

Parameterübergabe an Programme in Maschinensprache

Der SYS-Befehl läßt es zu, daß neben der Startadresse des Maschinenprogrammes auch Parameter übergeben werden.

Das kann zum Beispiel so aussehen: SYS adresse,p1,p2,p3,... . Wobei »adresse« Startadresse bedeutet und p1 bis p3 die Parameter sind. Für die Verarbeitung der Parameter ist es wichtig zwischen 1-Byte- und 2-Byte-Parametern zu unterscheiden, da unterschiedliche Interpreterroutinen nötig sind.

Hier sind die benötigten Interpreterroutinen:

Komma = \$AEFD

1byte = \$B79E

2byte = \$B7EB

Dabei ist die Routine »Komma« nötig, um die Kommas vor den Parametern zu erkennen und die Parameter zu trennen. Anschließend können dann die Parameter mit »1byte« oder »2byte« eingelesen werden.

Übergabe von 1-Byte-Werten

Die Befehle

JSR \$AEFD ; Komma

JSR \$B79E ; 1byte

laden den Parameter bei SYS...,p1 ins X-Register des Prozessors, wo er für die weitere Verarbeitung bereitsteht. Jetzt ist es zum Beispiel möglich p1 zu verarbeiten und mit JSR Komma : JSR 1byte den nächsten Parameter (p2) zu holen. Angewandt wird dies im Beispielprogramm Cursorsetzen.

Folgendes passiert: Nacheinander werden »zeile« (120,130) und »spalte« (160,170) eingelesen und schließlich verarbeitet. Die Routine \$FFFF0 setzt bei gelöschtem Carry-Flag den Cursor nach den Werten im X- und Y-Register an die gewünschte Position, wie schon in Ausgabe 7/84 beschrieben.

Aufruf des Hilfsprogrammes das im Kassettenpuffer liegt:

SYS 828,zeile,spalte

(siehe Listing 1)

SYS 828,10,5:PRINT CHR\$(42) setzt den Stern * auf Zeile 10, Spalte 5.

Übergabe von 2-Byte-Werten

Die Befehle

JSR \$AEFD ; Komma

JSR \$B7EB ; 2byte

bringen den ersten Parameter bei SYS...,p1,p2 auf die Zeropage Adresse \$14/\$15, und den zweiten ins X-Register, da die Routine \$B79E : 1byte (siehe oben) mit aufgerufen wird. Hierbei wird p1 nach Low- und High-Byte getrennt abgespeichert. Sein Wert kann also zwischen 0 und 65535 (= 2¹⁶) liegen. Der Interpreter benutzt diese Routine, um zum Beispiel bei POKE Speicherzellen mit bestimmten Werten zu laden.

Das nächste kleine Programm soll nun den Gebrauch der Routine 2byte = \$B7EB demonstrieren. Es erspart das doppelte POKEn von Bildschirm- und Farb-RAM. Der Aufruf hat die Form:

SYS 828,position, zeichen,farbe

Da »position« hier ein 2-Byte-Wert ist, muß die Routine \$B7EB benutzt werden, gleichzeitig wird dadurch »zeichen« ins X-Register geladen. »Farbe« wird dann später in Zeile 210 über \$B79E geladen (siehe Listing 2).

SYS828,1063,1,1 setzt ein weißes A in die rechte obere Ecke des Bildschirms.

Zum Schluß noch die Basic-Lader der Demoprogramme, für diejenigen, die keinen Assembler oder Monitor besitzen, aber die Routinen gebrauchen können. Sie liegen jeweils im Kassettenpuffer (ab 828). Durch Verändern der Werte in Zeile 110 lassen sie sich jedoch nach Bedarf verschieben.

(Markus Kuhn/rg)

```

10 REM ***** <225>
20 REM * MARKUS KUHN * <000>
30 REM * BAHLENSTR.52 * <057>
40 REM * 4 D'DORF 13 * <057>
50 REM * 0211/767519 * <075>
60 REM * * <031>
70 REM * C-64 + VC 1541 * <146>
80 REM ***** <039>
90 : <148>
100 REM BASICLADEPROGRAMM FUER DEMO 1 CURSORS
    ETZEN <015>
110 FOR I = 028 TO 850 <205>
120 READ X : POKE I,X : S=S+X : NEXT <069>
130 DATA 32,253,174, 32,158,183,138, 72, 32, <040>
    253,174, 32
140 DATA 158,183,138,168,104,170, 24, 32,240, <213>
    255, 96
150 IF S <> 3101 THEN PRINT "FEHLER IN DATAS
    !!" : END <215>
160 PRINT"OK": END <209>

```

Beispiel-Listing 1

```

10 REM ***** <225>
20 REM * MARKUS KUHN * <000>
30 REM * BAHLENSTR.52 * <057>
40 REM * 4 D'DORF 13 * <057>
50 REM * 0211/767519 * <075>
60 REM * * <031>
70 REM * C-64 + VC 1541 * <146>
80 REM ***** <039>
90 : <148>
100 REM BASICLADEPROGRAMM FUER DEMO 2 ZEICHEN
    SETZEN <056>
110 FOR I = 028 TO 856 <211>
120 READ X : POKE I,X : S=S+X : NEXT <069>
130 DATA 32,253,174, 32,235,193,138,160, 0, <026>
    145, 20,165
140 DATA 21, 73,220,133, 21, 32,253,174, 32, <036>
    158,183,138
150 DATA 160, 0,145, 20, 96 <251>
160 IF S <> 3396 THEN PRINT "FEHLER IN DATAS
    !!" : END <241>
170 PRINT "OK" : END <219>

```

Beispiel-Listing 2