

ERROR!#

Author : Erhard Pütz

Language: ACTION!

Im unten gezeigten Quellcode steht folgende Zeile:

```
INCLUDE "D:ERROR.LIB"
```

ERROR.LIB ist aus der original Runtime gekürzt. Da ist immer noch deutlich mehr drin, als das Programm unten braucht, aber es war das, was auf die Schnelle machbar war.

Wenn also aus dem unten gezeigten Quellcode ein fertiges COM-File werden soll:

- bitte eine Runtime oder Mini-LIBs (davon gibt es noch nicht alle) einbinden
- das kompilierte Programm abspeichern
- das Kompilat in z.B. BeWe-Softs Superpacker laden
- aus der INIT-Adresse eine RUN-Adresse machen
- CLR_80FF voranstellen mit INIT-Adresse voranstellen
- das Ganze als ERROR07.COM (oder ERROR.COM) abspeichern

CLR_80FF ist eine kleine Maschinenroutine, die die Speicheradressen \$80-\$FF nullt. Ich habe festgestellt, daß die nach dem Laden eines DOS sehr oft nicht Null sind, weil sie als beliebte freie ZP-Adressen von so allerlei Zeugs benutzt werden.

Allerdings werden sie auch von ACTION! und ACTION!-Programmen benutzt und wenn man nicht peinlich darauf achtet, daß man in seinem ACTION!-Programm alle Variablen sauber initialisiert, dann wird aus z.B.

```
WHILE a=0
```

schon mal eine Schleife, die überhaupt nicht ausgeführt wird.

Also hier noch mal ein Hinweis zur sauberen Variableninitialisierung:

Beispiel 1 (so ist es schlecht, weil die Variable nur beim Kompilieren gesetzt wird):

```
MODULE
BYTE I=[0]
...
```

Beispiel 2 (so ist es viel besser, weil die Speicherstelle bei jedem Programmstart gesetzt wird):

```
MODULE
BYTE I
...
PROC MAIN()
    I=0
...
RETURN
```

ERROR07.ACT

Und hier nun das, worum es eigentlich geht:

```
*****
;*
;* Dieses Programm soll numerische
;* Fehlermeldungen von DOS in den
```

```

;* entsprechenden Text umwandeln.      *
;*                                     *
;* Betriebssystem: Sparta DOS 3.x      *
;*                               BeWeDOS *
;*                                     *
;* Author      : Erhard Puetz         *
;* Web         : www.atari-central.de  *
;* Datum      : 2012-06-08            *
;* Version    : 0.7 vom 2012-08-11    *
;*                                     *
;* Dateiname:  ERRORxx.COM            *
;*           :  xx = Version          *
;*                                     *
;*****

```

MODULE

```

SET 14 = $4000 ;compiled program start address
SET $491 = $4000

```

```

BYTE RTS=[$60] ;dummy RTS as first byte

```

```

;-- Angepasste Runtime mit nur den Routinen,
;-- die das Hauptprogramm verwendet plus den
;-- Routinen, von denen andere abhängig sind.

```

```

INCLUDE "D:ERROR.LIB"

```

```

;--

```

MODULE

```

CHAR ARRAY VERSION = "Version 0.7"
CARD ARRAY MSG(43)

```

```

PROC SETERR()

```

```

MSG(00)="128 ($80) BREAK abort"
MSG(01)="129 ($81) IOCB already open"
MSG(02)="130 ($82) Non-existent device"
MSG(03)="131 ($83) Not open for read"
MSG(04)="132 ($84) Invalid IOCB command"
MSG(05)="133 ($85) Channel not open"
MSG(06)="134 ($86) Bad IOCB number"
MSG(07)="135 ($87) Not open for write"
MSG(08)="136 ($88) End of file"
MSG(09)="137 ($89) Truncated record"
MSG(10)="138 ($8A) Device timeout"
MSG(11)="139 ($8B) Device NAK"
MSG(12)="140 ($8C) (n/a)"
MSG(13)="141 ($8D) (n/a)"
MSG(14)="142 ($8E) (n/a)"
MSG(15)="143 ($8F) (n/a)"
MSG(16)="144 ($90) Device Done Error (Bad Sector, Disk write protected"
MSG(17)="145 ($91) (n/a)"
MSG(18)="146 ($92) Function not implemented in handler"
MSG(19)="147 ($93) (n/a)"
MSG(20)="148 ($94) Not a SpartaDOS disk"

```

```
MSG(21)="149 ($95) Disk not SpartaDOS 2.x"
MSG(22)="150 ($96) Directory not found"
MSG(23)="151 ($97) File (or directory) with same name exists"
MSG(24)="152 ($98) Not a binary file"
MSG(25)="153 ($99) (n/a)"
MSG(26)="154 ($9A) (n/a)"
MSG(27)="155 ($9B) (n/a)"
MSG(28)="156 ($9C) (n/a)"
MSG(29)="157 ($9D) (n/a)"
MSG(30)="158 ($9E) (n/a)"
MSG(31)="159 ($9F) (n/a)"
MSG(32)="160 ($A0) Drive number error"
MSG(33)="161 ($A1) (n/a)"
MSG(34)="162 ($A2) Disk full"
MSG(35)="163 ($A3) Illegal wild card in filename"
MSG(36)="164 ($A4) File erase protected"
MSG(37)="165 ($A5) File name error (illegal characters in filename)"
MSG(38)="166 ($A6) Position range error"
MSG(39)="167 ($A7) Cannot delete directory"
MSG(40)="168 ($A8) Illegal DOS command / not implemented"
MSG(41)="169 ($A9) Disk is write locked"
MSG(42)="170 ($AA) File not found"
```

```
RETURN
```

```
PROC FEHLER(BYTE I)
```

```
    BYTE ARRAY ERRMSG
```

```
    SETERR()
```

```
    IF I>127 AND I<171 THEN
```

```
        ERRMSG=MSG(I-128)
```

```
        PRINTE(ERRMSG)
```

```
    ELSE
```

```
        PRINTB(I)
```

```
        PRINT(": ")
```

```
        PRINTE("Nur Fehler von 128 bis 170")
```

```
    FI
```

```
    PUT(155)
```

```
RETURN
```

```
BYTE FUNC CHK_DOS()
```

```
    ;RETURNS DOS OK 0=NO 1=YES
```

```
    BYTE DFLAG=$700
```

```
    BYTE DVER=$701
```

```
    IF DFLAG='S AND DVER=$32 OR
```

```
        DFLAG='S AND DVER=$33 OR
```

```
        DFLAG='R AND DVER=$10 THEN
```

```
        RETURN(1)
```

```
    FI
```

```
    PRINTE("Not SpartaDOS 3.X, BW-DOS")
```

```
    PRINTE("or RealDOS 1.0")
```

```
RETURN(0)
```

```

PROC ZCRNAME()
;
;Diese Routine enthaelt keinen ACTION-Code.
;Die Adresse von ZCRNAME wird durch CRUNCH
;auf SpartaDOS's COMTAB+3 geaendert. Beim
;Aufruf von ZCRNAME wird dann der dortige
;Maschinencode ausgefuehrt.
;
RETURN ;Dummy RETURN, ML ends with RTS

CARD FUNC CRUNCH()
;
;Diese Routine dient zum Auslesen der Parameter,
;die in der Kommandozeile des DOS uebergeben wurden.
;
  BYTE I
  CARD COMTAB=$0A,
        COMFNAM,
        PAR
  BYTE POINTER BUFOFF,
              LBUF

  ZCRNAME=COMTAB+3
  BUFOFF =COMTAB+10
  COMFNAM=COMTAB+33+2
  LBUF   =COMTAB+63

  PAR=0

;Errechne Laenge der Eingabezeile -> "I"
  I=0
  WHILE LBUF^#155 DO
    I==+1
    LBUF==+1
  OD

  IF BUFOFF^ < I THEN
    ZCRNAME()
    PAR=VALB(COMFNAM)
    ;GET ALL PARMS AND PRINT THEM
    PRINTE(" ")
    WHILE BUFOFF^ < I DO
      ZCRNAME()
      PRINT("* ueberz. Parameter: ")
      PRINTE(COMFNAM)
    OD
  ELSE
    PRINTE(" ")
    PRINT("(c) 2012 Erhard Puetz, ")
    PRINTE(VERSION)
    PRINTE(" ")
    PRINTE("Syntax:")
    PRINTE(" ")
    PRINTE("ERROR num")
    PRINTE(" ")
    PRINTE("num   Eine Nummer on 128 - 170")
    PRINTE(" ")

```

```
PAR=$8000
FI
RETURN(PAR)

PROC MAIN()
;
;Programm startet hier.
;Bei gueltiger DOS Version Auslesen der
;Kommandozeile und Uebergabe der Fehlernummer
;zur Ausgabe der Fehlermeldung.
;
BYTE VALID,
    ERR_NO
CARD CRUNCH_RESULT

VALID=CHK_DOS()

IF VALID=1 THEN
    CRUNCH_RESULT=CRUNCH()
    IF CRUNCH_RESULT < 255 THEN
        ERR_NO=CRUNCH_RESULT
        FEHLER(ERR_NO)
    FI
FI
RETURN
```