Name> findet sich bei der Abarbeitung der mit dem Aufruf übergebene Name der zu bearbeitenden Datei. Es kann aber vorkommen, daß eine Befehlsfolge programmiert werden soll, bei der nicht unbedingt ein Name übergeben werden soll. Wir wollen annehmen, daß in diesem Fall ein vorher bestimmter Platzhalter eingesetzt wird. Die Definition »DEF Name Name2« bestimmt, daß, wenn kein Name beim Aufruf übergeben wird, ein definierter Platzhalter Verwendung findet. In diesem Fall muß das »/A« hinter .KEY entfallen, damit nicht zwingend ein Übergabeparameter angefordert wird:

```
.key name
.def name xxx
COPY <NAME> TO ram:
TYPE ram:<Name>
COPY ram:<Name> TO prt:
COPY ram:<Name> TO df1:
```

Gibt der Anwender jetzt beim Aufruf mit EXECUTE keinen Namen mehr an, setzt der Amiga anstelle von <Name> die mit .DEF bestimmte Zeichenkette »xxx« ein.

Das RAM wird aktiv

Speziell für den Amiga 1000 sind einige RAM-Erweiterungen zum internen oder externen Anschluß erhältlich, die teilweise den Nachteil aufweisen, daß sie nicht autokonfigurierend sind. Das bedeutet, daß sie vom Betriebssystem beim Initialisieren nicht automatisch erkannt werden. Um diese Speichererweiterungen dem Amiga doch noch zu-

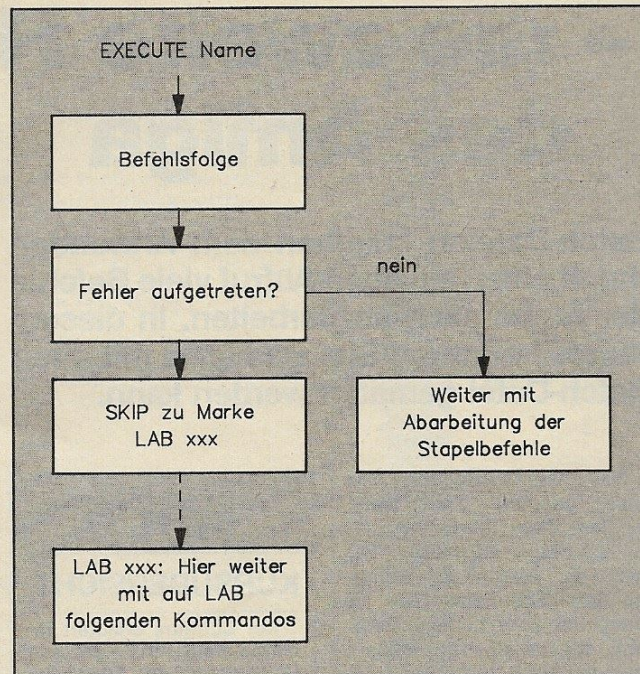


Bild. Die SKIP/LAB-Kombination funktioniert in etwa wie der von Basic her gewohnte Sprungbefehl GOTO

gänglich zu machen, gibt es den sowohl im Direkt- wie auch im Programmmodus einsetzbaren Befehl ADDMEM.

Er teilt dem Amiga mit, daß neuer Speicher einzubinden ist und an welcher Speicheradresse er beginnen soll. Dazu muß die Startadresse des neuen Speicherbereichs, also wo man ihn gerne haben will, und die Länge des RAM-Bereichs +1 angegeben werden. Zum Beispiel könnte der Aufruf für 1 MByte mehr Speicher, beginnend ab \$200000, so lauten:

```
ADDMEM 200000 300000
```

Nach Bestätigung des Befehls mit <RETURN> steht dem Anwender der neue RAM-Bereich sofort zur Verfügung.

Von Zeit zu Zeit kann es nötig werden, daß mehrere Einzeldateien (etwa Textfiles) zu einem einzigen File verbunden werden müssen. Damit die Sache nicht zu umständlich wird, stellt das CLI den Befehl JOIN zur Verfügung. Er erlaubt, maximal 15 Einzeldateien zu einer neuen Gesamtdatei »zusammenzuschweißen«. Die einzige Bedingung beim Aufruf ist, daß die neue Datei einen anderen Namen als eine der Einzeldateien erhalten muß. Die zu verbindenen Dateien müssen in der Reihenfolge, wie sie verbunden werden sollen, dem Befehl folgen:

```
JOIN Text1 Text2 AS Text3
```

Die Dateien »Text1« und »Text2« werden zu der neuen Datei »Text3« vereint.

der sowohl im Direkt- wie auch im Programmmodus eingegeben werden kann und das Multitaskingsystem nicht an seiner Entfaltung hindert.

Aber RUN kann noch mehr: Mit einem einzigen Aufruf lassen sich auch mehrere Prozesse starten. Dazu muß nur zwischen jedem neuen Befehl ein »+« gesetzt werden.

```
RUN COPY df0:Disk.info
TO df0:Test.info+
DELETE df0:Test.info+
ECHO "Fertig!"
```

Während der Anwender nach dem Aufruf ungestört mit dem CLI weiterarbeiten kann, kopiert das Betriebssystem die Datei »Disk.info« in das File »Test.info« und löscht die Kopie anschließend. Erst durch die Meldung »Fertig!« bekommt der Benutzer mit, daß der Vorgang abgeschlossen ist.

Ein selten benutzter Befehl des CLI ist EDIT. EDIT ist ein wenig komfortabler Zeileneditor, der nur gestattet, vorhandene, sequentiell aufgebaute Dateien zu bearbeiten. Es ist nicht möglich, von EDIT aus neue Dateien zu erzeugen. Da er außerdem nur mit Binärdateien etwas anfangen kann und auch sonst relativ selten eingesetzt wird, gehen wir in diesem Kurs nicht näher darauf ein. Er wird nur der Vollständigkeit halber erwähnt. Hauptsächlich gedacht ist dieses zeilenorientiert arbeitende Programm wohl für Programmierer, die beispielsweise Maschinenprogramme oder ähnliches editieren will. Wer genaueres über diesen Befehl erfahren möchte, sei auf das Amiga-DOS-Handbuch verwiesen, das dem Amiga beiliegt.

Der letzte CLI-Befehl, der in diesem Kursteil besprochen wird, ist VERSION. Der Befehl ist ohne Parameter anzugeben. Er zieht die Versionsnummern der Kickstart- und der Workbench an. So können Sie jederzeit erfahren, ob Sie die aktuellen Versionen besitzen (derzeit Kickstart 33.180/Workbench 33.60).

Sie sollten nun über genügend Wissen verfügen, um eigene Stapeldateien zu programmieren. Bis wir im nächsten Teil des Kurses auf neue Befehle für Zusatzgeräte (etwa Festplatten) eingehen, können Sie zur Übung mit den einzelnen Befehlen experimentieren und eigene Batch-Dateien erzeugen. (dm)

Einer der wohl am häufigsten gebrauchten Befehle im CLI ist RUN. Dieses äußerst nützliche Programm erlaubt es, vom CLI aus ein Programm (einen Task/Prozeß) zu starten, das abläuft, ohne das CLI zu blockieren. Um dies verständlicher zu machen, geben Sie bitte folgendes ein:

```
Preferences <RETURN>
```

Sobald der Preferences-Screen erschienen ist, legen Sie ihn mittels des Hintergrundschalters oben rechts nach hinten, so daß das CLI wieder sichtbar ist. Versuchen Sie, im CLI irgend einen Befehl einzugeben. Sie werden feststellen, daß das CLI keine Eingaben mehr verarbeitet. Holen Sie Preferences wieder in den Vordergrund und beenden das Programm mit dem Schließschalter links oben.

Nun probieren wir folgenden Befehl aus:

```
RUN Preferences
<RETURN>
```

Wiederholen Sie die Prozedur und geben einen Befehl im CLI ein. Sie sehen, das eben gestartete Preferences-Programm arbeitet im Hintergrund, während das CLI weiterhin Eingaben annimmt und verarbeitet. Der Befehl RUN startet also vom CLI aus ein Programm und gibt die Kontrolle anschließend zurück, so daß das Command Line Interface nicht blockiert ist. RUN ist ein äußerst nützlicher Befehl,

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