

und kann damit auf die Diskette zurückgeschrieben werden. Auch hier ist es möglich, noch das BIOS durch ein eigenes auszutauschen, ehe es verwendet wird.

Bei dem im Handel befindlichen Original-CP/M-Betriebssystem wird nach MOVCPM das BIOS für das Intel MDS 800 System geladen. MOVCPM darf aber von lizenzierten Händlern konfektioni-ert werden. Das gilt zum Beispiel für den mc-CP/M-Computer. Dort wird nach dem MOVCPM sofort ein ablauffähiges BIOS bereitgestellt. Das Gleiche gilt für das BOOT-Programm. Das CP/M wird normalerweise als 20-KByte-System ausgeliefert, um alle Möglichkeiten auch für kleinere Systeme offen zu halten.

Der nun der Ablauf zur Erzeugung eines 60-KByte-Systems.

Zuerst der Befehl

```
MOVCPM 60 *
```

Danach gibt das Programm MOVCPM folgende Meldung aus:

```
READY FOR „SYSGEN“ OR
```

```
„SAVE 32 CPM60.COM“
```

Es kann hier also auch eine .COM-Datei gebildet werden, die sich aber nicht direkt starten läßt. Besser aber ist es, das System sofort nach dem Booten zu haben. Daher wird dann SYSGEN aufgerufen. Die Quelle wird mit CR beantwortet und die Zieldiskette angegeben. Danach befindet sich dort das 60-KByte-System. Die Meldung beim Booten, die die Größe des CP/Ms angibt, wird im BIOS automatisch vom MOVCPM-Programm aktualisiert, so daß nach dem Kaltstart die neue Größe des Systems auch gemeldet wird.

MOVCPM kann auch mit zwei Sternen aufgerufen werden

```
MOVCPM **
```

Dann wird ein maximales CP/M erzeugt und für die SYSGEN-Operation vorbereitet.

Die Kommandodateien .SUB

Werden häufige Kommandosequenzen immer wieder benötigt, so können diese Kommandos in eine Datei geschrieben werden und mit dem Programm SUBMIT werden sie dann automatisch nach-

einander ausgeführt. Die Datei muß dabei die Extension .SUB besitzen. SUBMIT kann mehrere Parameter erhalten. Der erste Parameter ist der Name der abzuarbeitenden Datei. Dann können noch Parameter angegeben werden, die an die Kommandos in der Datei als lokale Parameter weitergereicht werden können. Die Parameter haben in dieser Datei die feste Bezeichnung \$1, \$2, \$3, \$4, ... \$n. Nach dem Start der SUBMIT-Funktion wird eine Zwischendatei \$\$\$SUB erzeugt. Danach wird ein Reboot (Warm Start) durchgeführt. Nun sucht der CCP (Consol Command Prozessor) auf dem Laufwerk A nach einer solchen Datei. Findet er diese, so werden die einzelnen darin befindlichen Befehle ausgeführt. Die Datei \$\$\$SUB muß also auf dem Laufwerk A abgelegt worden sein. Die Ausführung kann durch Eingabe des Zeichens „Rubout“, während der CCP ein neues Kommando aus dieser Datei liest und dabei auf dem Bildschirm anzeigt, abgebrochen werden.

Beispiel: Es soll eine Programm-Übersetzung mit nachfolgendem Laden durchgeführt werden. Dabei soll die zu übersetzende Datei beim Start dieser Kommandodatei angegeben werden.

Inhalt der Kommandodatei:

```
ASM $1
LOAD $1
ERA $1.HEX
```

Die Datei soll den Namen UEBER.SUB erhalten. Nun kann z. B. mit der Sequenz SUBMIT UEBER TEST die (schon

existente) Datei TEST.ASM übersetzt werden und es wird eine Datei TEST.COM erzeugt. Die Zwischendatei TEST.HEX wird gelöscht. Im Betriebssystem 2.2 gibt es noch eine weitere Datei, die dazugehört, XSUB.COM. Mit SUBMIT war es nämlich nicht möglich, Konsol-eingaben in die Kommando-Datei einzutragen. XSUB behebt diesen Mangel. Dazu wird XSUB als erster Befehl in eine Kommandodatei geschrieben. Nun ist folgende Sequenz möglich:

```
XSUB
DDT
I$1.HEX
R
G0
SAVE 2 $2.COM.
```

Wenn die Datei mit dem Namen SICHERE.SUB angelegt wurde, so kann durch die Sequenz SUBMIT SICHERE TEST NEUTEST eine Datei TEST.HEX in eine Datei NEUTEST.COM gewandelt werden, wobei diese Sequenz nicht das LOAD-Kommando ersetzen kann, da nur konstant ein Block abgelegt wird.

XSUB lädt sich selbst unter den Bereich des CCP und meldet sich von dort mit (xsub active) nach jedem Warm-Start. So lange, bis ein Kaltstart durchgeführt wird.

Im nächsten Teil werden die BDOS-Befehle geschildert werden.

(Fortsetzung folgt)

