

CLI – Die zweite Haut des Amiga

Batch-Dateien oder auch Stapeldateien sind kleine Programme, die verschiedene Befehle enthalten, die der Reihe nach abgearbeitet werden. Da diese Batchdateien sehr einfach mit Hilfe eines normalen Editors zu erstellen sind, bereitet ihre Programmierung auch dem Einsteiger wenig Probleme. Stapeldateien werden wohl hauptsächlich dazu eingesetzt, um während des Computerstarts benötigte Programme automatisch in den Speicher laden zu lassen. Bestes Beispiel dazu ist beim Amiga die »Startup-Sequence« im Unterverzeichnis »s«. Das Amiga-Betriebssystem sieht bei jedem Neustart des Computers auf der Diskette nach, ob das »s«-Verzeichnis vorhanden ist und ob sich in diesem Directory die Startdatei befindet. Ist sie vorhanden, wird sie geladen und die einzelnen Befehle, die in diesem Programm enthalten sind, der Reihe nach abgearbeitet. Wie dies in etwa funktioniert, zeigt Ihnen Bild 1.

Dieser Teil des Kurses beschäftigt sich hauptsächlich mit CLI-Befehlen, die sich hervorragend zum Einbau in eine Batch-Datei eignen, auch wenn die meisten dieser Befehle genauso gut im Direktmodus eingegeben werden können. Wie Batch-Dateien mit Hilfe des CLI-Editors ED erstellt und bearbeitet werden können, entnehmen Sie bitte dem dritten Teil dieses Kurses in Ausgabe 10/87.

Beginnen wir zuerst mit einem der einfachsten Befehle, mit ECHO, das der Ausgabe eines Textes dient. ECHO ist eigentlich nur sinnvoll, wenn es in eine Batch-Datei eingebunden wird (Wer will schon im Direktmodus einen Text ausgeben?). Ein Beispiel für einen Einsatz von ECHO: Sie möchten gerne aktuell darüber informiert sein, was der Amiga gerade in der Startdatei bearbeitet und wünschen eine Ausgabe eines betreffenden Textes im CLI-Fenster. Dazu wird einfach der ECHO-Befehl vor jeden Befehl in der Batch-Datei gesetzt. Erstellen wir uns gleich eine Stapeldatei. Geben Sie dazu bitte folgendes ein:

```
ED RAM:Test
```

Wie erstellt man Batch-Dateien, welche Befehle eignen sich zum Einbinden in eine Stapeldatei? In dieser Folge erfahren Sie, wie Sie beispielsweise die Startdatei »Startup-Sequence« nach Ihren Wünschen verändern können.

Damit erstellen Sie im RAM eine kleine Stapeldatei, die später mit

```
EXECUTE RAM:Test
```

ausgeführt wird. Da wir nur den ECHO-Befehl kennenlernen wollen, genügt eine kleine Befehlsfolge. Zu beachten ist bei ECHO nur, daß der auszugebende Text in Anführungszeichen stehen und vom Rest durch ein Leerzeichen getrennt sein muß. Tippen Sie bitte folgende Zeilen in ED ein und beenden den Editor anschließend mit <ESC X> <RETURN>:

```
ECHO "Lese Directory"
ECHO ""
DIR df0:
ECHO "Erstelle Info"
ECHO ""
INFO
ECHO "Lade Workbench"
ECHO ""
LOADWB
```

Diese kleine Batch-Datei tut folgendes: Die ersten beiden Zeilen geben auf dem Bildschirm den Text »Lese Directory« und eine Leerzeile aus. Nach Ausgabe dieser Meldung liest der Amiga das Inhaltsverzeichnis der in das interne Laufwerk (df0) eingelegten Diskette und zeigt es am Bildschirm an. Als nächstes gibt der Computer wieder einen Text aus: »Erstelle Info«, gefolgt von einer Leerzeile. Der darauf folgende Befehl INFO informiert Sie über die angeschlossenen Laufwerke und Geräte. Nach der dritten Textausgabe »Lade Workbench« wird noch einmal die Workbench geladen und die Diskettensymbole »erneuert«. Der ECHO-Befehl kann jederzeit in eine Batch-Datei eingebunden werden, um beispielsweise aktuelle Tätigkeiten oder benutzereigene Meldungen (etwa »Hallo User...«) anzuzeigen.

Ebenfalls zum Einbinden in eine Batch-Datei ist FAILAT gedacht. FAILAT dient dem Anheben des Fehlerniveaus beim Abarbeiten einer Stapeldatei. Erläutern wir dies genauer:

Jedesmal, wenn bei der Ausführung eines Stapeldatei-Befehls ein Fehler auftritt, liefert Amiga-DOS je nach Schwere der Fehlfunktion einen Wert zurück. Je nach Höhe des eingestellten Fehlerniveaus wird entweder auf den Fehler reagiert, oder er wird ignoriert. Standardeinstellung bei der Ausführung einer Batch-Datei ist der Wert 10. Dies bedeutet, daß die weitere Bearbeitung der Stapeldatei unterbrochen wird, wenn ein Fehler mit einem größeren Wert als 10 gemeldet wird. So gibt es beispielsweise leichte Fehler mit dem Wert 5, bei denen vom Amiga-DOS Warnungen ausgegeben werden. Will man aber nicht, daß die Weiterführung, aus welchen Fehlergründen auch immer, abgebrochen wird, kann man das Fehlerniveau erhöhen. So sichert etwa

```
FAILAT 30
```

oder ein höher angesetztes Fehlerniveau, daß die Abarbeitung der Batch-Datei nicht gestoppt wird. Nach Beendigung der Befehlsausführung der Stapeldatei wird der Wert von Amiga-DOS automatisch wieder auf 10 gesetzt. Das Abbruchniveau ist also nur solange gültig, wie die Stapeldatei abgearbeitet wird.

Ein weiterer Befehl, der aber nur für Besitzer von PC-Erweiterungen (etwa der Janus-PC/AT-Karte) Bedeutung besitzt, ist BINDDRIVERS. Dieser Befehl läßt sich sowohl im Direktmodus wie auch in Stapeldateien verwenden.

BINDDRIVERS, das ohne zusätzliche Parameter eingegeben wird, ermöglicht erst die Einbindung des PC-Systems

in das Amigasystem. Es ist quasi das Kommando, den PC/AT zum Leben zu erwecken und seine Tätigkeit als Task in das Amigasystem einzuflechten. Dieser Befehl sollte, wenn er benötigt wird, immer am Anfang der auszuführenden Batch-Datei stehen (nach FAILAT, wenn vorhanden).

Ebenso leicht zu erklären ist die Anweisung STACK. Jedem CLI-Befehl wird vor seiner Ausführung ein bestimmter Spei-

Teil 4

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 (FILENOM, 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 befaßt sich mit Befehlen, die sich zum Einbau in eine Batch-Datei anbieten (AS-SIGN, ECHO, FAILAT, IF, THEN, ELSE, ENDF, STACK, PROTECT, BINDDRIVERS)

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

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

cherbereich (Stapelspeicher) reserviert, den er, falls zur Befehlsausführung nötig, verwenden kann. Im Normalfall sind dies 4000 Byte. Es gibt aber einige CLI-Befehle, die mehr Platz für die Ablage von Daten benötigen (etwa DIR oder SORT). Um diesen Befehlen mehr Stapelspeicher zu gewähren, dient STACK. Will man die Größe des aktuellen Stapelspeichers erfahren, ist STACK ohne Parameter einzugeben. Soll dagegen der Stapel verändert werden, muß man, durch ein Leerzeichen getrennt, die Größe des neuen Bereichs angeben (diese Größe sollte aber mindestens 4000 Byte betragen). Ein Beispiel:

STACK 8000

Nach Eingabe dieses Kommandos stehen den CLI-Befehlen nun 8000 Byte Stapelspeicher zur Verfügung.

Gegen Diskettenwechsel

Gerade für Besitzer von nur einem Laufwerk oder einer zusätzlichen Festplatte ist folgender Befehl geeignet: ASSIGN. Doch um den Befehl zu verstehen, ist erst eine Erklärung über einen Teil des Amiga-Betriebssystems nötig.

Für das Amiga-DOS existieren verschiedene Standardverzeichnisse (Directories), in denen nach bestimmten Dateien gesucht wird. So sucht das Betriebssystem beispielsweise jeden CLI-Befehl im »c«-Directory und Zeichensätze im »fonts«-Ordner. Beim Start mit einer Workbench-Diskette sind diese Verzeichnisse, wenn nicht anders angegeben, auf die Startdiskette fixiert. Dies bedeutet: auch wenn Sie die Startdiskette entnehmen und eine andere Diskette einlegen, auf der die gleichen Dateien enthalten sind, wird der Amiga, wenn man beispielsweise einen CLI-Befehl eingibt, immer das Einlegen der Startdiskette verlangen. Das bedeutet, daß er es nicht registriert, wenn auch auf der eingelegten Diskette der gleiche Befehl im »c«-Directory vorhanden ist. Geben Sie bitte zum weiteren Verständnis folgendes ein:

ASSIGN

Es wird eine Liste ausgegeben, anhand der man erkennen kann, welche Ordner (Verzeichnisse) auf welches Gerät fixiert sind. Im Normalfall sind alle Verzeichnisse auf die Startdiskette festgelegt!

Besitzer von nur einem Laufwerk kennen sicher das Problem: Sie möchten etwa von einer anderen Diskette das Directory lesen, aber der Amiga verlangt immer wieder die Startdiskette und nichts klappt. Abhilfe schafft nun folgendes:

Sie kopieren die Befehle, die öfter benötigt werden (beispielsweise DIR, LIST, INFO) ins RAM und lassen die Befehle von dort aus ausführen. Bei allen Befehlen, die sich dann im RAM befinden, wird der Amiga nun nicht mehr das Einlegen der Startdiskette verlangen. Alles, was zum Gelingen dieser Prozedur nötig ist, ist außer dem Vorhandensein der Befehle im RAM die Mitteilung an den Amiga, wo diese Befehle nun zu finden sind. Dies macht ASSIGN. Probieren wir dies nun im kleinen Rahmen durch. Geben Sie bitte die folgenden Befehle ein, die natürlich auch in einer Batch-Datei stehen können:

```
MAKEDIR RAM:c
COPY c/dir RAM:c
COPY c/list RAM:c
COPY c/info RAM:c
COPY c/assign RAM:c
```

Nun befinden sich die Befehle DIR, LIST, INFO und ASSIGN im RAM, und zwar im zuvor mit MAKEDIR erzeugten »c«-Verzeichnis. Doch der Amiga greift noch immer auf die Startdiskette zu, da wir ihm noch nicht mitgeteilt haben, daß die Befehle jetzt im RAM stehen. Wenn Sie als Abschluß

```
ASSIGN c: RAM:c
```

eingeben, werden die CLI-Befehle von jetzt ab aus dem RAM geholt (natürlich nur diejenigen, die ins RAM kopiert wurden). Es empfiehlt sich, immer ASSIGN in das neue Verzeichnis mitzukopieren, da mit

```
ASSIGN c: df0:c
```

wieder auf ein anderes Gerät fixiert werden kann. Und das ist leider nicht mehr so einfach möglich, wenn sich ASSIGN nicht im aktuellen Ordner befindet.

Ebenso können natürlich auch alle anderen Verzeichnisse wie etwa »devs« oder »libs« neu fixiert werden. Wieviele Dateien Sie auf andere Geräte festlegen, ist nur durch den Speicherbereich der betreffenden Einheiten eingeschränkt.

Für Festplattenbesitzer ist der ASSIGN-Befehl natürlich noch interessanter. Sie können sämtliche Standard-Verzeichnisse (etwa devs, l, libs, fonts und c) auf ihre Festplatte installieren und in die Startdatei

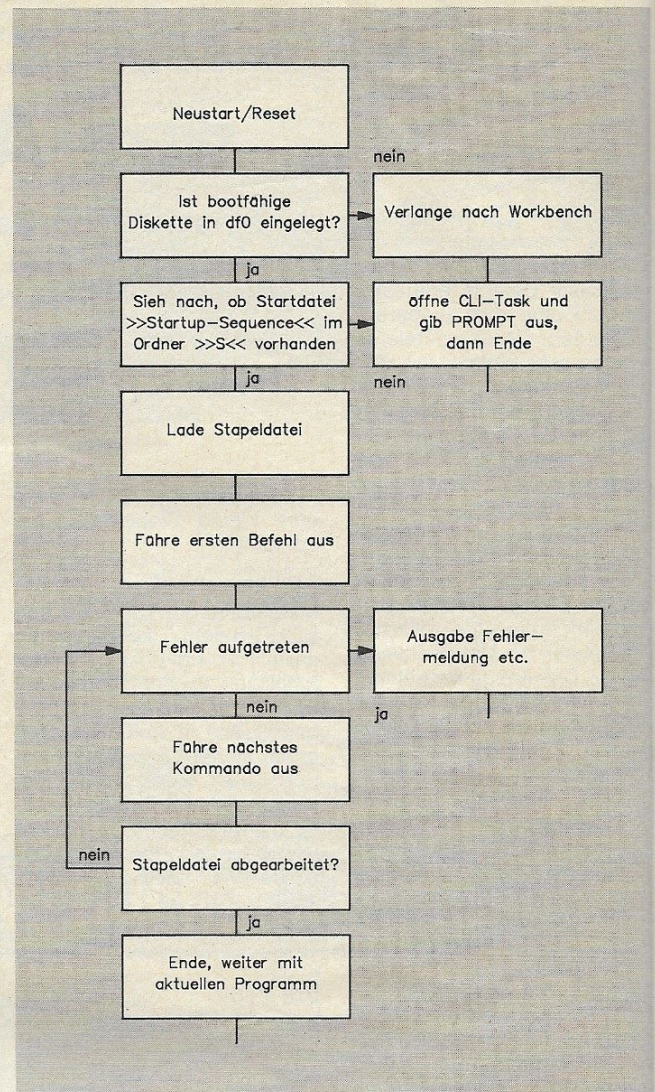


Bild 1. Die Startdatei »Startup-Sequence« wird vom Amiga-DOS beim Start in dieser Art und Weise behandelt

der Workbench-Diskette einige Zeilen einbauen, die sämtliche Ordner auf die Festplatte fixieren. Danach wird (sofern sich der betreffende Befehl wirklich im »c«-Ordner der Platte befindet) nicht mehr das Einlegen der Startdiskette verlangt, was häufigen Diskettenwechsel erspart. Folgendes, in die Startdatei eingebundene Beispiel würde einige Standardverzeichnisse (soviel Sie wollen) auf die Festplatte »dh0« festlegen (für Amiga 2000-Besitzer mit PC-Festplatte wäre dies das Laufwerk »jh0«):

```
...
ASSIGN c: dh0:c
ASSIGN devs: dh0:devs
ASSIGN l: dh0:l
ASSIGN libs: dh0:libs
ASSIGN fonts: dh0:fonts
ASSIGN sys: dh0:sys
...
```

Wichtig bei ASSIGN ist die

richtige Syntax, also die Schreibweise. Es muß immer der Quellordner zuerst angegeben werden. Danach folgt das Zielgerät und ohne Leerzeichen dazwischen der Zielordner.

Schleifenprogrammierung mit dem CLI

Was nun folgt, ist ein umfangreiches und äußerst nützliches Werkzeug des CLI, mit dem man Batch-Dateien erstellen kann, die abhängig von Entscheidungen verschiedene Dinge erledigen können. Das Werkzeug ist die Schleifenprogrammierung, die man mit den Befehlen IF, THEN, ELSE und ENDIF verwirklichen kann. Diese vier Befehle lassen sich nur in Batch-Dateien verwenden.

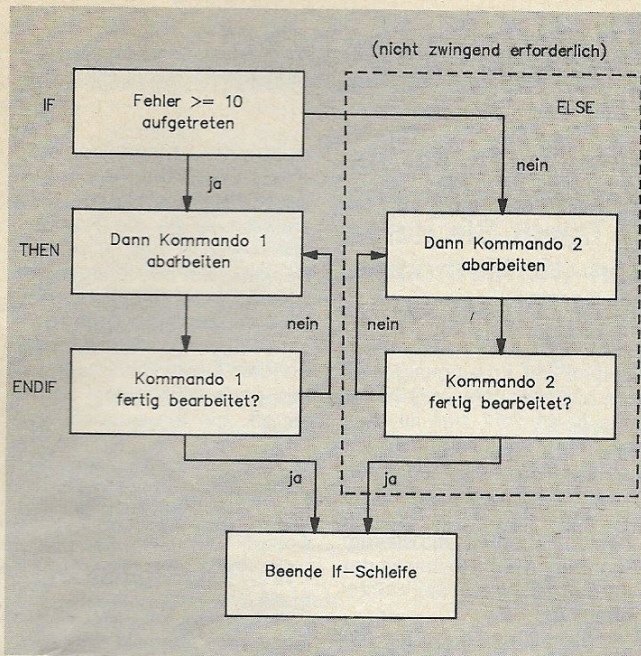


Bild 2. Wenn in der Batch-Datei ein Fehler ≥ 10 aufgetreten ist, wird Kommando 1 abgearbeitet (IF-THEN)

den, im Direktmodus führen sie zu keinem Ergebnis.

IF prüft auf ein Ereignis. Ist dieses wahr, so wird eine vorher bestimmte Befehlsfolge ausgeführt, die im Regelfall hinter THEN steht. THEN muß nicht unbedingt angegeben werden. Fehlt THEN, werden die folgenden Anweisungen so lange ausgeführt, bis das Betriebssystem auf die Befehle ELSE oder ENDIF trifft.

Stimmt das Ergebnis nicht, erfolgt die Ausführung einer anderen Befehlsfolge, die hinter ELSE beginnt und durch ENDIF abgeschlossen wird.

Um nun die einzelnen Ereignisse genauer zu spezifizieren, werden hinter IF verschiedene Bedingungen angegeben. Drei Schlüsselworte stehen für die Abfrage des aufgetretenen Fehlerwertes (siehe FAILAT). Diese drei Bedingungen sind WARN für einen Fehlerwert ≥ 5 , ERROR bei einem Fehlerlevel ≥ 10 und FAIL, wenn ein Wert ≥ 20 zurückgemeldet wurde (dies funktioniert aber nur, wenn der Fehlerlevel mit FAILAT auf die gewünschten Gegebenheiten angepaßt wurde). Ein Beispiel (siehe dazu auch Bild 2):

```

IF Error THEN (Kommando 1)
ELSE (Kommando 2)
ENDIF
oder
IF Error THEN (Kommando 1)
ENDIF
  
```

Zur Erklärung des Beispiels: Es wird geprüft, ob ein Fehler mit dem Wert ≥ 10 aufgetreten ist. Wenn ja, führt der Amiga »(Kommando 1)« aus. Ist dagegen die Bedingung nicht erfüllt worden, fährt der Amiga mit der normalen Abarbeitung der Befehlsfolge fort. Wenn es Sie wundert, daß zwei verschiedene Möglichkeiten zur Auswahl stehen, so können wir Sie beruhigen. Im obigen Fall kann die ELSE-Zeile auch weggelassen werden, wenn nicht zwingend eine Reaktion bei nicht erfüllter Bedingung gewünscht wird. Falls in diesem Fall normal weitergemacht werden soll, entfällt die ELSE-Zeile.

Es gibt aber noch weitere Bedingungen: Eine davon ist EXISTS. Bei Angabe dieses Schlüsselworts prüft der Amiga, ob auf einer Diskette eine bestimmte Datei vorhanden ist. Ein weiteres Beispiel (siehe auch Bild 3):

```

IF EXISTS Preferences
THEN (Kommando 1)
ELSE (Kommando 2)
ENDIF
  
```

Ist die Datei Preferences auf der aktuellen Diskette vorhanden, wird »(Kommando 1)« ausgeführt. Wenn nicht, arbeitet der Amiga »(Kommando 2)« ab. Auch hier gilt: Wenn nicht zwingend eine Reaktion auf ein Nichtvorhandensein von »Preferences« nötig ist, kann die ELSE-Zeile entfallen. Die dritte Bedingung ist EQ. Bei Angabe

dieser Bedingung prüft Amiga-DOS, ob zwei Dateien gleich sind (Bild 4):

```

IF (Datei 1) EQ (Datei 2)
THEN (Kommando 1)
ELSE (Kommando 2)
ENDIF
  
```

Stimmen die Dateien »(Datei 1)« und »(Datei 2)« überein, wird »Kommando 1« abgearbeitet. Ansonsten gilt das gleiche wie bereits gesagt.

Da aber möglicherweise auf eine Nichtübereinstimmung geprüft werden könnte, läßt sich auch ein Teil der Abfrage negieren, das heißt, umkehren. Die Bedingung dafür lautet NOT.

```

IF (Datei 1) NOT EQ
(Datei 2) THEN
(Kommando 1)
ELSE (Kommando 2)
ENDIF
  
```

(Kommando 2) würde in diesem Fall nur ausgeführt werden, wenn die Dateien gleich sind (Bild 4).

Die IF-THEN-ELSE-ENDIF-Konstruktionen können auch ineinander verschachtelt werden, was eine komplexe Batch-

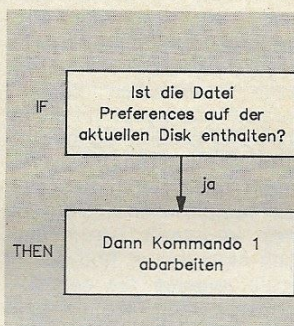


Bild 3. Um festzustellen, ob eine Datei auf der aktuellen Diskette enthalten ist, dient die IF-Bedingung EXISTS (den Rest des Bildes entnehmen Sie bitte Bild 2)

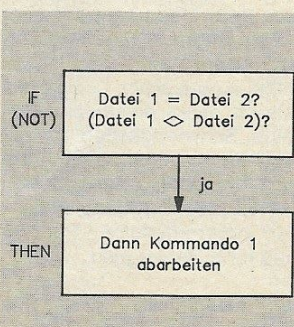


Bild 4. Durch eine gezielte IF-Abfrage läßt sich auch feststellen, ob zwei Dateien den gleichen Inhalt besitzen (den Rest des Bildes entnehmen Sie bitte Bild 2)

Programmierung ermöglicht. Nach dieser etwas schwerer verdaulichen Befehlskombination zum Abschluß noch ein Kommando, welches leicht zu verstehen und sehr nützlich ist: PROTECT. Damit können Dateien gegen Schreiben und Löschen, Lesen und Aufrufen geschützt werden. Wenn Sie sich mit dem LIST-Befehl ein Directory ausgeben lassen, können Sie nach dem Datum vier verschiedene Buchstaben in jeder Zeile sehen, die den Status der Datei anzeigen. Diese vier Buchstaben (r-w-e-d) haben folgende Bedeutung:

- »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

Datenschutz eingebaut

Mit dem PROTECT-Befehl können nun gezielt die einzelnen Flags (Schalter) beeinflusst werden. Dabei ist die Schreibweise wie folgt:

```

PROTECT NAME Datei
FLAGS xxxx
  
```

Es werden nur die Flags gesetzt, die auch angegeben sind. Um also ein Flag der Datei zu löschen, müssen die verbleibenden drei Flags gesetzt werden, das vierte wird automatisch gelöscht:

```

PROTECT NAME df1:Test
FLAGS re
  
```

würde die Datei »Test« gegen Löschen und Überschreiben schützen, da nur die Flags »r« und »e« gesetzt werden.

Bei der PROTECT-Methode ist allerdings eines zu beachten. Wenn Sie mit »Snapshot« die Lage einzelner Icons in einem Window dauerhaft festhalten wollen, dürfen die »info«-Files nicht Schreibgeschützt sein! Dies liegt daran, daß in den Incon-Dateien die Lage des später auf dem Workbench-Screens erscheinenden Bildsymbols eingetragen wird.

Damit findet dieser Teil des Kurses ein Ende. Im nächsten Teil behandeln wir einige Befehle, die sich zum Einbau in eine Batch-Datei eignen und erklären an einem Beispiel, was alles zu einer Startdatei gehört. (dm)

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