

Basic-Zeilen genau betrachtet

Wenn Sie eine Basic-Zeile eingeben und anschließend RETURN drücken, legt der Interpreter die Zeile im Speicher ab. Will man Basic-Zeilen durch ein Maschinenprogramm erzeugen oder in einem Basic-Programm nach bestimmten Befehlen suchen, muß man wissen, wie und wo Basic-Zeilen gespeichert sind.

In den Speicherzellen 43/44 steht die Anfangsadresse des Basic-Speichers. Die Adresse, bei der Basic beginnt, erfahren Sie durch die Abfrage PRINT PEEK(43)+256*PEEK(44). Sie erhalten den Wert 2049. Das ist die Adresse, ab der die erste Basic-Zeile vom Interpreter gespeichert wird. Wie Basic-Zeilen im Speicher vorliegen, soll am folgenden Beispielprogramm erläutert werden.

```
10 Print "PROBE"
300 REM C - 64
1200 A$ = "A = A"
50000 END
```

Bild 1 und 2 zeigen, wie der Interpreter diese vier Basic-Zeilen ab der Adresse 2049 speichert. Die Werte in Klammern sind die Startadressen der einzelnen Basic-Zeilen und dienen nur dem besseren Überblick. Alle Werte sind dezimal und hexadezimal angegeben.

Die ersten beiden Bytes jeder Zeile heißen Linkpointer oder einfach Verbindungszeiger, manche sprechen auch von Koppeladressen. Der Wert dieser beiden Bytes entspricht immer der Startadresse der nächsten Zeile. Das erste Byte ist das Low- und das zweite das High-Byte der Adresse. In Bild 1 ergeben die ersten 2 Byte der ersten Zeile den Wert

15+8x256=2063, also die Adresse, bei der die zweite Zeile beginnt. Der Linkpointer der vierten Zeile (47+8x256) zeigt auf die Speicherstelle 2095. Die beiden Null-Codes in 2095 und 2096 (der Linkpointer der fünften Zeile) signalisieren dem Interpreter das Programmende.

Byte 3 und 4 jeder Zeile ergeben die Zeilennummer der jeweiligen Basic-Zeile. In der ersten Zeile ergeben diese 2 Byte $10 + 0 \times 256 = 10$.

Im Beispielprogramm steht, nach dem Linkpointer und der Zeilennummer, als fünftes Byte, das Token für PRINT (99). Um Speicherplatz zu sparen und eine schnelle Programmabarbeitung zu erzielen, werden alle Basic-Befehle (PRINT, REM, END,...) in nur 1 Byte übersetzt. Ein 1-Byte-Befehlswort heißt Token. Der Interpreter holt sich die Information, welches Token einem bestimmten Befehlswort entspricht, aus einer Zuordnungstabelle im ROM. Die Tabelle beginnt ab Adresse 41118 (\$A090). Bild 3 zeigt einen Überblick über alle Basic-Befehle und deren Token. Die Leerstelle nach der Zeilennummer, die nach dem LIST-Befehl am Bildschirm zu sehen ist, wird nicht berücksichtigt. Die Leerstelle zwischen PRINT und dem Anführungszeichen steht als ASCII-Code 32 im sechsten Byte. Auf die gleiche Weise werden das Anführungszeichen, das Wort PROBE und das Schlußzeichen im ASCII-Code gespeichert. Das Beispiel zeigt, daß der Interpreter alle Zeichen, außer Befehlswörtern, im ASCII-Code speichert. Das gilt auch für Befehlswörter in Anführungszeichen ("PRINT"). Diese Unterscheidung können Sie in der dritten Zeile deutlich erkennen. Das erste Gleichheitszeichen stuft der Interpreter als Operator ein (Token 178, \$B2), das Gleichheitszeichen in Anführungsstrichen als ASCII-Code 61 (\$3D). Eine Aufstellung der ASCII-Codes finden Sie Ihrem Handbuch zum C 64 auf Seite 135.

Das Ende jeder Programmzeile ist durch eine Null markiert. Der Interpreter erkennt daran das Zeilenende und nimmt sich die nächste Zeile vor. Enthalten die beiden Bytes für den Linkpointer Null-Codes, ist für den Interpreter das Programm zu Ende. Im Beispiel sind es die Adressen 2095 und 2096 (\$082F, \$0830).

	Link- pointer	Zeilen- nummer	Basic-Text
(0801) (2049)	0F 08 15 8	0A 00 10 0	99 20 22 50 52 4F 42 45 22 00 153 32 34 80 82 79 66 69 34 0
			Print " P R O B E "
(080F) (2063)	1C 08 28 8	2C 01 44 1	8F 20 43 20 2D 20 36 34 00 143 32 67 32 45 32 54 52 0
			REM C - 64
(081C) (2076)	29 08 41 8	B0 04 176 4	41 24 B2 22 41 3D 41 22 00 65 36 178 34 65 61 65 34 0
			A \$ = " A = A "
(0829) (2089)	2F 08 47 8	50 C3 80 195	80 00 128 0
			END
(082F) (2095)	0 0		

Bild 1. So legt der Interpreter Basic-Zeilen im Speicher ab.

PC	SR	AC	XR	YR	SP	NV-BDIZC
;COOB	B0	C2	AF	00	F6	10110000
.MO800						
:0800	00	0F	08	0A	00	99 20 22
:0808	50	52	4F	42	45	22 00 1C PROBE"...
:0810	08	2C	01	8F	20	43 20 2D
:0818	20	36	34	00	29	08 B0 04 64.)...
:0820	41	24	B2	22	41	3D 41 22 A\$. "A=A"
:0828	00	2F	08	50	C3	80 00 00 ./..P-...
.X						
READY.						
L5						
10 PRINT	"PROBE"					
300 REM	C - 64					
1200 A\$ =	"A=A"					
50000 END						
READY.						
HARDCOPY						

Bild 2. Das kleine Basic-Programm mit SMON betrachtet.

Mehr Verständnis für den NEW-Befehl

Im Handbuch steht, daß NEW das Programm im Speicher löscht. Das ist nur bedingt richtig, denn der NEW-Befehl löscht nicht das ganze Programm, sondern schreibt nur zwei Null-Codes in die Speicherstellen 2049 (\$801) und 2050 (\$802). Sie können das mit einem Monitor, zum Beispiel dem SMON, überprüfen. Außerdem werden die Zeiger für Variablenanfang, Feldanfang und Feldende auf die Adresse 2051 gesetzt.

Zum Beweis dieser Aussage sollten Sie einmal das Beispielprogramm eintippen und die folgenden Direktbefehle eingeben.

```
NEW
POKE 2049,15
POKE 2050,8 Linkpointer
POKE 45,49
POKE 46,8 Variablen-Anfang
POKE 47,49
POKE 48,8 Feld-Anfang
POKE 49,49
POKE 50,8 Feld-Ende
LIST
```

Die Werte für den ersten Linkpointer und die Variablenzeiger werden mit diesen acht POKE-Befehlen wieder hergestellt. Sie gelten nur für dieses Beispielprogramm. Geben Sie nach NEW eine neue Basic-Zeile ein, dann kann das gelöschte Programm nicht mehr gerettet werden.

Als die Bilder laufen lernten ...

Zaubern Sie Bewegung auf Ihren Bildschirm! Mit diesem Programm können Sie bequem Blockgrafik und Text in frei definierbaren Bildschirmbereichen scrollen. Und das in vier Richtungen.

Mit einem einzigen SYS-Befehl wird sowohl der Zeichen- als auch der Farbcode gescrollt. Dabei können Sie die Größe und Lage des Scrollbereiches frei wählen. Wenn Zeichen aus einem Bereich hinausgeschoben werden, tauchen Sie an der gegenüberliegenden Seite wieder auf.

Mit SYS 50550,r,za,ze,s,l wird das Maschinenprogramm aufgerufen. Dabei werden Parameter für Richtung (r), Zeilenan-

Richtung	r	L,R,H,T
Zeilenanfang	za	1 ... 25
Zeilenende	ze	1 ... 25, za < ze
Spalte	s	1 ... 40
Zeilenlänge	l	1 ... 40 s+l < 41
Syntax		SYS50550,r,za,ze,s,l

Bild 1. Die Parametergrenzen

Wenn Sie nun am Ende dieses Satzes drei Nullen finden, dann wissen Sie

- a) der Satz ist zu Ende und
 - b) der Artikel ist zu Ende.
- 000

(J.Effenberg/hm)

Befehl	Token	Befehl	Token	Befehl	Token
	DEZ	DEZ	HEX	DEZ	HEX
END	128	80	CONT	154	9A
FOR	129	81	LIST	155	9B
NEXT	130	82	CLR	156	9C
DATA	131	83	CMD	157	9D
INPUT #	132	84	SYS	158	9E
INPUT	133	85	OPEN	159	9F
DIM	134	86	CLOSE	160	A0
READ	135	87	GET	161	A1
LET	136	88	NEW	162	A2
GOTO	137	89	TAB	163	A3
RUN	138	8A	TO	164	A4
IF	139	8B	FN	165	A5
REST.	140	8C	SPC	166	A6
GOSUB	141	8D	THEN	167	A7
RETURN	142	8E	NOT	168	A8
REM	143	8F	STEP	169	A9
STOP	144	90	+	170	AA
ON	145	91	-	171	AB
WAIT	146	92	*	172	AC
LOAD	147	93	/	173	AD
SAVE	148	94	!	174	AE
VERIFY	149	95	AND	175	AF
DEF	150	96	OR	176	B0
POKE	151	97	>	177	B1
PRINT #	152	98	=	178	B2
PRINT	153	99	<	179	B3
SGN	180	B4			
INT	181	B5			
ABS	182	B6			
USR	183	B7			
FRE	184	B8			
POS	185	B9			
SQR	186	BA			
RND	187	BB			
LOG	188	BC			
EXP	189	BD			
COS	190	BE			
SIN	191	BF			
TAN	192	C0			
ATN	193	C1			
PEEK	194	C2			
LEN	195	C3			
STR\$	196	C4			
VAL	197	C5			
ASC	198	C6			
CHR\$	199	C7			
LEFT\$	200	C8			
RIGHT\$	201	C9			
MID\$	202	CA			
GO	203	CB			

Bild 3. Basic-Befehle und deren Token

fang (za), Zeilenende (ze), Spalte (s) und Zeilenlänge (l) übergeben. Die Parameter müssen innerhalb der Grenzen von Bild 1 liegen. Angaben außerhalb dieser Grenzen fängt das Programm ab und gibt die entsprechende Fehlermeldung aus.

Listing 2 ist ein Demo-Programm, das Ihnen die verblüffende Wirkung des Scrollens in vier Richtungen zeigt.

(J. Effenberg/hm)

```
PROGRAMM : ROLLING          C576 C815
-----
C576 : 20 FD AE B1 7A C9 4C D0 3F
C57E : 07 E9 4C 8D A8 02 F0 22 E1
C586 : C9 52 D0 07 E9 51 8D A8 3E
C58E : 02 D0 17 C9 48 D0 07 E9 F2
C596 : 46 8D A8 02 D0 0C C9 54 4A
C59E : F0 03 4C 16 C7 E9 51 8D 12
C5A6 : A8 02 E6 7A D0 02 E6 7B 08
C5AE : 20 F1 B7 CA 30 09 E0 19 0F
C5B6 : B0 05 8E A9 02 50 03 4C 09
C5BE : 20 C7 20 F1 B7 CA 30 F7 8B
C5C6 : E0 19 B0 F3 8E AA 02 EC FD
C5CE : A9 02 F0 05 B0 0D 4C 2A 4E
C5D6 : C7 AD A8 02 C9 02 90 03 D3
C5DE : 4C 52 C7 20 F1 B7 CA 30 B2
C5E6 : 04 E0 28 90 03 4C 34 C7 69
```

Listing 1. »Als die Bilder laufen lernten...«. Das Programm muß mit dem MSE auf Seite 68 eingegeben werden