

BESCHREIBUNG DES PASCALCOMPILERS FÜR Z9001

HERSTELLEN DER BETRIEBSBEREITSCHAFT

Top of RAM           Der über RAMTOP gelegene Speicherplatz ist geschützt.  
Bedingung: <65536  
Top of RAM for 'T'   Vorgabewert ist Top of RAM  
Table size?         Speicherplatz für Symboltabelle  
                    Vorgabewert ist Top of RAM/16  
                    Kaltstart mit dem Kennwort PASNEW  
                    Warmstart mit dem Kennwort PASOLD

BESONDERHEITEN GEGENÜBER STANDARDPASCAL

Vordefinierte Namen:

MAXINT           :CONST MAXINT=32767

Vordefinierte Prozeduren:

MARK(v1)         :RELEASE(v1) :Freigabe des Speicherplatzes dynamischer  
                  Variablen, die nach MARK(v1) eröffnet wurden.  
                  v1:Zeigervariable  
USER(v)          :Aufruf eines Maschinenprogrammes an der Adresse v  
                  Bedingung: -32767<=v<=32767  
HALT             :stoppt die Programmausführung (zur Fehlersuche)  
PAGE             :Bildschirmlöschen bzw. Blattvorschub  
POKE(x,y)        :Schreiben auf Speicherplätze  
                  x:Anfangsadresse (Typ INTEGER)  
                  y:Ausdruck (jeder Typ außer SET)  
TOUT(n,a,l)      :Auslagerung von Variablen auf Band  
                  n:Name (ARRAY(1..8) of CHAR)  
                  a:Anfangsadresse (Typ INTEGER)  
                  l:Länge (Typ INTEGER)  
                  Bsp. TOUT('VAR',ADDR(V),SIZE(V))  
TIN(n,a)         :Einlesen von Variablen, die mit TOUT gespeichert  
                  wurden.  
                  n:Name  
                  a:Anfangsadresse (Typ INTEGER)  
                  Bsp. TIN('VAR',ADDR(V))  
OUT(p,c)         :Portausgabe  
                  p:Adresse (Typ INTEGER)  
                  c:Parameter (Typ CHARAKTER)  
WRITE(e,m,H)     :Bildschirmausgabe  
                  e:Ausgabegröße (bei Hexausgabe Typ INTEGER)  
                  m:Länge  
                  H:Kennzeichen für Hex-Ausgabe

vordefinierte Funktionen:

RANDOM           :Pseudozufallszahl 0<=x<=255  
ADDR(V)          :Ermittlung der Speicheradresse der Variablen V  
                  (Typ INTEGER)  
PEEK(x,t)        :Lesen von Speicherzellen  
                  x:Speicheradresse (Typ INTEGER)  
                  t:Typ, den die Fkt. liefern soll (alle Typen möglich)  
SIZE(V)          :Anzahl der durch die Variable V belegten Bytes  
INP(p)           :Porteingabe (Typ CHAR)  
                  p:Portadresse (Typ INTEGER)  
INCH             :Tastaturabfrage (Typ CHAR)  
ENTIER(x)        :größte Integerzahl<=x

Allgemeines:

-Bereich von REAL-Zahlen: 5.9E-39<=x<=3.5E38 nicht überschrei-  
ten, weil eine Zahlenkonvertierung erfolgen kann, für die  
keine Fehlermeldung ausgegeben wird.  
-Der Vergleich von Zeigern mittels '<>' führt immer zu TRUE.  
  Statt dessen NOT( = ) verwenden.  
-keine FILE-Implementierung vorhanden  
-RECORD-Typen dürfen keinen VARIANT-Teil enthalten.  
-PROCEDURES und FUNCTIONS sind als Parameter nicht zugelassen.  
-STRING: Länge<=256  
-SET: Grundtyp<=256 Elemente

- leere CASE-Anweisungen führen zu Fehlermeldungen.
- Case-anweisungen können mit ELSE oder END abgeschlossen werden
- 'Eckige Klammer auf' wird durch 'SHIFT 0' erzeugt
- 'Eckige Klammer zu' wird durch '@' oder 'SHIFT @' erzeugt
- '0' wird durch '^' oder 'SHIFT ^' erzeugt
- Hexadezimalzahlen werden durch '\*' gekennzeichnet (Bsp.:\*A2FF)

## EDITOR-KOMMANDOS

---

- I n,m :automatische Zeilennumerierung  
n:Anfangswert  
m:Schriftweite
- L n,m :Listen von n bis m (Standartwert n=1,m=MAXINT)
- K n :Zeilenzahl von LIST (Standartwert n=15)
- D (n,m) :Zeilen löschen von n bis m  
Löschen einzelner Zeilen durch n=m oder Eingabe der Zeilennummer und ENTER
- M n,m :Kopieren von Zeile n nach m (n bleibt erhalten)
- N (n,m) :Neue Zeilennumerierung  
n:erste neue Zeilennummer  
m:Zeilenabstand
- F n,m,f,s :Suchen des String f in n<x<m  
beim Finden:Unterbrechung, Übergang in den EDIT-Mode und Ersetzen von f durch s mittels EDITOR-Befehl S möglich
- E n :Edit-Mode von Zeile n  
Q :bricht den Editor ab  
R :Rückholung der alten Zeile  
L :Rücksprung zum Zeilenanfang und Listen der gesamten Zeile  
K :Zeichen löschen  
Z :löschen aller rechtsstehenden Zeichen  
F :sucht das nächste Auftreten vom vorher mit F n,m,f,s definierten String f  
S :ersetzt den gefundenen String f mit s  
I :Zeichen einfügen  
X :Spung zum Zeilenende und Übergang in den I-Modus  
C :Zeichen überschreiben
- P n,m,s :Der Zeilenbereich von n bis m wird unter dem File-namen s auf Band gespeichert.
- G ,,s :Suchen eines Files mit dem Namen s auf Band  
Beim Finden eines falschen Namen Drucken von:'FOUND' mit Namen und Fortsetzen der Suche.
- C n :Compilieren von n ab (Standartwert: kleinste Zeilennummer); bei Fehlern wird das Compilieren unterbrochen und eine Fehlernummer ausgegeben.  
Mit E kann die Zeile, mit P die vorige Zeile in den Edit-mode übernommen werden. Alle anderen Tasten setzen das Compilieren fort.  
Bei richtiger Compilierung erscheint RUN? , das -mit 'Y' beantwortet- den Programmstart bewirkt.  
Alle anderen Tasten führen in den EDITOR.
- R :Starten des übersetzten Programmes
- T n :Compilieren des Quelltextes;  
Bei erfolgreicher Übersetzung erscheint OK?.  
Wenn diese Frage mit 'Y' beantwortet wird, wird der Objektcode an das Ende der RUNTIMES (enthalten arithmetische u.a. Routinen, die zum Laufen compilierter Programme notwendig sind) verschoben und mit diesen auf Band gespeichert.Der Compiler wird dabei zerstört! Jede andere Taste führt in den EDITOR.

## RUNTIME-FEHLERMELDUNGEN

---

Auftreten von Fehlern während des Programmlaues in der Form: 'at PC=xxxx'; wobei xxxx die Speicheradresse ist, an der der Fehler auftrat. Runtimefehler stoppen die Programmausführung.  
Fehlerliste:

1. Halt
2. Überlauf
3. Out of RAM
4. Division durch Null

5. Index zu niedrig
6. Index zu hoch
7. Fehler beim Aufruf einer mathematischen Routine
8. Zahl zu groß
9. eine Zahl wurde erwartet
10. die Zeile ist zu lang
11. ein Exponent wurde erwartet

#### FEHLERMELDUNGEN, DIE DURCH DEN COMPILER ERZEUGT WERDEN

---

1. die Zahl ist zu groß
2. ein Semikolon wird erwartet
3. nichtdeklarieter Name
4. ein Name wird erwartet
5. in der Deklaration einer Konstanten muß '=' und nicht ':=' stehen
6. '=' wird erwartet
7. eine Anweisung darf nicht mit diesem Namen beginnen
8. ':=' wird erwartet
9. ')' wird erwartet
10. falscher Typ
11. '.' wird erwartet
12. ein Faktor wird erwartet
13. eine Konstante wird erwartet
14. dieser Name ist keine Konstante
15. 'THEN' wird erwartet
16. 'DO' wird erwartet
17. 'TO' oder 'DOWNTO' wird erwartet
18. '(' wird erwartet
19. dieser Ausdruck kann nicht geschrieben werden
20. 'OF' wird erwartet
21. ',' wird erwartet
22. '::' wird erwartet
23. 'PROGRAM' wird erwartet
24. eine Variable wird erwartet
25. 'BEGIN' wird erwartet
26. bei diesem READ-Aufruf wird eine Variable erwartet
27. Ausdrücke dieses Typs können nicht verglichen werden
28. eine INTEGER- oder REAL-Zahl wird erwartet
29. dieser Variablentyp kann nicht gelesen werden
30. dieser Name ist kein Typ
31. bei reellen Zahlen wird ein Exponent verlangt
32. ein skalarer Ausdruck (kein numerischer) wird erwartet
33. Leerstrings sind nicht erlaubt
34. 'eckige Klammer auf' wird erwartet
35. 'eckige Klammer zu' wird erwartet
36. ein ARRAY-Index muß vom Typ SKALAR sein
37. '..' wird erwartet
38. in der Deklaration eines ARRAYs wird 'eckige Klammer auf' und nicht ',' verlangt
39. die Obergrenze ist größer als die untere
40. SET muß <=256 sein
41. das Ergebnis der Funktion muß vom Typ Name sein
42. im SET wird ',' oder 'eckige Klammer zu' erwartet
43. im SET wird '..' oder ',' oder 'eckige Klammer zu' erwartet
44. der Parameter muß vom Typ Name sein
45. ein leeres SET kann nicht erster Faktor in einer Nicht-Zuweisungs-Anweisung sein
46. ein Skalartyp (einschließlich reellen Zahlen) wird erwartet
47. ein Skalartyp (außer reellen Zahlen) wird erwartet
48. die SET's sind nicht verträglich
49. SET's können nicht mit '<' oder '>' verglichen werden
50. 'FORWARD', 'LABEL', 'CONST', 'VAR', 'TYPE' oder 'BEGIN' werden erwartet
51. eine Hexadezimalziffer wird erwartet
52. SET's können nicht mit POKE gespeichert werden
53. das ARRAY ist zu groß
54. bei der Definition von RECORD's wird END oder ';' erwartet
55. ein Feldname wird erwartet
56. nach WITH wird eine Variable erwartet
57. die Variable bei WITH muß vom RECORD-Typ sein
58. der Feldname wurde nicht mit der WITH-Anweisung in Verbindung gebracht
59. nach LABEL wird eine vorzeichenlose INTEGER-Zahl erwartet
60. nach GOTO wird eine vorzeichenlose INTEGER-Zahl erwartet
61. diese Marke ist in einer falschen Programmebene
62. die Marke ist nicht vereinbart

- 63. der Parameter von SIZE muß eine Variable sein
- 64. auf Zeiger kann nur der Gleichheitstest angewendet werden
- 67. der einzig zulässige WRITE-Parameter für INTEGER-Zahlen mit zwei Doppelpunkten ist e:m:H
- 68. Strings dürfen keine Zeilenendezeichen enthalten
- 69. der Parameter von NEW, MARK oder RELEASE muß eine Zeigervariable sein
- 70. der Parameter von ADDR muß eine Variable sein

AUSZUG AUS OS-DOKUMENTATION KC 85/1  
\*\*\*\*\*

## 1.2. Monitorbestandteile

---

### 1. Console Command Programm (CCP)

---

-ähnlich CP/M-80

- Fkt.: -Steuerung des KC im OS-Mode
  - Kdo.- und Parameterübernahme
  - Verzweigen zur Kdo.-ausführung, Verwaltung der Gerätetreiber
  - Ausgabe von Fehlernachrichten
- Inhalt: -Monitorgrundschleife zur Kdo.-anforderung
  - Kdo.-entschlüsselung und -ausführung
  - Subroutinen zur Kdo.- und Parameterübernahme und deren Prüfung, Speichertest und Speichermodifikation, Test auf zulässige Geräteadressierung bei Zeichen-E/A
  - allgemeine Hilfsroutinen
- Eintritt: -automatisch nach:
  - Einschalten und RESET
  - Warmstart JMP 0, Systemruf "0", "STOP" im OS-Mode
  - Rückkehr von Kdo.-Ausführung
  - Rückkehr aus Anwenderprogrammen

### 2. Basic Operating System (BOS)

---

- Fkt.: -Steuerprogramm für Ausführung von Systemrufen und deren Prüfung
  - Übernahme von Parametern
  - Verzweigen zur Ausführung
  - Rückgabe von Parametern
  - Fehlermeldung
- Inhalt: -zentraler Eintrittspunkt zur Rufnummerentschlüsselung, Parameterübernahme und Registerrettung
  - Adressvektor zur Adressierung der ausgewählten Systemprogramme
  - Systemprogramme zur String- und Kassetten- E/A
  - Fehlerbehandlung
  - zentraler Austrittspunkt mit Rückgabe von Parametern Ausgabe von Fehlermeldungen und Registerregenerierung
- Eintritt: -nach CALL 0005
  - Rufnummer in C (0...33)
  - Parameter in DE bzw. E
- Austritt: -Parameter in BC bzw. A

### 3. Basic I/O System (BIOS)

---

- Fkt.: -übersetzt Rufe aus CCP und BOS in Unterprogramme zur Einzelzeichen-E/A
  - verbindet logische und physische Geräte und übergibt die Zeichen an die entsprechenden Treiberroutinen
  - übergibt übergeordnetem Steuerprogramm Parameter und spezielle Fehlermeldungen
- Inhalt: -Sprungvektor zur Direktadressierung spezieller Monitorprogramme
  - Programme zur Initialisierung des KC bei Warm- und Kaltstart
  - Programme zur Einzelzeichen-E/A
  - Standardtreiber für Tastatur, Bildschirm und Kassetten-E/A

- spezielle Unterprogramme zur Ausführung von System- und Direktrufen (teilweise von BOS mit benutzt)
- Hilfsprogramme (z.T. für CCP und BOS)

Eintritt: -nach CALL 0F000H+(n\*3) (n=0...22)

- Parameter in BC bzw. C

Austritt: -Parameter in BC bzw. A

### 1.3. Speicheraufteilung

-----

- abweichend von CP/M-80:
  - Systemarbeitszellen 0...0FFH
  - interner Puffer und Stackbereich 100H...2FFH
  - erste von Anwender nutzbare Adresse 300H
- auf RESTART-Adressen jeweils drei Byte zur Aufnahme eines Sprungbefehls frei, um Arbeit mit RST-Befehlen zu sichern, Ausnahme: adr 0...2 mit Sprung zu Warmstart belegt -->
  - RST 0 = Steuerung an CCP; adr 66H (NMI) wird bei Kassetten-E/A genutzt,
- im nichtgenutzten BWH-Speicher befinden sich Systemzellen für Systemschutz und aktuelle Gerätekonfiguration (0EF00)

RAM - BELEGUNG  
\*\*\*\*\*

Adresse Hex Dez	Start Hex Dez	
00	0	JMP WBOOT
03	3	BER 1 frei
04	4	IOBYT: DB 0; I/O-Byte zur Zuweisung log.-->phys. Gerät
05		JMP BOS ; Sprung zum BOS-Eintritt
08		BER 3 frei für RST 8
0B	11	SPSV: DA 0 Merkregister für aktuellen Anwender- SP bei Eintritt in BOS
0D	13	BCSV: DA 0 "=" für BC
0F	15	ASU: DB 0 "=" für A
10	16	BER 3 frei für RST 10
13	19	IOYR: DB 0 Puffer für Datenwort Joystick 1 (rechts)
14	20	IOYL: DB 0; "=" (links)
15	21	LISM: DB 0 Schalter für Hardcopy - 0=aus, 1= Copy auf akt. LIST-Gerät
16	22	BSW: DB 0 Schalter für Tastaturbeep, 0=aus, 1=an
17	23	CLOSM: DB 0 Merzelle für Farbsteuerzeichen
18	24	BER 3 frei für RST 18
1B	27	DMA: DA 80H ; Pointer auf Puffer für Kassetten E/A
1D	28	HOUR: BER 1 Puffer für Stunden
1E	30	MIN: BER 1 "=" Minuten
1F	31	SEC: BER 1 "=" Sekunden
20	32	BER 3 frei für RST 20H
23	35	COUNT: DB 0; Zähler zur Tastaturentprellung und REPEAT-Fkt. der Tastatur
24	36	LAKEY: DB 0 letztes Zeichen von Tastatur
25	37	KEYBU: DB 0 Tastaturpuffer
26	38	SHLOCK: DB 0 Schalter für SHIFT-LOCK
27	39	ATRIB: DB 2 akt. Farbattribut
28	40	BER 3 frei für RST 28H
2B	43	CHARP: DB 1 akt. Spalte Cursor
2C	44	LINEP: DB 1 akt. Zeile Cursor
2D	45	CURS: DA 0EC00H akt. phys. Adresse Cursor
2F	47	PU: BER 1 ; Hilfszelle
30	48	BER 3 ; frei für RST 30
33	51	WORKA: BER 1; Hilfszelle
34	52	BUFFA: BER 1 Puffer für Attribut des vom Cursor überlagerten Zeichens
35	53	PU: BER 1 Hilfszelle
36	54	EOR: BER 2 Pointer auf letzte für Anwender freie Adresse
38	56	BER 3 frei für RST 38H
3B	59	P1ROL: DB 0 erste zu rollende Zeile-1
3C	60	P2ROL: DB 0 letzte zu rollende Zeile+1
3D	61	P3ROL: DB 0 erste zu rollende Spalte-1
3E	62	P4ROL: DB 0 letzte zu rollende Spalte+1
3F	63	BUFF: BER 1; Puffer für vom Cursor überschriebenes Zeichen
40	64	PARBU: BER 2 Hilfszelle für Parameterüberprüfg.
42	66	BER 27 frei für DEBUGGER, ... IDAS usw.
5C	92	FCB: BER 36 File Control Block FNAME: EQU FCB ; Programmname FTYP: EQU FCB+8 ; Programmtyp 3 Byte AADR: EQU FCB+17 ; Anfangsadresse des Programms EADR: EQU FCB+19 ; Endadresse "=" SADR: EQU FCB+21 ; Startadresse "=" SYSBY: EQU FCB+23 ; Systemschutzbyte 0=ungeschützt, 1=System vor WRITE geschützt, rücksetzen nur durch RESET
80	128	CONBU: DB 80; Consolepuffer und Standardpuffer für Kassetten-E/A
81	129	DB 0 akt. Länge des Eingabe-String
82	130	BER 80
D2	210	BER 46 frei
100	256	INTLN: DB 0 interner Stringpuffer, akt. Stringlänge
101	257	BER 80 BOSSP: EQU 1C0H; Stack-Top für BOS CPSP: EQU 20H; "=" für CCP INTV: EQU 200H; Interruptvektor ORG INTV
200	512	DA IKAKT Kassette schreiben
202	514	DA 0 frei für Anwender CTC Kanal 1
204	518	DA ICTC Entprellen Tastatur Verteiler Systemuhr
206	520	DA INUHR ; Sekundentakt Uhr
208	520	DA INTP Tastaturinterrupt
20A	522	DA IKEP Kassette lesen
20C	524	BER 74H frei für Anwender Interrupt
280	768	SYSP: BER 80H für Erweiterung Systempointer, z.B. LIST-Driver

BILDWIEDERHOLSPEICHER  
\*\*\*\*\*

Adresse Hex	Start Dez	
EFC0	61376	SYSSB: DB 0; Systemschutzbyte, 0=kein Schutz, 1=vor WRITE geschützt, Rücksetzen nur durch RESET
EFC1	61377	MAPAR: BER 8 MAPPING-AREA für Speicherkonfiguration, ein Bit je 1k Speicher, 1=RAM, 0=ROM/WRITE-gesch.
EFC9	61385	ATTYC: DA -1 Adressentabelle für Gerätetreiber
EFCB	61387	ACRTC: DA CRT CRT-Driver für CONST
EFCD	61389	ABATC: DA BAT BAT- "="
EFCF	61391	AUC1: DA -1 UC- "="
EFD1	61393	ATTYR: DA -1 TTY-Driver für READ
EFD3	61395	ARDR: DA -1 RDR- "="
EFD5	61397	AUR1: DA -1 UR1- "="
EFD7	61399	AUR2: DA -1 UR2- "="
EFD9	61401	ATTYP: DA -1 TTY-Driver für PUNCH
EFDB	61403	APUN: DA -1 PUN- "="
EFDD	61405	AUP1: DA -1 UP1- "="
EFDF	61407	AUP2: DA -1 UP2- "="
EFE1	61409	ATTYL: DA -1 TTY-Driver für LIST
EFE3	61411	ALST: DA -1 LST- "="
EFE5	61413	ACRTL: DA -1 CRT- "="
EFE7	61415	AUL: DA -1 UL- "="
EFE9	61417	TXCON: DA PHYDU+2 Adresse eines Strings des aktuellen phys. Consolegerätes, String wird im ASGN-Kdo. nach dem log. Gerätenamen ausgegeben, String wird auf CONST ausgegeben
EFEB	61419	TXRDR: DA PHYDU analog TXCON
EFED	61421	TXPUN: DA PHYDU analog TXCON
EFEF	61423	TXLST: DA PHYDU analog TXCON

## 2.1. Console Command Programm

- realisiert Anforderung Übernahme und Prüfung aller Kdo's, residente Kdo's werden durch CCP selbst ausgeführt; wird ein Kdo eingegeben, daß weder resident noch transient ist, wird es als Name eines Anwenderprogramms interpretiert. Das Programm wird dann von Kassette eingelesen und entsprechend Information im File Control Block (FCB) plaziert und gestartet (s.a. 2.2.). Bei Rückkehr mit Fehler in Monitorgrundschleife wird entsprechend Fehlercode eine Fehlerausschrift gegeben (s. 4.)
- CCP kann vom Anwender umgangen werden, indem im Speicher das transiente Kdo "\*" abgelegt wird --> Initialisierungsroutine WBOOT verzweigt zur Ausführung des Kdo's unter Umgehung des CCP (=Abschaltung des CCP) und Steuerung durch eigenes Programm
- eigener Stackbereich ab 200H, Länge 40H, ist gleichzeitig Stackbereich eines Anwenderprogramms, sofern kein eigener Stack eingerichtet wird; für Anwender verfügbare Stacktiefe 3CH, Stacktop mit Eintritt in Anwenderprogramm ist SP=1FCH, auf Adresse 1FCH steht Adresse der Fehleranzeige (ERDIS), auf 1FFEh Adresse des CCP (GOCP)

### 2.1.1. Residente Kommandos

- TIME
- CLOAD
- ASGN --> Zuweisung physischer Geräte und deren Treiber zu logischen Geräten des KC
- \*logische Geräte: CONST: interaktives Consoleaggregat  
Tastatur/BS  
READER: Eingabegerät  
PUNCH: Ausgabegerät  
LIST: Ausgabegerät, wird mit >CONTR P< dem Gerät CONST parallel geschaltet
- \*physische Geräte: CRT: Standardtastatur und BS, auch für LIST zulässig  
BAT: Batch-Mode der Concole, nur für CONST; Eingabegerät ist der aktuelle READER, Ausgabegerät der

aktuelle LISTER; weitere physische Geräte unter Angabe eines Programmnamens beliebig wählbar; angegebenes Programm wird eingelesen und als Treiber für ein physisches Gerät interpretiert; je logisches Gerät sind gleichzeitig 4 physische Geräte wählbar; Verbindung log.-->phys. Gerät erfolgt über I/O-Byte (04H)

Aufbau des I/O-Byte:

-Unterteilung in 4 Bereiche je 2 Bit, jeder Bereich einem log. Gerät zugeordnet

Bit	7	6	5	4	3	2	1	0	I/O-Byte
									LIST PUNCH READER CONST

-durch Monitor realisiert: CONST: 1-->CRT zugewiesen

2-->BAT zugewiesen

LIST: 1-->CRT zugewiesen

-zugehörige Treiberadressen in Adressentabelle für Gerätetreiber (0EFC9H-0EFE8H)

-bei Anzeige der Gerätezuweisung wird nach Gerät ein String ausgegeben, der die Zuweisung verdeutlicht

-Adresse des auszugebenden String:

0EFE9H für CONST  
 0EFEAH für READER  
 0EFECH für PUNCH  
 0EFEFH für LIST

-Anzeige der aktuellen Gerätezuweisung:

ASGN >ENTER<

-Console wird Standardtastatur und BS zugewiesen:

ASGN CONST:=CRT

-Console-E/A über READER und LIST:

ASGN CONST:=BAT

-->READER und LISTER müssen vorher zugewiesen sein, ansonsten Fehler !!

-LIST auf BS:

ASGN LIST:=CRT

-Anwendergeräte:

ASGN log. Gerät:=name.typ

-dem Monitor sind Initialisierungsparameter zurückzuliefern

## 2.2. Basic Operating System

---

-Steuerprogramm zur Ausführung spezieller Systemrufe des Anwenders; macht Ausführung komplexer Funktionen durch Monitor möglich, ohne interne Schnittstellen zu kennen, Aufruf generell unter CALL 0005; Auswahl des gewünschten Systemrufes über C-Register, dessen Inhalt den Systemruf adressiert; einige Systemrufe erwarten Eingabeparameter und/oder liefern Ausgabeparameter zurück:

- a) Eingabeparameter Bytewert in E  
 Wortwert in DE
- b) Ausgabeparameter Bytewert in A  
 Wortwert in BC

-bei Fehler Rückkehr mit CY=1 und Fehlercode in A

-alle Register werden gerettet, sofern sie nicht Parameter oder Fehlercode übermitteln

-eigener Stack des BOS ab 1C0H



### 2.2.1. BOS - Jumpvektor (ab 0F045H)

Ruf-Nr.	Name	Bemerkungen
0	INIT	Kaltstart, volle Initialisierung wie RESET
1	CONSI	Eingabe eines Zeichens von Console
(F047)		Ausgang Zeichen in A
2	CONSO	Ausgabe eines Zeichens zu Console
(F049)		Eingang Zeichen in E
3	READI	Eingabe eines Zeichens vom READER
(F04B)		Ausgang Zeichen in A
4	PUNO	Ausgabe eines Zeichens zu PUNCH
(F04D)		Eingang Zeichen in E
5	LISTO	Ausgabe eines Zeichens zu LIST
(F04F)		Eingang Zeichen in E
6	GETS	Eingabe von Joystick
(F051)		Ausgang Joystick links in B Joystick rechts in C
7	GETIO	Lesen I/O-Byte
(F053)		Ausgang I/O-Byte in A
8	SETIO	Setzen I/O-Byte
(F055)		Eingang I/O-Byte in E
9	PRNST	Stringausgabe zu CONSOLE
(F057)		Eingang \$-Adresse in DE
10	RCONB	\$-Eingabe von CONSOLE
(F059)		Eingang \$-Adresse in DE
11	CSTC	Abfrage CONSOLE-Status
(F05B)		Ausgang Status in A
12	RETVN	Abfrage Versionsnummer des Monitors
(F05D)		Ausgang Versionsnummer in BC
13	OPENR	OPEN für Kassette lesen
(F05F)		Eingang Name und Typ im FCB
14	CLOSR	CLOSE für Kassette lesen
(F061)		Ausgang Adresse des File-Adressfeldes in BC
15	OPENW	OPEN für Kassette schreiben
(F063)		Eingang Name, Typ, Adressfeld im FCB Ausgang Nummer des geschriebenen Blocks in A (00)
16	CLOSW	CLOSE für Kassette schreiben
(F065)		Ausgang Nummer des geschriebenen Blocks in A (FF)
17	GETCU	Abfrage der logischen Cursoradresse
(F067)		Ausgang Zeile in B (0 ... 23) Spalte in C (0 ... 39)
18	SETCU	Setzen logische Cursoradresse
(F069)		Eingang Zeile in D, Spalte in E
19	BOSER	nicht belegt
(F06B)		
20	READS	Lesen eines Blocks von Kassette
(F06D)		
21	WRITS	Schreiben eines Blocks auf Kassette
(F06F)		Ausgang Nummer des Geschriebenen Blocks in A (01 ... FE)
22	SETTI	Stellen Systemuhr
(F071)		Eingang Stunden in A, Minuten in B, Sekunden in C
23	GETTI	Abfragen Systemuhr
(F073)		Ausgang Stunden in A ... (s.o.)
24	PRITI	Ausgabe der Systemuhrzeit
(F075)		Eingang Adresse des Ausgabebereiches in DE
25	INITA	Initialisieren der Tastatur
(F077)		keine Parameter
26	SETDM	Setzen Pufferadresse für Kassetten-E/A
(F079)		Eingang Pufferadresse in DE
27	GETM	Abfragen der Speicherkonfiguration
(F07B)		Eingang zu prüfende Adresse in DE Ausgang Belegung der Adresse in A, 0=PROM/ leer/geschützt, 1=RAM
28	SETM	Setzen der Speicherkonfiguration
(F07D)		Eingang zu belegende Adresse in DE, Belegung der Adresse in A, 0=geschützt, 1=RAM
29	DCU	Löschen Cursor
(F07F)		keine Parameter
30	SCU	Setzen Cursor
(F081)		keine Parameter
31	COEXT	Steuerzeichen aus String entfernen und Anhängen von 00
(F083)		Eingang \$-Pufferadresse in DE
32	BOSER	nicht belegt
(F085)		
33	RAND	Lesen eines beliebigen Blocks von Kassette
(F087)		