Wühlereien im Betriebssystem

Für alle Assembler-Programmierer zeigt der folgende Artikel die Verwendung der wichtigsten ROM-Routinen im C16.

Bevor wir anfangen können, müssen wir noch ein paar Dinge klären.

1) Dieser Artikel geht davon aus, daß Sie Maschinensprachekenntnisse haben.

2) Die vorgestellten Listings werden mit dem integrierten Monitor TEDMON eingegeben. Sie beginnen im Speicher bei $1001, dem Anfang des Basic-Speichers. Damit unsere Programme dort einigermaßen sicher sind, geben Sie bitte im Monitor folgende Befehle ein:

```
>002B 01 11 (setzt Basic-Start herauf)
>1100 00
```

3) Die Programme werden von Basic aus gestartet:

```
SYS DEC("1001")
```

Kernel-Einsprünge

Ab $FF81 stehen im Speicher die sogenannten Kernel-Einsprünge. Diese sind in der Wirkung bei allen Commodore-Computern gleich, was ein eventuell anfallendes Umschreiben (zum Beispiel auf den C64 oder den Plus/4) erleichtert. Wir besprechen hier in Kurzform die wichtigsten; einige davon kennen Sie schon aus besagtem Artikel im C16-Sonderheft 3, 1986.

VIRES ($FF81)

»VIRES« initialisiert den Video-Chip und den Bildschirm-Editor. Diese Routine wird einfach über »JSR $FF81« aufgerufen und benötigt keinerlei Parameter.

IORES ($FF84)

»IORES« initialisiert die Ein-/Ausgabe-Bausteine des C16. Auch hier sind keine Parameter notwendig.

STMSG ($FF90)

Mit dieser Routine können Sie die Ausgabe von Systemmeldungen (Fehlermeldungen etc.) teilweise oder ganz unterdrücken.

Fehlermeldungen sind Meldungen wie »I/O ERROR #«, Systemmeldungen sind »LOADING« und so weiter. Entscheidend ist der Inhalt des Akkus vor dem Aufruf von »STMSG«:

Akku	Fehlermeldungen?	Systemmeldungen?
$00	ja	ja
$40	nein	ja
$80	ja	nein
$C0	nein	nein

So kann man also alle Meldungen ausschalten:

```
LDA #$C0
JSR $FF90
```

SETLFS ($FFBA)

Diese Routine muß man vor »OPEN«, »LOAD« oder »SAVE« zur Angabe der Fileparameter aufrufen. Sie wurde bereits im Sonderheft 3, 1986 vorgestellt.

SETNAM ($FFBD)

Es gilt das gleiche wie für »SETLFS«.

OPEN ($FFC0)

Diese Routine dient dem Öffnen eines Files, das vorher durch »SETLFS« und »SETNAM« (wie vor »LOAD«) bestimmt wurde. In den Registern übergibt man keine Parameter.

CLOSE ($FFC3)

Gegenstück zu »OPEN«. Damit schließt man ein File wieder. Im Akku muß beim Aufruf die logische Filenummer stehen.

CHKIN ($FFC6)

Mit dieser Routine kann man alle Eingaben von »BASIN« oder »GETIN« statt von der Tastatur von einem vorher geöffneten File erfolgen lassen. Im X-Register übergibt man der »CHKIN«-Routine die logische Filenummer dieses neuen Eingabe-Files.

CKOUT ($FFC9)

Wie »CHKIN«, nur daß das Ausgabegerät für »BSOUT« gesetzt wird. Im X-Register übergibt man die Filenummer der Ausgabedatei.

CLRCH ($FFCC)

Wenn »CHKIN« oder »CKOUT« verwendet wurde, kann man über »CLRCH« wieder die normalen Ein-/Ausgabegeräte (Eingabe: Tastatur; Ausgabe: Bildschirm) einstellen lassen.

BASIN ($FFCF)

Wurde schon im Sonderheft 3, 1986 vorgestellt. Es erfolgt die Eingabe über Tastatur oder dem über »CHKIN« angewählten Gerät.

BSOUT ($FFD2)

Wurde auch schon vorgestellt. Ein Zeichen, das im Akku übergeben wird, wird auf dem Bildschirm oder dem über »CKOUT« eingestellten Gerät ausgegeben. Listing 1 und 2 zeigen Beispiele mit den Ausgabe-Routinen.

LOAD ($FFD5)

Auch nichts Neues. Zu dieser Load-Routine soll an dieser Stelle aber noch ein Hinweis gegeben werden.

Wenn man ein Basic-Programm laden will, muß man:

a) dessen Endadresse dem Computer mitteilen und

b) die Linkpointer (siehe Artikel »Durchblick mit dem Monitor« in diesem Heft) neu berechnen lassen.

Um beides zu bewerkstelligen, läßt man dem Aufruf der

```
. 1001  20 E7 FF JSR $FFE7
. 1004  A9 01    LDA #$01
. 1006  A2 04    LDX #$04
. 1008  A0 00    LDY #$00
. 100A  20 BA FF JSR $FFBA
. 100D  A9 00    LDA #$00
. 100F  20 BD FF JSR $FFBD
. 1012  20 C0 FF JSR $FFC0
. 1015  A2 01    LDX #$01
. 1017  20 C9 FF JSR $FFC9
. 101A  A2 00    LDX #$00
. 101C  BD 2F 10 LDA $102F,X
. 101F  20 D2 FF JSR $FFD2
. 1022  E8       INX
. 1023  E0 08    CPX #$08
. 1025  D0 F5    BNE $101C
. 1027  20 CC FF JSR $FFCC
. 102A  A9 01    LDA #$01
. 102C  4C C3 FF JMP $FFC3

>102F 44 52 55 43 4B 45 52 0D :
```

Listing 1. Beispieltext auf Drucker ausgeben

Load-Routine unmittelbar folgende Befehle folgen:

```
STX $2D     ;Lo-Byte der Endadresse setzen
STY $2E     ;Hi-Byte der Endadresse setzen
JSR $8818   ;Linkpointer berechnen lassen
```

Beim Laden von Maschinenprogrammen ist dies natürlich nicht erforderlich.

SAVE ($FFD8)

Siehe Sonderheft 3, 1986.

STOP ($FFE1)

Mit »JSR $FFE1« fragt man die Stop-Taste ab. Wurde sie gedrückt, ist das Zero-Flag im Prozessor-Statusregister gesetzt, das heißt ein BEQ-Befehl würde ausgeführt werden. Das Programm in Listing 3 wartet, bis die Stop-Taste gedrückt wird.

GETIN ($FFE4)

Siehe Sonderheft 3, 1986.

CLALL ($FFE7)

Diese Routine schließt alle offenen Kanäle. Dabei ist jedoch zu beachten, daß dies nicht wie bei »CLOSE« geschieht. Die entsprechenden Geräte werden von »CLALL« nicht angesprochen, das heißt die Kanäle werden nur innerhalb des Computers geschlossen, während ein File auf Diskette/Kassette für das entsprechende Gerät nach wie vor offenbleibt.

Diese Routine kann man am Anfang eines Programms verwenden, nicht aber als Ersatz für »CLOSE« ($FFC3) betrachten.

PLOT ($FFF0)

Siehe Sonderheft 3, 1986.

Die bisherige Aufzählung sollte nur eine grobe Zusammenfassung sein. Auf einige Routinen gehen wir nun näher ein.

Eine Anwendung der Routinen »CLALL«, »SETPAR«, »SETNAM«, »OPEN«, »CKOUT«, »BSOUT«, »CLRCH« und »CLOSE« ist Listing 1. Zunächst werden alle noch offenen Kanäle geschlossen ($1001). Dann werden 1 als Filenummer, 4 (Drucker) als Gerätenummer und 0 als Sekundäradresse gesetzt ($1004-$100A). Bei $100D/$100F wird dem Computer mitgeteilt, daß wir keinen Filenamen wollen (Länge des Filenamens = 0).

Das File kann geöffnet ($1012) und die Ausgabe des Computers darauf umgeleitet werden ($1015/$1017). Eine kleine Schleife gibt den Text ab $102F über »BSOUT« aus ($101A-$1025). »CLRCH« ($1027) setzt wieder den Bildschirm als Ausgabegerät. Schließlich wird das File geschlossen ($102A/$102C) und das Programm beendet (»JMP $FFC3« entspricht »JSR $FFC3« und »RTS«).

Wie die anderen Beispielprogramme, wird Listing 1 mit `SYS-DEC("1001")` gestartet.

Erleichterte Bildschirmausgabe

Wie wir wissen, gibt »BSOUT« das im Akku übermittelte Zeichen auf das aktuelle Ausgabegerät aus; dieses ist der Bildschirm, solange nicht über »CKOUT« ein anderes Ausgabegerät gesetzt ist. Will man gleichzeitig Texte aus dem Bildschirm und das über »CKOUT« eingesetzte Gerät ausgeben, hilft die Routine

BSOUTSCREEN ($DC49)

Diese Routine arbeitet wie »BSOUT«, aber die Ausgabe erfolgt hier in jedem Fall auf dem Bildschirm.

Die Bildschirmausgabe kann auch noch durch weitere Routinen erleichtert werden:

CLEAR ($D88B)

ist eine Abkürzung für

```
LDA #$93    ;$93 = 147 = Code für »Clear Home«)
JSR $FFD2
```

und ist – wie »BSOUTSCREEN« – von »CKOUT« unabhängig.

HOME ($D89A)

ist eine Abkürzung für

```
LDA #$13    ;$13 = 19 = Code für »Home«)
JSR $FFD2
```

und ist wie »CLEAR« von »CKOUT« unabhängig.

SETCR ($D83B)

ist eine Abkürzung für

```
CLC
JSR PLOT
```

und setzt somit den Cursor an die in X und Y angegebene Position.

GETCR ($D849)

ist eine Abkürzung für

```
SEC
JSR PLOT
```

und holt die Cursorposition nach X und Y.

Diese neuen Routinen zur Bildschirmausgabe werden in Listing 2 angewendet, um den Text »(C)64'ER« in die Mitte des Bildschirms zu schreiben.

Ich möchte ausdrücklich darauf hinweisen, daß die Routinen »CLEAR«, »HOME«, »SETCR« und »GETCR« keine Standard-Vektoren sind und somit nur für den C16 gelten.

Jetzt haben Sie die wichtigsten Betriebssystemroutinen kennengelernt. Nicht so bekannt sind vier recht interessante Routinen des TEDMON. Diese Routinen kann man zur hexadezimalen Ausgabe von Zahlen verwenden.

PUTHEX ($FB10)

Im Akku übergeben wir einen Byte-Wert, der dann über JSR $FB10 zweistellig hexadezimal ausgegeben wird.

```
. 1001  20 8B D8 JSR $D88B
. 1004  A2 0A    LDX #$0A
. 1006  A0 0F    LDY #$0F
. 1008  20 3B D8 JSR $D83B
. 100B  A2 00    LDX #$00
. 100D  BD 1C 10 LDA $101C,X
. 1010  20 49 DC JSR $DC49
. 1013  E8       INX
. 1014  E0 08    CPX #$08
. 1016  D0 F5    BNE $100D
. 1018  20 9A D8 JSR $D89A
. 101B  60       RTS

>101C 28 43 29 36 34 27 45 52 :
```

Listing 2. Bildschirmausgabe mit den neuen ROM-Routinen

```
. 1001  20 E1 FF JSR $FFE1
. 1004  D0 FB    BNE $1001
. 1006  60       RTS
```

Listing 3. Abfragen der Run/Stop-Taste

```
. 1001  A9 34    LDA #$34
. 1003  A2 12    LDX #$12
. 1005  20 FF FA JSR $FAFF
. 1008  A9 A0    LDA #$A0
. 100A  20 05 FB JSR $FB05
. 100D  A9 A1    LDA #$A1
. 100F  20 10 FB JSR $FB10
. 1012  A9 2E    LDA #$2E
. 1014  4C D2 FF JMP $FFD2
```

Listing 4. Beispiel zur Anwendung der Routinen des TEDMON

Beispiel:

```
LDA #$D5
JSR $FB10
```

Auf dem Bildschirm erscheint »D5«.

PUTHXS ($FB05)

Diese Routine entspricht »PUTHEX«, allerdings wird nach der Hex-Darstellung des Bytes noch ein Leerzeichen ausgegeben.

MAKHEX ($FB20)

Die Routinen »PUTHEX« und »PUTHXS« bedienen sich dieser Routine.

Im Akku übergibt man »MAKHEX« den umzurechnenden Wert. Man erhält dann im Akku den ASCII-Code der ersten Stelle der zweistelligen Hex-Zahl, im X-Register die zweite Stelle als ASCII-Code zurück.

Beispiel:

```
LDA #$A5
JSR $FB20
```

Im Akku steht dann $41 (ASCII-Code von A), im X-Register $35 (ASCII-Code von 5).

PUTWRD ($FAFF)

Mit dieser Routine ist auch die Ausgabe von 2-Byte-Werten kein Problem. Im Akku wird das Low-Byte, im X-Register das High-Byte der Zahl, die vierstellig hexadezimal ausgegeben werden soll, übergeben.

Listing 4 wendet die aufgezeigten TEDMON-Routinen an.

Der letzte ROM-Einsprung, mit dem wir uns genauer beschäftigen wollen, ist

RESET ($FFF9)

»JMP $FFF9« hat die gleiche Wirkung wie das Drücken der Reset-Taste. Es erfolgt ein Sprung zum Reset-Vektor, wodurch der Einschaltzustand hergestellt wird.

Die Tabelle in Bild 1 zeigt Ihnen noch einmal alle ROM-Routinen zusammengefaßt. Viel Spaß und viel Erfolg bei deren Anwendung. (Florian Müller/ks)

Adresse	Label	Funktion
$8818	LPUPDT	LinkPointer aktualisieren
$D83B	SETCR	Cursor setzen
$D849	GETCR	Cursor holen
$D88B	CLEAR	Bildschirm löschen
$D89A	HOME	Cursor in HOME-Position
$DC49	BSOUTSCREEN	Zeichen auf Bildschirm ausgeben
$FAFF	PUTWRD	2-Byte-Zahl hexadezimal ausgeben
$FB05	PUTHXS	Bytewert hexadezimal mit Space ausgeben
$FB10	PUTHEX	Bytewert hexadezimal zweistellig ausgeben
$FB20	MAKHEX	ASCII-Code der Hex-Darstellung berechnen
$FF81	VIRES	Video-Chip und Editor initialisieren
$FF84	IORES	I/O-Bausteine initialisieren
$FF90	STMSG	Flag für System-/Fehlermeldungen setzen
$FFBA	SETLFS	File-Parameter setzen
$FFBD	SETNAM	File-Namen setzen
$FFC0	OPEN	File öffnen
$FFC3	CLOSE	File schließen
$FFC6	CHKIN	Eingabegerät setzen
$FFC9	CKOUT	Ausgabegerät setzen
$FFCC	CLRCH	Tastatureingabe und Bildschirmausgabe setzen
$FFCF	BASIN	Zeichen von Eingabegerät holen
$FFD2	BSOUT	Zeichen auf Ausgabegerät ausgeben
$FFD5	LOAD	Programm laden
$FFD8	SAVE	Programm speichern
$FFE1	STOP	Stop-Taste abfragen
$FFE4	GETIN	Zeichen eingeben
$FFE7	CLALL	alle offenen Kanäle schließen
$FFF0	PLOT	Cursor setzen (C=0) oder holen (C=1)
$FFF9	RESET	Software-Reset auslösen

Bild 1. Die Routinen und ihre Einsprünge. Die genaue Anwendung der Routinen entnehmen Sie bitte dem Artikel.

Für den Preis lohnt sich kein Selbstbau!

64K-Speicher-erweiterung 89.-

64K-Byte auf einem Steckmodul, daß nur in den Erweiterungsport eingesteckt werden muß. Kein Löten oder sonstige Arbeiten erforderlich.
Die Erweiterung wird voll von Basic- und Maschinenprogrammen unterstützt.

● NÜTZLICHES ZUBEHÖR:

Joystickadapter 11.50
Anpassung für jedes Joystick (oder Maus) an die C16-Steckernorm

● JOYSTICKS:

Quickshot I 11.90
Quickshot II 16.90
Quickshot IX 44.90
(Microschalter)
Quickgun III 37.90
(Microschalter)
Rauchglas-abdeckhaube 9.90
Formschöne Abdeckung aus rauchfarbenem Plexiglas. Passend für C16/C20/C64.
Datenrecorder 39.50
Kompatibel zur COMMODORE-Datasette.
Direkt an C20/64/128 anschließbar.
An C16/116/+4 über Cassettenportadapter.
Cassettenportadapter 12.50
Anschlußadapter für COMMODORE-Datasette und Kompatible an C16/116/+4.

● DISKETTEN + ZUBEHÖR:

No-Name 1D 14.90
10er Pack 5,25"
No-Name 2D 18.90
10er Pack 5,25"
Diskettenbox 19.90
80-100 Disk, mit Schloß und Rauchglasdeckel
Diskettenlocher 6.95
(Kunststoff)
Diskettenlocher 9.90
(Ganzmetall)
Aufklebetaschen 8.90
50iger Pack für DIR-Listings auf Disk-Hülle.

● EPROMs:

2764 250ns 6.90
27128 250ns 8.00
27256 250ns 14.90
nur 1. Wahl von Markenherstellern!

Achtung!! Wir liefern das komplette Zubehör für C64/C128, Amiga, Schneider und Atari, sowie elektronische Bauteile. Bitte fordern Sie unseren Katalog an!!!

Besuchen Sie unser Ladenlokal in Köln:

DELA Elektronik
Maastrichter Str. 23 · 5000 Köln 1
☎ 0221/51 70 81
Öffnungszeiten: Mo.-Fr. von 10 - 18.30, Sa. von 10 - 14 Uhr

Bestellungen (soweit vorrätig) bis 12 Uhr werden am selben Tag verschickt. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu unseren bekannten Verkaufs- und Lieferbedingungen.

Nachnahmeversand NN-Spesen 7.50 DM bei Vorkasse 3.- DM. Bei Auslandsbestellungen unter 30.- DM Lieferung nur bei Vorkasse auf Postgiroamt Köln (BLZ 370 500 00) Kto. 321095-501.

Vertrieb für die Schweiz: Achim Wengeler, Steinhauser Straße 44, CH 6300 Zug, Tel. 042/413926
Vertrieb für Österreich: ASM Engineering & Consulting, Wollgasse 39/9, A 1060 Wien, Tel. 0222/565241
Vertrieb für Italien: Firma Manhofer, Via Lorenzo il Magnifico 148, I 00162 Roma, Tel. 6/4270418
Vertrieb für Berlin: Elektronik A-Z, Stresemannstraße 95, 1000 Berlin

Vertrieb für Holland: GIMA-Printservice, Tel. 077/870937, Pb. 186-5900 AD-Venlo
Vertrieb für Belgien: Second-Software-Service, Diestestraat 133/8, 3000 Leuven
Vertrieb für Dänemark: D/C Trading, Sondergade 24, 9240 Nibe