Programme und Kommandos

Speicheraufteilung

Der U880 (Z80) adressiert einen Speicherbereich von 0000h - 0FFFFh (64 kByte). Den Speicher teilen sich Betriebssystem, Bildwiederholspeicher, Anwendungsprogramme und Daten.

Der Z9001 verwaltet den Speicher nicht selbst. Der Anwender muss selbst die Speicheraufteilung und -nutzung im Blick haben. Die **Programme** haben feste Speicherbereiche (Programmstandorte und Arbeitsspeicher). Gerade bei der Nutzung von Treiber-Programmen ist deshalb darauf zu achten, dass sich verschiedene Treiber und die restlichen Programme nicht in den Speicherbereichen überschneiden!

Standard-Aufteilung mit 2 gesteckten 16K-RAM-Modulen:

<ditaa noedgesep>

```
FFFF          +-----------+
              | BOS       |       Z9001 Betriebssystem
F000          +-----------+
              | BWS +FAS  |       Bild- +Farbattributspeicher
E800          +-----------+
              |           |
              | ROM-Modul |       ROM-Basic (oder IDAS-Modul o.a.)
              |           |
C000          +-----------+
              |           |
              | RAM-Modul |       16K-RAM-Modul
              |           |
8000          +-----------+
              |           |
              | RAM-Modul |       16K-RAM-Modul
              |           |
4000          +-----------+
              |           |
              | RAM       |       System-RAM
              |           |
0300          +-----------+
              | System    |       BOS-Arbeitsbereich
0000          +-----------+
```

</ditaa>

Einige Programme und ihre Speicherbereiche:

```
Programmname    anfangsadresse-endadresse (hexadezimal)

IDAS            0400-1BFF  interpretativer Dialogassembler
```

```
ASM             1000-37FF  Assembler + Editor, EDAS
ZM30        3000-3DFF  Zusatzmonitor
ZMA8        A800-B5FF
EPROM2A         2A00-2DFF  EPROM-Programmierung
EPROM6A         6A00-6DFF
EPROMA2         A200-A5FF
BASIC           C000-E7FF  ROM-Basic
```

Dateien

Programme können in ROM-Modulen enthalten sein (IDAS-Modul, ASM-Modul, BASIC-Modul u.a.). Diese Programme sind sofort nach dem Einschalten des Kleincomputers verfügbar und brauchen nicht erst geladen zu werden.

Sonstige Programme werden von Kassette (oder von anderen externen Speichermedien wie Diskette oder USB-Stick) in den Speicher geladen. Programme und Daten sind in diesem Fall in **Dateien** organisiert.

Das „Dateisystem“ einer Kassette ist ganz einfach aufgebaut: alle Dateien liegen hintereinander auf der Kassette. Der Anwender muss selbst den Überblick behalten, an welcher Position der Kassette welches Programm abgelegt ist. Dieses ist üblicherweise auf den Kassetteneinlegern vermerkt.

KLEINCOMPUTER robotron KC 87
Programmkassette R 0112

A
R + HANOI
R + FPLOT
R + MOSAIK
ZM 30
EPROM 2 A
EPROM 6 A
EPROM A 2
R + DEMO 1
R + DEMO 2
R + DEMO 3

B
R + DEMO 1
Test für R + DEMO 1
R + DEMO 2
Test für R + DEMO 2
R + DEMO 3
Test für R + DEMO 3

Platte 1

PROGRAMMKASSETTE R 0112

Auf der Seite A sind die ersten 7 Programme zweimal in der angegebenen Reihenfolge aufgezeichnet. Anschließend folgen die 3 Demo-Programme.

Kassetten-Dateien haben einen 128 Byte großen Anfang, der nicht zum Programm gehört (Block 0, enthält den FileControlBlock FCB), s. Kassetteninterface. In diesem sogenannten Kopfblock stehen Programmname, Typ, der Speicherbereich, in den das Programm geladen werden soll (Anfangsadresse, Endadresse), sowie die Startadresse, auf der das Programm nach dem Laden mit `jp startadresse` gestartet wird. Ausführbare Maschinencode-Programme haben den Typ COM, Assembler-Quelltexte den Typ TXT.

Hinweis: BASIC-Programme und -Daten haben eigene Kassettenroutinen, ein eigenes Dateiformat und verzichten auf diesen Kopfblock!

Laden

Um ein Programm von Kassette zu laden, gibt man einfach den Programmnamen am Prompt ein. Der

Typ COM wird automatisch ergänzt, kann aber ebenfalls mit angegeben werden:

```
>ZM30
```

oder

```
>ZM30.COM
```

Es erscheint die Aufforderung

```
start tape
```

Nach Enter wird auf Kassette der nächste Kopfblock geladen. Steht in ihm derselbe Programmname wie eingegeben und als Typ COM, wird das Laden fortgesetzt. Andernfalls gibt es eine Fehlermeldung.

Nach dem Laden startet das Programm automatisch[1)]!

Kommandos

Um nicht jedesmal ein Programm von Kassette laden zu müssen, um es zu starten, oder um Programme zu starten, die im ROM-Modul stehen, enthalten Maschinencode-Programme ein oder mehrere **Kommandos**, die vom OS aus durch Eingabe des Kommandonamens gestartet werden. Nach Laden des Programms stehen die enthaltenen Kommandos dauerhaft bereit, bis der Speicherbereich durch ein anderes Programm überschrieben wird.

Meist sind der Programmname und (einziger) Kommandoname identisch. Es gibt praktisch keinen Unterschied zwischen Programm und Starten des Kommandos. Beim erstmaligen Aufruf wird das Programm geladen und gestartet, beim erneuten Aufruf wird das gleichnamige Kommando im Speicher gefunden und gestartet.

Programme wie Assembler von Kassette R 0121 u.a. enthalten dagegen mehrere Kommandos, die erst nach dem Laden des Programms zur Verfügung stehen. Hier muss zweistufig gearbeitet werden:

```
>ASM       Laden des Assembler-Programmpakets in den Speicher
start tape
...
>EDIT      Dieses Kommando ist erst nach Laden des Assemblers verfügbar. Es
startet den Editor.
>ASM       Dieses Kommando ist erst nach Laden des Assemblers verfügbar. Es
startet den eigentlichen Assembler.
```

Kommando ASM und Programmname ASM sind gleich[2)], bewirken aber unterschiedliches.

Wenn ein eingegebenes Kommando nicht im Speicher gefunden wird, versucht das Betriebssystem, ein gleichnamiges Programm mit Typ COM von Kassette zu laden.

Manchmal sind Programmname und Kommando NICHT gleich. Damit kann gezielt ein Laden des Programms vom Start des Kommandos unterschieden werden. Der Zusatzmonitor wird z.B. mit

```
>ZM30
```

von Kassette geladen, das Kommando zum erneuten Start ist jedoch

```
>ZM
```

Kommandos können **Parameter** übergeben werden. Diese sind durch mindestens ein Leerzeichen vom Kommandoname getrennt. Beispiel ist das System-Kommando TIME: mit Parameter wird die Uhrzeit gesetzt, ohne Parameter die Zeit angezeigt.

```
>TIME 07:30    Uhrzeit setzen
>TIME          Uhrzeit anzeigen
07:30:03
>
```

Mega-Modul

Die Mega-Flash-Software stellt neben der Kassette ein weiteres spezielles Dateisystem zur Verfügung, von dem Programme geladen werden können. Im Mega-Modul gibt es zum einen Programme, die im ROM-Bereich des Modul stehen. Deren Kommandos können direkt aufgerufen werden, als wäre das entsprechende 10K-Modul gesteckt. Das betrifft z.B. BASIC und EDIT, ASM oder IDAS oder auch die Modul-eigenen Kommandos DIR, DUMP, SAVE, usw.

Daneben enthält das Mega-Modul eine Vielzahl von Programmen, die in den RAM-Speicher geladen werden müssen. Diese Programme sind als FA-Dateien im Mega-Modul abgelegt[3)]. FA-Dateien enthalten anstelle des Blocks 0 der Kassettendateien einen 32 Byte großen Bereich, in dem Programmname, Typ, Anfangs-, End-, und Startadresse stehen. Die Modulsoftware (Systemerweiterung '#') steuert den Ladevorgang.

Wird ein Programmname eingegeben, sucht die Modulsoftware zuerst einen passenden Kommandonamen im Speicher. Wird hier nichts gefunden, wird ein passender FA-Dateiname im Mega-Modul (typ COM) gesucht. Wird er gefunden, wird das Programm in den RAM-Speicher geladen und dort gestartet. Wird er nicht gefunden, wird nun auf Kassette gesucht. Es erscheint „start tape“.

Die Modulsoftware unterstützt im Unterschied zum OS die **Übergabe von Parametern** an zu ladende Programme. Bei Start des Programms werden die Parameter übergeben. Damit können Programme/Kommandos praktisch ohne Unterschied genutzt werden, egal ob sie schon in den RAM geladen wurden oder noch aus dem Modul geladen werden müssen. OS-SAVE (adaptiert von R 0111) ist ein Beispiel dafür:

```
>OS-SAVE TEST.BIN 1000,10FF,0FFFF
```

lädt erst OS-SAVE in den RAM-Speicher und startet es mit den übergebenen Parametern.

Als Besonderheit können auch **BASIC-Programme** als FA-Dateien im Mega-Modul abgelegt werden. Diese BASIC-Programme werden ebenso wie die Maschinencodeprogramme direkt am OS-Prompt durch Eingabe des Programmnamens geladen und gestartet. Sie werden nicht in BASIC geladen und können hier auch nicht mit CLOAD„...“ geladen werden. Beispiel:

```
>TATUM
```

initialisiert BASIC, lädt das BASIC-Programm TATUM und startet es mit RUN.

Disketten und USB

Mit den Erweiterungen Disk-OS und OS-Erweiterung USB+SD können Programme und Daten nicht nur auf Kassette, sondern auch auf Disketten bzw. USB-Speichermedien gespeichert und geladen werden.

Bei Disk-OS kommt das CP/M-Dateisystem zum Einsatz. Die Disketten sind 100% kompatibel zu CP/A und können parallel genutzt werden[4)]. BASIC-Programme können unter beiden Systemen (OS und CP/A) gleichwertig benutzt werden.

Bei OS-Erweiterung USB+SD werden FAT32-formatierte Datenträger genutzt. Die USB-Speichermedien können unter DOS/Windows gelesen und beschrieben werden[5)].

Analog zur Kassettenaufzeichnung werden Programme und Daten mit einem Block 0 (Kopfblock) gespeichert. Zusätzlich gibt es noch den Dateinamen, der zum Programmnamen identisch sein muss.

Wird ein Programmname eingegeben, sucht die Erweiterungssoftware zuerst einen passenden Kommandonamen im Speicher. Wird hier nichts gefunden, wird ein passender FA-Dateiname im Mega-Modul (typ COM) gesucht. Wird er gefunden, wird das Programm in den RAM-Speicher geladen und dort gestartet. Wird hier nichts gefunden, wird der Programmname als Datei auf Diskette bzw. USB-Speichermedium gesucht. Wird er gefunden, wird das Programm in den RAM-Speicher geladen und dort gestartet.

Wird er nicht gefunden, wird nun auf Kassette gesucht. Es erscheint „start tape“.

Im BASIC wird ganz normal mit den Kommandos CLOAD/LIST#1/CSAVE gearbeitet. Das Speichern und Laden erfolgt nur auf Diskette bzw. USB-Speichermedium, die Kassette wird nicht genutzt.

Um einen Überblick über die Dateien zu behalten, besitzen beide Erweiterungen das Kommando DDIR zum Anzeigen der Dateien. Auch unter BASIC gibt es BASIC-Befehl DIR zum Anzeigen der BASIC-Dateien. In OS-Erweiterung USB+SD gibt es außerdem das Kommando CD zum Verzeichniswechsel.

1) der Start erfolgt auch, wenn der Typ nicht COM ist! Zum Laden ohne automatisches Starten gibt es das Systemkommando CLOAD.

2) eigentlich ist der volle Programmname ASM.COM und damit nicht gleich dem Kommando

3) Details zu FA stehen hier: software_ins_modul_integrieren

4) Intern werden Maschinencode-Programme vom Typ COM werden als Typ KCC gespeichert, um eine Verwechslung mit CP/M-Programm zu verhindern. Der Anwender muss sich nicht darum kümmern, er speichert/lädt mit COM!

5) Dateinamen dürfen nur 8 Zeichen lang sein!

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z9001/software/os/kommandos?rev=1686667009

Last update: **2023/06/13 14:36**