BEDIENUNGSANLEITUNG

In diesem Abschnitt werden der Kommandointerpreter, der Kommandoeditor, der Unterschied zwischen internen und externen Kommandos und der Batchprozessor erklärt. Daneben werden noch einige wichtige Begriffe erläutert.

DATEINAMEN

Der Dateiname identifiziert eindeutig eine Datei. Pro Laufwerk kann ein und derselbe Name nur einmal auftauchen. Der Name besteht aus mehreren Komponenten:

```
d:nnnnnnnn.eee
```

Es bedeuten:

"d:" identifiziert das Laufwerk. Das Laufwerk wird durch genau ein Zeichen gefolgt vom Doppelpunkt bestimmt. Die Zuordnung zwischen dem Laufwerksbezeichner und dem tatsächlichen Laufwerk geschieht laut nachfolgender Tabelle.

Laufwerksbezeichner	Physikalisches Laufwerk
0	Ramdisk
1	Diskette 1
2	Diskette 2
3	Diskette 3
4	Diskette 4
A	Festplatte Partition A
B	Festplatte Partition B
.	
.	
.	
Z	Festplatte Partition Z

Die Angabe des Laufwerks im Dateinamen ist nicht zwingend. Liegt keine Angabe vor, dann wird das aktuelle Laufwerk angenommen. Der Bezeichner des aktuellen Laufwerks ist immer im Systemprompt des Kommandointerpreters enthalten, also vor dem ">".

"nnnnnnnn" identifiziert den eigentlichen Namen der Datei. Der Name besteht aus ein bis acht Zeichen. Da in allen Kommandos Klein- in Großbuchstaben umgewandelt werden, besteht kein Unterschied dazwischen. Sonderzeichen werden akzeptiert, sollten aber nicht verwendet werden. Das Zeichen "." wird nicht akzeptiert, da es ein Trennzeichen im Dateinamen ist. Die Zeichen "*" und "?" werden vom Kommandointerpreter als Jokerzeichen gewertet und sollten daher auf keinen Fall im Dateinamen auftauchen. Das Zeichen "@" hat eine Sonderbedeutung bei den sogenannten Antwortdateien, siehe Kommando "ass".

"eee" identifiziert den Dateityp. Dieser kann entfallen oder er besteht aus ein bis drei Zeichen. Hierbei gilt das für den Dateinamen gesagte. Einige Dateitypen werden von JADOS mit festen Bedeutungen belegt. Dies sind:

Dateityp	Bedeutung
68K	Programmdatei, die ab Ladeadresse läuft
BAT	Batchdatei
COM	Programmdatei, die überall lauffähig ist
OBJ	Objektdatei, reserviert für den Linker
SYS	Systemdatei. Z.B. JADOS.SYS, CONFIG.SYS

Die Dateitypen können benutzt werden, um Klassen von Dateien zusammenzufassen. Z.B. kann man TXT für Textdateien, ASM für Assemblerquelldateien, PAS für Pascalquelldateien usw. nehmen.

Zwischen dem Namen und dem Typ wird als Trennzeichen der "." genommen.

Beispiele für gültige Dateinamen sind:

```
FORMAT.68K
FORMAT.ASM
B:FORMAT.ASM
BRIEF1.TXT
BRIEF_2.TXT
```

Beispiele für ungültige Dateinamen sind

```
BRIEF..
LANGER_BRIEF
8:BRIEF.TXT
```

LADEADRESSE

Dieser Begriff taucht des öfteren im Zusammenhang mit Programmen auf. Die Ladeadresse ist identisch mit dem Beginn des Benutzerspeichers. Bei Systemen mit Bootkarte ist die Ladeadresse gleich $400. Bei den anderen Systemen mit dem Grundprogramm ab Adresse 0 liegt die Ladeadresse um $4000 hinter dem Start der Grundprogrammvariablen. Beim Grundprogramm 6.2 wäre dies die Adresse $14000.

Die Programmdateien vom Typ 68K werden grundsätzlich auf die Ladeadresse geladen und dort gestartet.

Der KOMMANDOEDITOR

Zur Eingabe der Kommandos steht ein komfortabler Zeileneditor zur Verfügung, der eine volle Cursorsteuerung und zahlreiche Korrekturmöglichkeiten bietet. Im folgenden werden die Möglichkeiten des Editors erläutert.

Ctrl S	--> Cursor um ein Zeichen nach links
Ctrl D	--> Cursor um ein Zeichen nach rechts
Ctrl A	--> Cursor auf Wortanfang nach links
Ctrl F	--> Cursor auf Wortanfang nach rechts
TAB	--> Cursor auf Wortanfang nach rechts
Ctrl G	--> Zeichen unter dem Cursor löschen und Rest der Zeile nach links
Ctrl T	--> Alle Zeichen ab dem Cursor löschen
BACKSP	--> Zeichen links vom Cursor löschen und Rest der Zeile nach links
Ctrl V	--> Einfügemodus ein-/ausschalten
RETURN	--> Kommandoeingabe abschließen
Ctrl U	--> Kommandoeingabe abbrechen
Ctrl C	--> Kommandoeingabe abbrechen und Warmstart ausführen
Ctrl E	--> Kommandoeingabe abbrechen und voriges Kommando sichtbar machen
Ctrl X	--> Kommandoeingabe abbrechen und nächstes Kommando sichtbar machen

Es ist ein sogenannter "Historypuffer" eingebaut, in dem die letzten zwanzig Kommandos abgelegt sind. Diese können mit den Steuerfunktionen Ctrl E und Ctrl X sichtbar gemacht werden. Jedes gültige Kommando (mit RETURN abgeschlossen) wird in den Buffer übernommen. Das älteste Kommando wird aus dem Buffer gestrichen (wenn der Buffer voll ist). Mit Ctrl E "wandert" man in Richtung Vergangenheit durch den Buffer. Vom ältesten Eintrag kommt man dann wieder zum neuesten Eintrag. Mit Ctrl X wandert man in Richtung auf den neuesten Eintrag, wobei man beim ältesten Eintrag beginnt. Hat man ältere Kommandos sichtbar gemacht, so steht der Cursor auf Position 1. Das Kommando kann dann direkt mit RETURN ausgeführt werden oder nach den Erfordernissen modifiziert werden.

Der KOMMANDOINTERPRETER

JADOS ist ein Betriebssystem, das mit Kommandos bedient wird, die mit dem weiter oben beschriebenen Zeileneditor eingegeben werden. Die Kommandos nebst den oft benötigten Parametern werden von einem Interpreter bearbeitet und in die dazu nötigen internen Funktionen von JADOS umgesetzt. In der Regel werden die Kommandos mit der Tastatur erzeugt, es gibt aber auch die weiter unten beschriebene Möglichkeit, mehrere Kommandozeilen zu einer Batchdatei zusammenzufassen.

Die Bereitschaft des Interpreters, Kommandos entgegenzunehmen, wird durch das Systemprompt dargestellt. Dieses besteht aus der Nummer des momentan aktuellen Laufwerks und dem Zeichen ">". Wenn also das Laufwerk 1 aktuell ist, dann lautet das Prompt

1>

und wenn die Festplattenpartition A aktuell ist, dann

A>

Sofort nach dem Systemstart und immer dann wenn ein Kommando vollständig abgearbeitet oder sonstwie beendet wurde, erscheint das Prompt.

Ein Kommando besteht aus einem zusammenhängenden Wort von maximal acht Zeichen und eventuell Parametern. Es gibt die sogenannten internen Kommandos, die aus maximal fünf Zeichen bestehen und die fest in JADOS eingebaut sind, und die externen Kommandos, die entweder die Namen von Programmdateien oder einer Batchdatei sind. Bei der Interpretation haben die internen Kommandos die höchste Priorität. Wenn also ein Programm den gleichen Namen hat wie ein internes Kommando, dann hat immer das interne Kommando den Vorrang. In der Regel benötigt ein Kommando weitere Parameter, wie z.B. den Namen einer Datei. Das Kommandowort und die Parameter sind durch mindestens ein Leerzeichen zu trennen. Die Länge eines Parameters liegt zur Zeit bei maximal 14 Zeichen.

Der erfahrene Benutzer kennt die Kommandos und auch die nötigen Parameter und wird alles zusammen in die Kommandozeile eingeben. Dies ist auch der schnellste Weg. Wer sich aber nicht alles merken will, der braucht nur den Kommandonamen eingeben. Die Kommandos sind in der Regel so programmiert, daß fehlende Parameter interaktiv vom Anwender erfragt werden.

Bei den Kommandos wird eine strenge Hierarchie eingehalten. Dabei geht der Interpreter der Reihe nach folgende Möglichkeiten durch:

Internes Kommando

Speicherresident

68K - Programm

COM - Programm

BAT - Datei

Daß es sich um ein externes Kommando handelt, erkennt JADOS am Dateityp. Der Typ wird bei der Eingabe des Kommandos nicht mit angegeben und wird auch nicht akzeptiert. Um das Programm BRDRUCK.68K aufzurufen, wird lediglich "brdruck" eingegeben.

Falls ein eingegebenes Kommando weder als internes noch externes Kommando existiert, erfolgt die Fehlermeldung

Falsches Kommando

Es kann mit dem internen Kommando "dpath" ein Suchpfad auf bis zu 8 Laufwerke eingerichtet werden. Der Interpreter sucht dann auf allen im Pfad angegebenen Laufwerken nach der gewünschten Programmdatei.

Programm des Typs 68K

Diese Programmdateien werden von JADOS auf die Ladeadresse geladen und dann von dort gestartet. Das Programm muß also am Anfang ausführbaren Maschinencode besitzen. Es darf aber auch den aus dem Grundprogramm bekannten Bibliotheks-vorspann besitzen, der unter anderem auch die tatsächliche Startadresse des Programms bestimmt. JADOS akzeptiert aber nur Programme, die als relokativ markiert sind. Das Programm muß zur Beendigung den Maschinenbefehl "RTS" benutzen, um in JADOS zurückzukommen.

Ein 68K-Programm sollte relokativ sein, d.h. an keine feste Ablaufadresse gebunden. Da es aber auf eine feste Adresse, in der Regel $400, geladen wird, darf es auch nicht relokativ sein.

Programm des Typs COM

Programme dieses Typs müssen zwangsläufig relokativ sein, da es ansonsten zu einem Rechnerabsturz kommt. Diese Art von Programmdateien werden nicht auf die Ladeadresse geladen, sondern soweit hinten wie möglich im Benutzerspeicher. Dies hat den Vorteil, daß der Bereich ab der Ladeadresse freibleibt. Die tatsächliche Ablaufadresse hängt vom Speicherausbau ab und davon, ob eine Ramdisk eingerichtet ist und ob und wieviele residente Programme eingerichtet wurden. Variablenspeicher dürfen solche Programme nur mit der Assembleranweisung "ds.b" reservieren oder auf dem Stackbereich anfordern, der noch vor dem COM-Programm liegt.

Residentes Programm

Ein 68K- oder COM-Programm kann resident, also dauerhaft, im Arbeitsspeicher eingerichtet werden. Bis zu 20 solcher Programme werden verwaltet. Falls es sich um ein COM-Programm handelt, dann wird es nach Eingabe seines Namens in der Kommandozeile auf der Speicheradresse gestartet, auf der es sich gerade befindet. Dies ist ein vorteilhafter Platz für Treiberprogramme, die auch nach Beendigung noch an ihrem Platz im Speicher verbleiben müssen, da sie bestimmte Aufgaben während der gesamten Einschaltdauer des Rechners verrichten müssen. Ein Beispiel für einen solchen Treiber könnte ein Hardcopyprogramm sein, welches sich in den Tastaturtrap "eingehängt" hat und nach einem bestimmten Tastendruck aktiv wird, sich ansonsten aber wie der normale Tastaturtrap benimmt.

Die 68K-Programme werden zuerst auf die Ladeadresse kopiert und erst dann gestartet. Die früher einmal vorhandene Grenze von 256 KByte ist aufgehoben.

Der BATCHPROZESSOR

In der Regel werden Kommandos nur über die Tastatur eingegeben. Es gibt aber die Möglichkeit, Sequenzen von Kommandos in einer Textdatei zusammenzufassen. Eine solche Datei wird Batchdatei oder Kommandodatei genannt und wird nach Aufruf automatisch abarbeitet. Immer wiederkehrende Kommandofolgen, z.B. bei einer Datensicherung, können auf diese Weise bequem und sicher ablaufen.

Dies entspricht der SUBMIT-Funktion des bekannten Betriebssystems CP/M 68K.

Die Kommandodateien sind ganz normale Textdateien, die mit dem Editor erstellt werden. Sie müssen vom Dateityp BAT sein, damit JADOS sie als Kommandodateien erkennen kann.

Jede Textzeile darf ein JADOS-Kommando enthalten. Dabei können alle interne und externe Kommandos verwendet werden. Lediglich die Verwendung von Kommandodateien ist eingeschränkt, da eine Verschachtelung von Kommandodateien nicht zulässig ist. Nur das letzte Kommando einer Batchdatei darf eine Kommandodatei sein ! Auf diese Weise können Kommandodateien verkettet werden.

Die Kommandos werden der Reihe nach abgearbeitet, bis das letzte Kommando bearbeitet worden ist. Der Benutzer kann die Batchverarbeitung unterbrechen, indem er die Taste <ESC> betätigt. JADOS fragt dann:

Stapelverarbeitung abbrechen (J/N) ?

Durch Eingabe von "J" wird die Stapelverarbeitung komplett beendet, ansonsten wird sie korrekt fortgeführt.

Eine Kommandodatei darf Kommandos mit formalen Parametern enthalten. Diese formalen Parameter heißen %1, %2, %3 und %4. Beim Aufruf einer Kommandodatei können bis zu 4 Parameter mitgegeben werden, die dann die formalen Parameter ersetzen. Dadurch wird eine hohe Flexibilität erreicht.

Ein Beispiel soll dies verdeutlichen:

Eine Textdatei (z.B. mit dem Namen dateixyz) soll viermal ausgedruckt werden. Man könnte nun viermal das Kommando PRINT dateixyz über die Tastatur eingeben und dazwischen jedesmal den Drucker auf eine neue Seite einstellen. Eine Kommandodatei erfüllt dies wesentlich eleganter. Um die Kommandodatei allgemeingültig zu halten, d.h. auf Textdateien unterschiedlichen Namens anwendbar zu machen, muß die Kommandodatei mit einem formalen Parameter arbeiten. Die folgende Kommandosequenz erfüllt alle Anforderungen:

```
FF
PRINT %1
FF
PRINT %1
FF
PRINT %1
FF
PRINT %1
FF
FF
```

Angenommen, die Kommandodatei habe den Namen DRUCKE4.BAT. Wird nun

DRUCKE4 dateixyz

eingegeben, so ersetzt JADOS den formalen Parameter %1 durch den aktuellen Parameter dateixyz.

Soll nun eine Textdatei mit dem Namen dateiabc viermal ausgedruckt werden, so muß man lediglich

DRUCKE4 dateiabc

eingeben. Die Kommandodatei konnte dabei unverändert übernommen werden. Damit hat sich der kleine Arbeitsaufwand, die Kommandodatei zu erstellen, bereits gelohnt!!

Die folgende Kommandodatei, sie sei mit DOSSAVE.BAT bezeichnet, mit deren Hilfe eine Datensicherung aller Dateien, die zur Entwicklung von JADOS dienten, ausgeführt wird, möge abschließend als Beispiel einer nützlichen Anwendung der Batchverarbeitung dienen:

```
bell
rem Diskette in 2: einlegen
pause
copy a:jados.a 2:
copy a:command.a 2:
copy a:cmdtools.a 2:
copy a:lineedit.a 2:
copy a:dostabs.a 2:
copy a:strtools.a 2:
copy a:misc.a 2:
copy a:filmanag.a 2:
copy a:filtools.a 2:
copy a:datim.a 2:
copy a:dosgrund.a 2:
copy a:urlader.a 2:
bell
rem Ende des Backups
```

Nach Aufruf der Kommandodatei "dossave" ertönt ein akustisches Signal. Danach fordert JADOS den Benutzer auf, die Zieldiskette in Laufwerk 2 einzulegen und die Leertaste zu betätigen. Danach werden alle Dateien, die zur JADOS-Entwicklung dienten von der Festplattenpartition A auf Floppy 2 kopiert.

Der 1. große Vorteil von Kommandodateien ist, daß sie immer wieder verwendet werden können. Eine Kommandofolge muß nur einmal erstellt werden. Dadurch spart man sehr viel "Tipparbeit". Der 2. große Vorteil ist, daß die Kommandos automatisch abgearbeitet werden. In dieser Zeit muß der Benutzer nicht unbedingt am Computer sitzen. Der 3. große Vorteil ist, daß man sich über die Kommandoabfolge keine Gedanken mehr machen muß. Dies hat man ja bei der Erstellung der Kommandodatei bereits getan.

Ein weiteres Beispiel soll zeigen, wie eine Kommandodatei vorteilhaft zur Entwicklung kleiner Programme eingesetzt werden kann:

```
EDIT %1.ASM
ASS %1.ASM
%1
```

Hierbei wird ein Entwicklungszyklus bestehend aus Editieren, Assemblieren und Starten des Programms durchgeführt.

INHALTSVERZEICHNIS

Dateinamen 2

Ladeadresse 4

Der Kommandoeditor 5

Der Kommandointerpreter 6

Programm des Typs 68K 7
Programm des Typs COM 7
Residentes Programm 8

Der Batchprozessor 9