

Rolf-Dieter Klein

CP/M – ein Betriebssystem für jedermann

Teil 2

Im ersten Teil dieser Serie wurden die Aufgaben des BIOS, des Basic Input Output System geschildert. BIOS ist eine Sammlung von Routinen, die den Kontakt zur Hardware des Computers herstellen. Zum Beispiel enthält das BIOS eine Routine, die es erlaubt, von der angeschlossenen Disk-Station einen Sektor zu lesen. BIOS hilft auch beim Schreiben auf den Bildschirm oder beim Schreiben auf einen Drucker. Die Arbeit mit BIOS allein wäre für einen Programmierer noch sehr mühsam. Vor allem die Verwaltung größerer Daten- und Programm-Mengen auf einem angeschlossenen Massenspeicher müßte ein Benutzer sich immer wieder neu ausdenken. Den eigentlichen Wert von CP/M machen deshalb die BDOS-Routinen aus, die gerade dieses Problem ein für allemal lösen.

Das BDOS besitzt genau einen Einsprung, der in Adresse 5 des Hauptspeichers steht. Parameter, die in Registern oder Speicherteilen an BDOS übertragen werden, geben an, was BDOS tun soll.

Beim Kaltstart wird die Adresse 5 von BIOS aus mit dem Sprungbefehl belegt.

Alle CP/M-Programme können also über diesen Sprung an das BDOS gelangen, und zwar unabhängig von der eigentlichen Lage des BDOS. Leider gibt es auch Ausnahmen. Manche Systeme verwenden eine andere Adresse für den Einsprung, die oberhalb von 4000H liegt.

Solche Systeme sind allerdings eine Minderheit. Es gibt daher auch nicht allzu viele Programme dafür. Auf dem mc-CP/M-Computer lassen sich im Prinzip beide Systeme fahren. Jedoch sei von der Verwendung des 4000H-Systems abgeraten, da ja auch der Adreßraum für An-

wenderprogramme sehr stark eingeschränkt ist (16 KBytes weniger).

Auf der Adresse 0 befindet sich ebenfalls ein Sprung, der auf den Warm-Boot-Teil des BIOS zeigt. Wird dieser angesprungen, so wird das CP/M-Betriebssystem neu geladen und gestartet.

g EA03

Die Aufgabe von BDOS

Das BDOS hat die Aufgabe, die Dateiverwaltung vorzunehmen. Was versteht man aber darunter? Beim BIOS konnte man nur über Sektor- und Spur-Nummer (= Adresse) auf einen Sektor der Diskette zugreifen. Wenn zum Beispiel das Programm „Basic-Interpreter“ 8 KByte umfaßt, ein Programm „Pascal“ 32 KByte, und beide Programme auf der Diskette festgehalten werden sollen, dann muß man zuerst jeweils die benötigte Anzahl von Sektoren ermitteln. Bei 128 Bytes pro Sektor wären das beim Basic-Interpreter 64 Sektoren und beim Programm

Pascal 256. Ein Track habe nun zum Beispiel 26 Sektoren und es gebe insgesamt 77 Spuren. Der Adreßbereich der Sektoren liegt dann zwischen 1 und 26 und bei den Tracks bei 0 bis 76. Dann wäre eine Zuweisung wie folgt möglich:

```
Basic
Interpreter: Start Track 0   Sektor 1
              Ende  Track 2   Sektor 13
Pascal:      Start Track 2   Sektor 14
              Ende  Track 12  Sektor 11
```

Man müßte dann eine Liste führen, auf der genau diese Tabelle steht. Daß das ganze sehr mühsam und fehlerträchtig ist, kann man sich denken.

Besser ist es, wenn der Computer selbst in der Lage ist, eine solche Liste zu führen. Die Liste nennt man dann Directory oder Inhaltsverzeichnis oder auch Datei-Katalog.

Wir haben hier also eine Möglichkeit gesehen, ein Directory aufzubauen. Dazu wird einmal der Name der Datei notiert und zum anderen der Start (Track, Sektor) und das Ende (Track, Sektor). Und so wird mit jeder Datei verfahren.

Dieses Verfahren wird oft genauso angewendet. Doch es hat auch Nachteile. Was passiert zum Beispiel, wenn man eine Datei mitten aus einer Gruppe von weiteren Dateien herauslöschen will? Es bleibt ein Loch übrig. Beispiel:

Vor dem Löschen:

Name	Start		Ende	
	Track	Sektor	Track	Sektor
Datei 1	1	1	4	5
Datei 2	4	6	10	3
Datei 3	10	4	23	16
Datei 4	23	17	40	2

Datei 2 soll beispielsweise gelöscht werden. Dann sieht das Inhaltsverzeichnis danach wie folgt aus:

Name	Start		Ende	
	Track	Sektor	Track	Sektor
Datei 1	1	1	4	5
Datei 3	10	4	23	16
Datei 4	23	17	40	2