

BESCHREIBUNG und
BEDIENUNGSANLEITUNG

INHALT

1. EINLEITUNG
2. ORGANISATION DES SYSTEMS
 - 2.1 Funktionen
 - 2.2 Speichereinteilung
 - 2.3 Tabellen des Betriebssystems
3. PROGRAMMIERUNG DES SYSTEMS
 - 3.1 Allgemeines
 - 3.2 Betriebssystem-Steuerbefehle
 - 3.3 Programmier-Vorschriften
 - 3.4 Beispiele
4. BEDIENUNG DES SYSTEMS
 - 4.1 Inbetriebnahme
 - 4.2 Bedienung
 - 4.3 Monitor
5. GENERIERUNG EINES ANWENDER SYSTEMS
6. TESTSYSTEM
7. DIENSTPROGRAMME

1. Ausgabe Oktober 1976
2. Ausgabe Dezember 1977

1. EINLEITUNG

Das Betriebssystem (BS) für das Modell TA1Ø69 ist als Mehrprogramm-Betriebssystem ausgelegt. Es ermöglicht den simultanen Ablauf von mehreren Anwenderprogrammen.

Im Betriebssystem sind alle Betriebsmittel zusammengefaßt, die für

- das Verarbeiten und Sichern von Dateien,
- das Erstellen und Testen von Programmen und
- den Ablauf von Programmen

notwendig sind.

Das Betriebssystem ist plattenorientiert, d.h., Datenträger der Systemkomponenten ist eine Magnetplatte.

Im Betriebssystem sind Modifikationen möglich, die über Zusatzsoftware-Anschlüsse eine individuelle Anpassung an das jeweilige Einsatzgebiet erlauben.

2. ORGANISATION DES SYSTEMS

2.1 Funktionen

Das Betriebssystem (BS) umfaßt eine Anzahl von Systemkomponenten, die sich in folgende Gruppen einteilen lassen:

Firmware:

- Ablaufteil
- BS-Routinen

Systemsoftware:

- Organisationsprogramm
- Testsystem
- Dienstprogramme
- Zusatzsoftware

2.1.1 Ablaufteil

Im Ablaufteil werden die Programmbefehle interpretiert und die gewünschten Operationen ausgeführt. Simultan dazu werden die peripheren Geräte gesteuert und E/A-Unterbrechungen erzeugt.

Weiterer Bestandteil des Ablaufteils ist die Unterbrechungsanalyse.

Hier werden alle aufgelaufenen Unterbrechungen untersucht und eine entsprechende Unterbrechungsbehandlung eingeleitet.

Eine Unterbrechung bewirkt in der Regel eine Prozeßumschaltung.

2.1.2 BS-Routinen

Die wichtigsten BS-Routinen sind:

- Routinen für die Unterbrechungsbehandlung,
- der Prozeßumschalter (PUM),
- SVC-Routinen;

Wichtigste Aufgabe der Unterbrechungsbehandlung ist u.a., blockierte Prozesse zu deblockieren, damit sie ihren Programmlauf wieder fortsetzen können. Ein Prozeß ist blockiert, wenn er z.B. auf das Fertigwerden einer E/A-Operation wartet (Befehl WAIT).

Folgende Unterbrechungen (Interrupts) können auftreten:

- E/A-Unterbrechung (Fertigwerden einer E/A-Operation)
- Zeit-Unterbrechung (TIMER abgelaufen)
- Externe Unterbrechung (DFÜ-Nachrichten angekommen)
- Anruf-Taste (Monitor-Anruf)
- Maschinenanlauf (nach dem Einschalten)
- System-Unterbrechung (Internfehler, Gerät belegt;)

Der Prozeßumschalter hat die Aufgabe, den höchst-prioren aller deblockierten Prozesse (dynamischer Status = "bereit") zu aktivieren, d.h., an das laufende System hinzuzuschalten. Sämtliche dazu notwendigen Informationen sind im dazugehörigen Prozeßkontrollblock (PCB) des zu aktivierenden Prozesses hinterlegt.

Mit den SVC-Routinen werden den Anwenderprozessen verschiedene Systemdienste zur Verfügung gestellt, welche die spezifischen Eigenschaften des Mehrfach-Betriebssystems berücksichtigen.

Solche Systemdienste sind u.a.:

- Nachrichtenaustausch unter verschiedenen Programmen;
- Warten auf Ereignisse wie z.B. Timer abgelaufen oder DFÜ-Nachrichten eingetroffen;
- Abfragen Gerätezustand;

2.1.3 Organisationsprogramm

Das Organisationsprogramm im Betriebssystem ist der MONITOR. Der MONITOR übernimmt u.a. das Einrichten der Speicherbereiche, das Laden, Starten und Beenden von Programmen und zeigt Status- und Fehlerhinweise an. Die gewünschten Funktionen müssen vom Bediener in Form von Monitor-Aufrufen eingegeben werden.

2.1.4 Testsystem

Das Testsystem erlaubt das Testen von beliebigen Anwenderprogrammen im Mehrfachbetrieb. Das Testsystem kann jederzeit durch einen Monitoraufruf hinzugeschaltet werden. Die genaue Beschreibung ist im Kapitel 'Testsystem' nachzulesen.

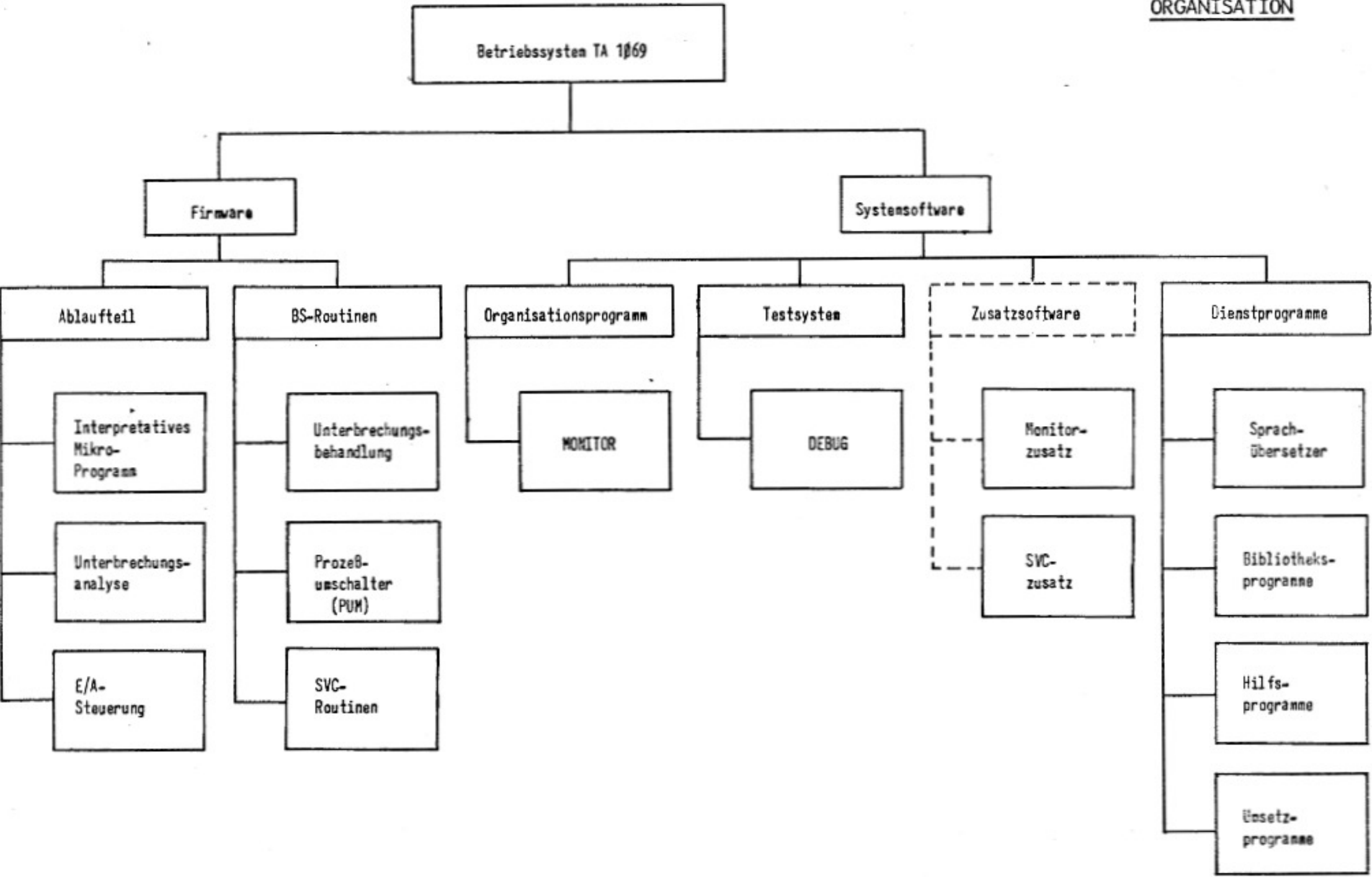
2.1.5 Zusatzsoftware

Standardmäßig ist das Betriebssystem ohne Zusatzsoftware ausgerüstet. Die Zusatzsoftware erlaubt eine individuelle Anpassung an spezielle Einsatzgebiete.

Es sind Modifikationen der Monitorfunktionen und der Anschluß zusätzlicher SVC-Routinen möglich. Nähere Informationen sind in einer getrennten Beschreibung nachzulesen.

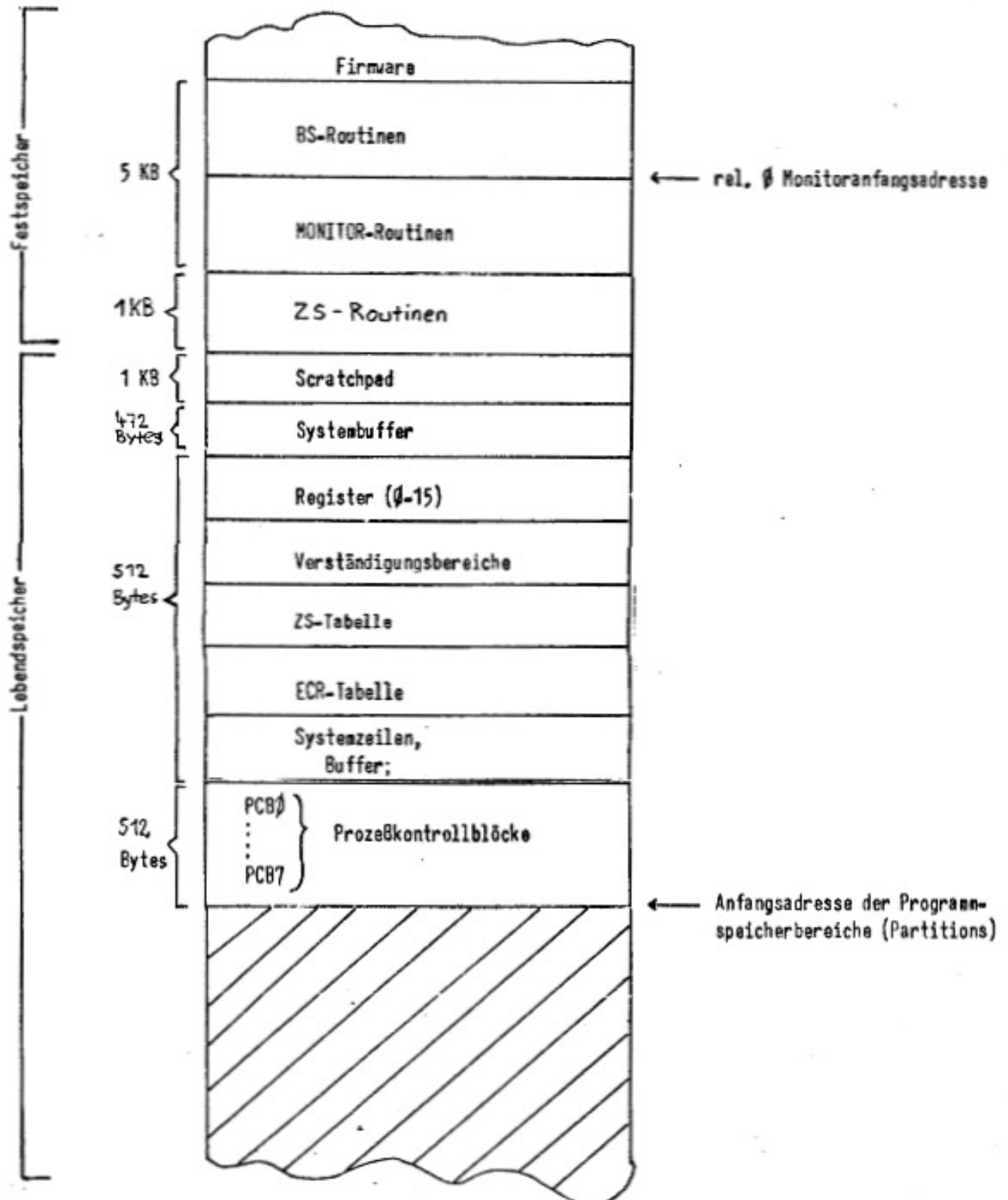
2.1.6 Dienstprogramme

Die Dienstprogramme unterstützen das Erstellen und Warten von Programmen und Dateien. Weitere Informationen sind im Kapitel 'Dienstprogramme' nachzulesen.



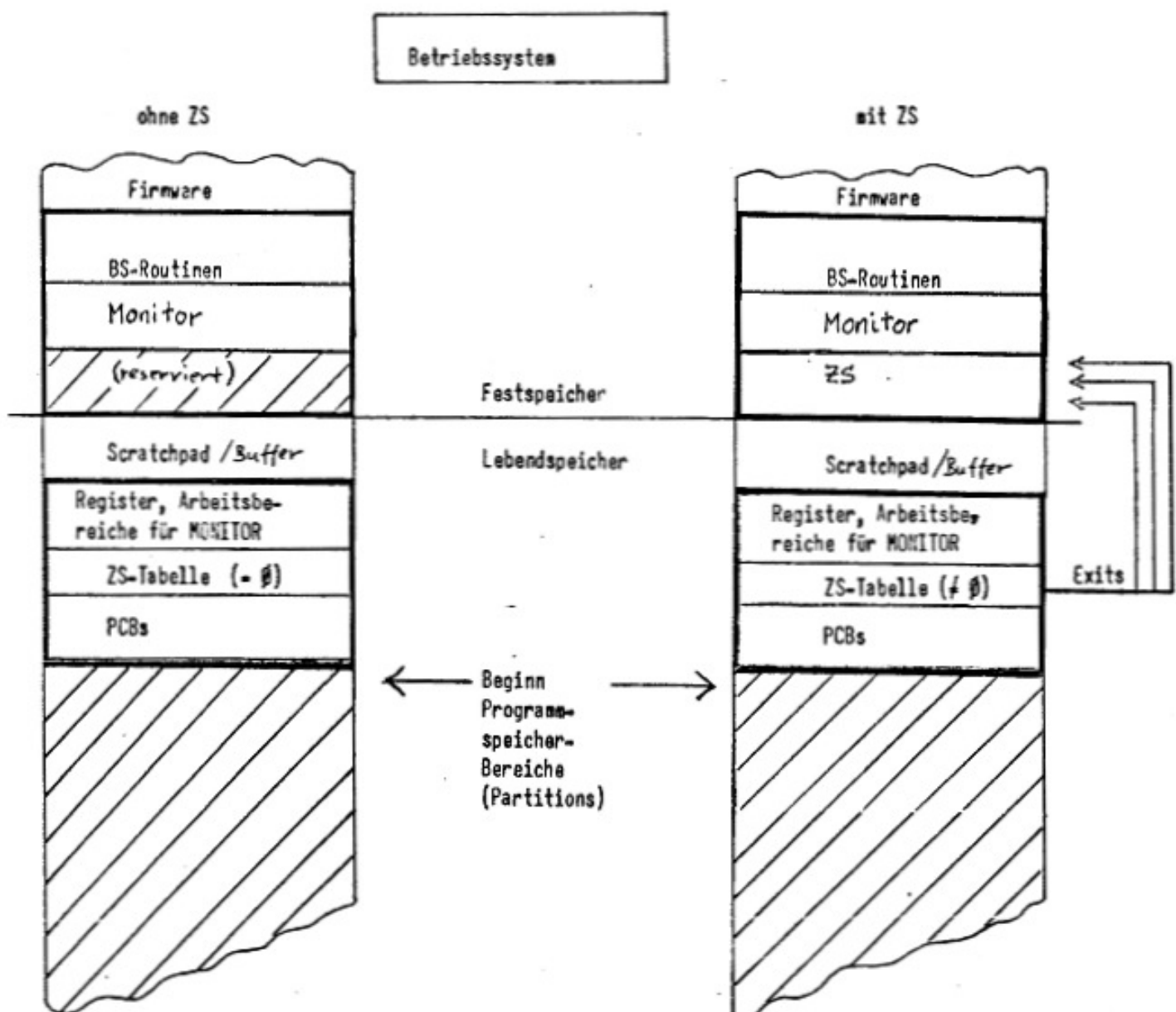
2.2 Speichereinteilung

Die folgende Abbildung zeigt die prinzipielle Speichereinteilung des Betriebssystems.



2.2.1. Schnittstelle "Betriebssystem/Zusatzsoftware"

Die Zusatzsoftware erlaubt Modifikationen des Standard-Betriebssystems. Die Schnittstelle ist mittels Exits realisiert, die sich in der ZS-Tabelle befinden.



Die Verbindung vom Betriebssystem bzw. MONITOR zur ZS geschieht über die ZS-Tabelle, wenn Einträge (EXITS) ungleich Ø vorliegen.

2.3 Tabellen des Betriebssystems2.3.1 ECR-Tabelle (Executive Communication Region)

Byte	Länge	Inhalt
1-5	5	BS-Identifikation (α -numerisch).
6	1	reserviert (= x/0200)
7-12	6	IPL-Identifikation (α -numerisch).
13	1	reserviert (= x/0200)
14	1	DS-Status (α -numerisch). x/0300 = Grundstellung (BS arbeitet ohne DFÜ) x/0301 = LOG-AUS; es darf kein DFÜ-Nachrichtenverkehr stattfinden, Sonderbedingungen ausgenommen. x/0302 = LOG-EIN; alle Formen des Nachrichtenverkehrs erlaubt. x/0304 = DFÜ-AUS; DFÜ gestört, kein Nachrichtenverkehr möglich. x/0308 = TEST; Alle Formen des Test-Nachrichtenverkehrs erlaubt. 1)
15	1	reserviert (= x/0200)
16-23	8	DS-Identifikation (α -numerisch). Name der Datenstation.
24	1	reserviert (= x/0200)
25-30	6	Tagesdatum (numerisch entpackt).
31-58	28	Frei belegbar für Zusatzsoftware. 2)

1) TEST beinhaltet LOG-EIN.

2) Die Belegung ist individuell, je nach ZS-Funktionen.

2.3.2 ZS-Tabelle (Zusatzsoftware)

Die Tabelle wird durch IPL (Inbetriebnahme des Systems) geladen.

Die Tabelle beinhaltet einige EXIT's, die eine Modifikation des Betriebssystems durch die Zusatzsoftware erlauben.

Ist ein Eintrag = \emptyset , wird die entsprechende Routine nicht ausgeführt.

Ist keine Zusatzsoftware vorhanden, sind sämtliche Eintragungen mit \emptyset geladen.

Byte	Länge	Inhalt
1- 2	2	Absolute Adresse einer Mikroroutine, auf die bei Erkennen eines SVC für Zusatzsoftware verzweigt wird.
3- 4	2	Adresse einer Routine (relativ zur Monitoranfangsadresse), auf die in der Monitoranlaufoutine bei IPL, NEUSTART oder CONTINUE verzweigt wird.
5- 6	2	Adresse einer Routine (relativ zur Monitoranfangsadresse), auf die verzweigt wird, wenn die Tastatureingaberoutine des Monitor nicht verwendet werden soll. Die Routine wird angesprungen, wenn die MON-Anruftaste gedrückt wird und muß eine Eingaberoutine für Monitoraufrufe beinhalten.
7- 8	2	Adresse einer Routine (relativ zur Monitoranfangsadresse), auf die verzweigt wird, wenn bereits eingegebene Monitoraufrufe getestet und ausgeführt werden sollen.
9-10	2	Adresse einer Routine (relativ zur Monitoranfangsadresse), auf die verzweigt wird, wenn vom Monitor eine Meldung auf die Systemzeile ausgegeben werden soll.
11-12	2	Adresse einer Routine (relativ zur Monitoranfangsadresse),

2.3.3 Aufbau des Prozeßkontrollblockes (PCB)

Byte	Länge	Inhalt
1	1	Dynamischer Status Bit 1 = 1 Prozeß ist "bereit" Bit 2 = 1 Prozeß ist "blockiert" Bit 3 = 1 Prozeß ist "aktiv" Bit 4 = 1 Prozeß "wartet"
2	1	Statischer Status Bit 1 = 1 Partition ist "eingesetzt" Bit 2 = 1 Prozeß ist "geladen" Bit 3 = 1 Prozeß ist "gestartet" Bit 4 = 1 Prozeß ist "unterbrochen" Bit 5 = 1 Prozeß ist "beendet" (SVC Terminate) Bit 6 = 1 Prozeß ist "fehlerhaft beendet" (Intern fehler) Bit 7 = 1 Prozeß ist "durch Bediener eingriff beendet" (MON-Aufruf CAN)
3	1	Priorität des Prozesses (binär)
4	1	Reserviert
5-12	8	Programmname (1-8 α-num. Zeichen)
13	1	Nummer des Internfehlers (binär)
14	1	Gerätenummer des belegten Gerätes (binär)
15-16	2	Partition-Anfangsadresse (absolut)
17-18	2	Partition-Endeadresse (absolut)
19-20	2	Adresse des Internfehlers (absolut)
21-22	2	Prozeßmaske (linkes Byte) Bit 1 = 1 Prozeß nach jeder Unterbrechungsverarbeitung fortsetzen Bit 2 = 1 Prozeß läuft unter Testbedingungen Bit 3 = reserviert für Monitor Bit 4 - 8 (reserviert) Prozeßmaske (rechtes Byte) Bit 1 = 1 E/A-Unterbrechung zulässig Bit 2 = (reserviert) Bit 3 = 1 Externe Unterbrechung zulässig Bit 4 = 1 Testunterbrechung zulässig Bit 5 = (reserviert) Bit 6 = reserviert für Monitor Anruf Bit 7 = reserviert für Anlauf Bit 8 = reserviert für Systemnachrichten

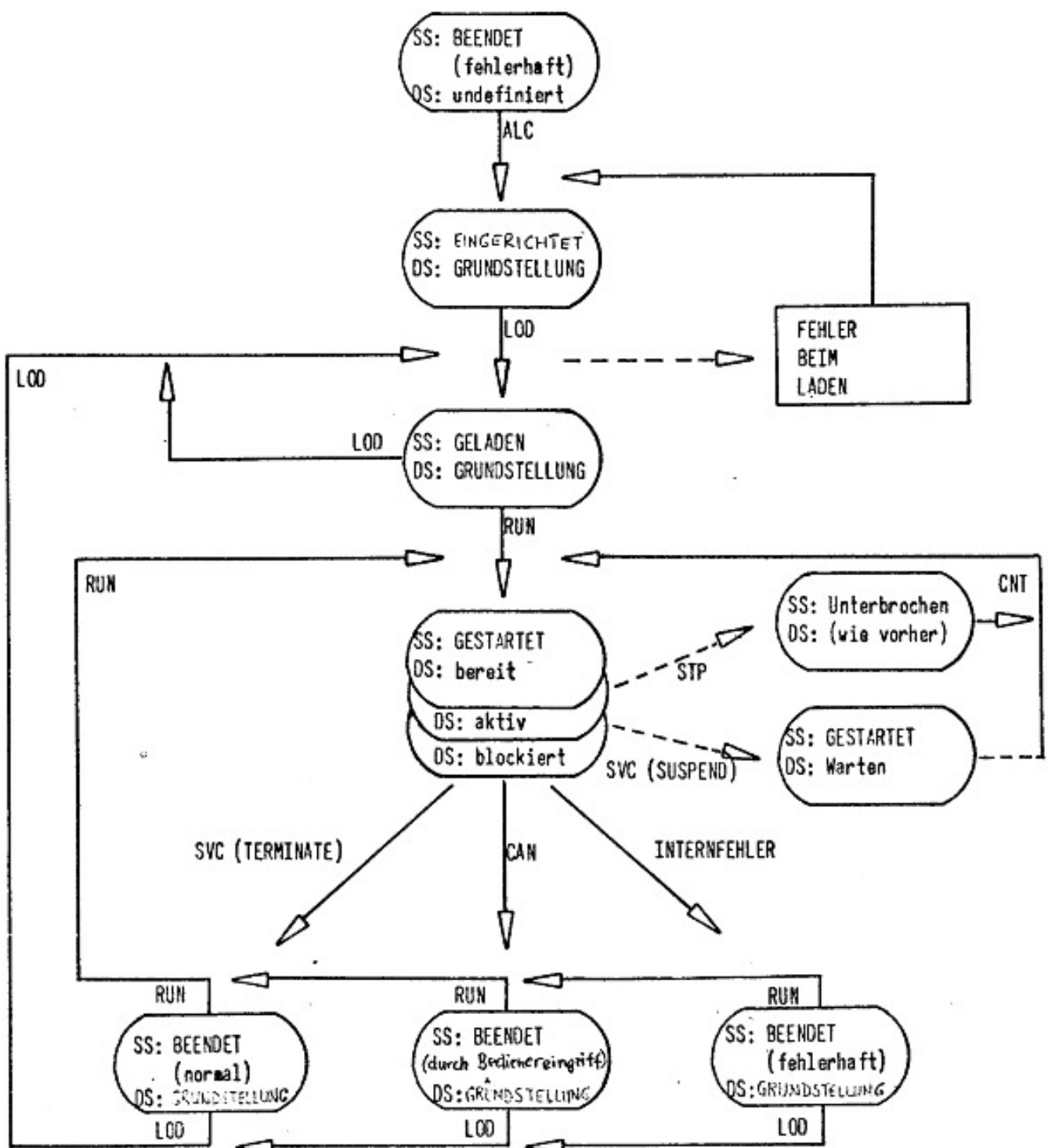
Byte	Länge	Inhalt
23	1	Ereignisbyte Bit 1 = 1 Data Transmission Bit 2 = 1 Continue Bit 3 = 1 Receive Message Bit 4 = 1 End of Send Message Bit 5 = 1 Time Bit 6-8 - reserviert für Monitor
24	1	Wait for transmission-byte (Priorität der Sender - binär)
25-26	2	VB-Adresse des Senders (absolut)
27-28	2	Reserviert für die Zeitüberwachung
29-44	16	Save-Area für Floppy-Disk
45-58	14	Save-Area für den Prozeß
59-60	2	Reserviert für Monitor
61-64	4	(reserviert für Erweiterungen)

Programmzustände im Betriebssystem

Der Programmzustand bzw. der Zustand des Programmbereichs (Partition) ändert sich durch Bedienereingriffe (Monitorkaufufe) oder durch die Aktivität des Programms selbst.

Der Programmzustand ist im Prozeßkontrollblock (PCB) als statischer Status (SS) und dynamischer Status (DS) vermerkt.

Programmzustände nach Inbetriebnahme des Systems



3. PROGRAMMIERUNG DES SYSTEMS

3.1. Allgemeines

Für die Programmierung ist der Grundbefehlssatz des Systems TA 1000 (ohne Betriebssystem) maßgebend. Für das Mehrfachbetriebssystem kommt eine Anzahl weiterer Befehle hinzu.

Eine eigene Befehlsgruppe bilden die Betriebssystem-Steuerbefehle. Sie sind nur in einem Mehrfach-Betriebssystem anwendbar.

Im Gegensatz zu einem Einprogrammssystem (TA 1000) müssen bei dem Mehrprogrammssystem zusätzlich bestimmte Programmierregeln berücksichtigt werden.

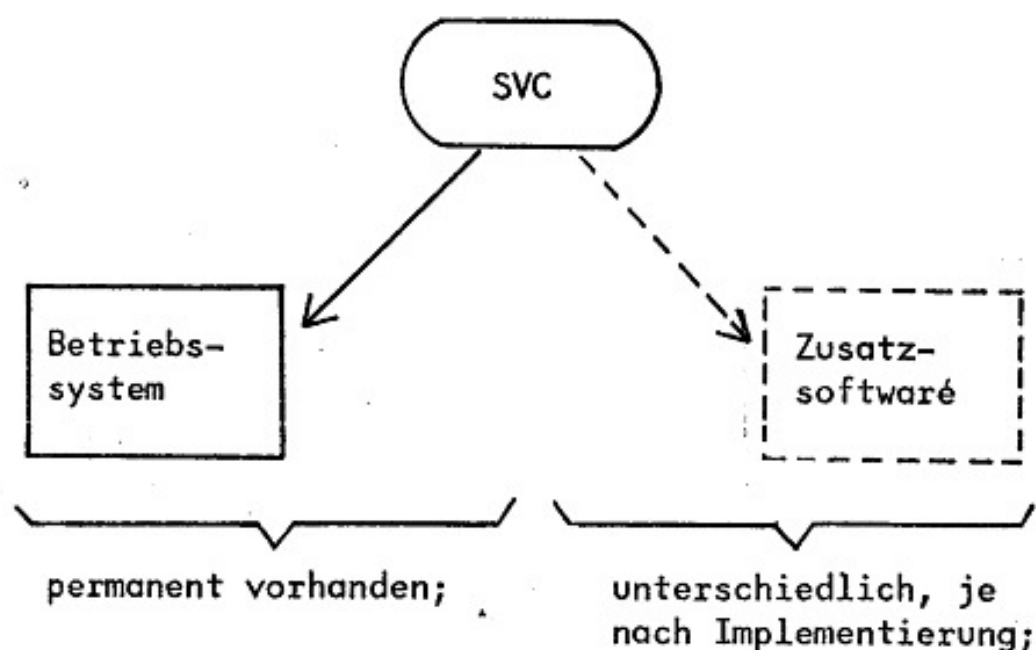
In den folgenden Abschnitten sind die Erweiterungen und Einschränkungen, die für das Mehrprogramm-Betriebssystem gelten, näher erläutert.

Betriebssystem-Steuerbefehle

1.1 Allgemeines

Im Multiprogramm-Betriebssystem stehen dem Anwender verschiedene Systemroutinen zur Verfügung. Diese Routinen berücksichtigen die spezifischen Eigenschaften des Betriebssystems. Um die Systemdienste in Anspruch nehmen zu können, müssen sie durch das Anwenderprogramm mit dem Befehl "SUPERVISOR CALL" (SVC, Aufrufen Betriebssystem) aktiviert werden. Der Befehl "SUPERVISOR CALL" wird in zwei Gruppen eingeteilt:

1. Aufruf von Systemroutinen, die unabhängig von der Anlagenausstattung permanent zur Verfügung stehen. Sie sind fester Bestandteil des Betriebssystems.
2. Aufruf von Systemroutinen, die abhängig von der Anlagenausstattung nur bei einer speziellen Implementierung zur Verfügung stehen. Sie sind Bestandteil der Zusatzsoftware.



1.2 SUPERVISOR CALL

Mnemocode	Operanden
SVC	c, { ∅ adresse direktooperand }

Beschreibung der Operanden

c: SVC-Code ($c \geq \emptyset$)

∅: der Eintrag wird nicht verwendet

adresse: Speicherstelle eines allgemeinen Datenfeldes

direktooperand: Direktoperand, abhängig vom SVC-Code.

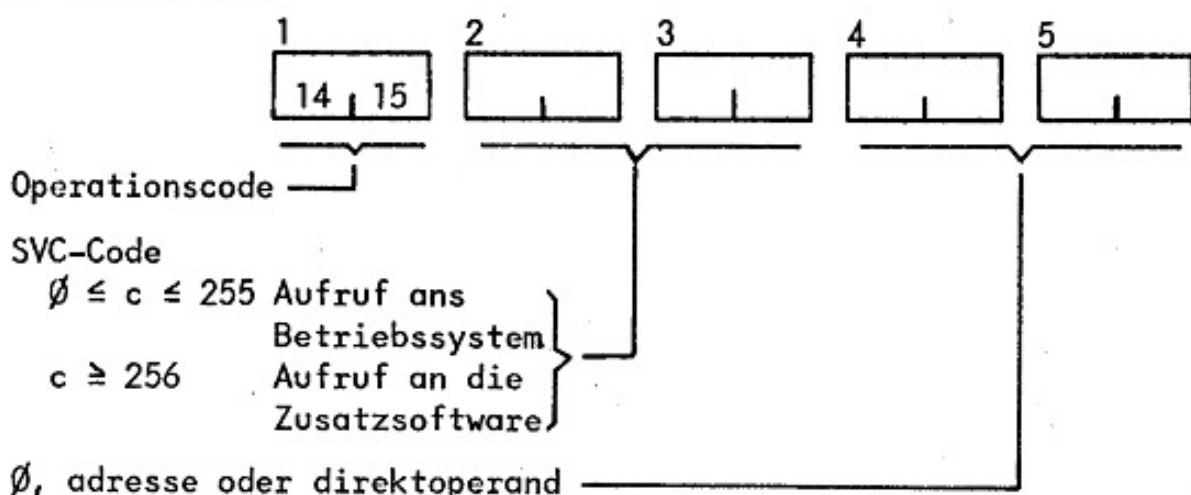
Die Operanden können als "direkte Adressen" oder "indizierte Adressen" gemäß den Assemblerkonventionen angegeben werden.

Befehlsbeschreibung

Der im ersten Operand enthaltene SVC-Code liefert den Anschluß an die Systemroutine, die durch den Befehl aufgerufen werden soll. Je nach Funktion der Routine muß im zweiten Operand die Adresse eines Verständigungsbereichs, ein Direktoperand oder, wenn der Eintrag nicht verwendet wird, der Wert ∅ angegeben werden.

Bedingungsschlüssel

Unterschiedlich, abhängig vom SVC-Code.

Maschinencode

Fehlermöglichkeiten

Internfehler 2: Direktoperand falsch,
falsche Werte im Datenfeld;

Verwendungsvorschriften

Der Befehl "SUPERVISOR CALL" ist unterteilt in Aufrufe an das Betriebssystem (SVC-Code \emptyset -255) und Aufrufe an die Zusatzsoftware (SVC-Code \geq 256).

Die Routinen für Aufrufe an das Betriebssystem sind standardmäßig immer vorhanden.

Bei den Aufrufen an die Zusatzsoftware ist zu beachten, daß ein bestimmter SVC an verschiedenen Anlagen u.U. unterschiedliche Funktionen auslöst, bzw. überhaupt nicht implementiert ist. Dies ist davon abhängig, ob und mit welcher Zusatzsoftware die Anlage ausgestattet ist.

Ist ein SVC-Aufruf im Betriebssystem nicht bekannt, wird der Befehl übergangen, d.h. die gewünschte Funktion wird nicht ausgeführt und das Programm hinter dem SVC fortgesetzt.

1.3 SVC-Aufrufe an das Betriebssystem

Folgende SVC-Aufrufe stehen zur Verfügung:

SVC-Name	SVC-Code	Funktion
SET MASK	∅	Maskierungsbyte des laufenden Programms setzen.
RESET MASK	1	Maskierungsbyte des laufenden Programms zurücksetzen.
TERMINATE	2	Laufendes Programm beenden.
SEND MESSAGE	3	Senden einer Nachricht an ein anderes Programm.
SUSPEND	4	Warten auf ein Ereignis.
GET ECR	5	ECR-Daten übernehmen.
TEST DVC	6	Testen Gerät-

1.3.1 SVC (SET MASK)

Mnenocode	Operanden
SVC	∅, direktoperand

Beschreibung der Operanden

∅: SVC-Code für Funktion SET MASK

direktoperand: Programmmaske in dezimaler Darstellung

Beschreibung der Programmmaske

Bit 0 = 1	E/A-Unterbrechungen nicht zugelassen.	} rechtes Byte
Bit 1 = ∅	reserviert (immer ∅).	
Bit 2 = 1	Externe Unterbrechung nicht zugelassen.	
Bit 3 = 1	Testunterbrechung nicht zugelassen.	
Bit 4-7 = ∅	reserviert (immer ∅).	
Bit 8 = 1	Prozeß nach jeder Unterbrechungsverarbeitung unbedingt fortsetzen.	} linkes Byte
Bit 9 = 1	Prozeß läuft unter Testbedingungen.	
Bit 10-15 = ∅	nicht verwendet (ohne Bedeutung).	

Befehlsbeschreibung

Der Befehl SVC (SET MASK) verknüpft die Programmmaske mit dem Maskierungsbyte des laufenden Prozesses. Entsprechend diesen Eintragungen wird die Prozeßverarbeitung gesteuert. Eine Funktion ist ein, wenn das zuständige Bit = 1 ist. Ist ein Bit = ∅, bleibt das entsprechende Bit im Maskierungsbyte unverändert.

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten keine; fehlerhaft gesetzte Bits werden ignoriert.

Anmerkung

Die Programmmaske darf auch in einem Register stehen, bzw. sich aus Registerinhalt und Displacement zusammensetzen. In diesem Fall ist als Direktoperand Register und Displacement anzugeben (indizierte Adresse, Bit 15 = 1).

1.3.2 SVC (RESET MASK)

Mnemocode	Operanden
SVC	1, direktoperand

Beschreibung der Operanden

1: SVC-Code für Funktion RESET MASK
 direktoperand: Programmaske in dezimaler Darstellung

Beschreibung der Programmaske

Bit 0	= 1	E/A-Unterbrechungen zugelassen.	} rechtes Byte
Bit 1	= 0	reserviert (immer 0).	
Bit 2	= 1	Externe Unterbrechungen zugelassen.	
Bit 3	= 1	Testunterbrechung zugelassen.	
Bit 4-7	= 0	reserviert (immer 0).	} linkes Byte
Bit 8	= 1	Prozeß nicht nach jeder Unterbrechungs- verarbeitung unbedingt fortsetzen.	
Bit 9	= 1	Prozeß läuft nicht unter Testbedin- gungen.	
Bit 10-15	= 0	nicht verwendet (ohne Bedeutung).	

Befehlsbeschreibung

Mit dem Befehl SVC (RESET MASK) können die mit SVC (SET MASK) gesetzten Funktionen zurückgesetzt werden. Eine Funktion ist aus, wenn das zuständige Bit = 1 ist. Ist ein Bit = 0, bleibt das entsprechende Bit im Maskierungsbyte unverändert.

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten keine; fehlerhaft gesetzte Bits werden ignoriert.

Anmerkung

Die Programmaske darf auch in einem Register stehen, bzw. sich aus Registerinhalt und Displacement zusammensetzen. In diesem Fall ist als Direktoperand Register und Displacement anzugeben (indizierte Adresse, Bit 15 = 1).

1.3.3 SVC (TERMINATE)

Mnemocode	Operanden
SVC	2,∅

Beschreibung der Operanden

2: SVC-Code für Funktion TERMINATE

∅: nicht benutzt (immer ∅)

Befehlsbeschreibung

Der Befehl SVC (TERMINATE) beendet das laufende Programm. Das Programm bleibt im Speicher stehen und kann jederzeit wieder gestartet werden. Es kann jedoch auch ein anderes Programm in die Partition geladen werden.

Nicht abgeschlossene EA-Geräte werden freigegeben. *)

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten keine.

*) Die Geräte werden nicht physikalisch entriegelt;

1.3.4 SVC (SEND MESSAGE)

Mnemocode	Operanden
SVC	3, adresse

Beschreibung der Operanden

3: SVC-Code für Funktion SEND MESSAGE

adresse: Speicheradresse des Verständigungsbereichs

Beschreibung des Verständigungsbereichs

Byte	Länge	Inhalt
1	1	Statusbyte (binär): Code \emptyset = Nachricht übernommen 1 = Empfänger nicht vorhanden 2 = Nachricht wurde nicht übernommen
2-4	3	reserviert.
5-6	2	Länge der Nachricht.
7-14	8	Name des Empfangsprozesses.
15-16	2	Adresse des Nachrichtenpuffers.

} wird vom System versorgt;

} vom Benutzer anzugeben;

Befehlsbeschreibung

Der Befehl SVC (SEND MESSAGE) startet die Übertragung der im Nachrichtenpuffer stehenden Daten des laufenden Prozesses an einen anderen namentlich bekannten Prozeß (Empfänger). Der Name des Empfängers, Länge und Adresse der Nachricht muß vor Aufruf des Befehls im Verständigungsbereich hinterlegt werden. Dem Befehl SVC (SEND MESSAGE) muß ein SVC (SUSPEND) mit den Ereignissen EOSM, TIME folgen. Dort wird auf die Übernahme der Nachricht gewartet. Das Statusbyte darf erst nach Ablauf des SVC (SUSPEND) geprüft werden.

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten

Internfehler 2: Falsche Befehlsfolge, SVC (SUSPEND) fehlt.

1.3.5 SVC (SUSPEND)

Mnemocode	Operanden
SVC	4, adresse

Beschreibung der Operanden

4: SVC-Code für Funktion SUSPEND

adresse: Speicheradresse des Verständigungsbereichs

Beschreibung des Verständigungsbereichs

Byte	Länge	Inhalt
1	1	Statusbyte (binär), enthält das aktuelle Ereignis nach Aufhebung des Wartezustandes.
3) { 2	1	Zulässige Ereignisse, auf deren Eintritt das Programm wartet: Bit 0 = 1 Wait for DATA TRANSMISSION (DT) Bit 1 = 1 Wait for CONTINUE (CNT) Bit 2 = 1 Wait for RECEIVE MESSAGE (RECV) Bit 3 = 1 Wait for END OF SEND MESSAGE (EOSM) Bit 4 = 1 Wait for TIME INTERRUPT (TIME) Bit 5-7 = 0 reserviert (immer 0)
4) { 3-4	2	Wartezeit *) für das Ereignis TIME in binärer Darstellung ($1 \leq \text{Wartezeit} \leq 65535$).
5) { 5-6	2	Länge des Empfangspuffers.
7-14	8	Name des Sendeprozesses nach Übernahme der Nachricht.
15-16	2	Adresse des Empfangspuffers.

- 1) Wird vom System versorgt.
- 2) Muß vom Benutzer angegeben werden.
- 3) Dieser Bereich ist immer anzulegen (konstanter Bereich).
- 4) Dieser Bereich ist nur anzulegen, wenn als zulässiges Ereignis Bit 4 = 1 (TIME) oder/und Bit 2 = 1 (RECV) angegeben ist.
- 5) Dieser Bereich ist nur anzulegen, wenn als zulässiges Ereignis Bit 2 = 1 (RECV) angegeben ist.

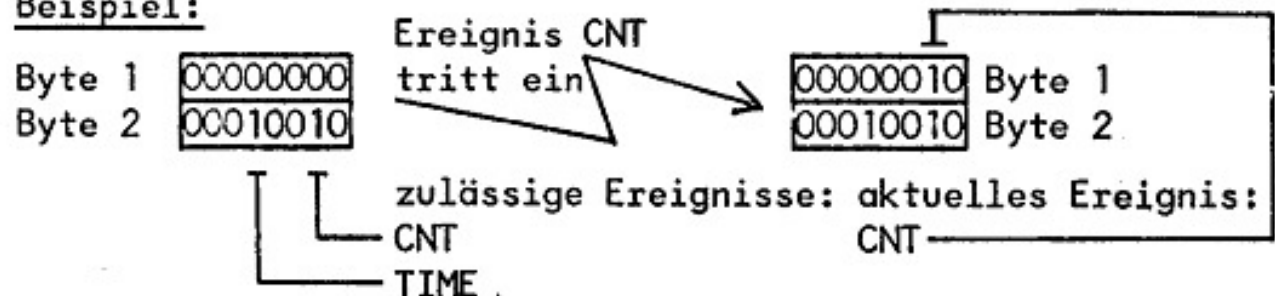
*) Eine Zeiteinheit entspricht ca. 128 ms.

Befehlsbeschreibung

Mit dem Befehl SVC (SUSPEND) wird das laufende Programm in einen definierten Wartezustand versetzt. Das Programm wird nur dann fortgesetzt, wenn eines der im Byte 2 angegebenen Ereignisse eintritt. Das bedeutet, daß mindestens ein Bit für ein zulässiges Ereignis vor Aufruf des Befehls gesetzt sein muß, da sonst das Programm den Wartezustand nicht mehr verlassen kann.

Im Byte 2 werden die zulässigen Ereignisse vom Benutzerprogramm definiert, d.h., das entsprechende Bit auf 1 gesetzt. Es dürfen mehrere Ereignisse angegeben werden. Tritt eines der angegebenen Ereignisse ein, wird das Programm wieder aktiviert und hinter dem Befehl SVC (SUSPEND) fortgesetzt.

Das aktuelle Ereignis wird im Byte 1 des Verständigungsbereich abgelegt und kann dort abgefragt werden. Es kann nur ein aktuelles Ereignis eintreten. Die Bitposition des aktuellen Ereignisses (Byte 1) ist entsprechend der Bitposition des zulässigen Ereignisses (Byte 2).

Beispiel:

Tritt ein Ereignis ein und das entsprechende Bit im Byte 2 ist nicht auf 1 gesetzt, wird das Ereignis ignoriert und das Programm bleibt weiter im Wartezustand.

Beschreibung Byte 2 (zulässige Ereignisse)

"Wait for DATA TRANSMISSION" (DT, Bit 0 = 1)

Das Programm wartet solange, bis das Ereignis "DFÜ-Nachricht eingetroffen" eintritt und die Nachricht für das wartende Programm bestimmt ist.

"Wait for CONTINUE" (CNT, Bit 1 = 1)

Das Programm wartet solange, bis der Bediener durch den Monitor-Aufruf CNT (=CONTINUE) den Wartezustand wieder aufhebt.

"Wait for RECEIVE MESSAGE" (RECV, Bit 2 = 1)

Das Programm wartet solange, bis ein anderer Prozeß eine Nachricht mit SVC (SEND MESSAGE) an das wartende Programm abgeschickt hat. Die Nachricht wird in den Empfangspuffer übertragen und der Name des Senders im Byte 7-14 hinterlegt. Ist der Empfangspuffer zu klein, werden Daten abgeschnitten. Die Länge der Nachricht steht im Register $x\phi$.

"Wait for END OF SEND MESSAGE" (EOSM, Bit 3 = 1)

Das Programm wartet solange, bis eine vorher mit SVC (SEND MESSAGE) abgesandte Nachricht von einem anderen Prozeß übernommen wurde. Kann die Nachricht nicht übernommen werden, wird der Wartezustand nicht aufgehoben. *) Ob die Nachrichtenübertragung erfolgreich war, kann im Verständigungsbereich (Statusbyte) des SVC (SEND MESSAGE) geprüft werden.

"Wait for TIME INTERRUPT" (TIME, Bit 4 = 1)

Das Programm wartet solange, bis die in Byte 3-4 angegebene Wartezeit abgelaufen ist.

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten keine.

Hinweis

Ein vorher gegebener SVC (SEND MESSAGE) wird durch jedes Ereignis beendet.

Die TIME-Funktion wird durch jedes Ereignis aufgehoben.

*) es wird empfohlen, EOSM mit CNT und/oder TIME zu kombinieren um aus dem Wartezustand gezielt herauszukommen.

1.3.6 SVC (GET ECR)

Mnemocode	Operanden
SVC	5, adresse

Beschreibung der Operanden

5: SVC-Code für Funktion GET ECR

adresse: Speicheradresse eines Datenfeldes

Beschreibung des Datenfeldes

Für das Datenfeld müssen 17 Bytes reserviert werden. Nach Ausführung des Befehls hat das Datenfeld folgenden Inhalt:

Byte	Länge	Inhalt
1	1	DS-Status (numerisch).
2	1	nicht verwendet (X/0200).
3-10	8	DS-Identifikation (α-numerisch).
11	1	nicht verwendet (X/0200).
12-17	6	Tagesdatum, wie in der Systemzeile angezeigt.

Befehlsbeschreibung

Mit dem Befehl SVC (GET ECR) werden 17 Bytes aus dem Betriebssystem-Verständigungsbereich (ECR = Executive Communication Region) in das im Befehl angegebene Datenfeld übertragen.

Bedingungsschlüssel unverändert.

Fehlermöglichkeiten keine.

1.3.7 SVC (TEST DVC)

Mnemocode	Operanden
SVC	6, gn

Beschreibung der Operanden

- 6 : SVC-Code für Funktion TEST DVC
 gn: Gerätenummer (Direktoperand od. Register)

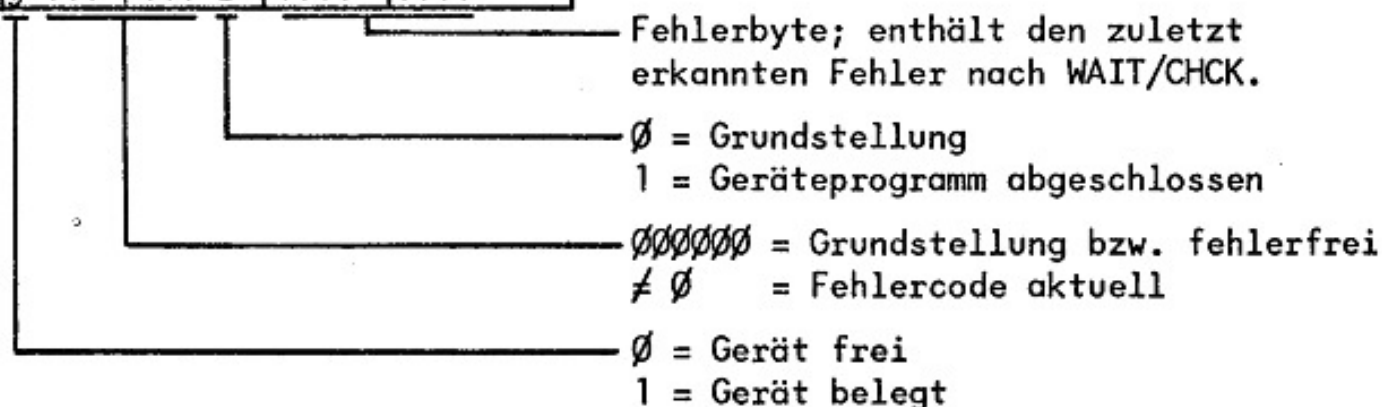
Befehlsbeschreibung

Mit dem Befehl SVC (TEST DVC) kann der aktuelle Zustand eines Gerätes geprüft werden.

Nach Ausführung des Befehls enthält das Register XØ das Status- und Fehlerbyte aus dem Geräte-Scratchpad. Entsprechend dem Gerätezustand werden die Bedingungsschlüssel gesetzt.

Inhalt Register XØ:

Linkes Byte		rechtes Byte	
g	fff	fff	r
		ffff	ffff

Bedingungsschlüssel

- H: Gerät ist z.Z. von einem anderen Programm reserviert-
 E: Gerät ist frei.
 O: Gerät ist nicht vorhanden (Register XØ unverändert).

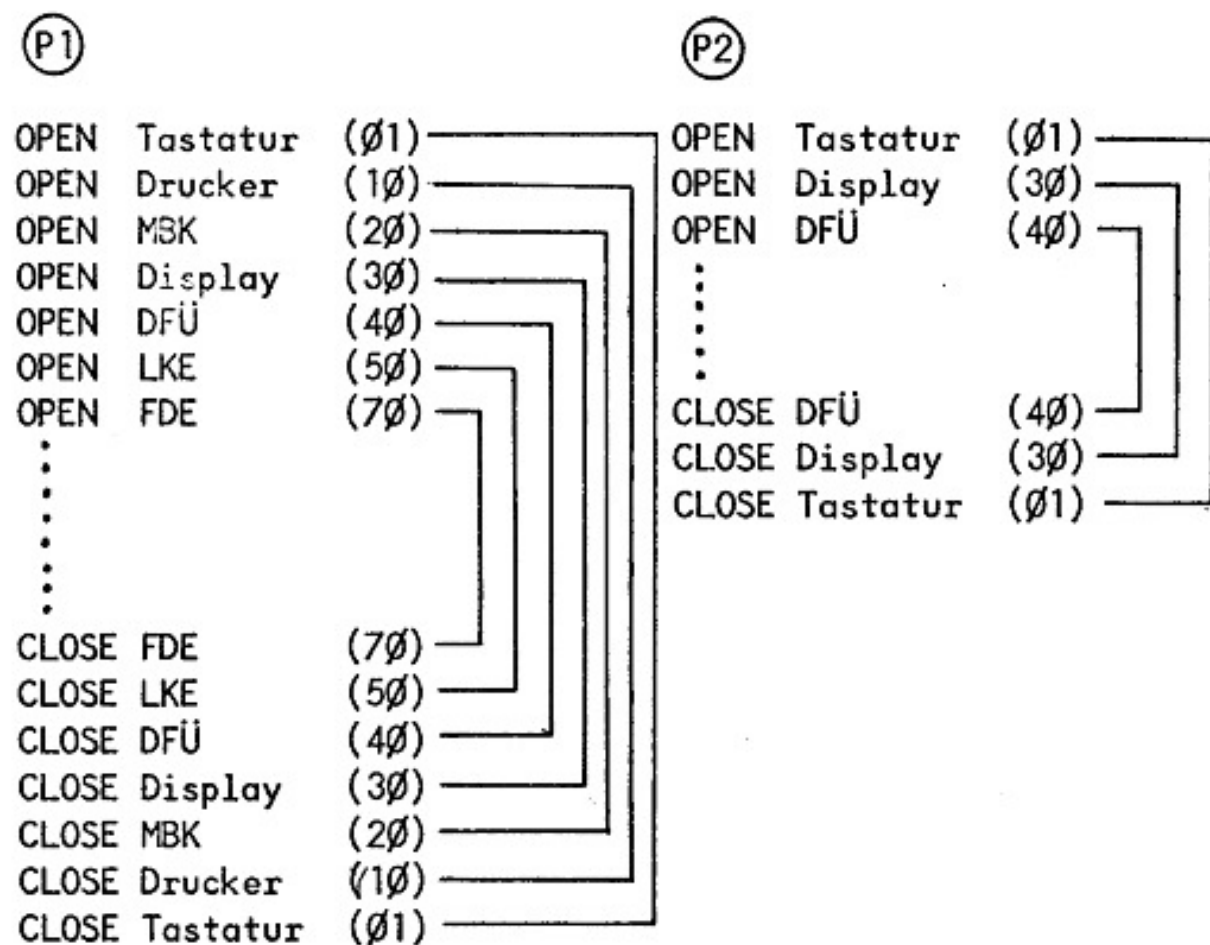
Fehlermöglichkeiten: Keine.

PROGRAMMIER - VORSCHRIFTEN1. Programmierung von Ein-Ausgabegeräten1.1 Allgemeines

Prinzipiell muß ein Programm alle Geräte, die es zur Verarbeitung der Daten benötigt, mit OPEN eröffnen und nach Beendigung der Arbeit mit CLOSE wieder schließen.

Jede Ein-Ausgabeoperation muß mit WAIT oder CHECK abgeschlossen werden.

Alle benötigten Geräte müssen nach Geräte-Nummern in aufsteigender Reihenfolge eröffnet und in absteigender Folge abgeschlossen werden.

Beispiele:

Wird diese Programmierregel bei seriell verwendbaren Geräten (TYP-I) nicht eingehalten, kann dies zu Systemblockaden führen.

1.2 Doppelbelegung von Geräten

Simultan verwendbare Geräte (Typ-II) dürfen von mehreren Programmen benutzt werden. Ein Gerät ist nur während einer E/A-Operation belegt.

Für seriell verwendbare Geräte (Typ-I) müssen bei der Programmierung bestimmte Vorschriften beachtet werden.

Zu diesem Gerätetyp zählen:

MBK, LKE, Display, Tastatur, Drucker

1.2.1 Ablauf

Während der Verarbeitung zwischen OPEN und CLOSE ist das Gerät nur einem Programm zugeordnet. Will ein zweites Programm während dieser Zeit das gleiche Gerät benutzen, wird ein Bedienerhinweis

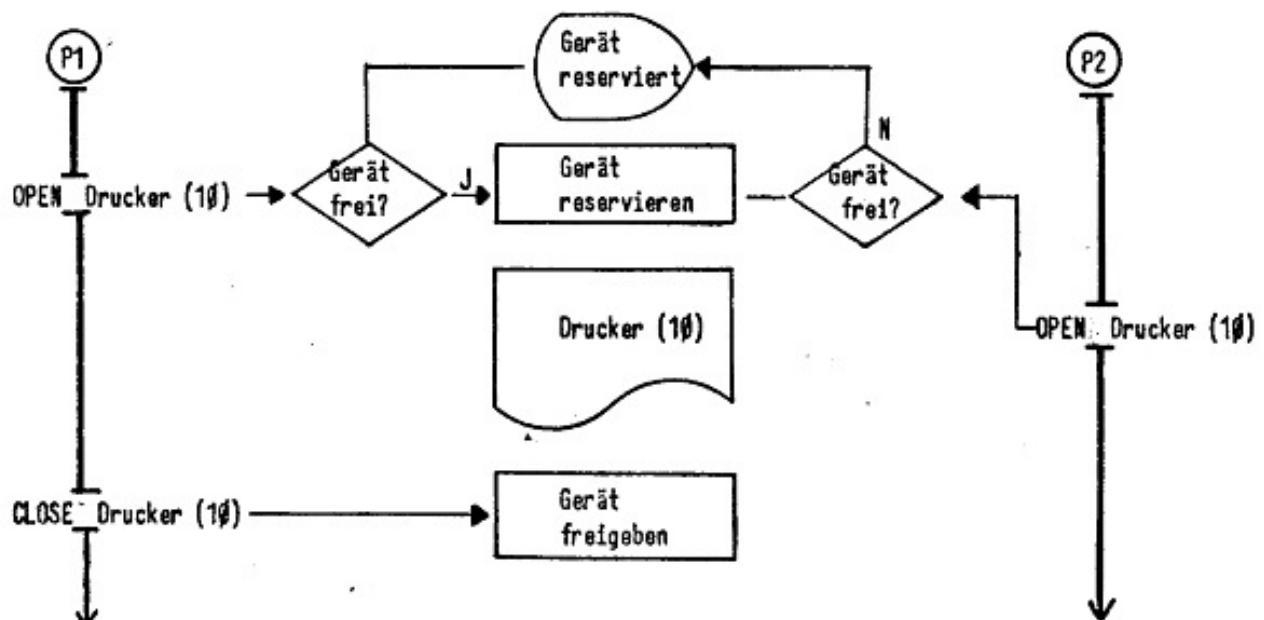
p/DVC-xx-RESRV

(Gerät xx reserviert)

ausgegeben und das anfordernde Programm solange blockiert, bis das Gerät durch CLOSE vom ersten Programm freigegeben wird.

Beispiel:

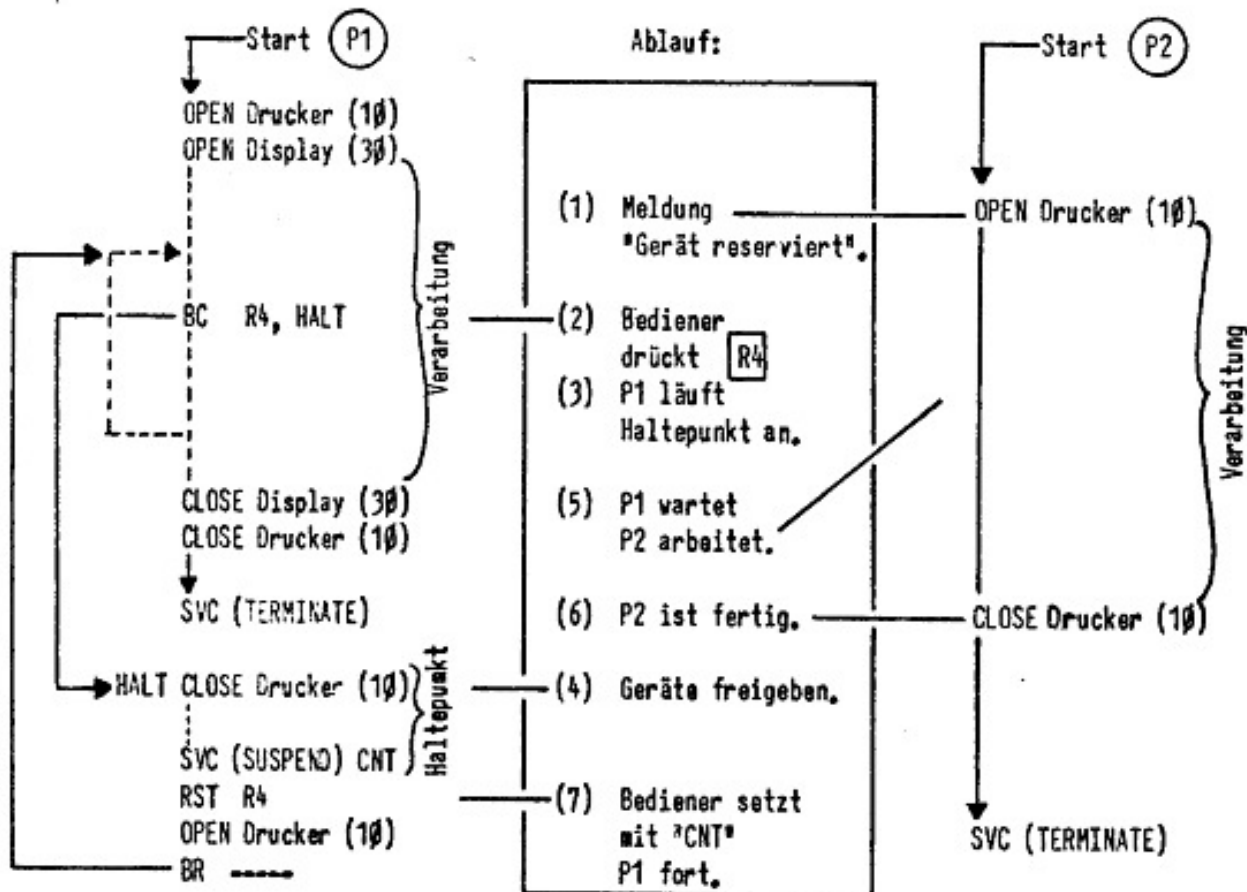
Programm P1 hat den Drucker zuerst eröffnet. Programm P2 will den Drucker etwas später ebenfalls eröffnen. P2 wird blockiert, bis P1 mit CLOSE den Drucker freigibt.



1.2.2 Haltepunkt

Soll ein Programm "P2" die Möglichkeit bekommen, zwischenzeitlich ein bereits reserviertes Gerät ebenfalls zu benutzen, so muß ein Programm "P1", welches das Gerät reserviert hat, so konstruiert sein, daß es, z.B. durch Bedieneringriff (2), das Gerät kurzfristig freigeben kann (3).

Beispiel:



Das Programm "P1" muß eine Routine anlaufen, in der das entsprechende Gerät freigegeben werden kann (Haltepunkt). Die Routine muß mit dem Befehl SVC (SUSPEND) abgeschlossen werden (4).

Das Programm wartet, bis ein Ereignis eintritt, welches das Programm wieder aktiviert. Während dieser Zeit kann das Programm "P2" das Gerät belegen (5). Nach Beendigung der Arbeit muß "P2" das Gerät wieder freigeben (6).

Mit dem Monitoraufruf "CNT" (7) wird das unterbrochene Programm wieder fortgesetzt. Das geschlossene Gerät wird wieder eröffnet und das Programm an einer geeigneten Stelle fortgesetzt.

1.2.3 Mehrfachzugriff auf Dateien

Ein Mehrfachzugriff von verschiedenen Programmen auf eine Datei im gleichen Verarbeitungszeitraum ist prinzipiell erlaubt.

Bei seriell verwendbaren Geräten (TYP-I) ist ein Mehrfachzugriff aus hardware-technischen und organisatorischen Gründen nicht zulässig (siehe vorhergehende Abschnitte).

Bei simultan verwendbaren Geräten (TYP-II) ist ein Mehrfachzugriff (z.B. mehrere Programme greifen auf eine Plattendatei zu) möglich, wird aber vom Betriebssystem nicht überwacht.

1.3 Programmierregeln

Neben den allgemeinen Regeln (siehe 1.1) müssen bei den nachfolgend beschriebenen Geräten noch einige Besonderheiten beachtet werden.

1.3.1 Tastatur (GN = 1)

BSK, ACB, ACC, RST-F:

Kein WAIT/CHECK erforderlich.

ACPS: Zusätzlich OPEN/CLOSE Drucker (GN = 1Ø) erforderlich.
 WAIT/CHECK mit Geräte-Nr. 1Ø.
 Verständigungsbereich für Drucker zuständig.

BWO: Zusätzlich OPEN/CLOSE Display (GN = 3Ø) erforderlich.
 WAIT/CHECK mit Geräte-Nr. 3Ø.
 Verständigungsbereich für Display zuständig.

1.3.2 Drucker (GN = 1Ø)

ACPS: Zusätzlich OPEN/CLOSE Tastatur (GN = 1) erforderlich.

OPEN: Verständigungsbereich muß angegeben werden.

1.3.3 Display (GN = 3Ø)

BWO: Zusätzlich OPEN/CLOSE Tastatur (GN = 1) erforderlich.

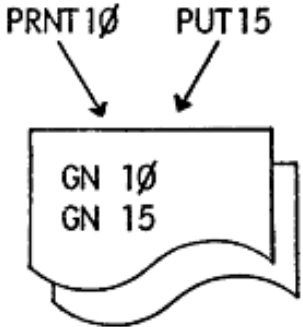
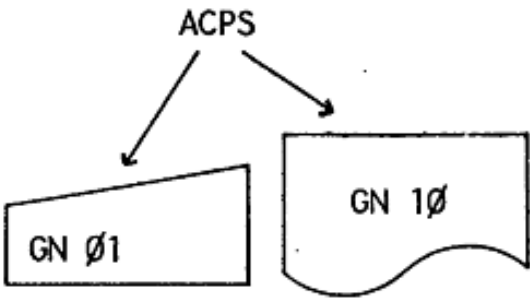
PUT (Zeile): Das Datenfeld muß innerhalb des Bildspeicherbereichs liegen.

1.3.4 Gerätekombinationen

Gerätekombinationen können durch physikalische Zusammenhänge (Hardware) oder logische Verbindungen (Software) begründet sein.

Beispiel:

Gerätekombinationen

physikalisch	logisch
1 Gerät (Drucker mit Belegzufuhr), aber 2 Geräte-Nummern und getrennte Programmierung möglich.	2 Geräte (Drucker und Tastatur), aber 1 Zugriffsbefehl (ACPS).
	

Bei der Programmierung können bei bestimmten Konstellationen unsinnige Systemwartezeiten auftreten:

Beispiel:

<pre> ┌ PRNT └ WAIT 10 PUT 15 WAIT 15 ┐ </pre>	Richtig.
<pre> ┌ PRNT └ PUT 15 WAIT 10 WAIT 15 ┐ </pre>	Möglich, aber PUT 15 wartet, bis PRNT fertig ist.

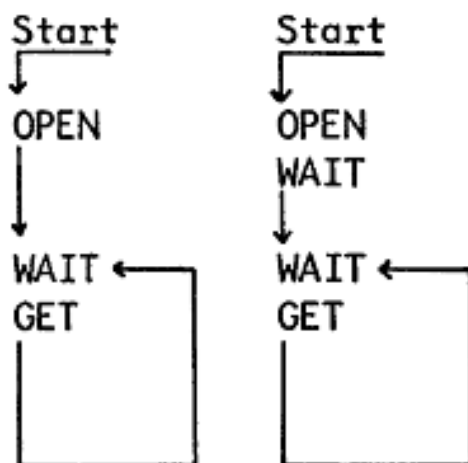
1.3.5 Geräteklasse III

Bei der Geräteklasse III (Parallelschnittstelle, DFÜ) muß das entsprechende Gerät unbedingt mit OPEN eröffnet werden.

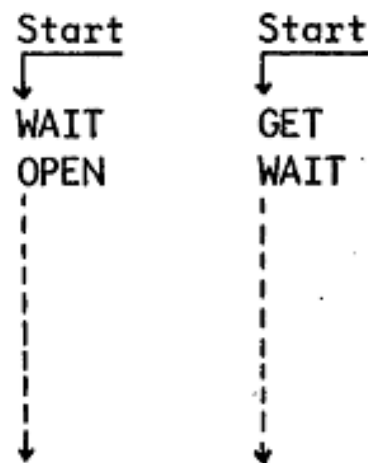
Ist vor OPEN ein sogenannter "blinder" WAIT/CHECK oder ein anderer Zugriffsbefehl programmiert, so kann dies zu einem falschen Programmablauf, bzw. zu falschen Ein-/Ausgabeoperationen führen.

Beispiele

(richtig)



(falsch)



2. Programmierung von Lampen, Rasttasten, Schleifen

Das Setzen bzw. Löschen von Lampen und Rasttasten kann vom Betriebssystem nicht überwacht werden. Dies bedeutet, daß eine Mehrfachbelegung von Lampen und Rasttasten durch verschiedene Programme nicht erkannt wird und zu Fehlinformationen für Bediener und Programme führen kann.

Lampen und Rasttasten müssen für Programme, welche im Multiprogramm-Betrieb laufen, eindeutig zugeordnet sein. D.h., mehrere Programme dürfen nicht die selben Lampen bzw. Rasttasten verwenden.

Zu beachten ist, daß Rasttastenschleifen wie auch grosse DEX-Schleifen, die keine unterbrechungserzeugende Operationen (E/A, SVC-SUSPEND) beinhalten, eine Prozeßumschaltung für längere Zeit verhindern können. Nämlich dann, wenn während dieser Zeit keine E/A-Operationen laufen oder keine Ereignisse für andere Programme die darauf warten, eintreten.

Solche Schleifen werden vom Betriebssystem nach einer bestimmten Wartezeit zwangsweise unterbrochen und versucht, die Steuerung an ein anderes Programm zu übergeben.

Aus diesen Gründen ist es auch nicht sinnvoll, Lampen-Blinkschleifen zu programmieren, da das Blinken unregelmäßig wird, bzw. zeitweise ganz unterbleibt. Dies ist davon abhängig, welche Operationen andere Programme simultan dazu ausführen.

3. Programmierung von SVC-Befehlen

SVC-Befehle sollen nur von Programmierern mit genauen Kenntnissen des Betriebssystems angewendet werden.

Die Programmierung von SVC-Befehlen sollte deshalb nur nach vorhergehender Rücksprache mit dem Systemberater erfolgen.

Eine Ausnahme bildet der Befehl SVC (TERMINATE). Er sollte immer bei Programmende durchlaufen werden (siehe folgenden Abschnitt).

3.1 SVC (TERMINATE)

Jedes Programm muß sich nach Abschluß seiner Tätigkeit selbst beenden. Dafür muß der Befehl SVC (TERMINATE) verwendet werden.

Alle eröffneten Geräte müssen vorher mit CLOSE abgeschlossen sein.

Beispiel:

```

┌STRT
├ OPEN  Tastatur (1)
├ OPEN  Display (3Ø)
├ .
├ .
├ .
├ CLOSE Display (3Ø)
├ CLOSE Tastatur (1)
├ .
├ .
├ SVC (TERMINATE)
└END

```

3.2 SVC (SET-MASK), SVC (RESET-MASK)

Die Befehle SVC (SET-MASK) und SVC (RESET-MASK) dürfen nur für Systemprogramme und Dienstprogramme verwendet werden.

Eine unsachgemäße Anwendung kann zum Systemzusammenbruch, bzw. zu undefinierten Ergebnissen führen.

4. BEDIENUNG DES SYSTEMS

Bei der Bedienung des Systems wird zwischen der Inbetriebnahme, d.h. das Einschalten und Anlaufen des Systems, und der eigentlichen Bedienung während des normalen Arbeitsablaufes unterschieden.

Modifikationen und Sonderbehandlungen sind in der getrennten Beschreibung 'Zusatzsoftware' nachzulesen.

4.1. Inbetriebnahme

Bei der Inbetriebnahme wird zwischen dem 'Urladen' (IPL), dem 'Neustarten' (NEUSTART) und dem 'Fortsetzen' (CONTINUE) unterschieden.

IPL

IPL ist immer dann notwendig, wenn

- das System erstmalig in Betrieb genommen wird,
- Internfehler auf Systemebene auftritt,
- nicht behebbare Systemblockaden auftreten;

Nach dem Einschalten der Maschine und Betätigen der IPL-Taste wird automatisch durch das System eine IPL-Anforderung über die Systemzeile ausgegeben.

Beim einem Standard-IPL wird nur das Betriebssystem in Grundstellung gebracht. Dabei ist kein Datenträger notwendig.

Bei einem speziellen IPL dient als Systemdatenträger eine Magnetplatte (z.B. Floppy Disk), auf der die notwendigen Systemdaten unter dem Namen "IPLxxx" in der Systembibliothek "SYSCOR" gespeichert sein müssen. Damit können Anwendersysteme, bestehend aus mehreren vorbereiteten Programmen in bestimmten Partitions, ohne weitere Bediene Maßnahmen gleichzeitig geladen und gestartet werden. Das Gerät der Partitions (ALC) entfällt dabei.

BEDIENUNG

Beim IPL-Vorgang wird automatisch die Funktion NEUSTART ausgeführt. Wird das Standard-IPL benutzt, müssen anschließend die Speicherbereiche (Partitions) für die Anwenderprogramme eingerichtet werden (ALC). Nach dem Einrichten können Programme geladen und gestartet werden (LOD, RUN, EXC.).

Der IPL-Vorgang kann auch durch den Monitoraufruf IPL eingeleitet werden, ohne daß die Anlage vorher ausgeschaltet werden muß (siehe Monitoraufruf IPL).

NEUSTART

NEUSTART ist dann sinnvoll, wenn die beim letzten Ausschalten der Maschine oder Netzausfall unterbrochenen Programmläufe nicht mehr weitergeführt werden sollen. Eine früher getroffene Speichereinteilung bleibt erhalten. Evtl. bereits vorhandene Programme werden in den Zustand "fehlerhaft beendet" gesetzt. Sie können jederzeit neu gestartet werden (Monitoraufruf RUN).

Mit NEUSTART wird der MONITOR in Grundstellung gebracht und gestartet. Damit ist das Betriebssystem arbeitsfähig. Jetzt können Programme geladen und gestartet oder die Speicherbereiche neu eingeteilt werden (ALC, LOD, RUN, EXC.).

CONTINUE

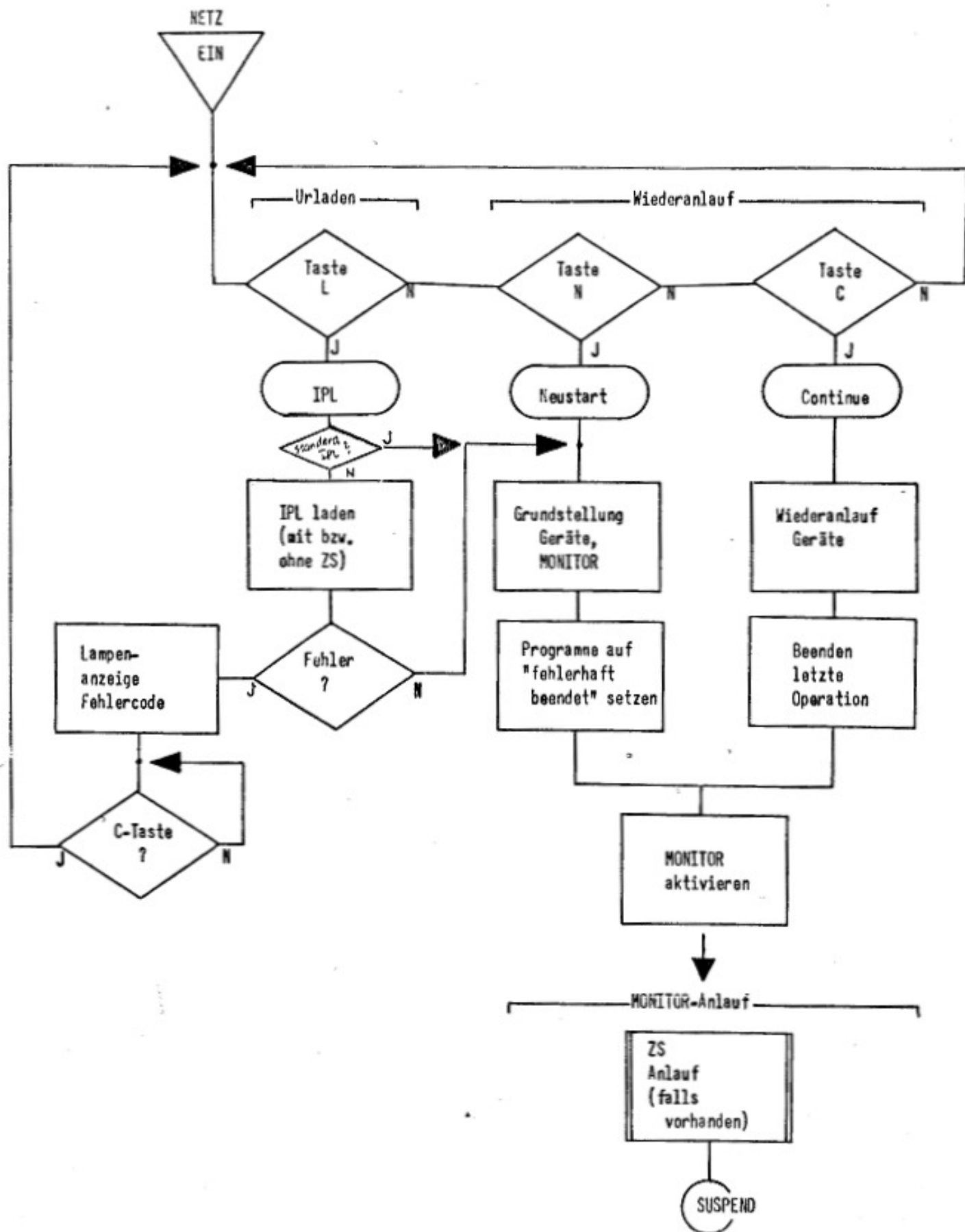
CONTINUE ist immer dann notwendig, wenn die beim letzten Ausschalten oder Netzausfall unterbrochenen Programmläufe unbedingt bei der unterbrochenen Stelle wieder fortgesetzt werden sollen.

6.10.76/EE 41/BS/2a

BEDIENUNG

Hinweis

Modifikationen und zusätzliche Bedienungsmaßnahmen sind der entsprechenden Beschreibung 'Zusatzsoftware' zu entnehmen.

Ablaufdiagramm "Maschinenanlauf"

4.1 BEDIENUNGAllgemeines

Wenn ein System für mehrere Anwendungsprogramme eingerichtet ist, bleibt dieser Zustand solange erhalten, bis eine Reorganisation des Systems durchgeführt wird. In eine Programmpartition können beliebige Programme geladen und gestartet werden.

Einrichten der Partitions

Das Einrichten der Partitions wird mit dem Monitoraufruf ALC vorgenommen. Die Partitions sind die Speicher- oder Laufbereiche der Anwenderprogramme. Es können je nach Bedarf zwischen 1 und 6 Partitions definiert werden. Die Partitions sind einzeln durchnummeriert. Die Nummerierung ist gleichzeitig die Priorität der in den Partitions laufenden Programme. Die niedrigste Partitionsnummer (= 0) hat die höchste Priorität, die höchste Partitionsnummer (= 7) hat die niedrigste Priorität.

Partition 0 = Priorität 0 (höchste Priorität)
Partition 7 = Priorität 7 (niedrigste Priorität)

Die Prioritäten 0 und 1 sind für das Betriebssystem reserviert und können vom Bediener nicht verwendet werden.

Nach der erstmaligen Inbetriebnahme der Anlage müssen unbedingt ein oder mehrere Speicherbereiche eingerichtet werden. Danach ist kein Einrichten der Bereiche auch nach Netzausfall oder Abschalten der Anlage mehr notwendig, es sei denn, die Speichereinteilung soll neu vorgenommen werden.

Laden und Starten von Programmen

Programme können mit dem Monitoraufruf LOD geladen und mit dem Monitoraufruf RUN gestartet werden. Soll ein Programm geladen und anschließend automatisch gestartet werden, ist der Monitoraufruf EXC zu verwenden.

Anhalten von Programmen

Programme können durch den Monitoraufruf STP für unbestimmte Zeit angehalten, d.h. in ihrem Programmlauf unterbrochen werden. Mit dem Monitoraufruf CNT kann der unterbrochene Programmlauf wieder fortgesetzt werden.

Beenden von Programmen

Gestartete Programme beenden sich normalerweise durch den Befehl SVC (TERMINATE) selbst. Es kann sich jedoch u.U. die Notwendigkeit ergeben, ein Programm durch Bedienereingriff zu beenden. Dies muß mit dem Monitoraufruf CAN erfolgen. Es können nur Anwenderprogramme beendet werden.

Inaktivität des Systems

Wenn das System untätig ist, d.h., zur Zeit keine Programmaktivitäten vorliegen, weil Programme auf das Fertigwerden von Ein/Ausgabeoperationen, das Eintreten von Ereignissen warten, oder noch kein Programm gestartet wurde, leuchtet die Systemanzeigelampe IDL . Sobald irgendein Programm aktiv wird, erlischt die Lampe wieder.

Internfehler

Internfehler werden, falls sie in einem Anwenderprogramm auftreten, vom Monitor in der Systemzeile angezeigt. Das Programm wird automatisch fehlerhaft beendet. Dies hat keinen Einfluß auf andere Programme. Sie laufen ungestört weiter.

Tritt ein Internfehler im Monitor auf (Speicher defekt, o.ä.), wird der Monitor ebenfalls fehlerhaft beendet. In diesem Fall leuchtet die Systemanzeigelampe INT und der Monitor ist nicht mehr ansprechbar. Das System muß mit IPL neu eingeleitet werden (siehe Inbetriebnahme des Systems).

Service-Routinen und Testhilfen

Zur Unterstützung des Bedieners stehen verschiedene Service-Routinen und Testhilfen zur Verfügung. Sie werden mit Hilfe von Monitoraufrufen aktiviert.

Folgende Funktionen können aufgerufen werden:

- Ändern des Tagesdatums (DAT)
- Nachrichten an ein Programm senden (MSG)
- Informationen über Partitiongröße und Programmzustand anzeigen (PSC)
- Informationen über den Zustand eines Gerätes anzeigen (DVC)
- Aktivieren des Testsystems DEBUG (TST)

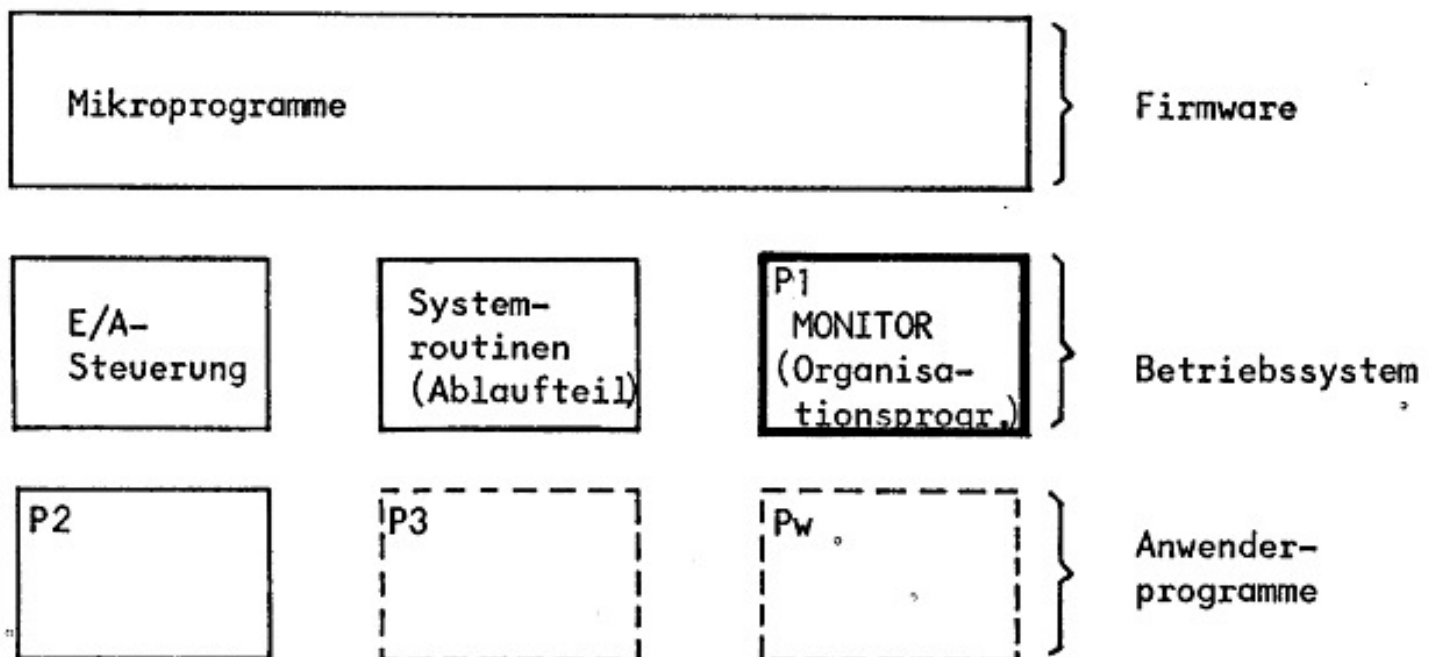
Die genaue Beschreibung der Monitoranweisung ist im Kapitel "MONITOR" angegeben.

4.3 MONITORAllgemeines

Der MONITOR ist das Organisationsprogramm des Betriebssystems (BS) :

Der MONITOR ist als selbständiges Programm unter der Programmpriorität 1 im Betriebssystem integriert.

Stellung des MONITOR im Betriebssystem:

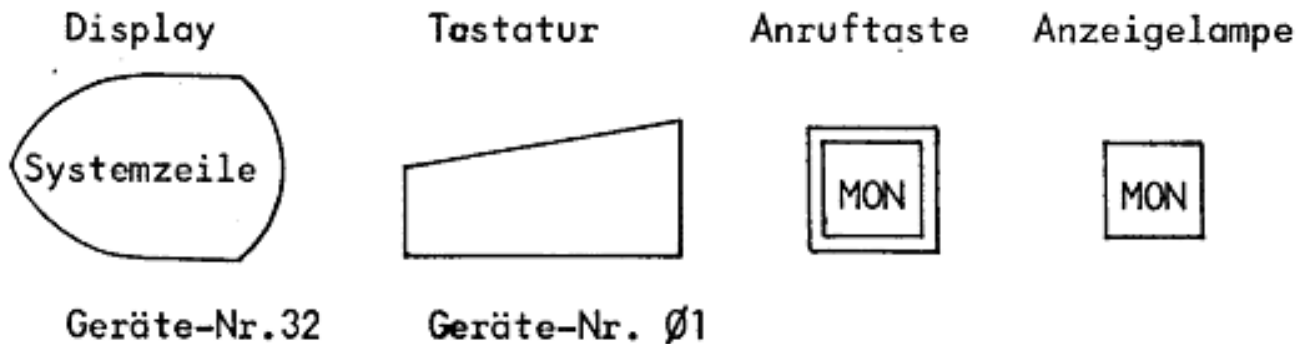


Der MONITOR übt im wesentlichen folgende Funktionen aus:

1. Steuerung der Kommunikation zwischen Bediener und Computer.
2. Laden und Starten von Programmen.
3. Beenden von Programmen.
4. Status- und Fehlerinformationen anzeigen.

Bedienungs- und Anzeigeelemente

Für die Bedienung des Monitors sind folgende Bedienungselemente vorhanden:



Die Systemzeile ist die erste Zeile des Displays und wird ausschließlich vom Monitor für Nachrichten an den Bediener und zur Anzeige der Monitoraufrufe benutzt.

Die Tastatur dient zur Eingabe der Monitoraufrufe.

Mit der Anruftaste wird der Monitor aktiviert. Sie muß bestätigt werden, wenn Monitoraufrufe eingegeben oder Nachrichten quittiert werden sollen.

Die Anzeigelampe MON leuchtet immer dann, wenn der Monitor aktiv ist.

Quittieren von Nachrichten

In der Systemzeile angezeigte Nachrichten (Systemmeldungen, Programm Meldungen, Fehlerhinweise) müssen vom Bediener quittiert werden. Dies ist notwendig, da bei gehäuftem Auftreten von Nachrichten wichtige Informationen unerkannt verloren gehen könnten. Eine nicht quittierte Nachricht ist durch das Zeichen "⌞" hinter dem Nachrichtenkopf gekennzeichnet.

Die Quittierung muß durch Drücken der Monitor-Anruftaste erfolgen. Die quittierte Nachricht wird durch das Zeichen "*" hinter dem Nachrichtenkopf gekennzeichnet.

↓
 Noch nicht quittierte Nachrichten

SYS ⌞ -----	
MSG ⌞ -----	
MON ⌞ -----	

↓
 Quittierte Nachrichten

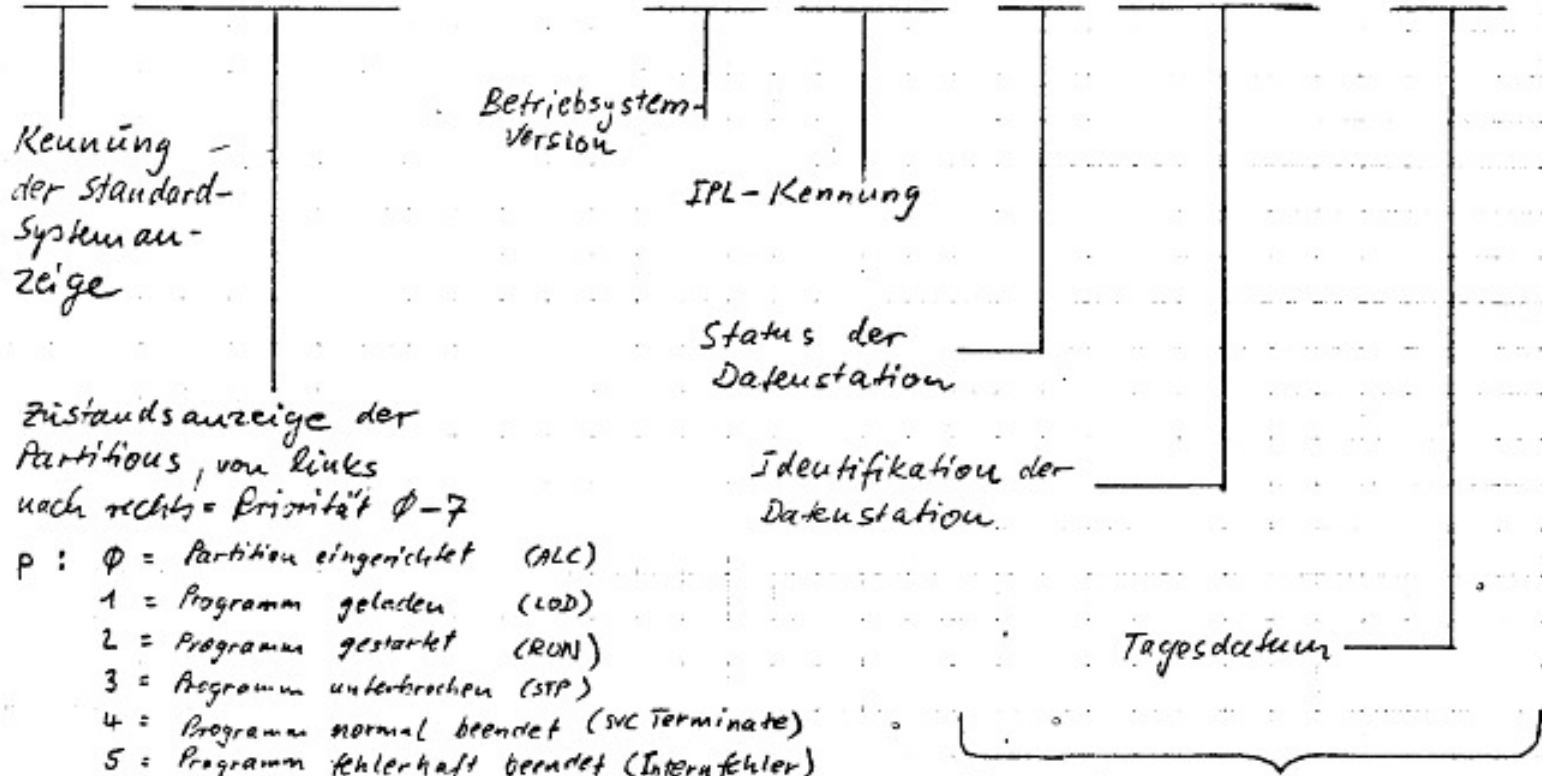
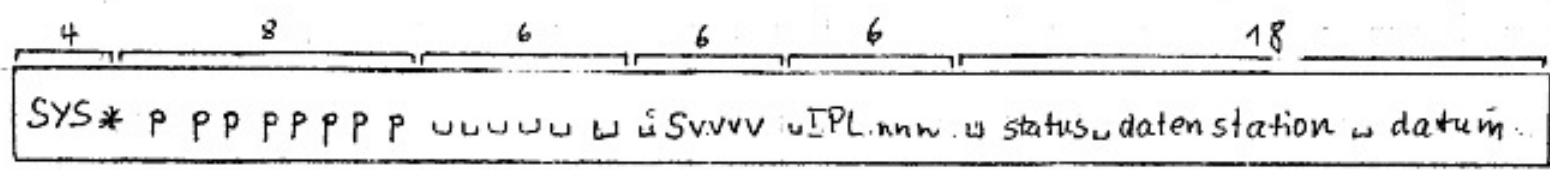
SYS * -----	
MSG * -----	
MON * -----	

MONITOR

Standard-Systemanzeige (SYS)

Die Standard-Systemanzeige ist permanent in der Systemzeile abgebildet und wird mit Verminderter Helligkeit angezeigt. Mit der Standard-Systemanzeige werden Informationen über den augenblicklichen Zustand der Maschine zur Verfügung gestellt.

Format:



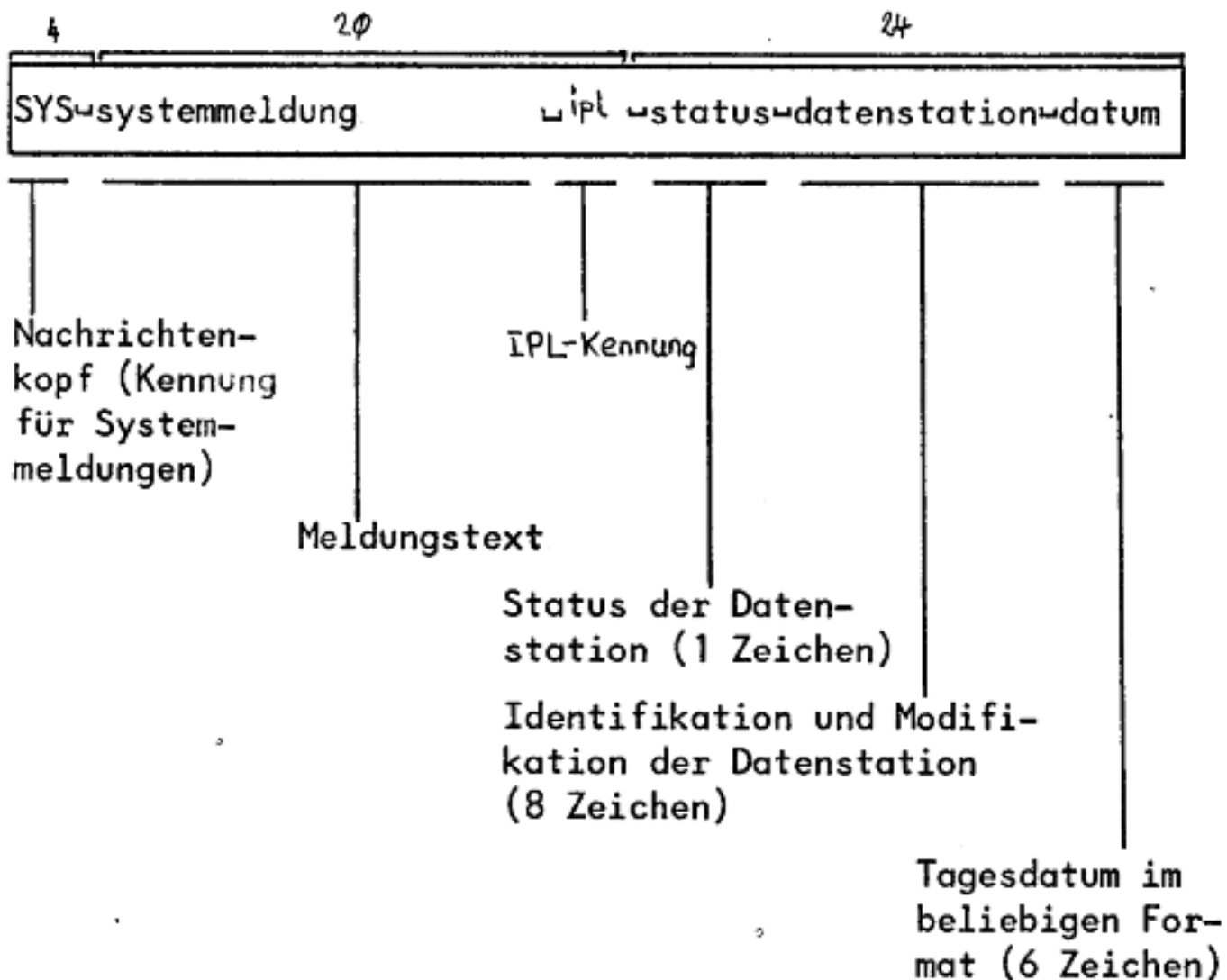
Zustandsanzeige der Partitions, von links nach rechts = Priorität 0-7

- p : 0 = Partition eingerichtet (ALC)
- 1 = Programm geladen (LOD)
- 2 = Programm gestartet (RUN)
- 3 = Programm unterbrochen (STP)
- 4 = Programm normal beendet (SVC Terminate)
- 5 = Programm fehlerhaft beendet (Interufehler)
- 6 = Programm abgebrochen (CAN)
- 8 = Grundstellung bzw. undefiniert

(siehe Systemmeldungen)

Systemmeldungen (SYS)

Systemmeldungen werden ausschließlich vom Betriebssystem, bzw. vom Monitor in der Systemzeile angezeigt. Die Meldungen sind im Nachrichtenkopf mit "SYS" gekennzeichnet und müssen stets vom Bediener quittiert werden.

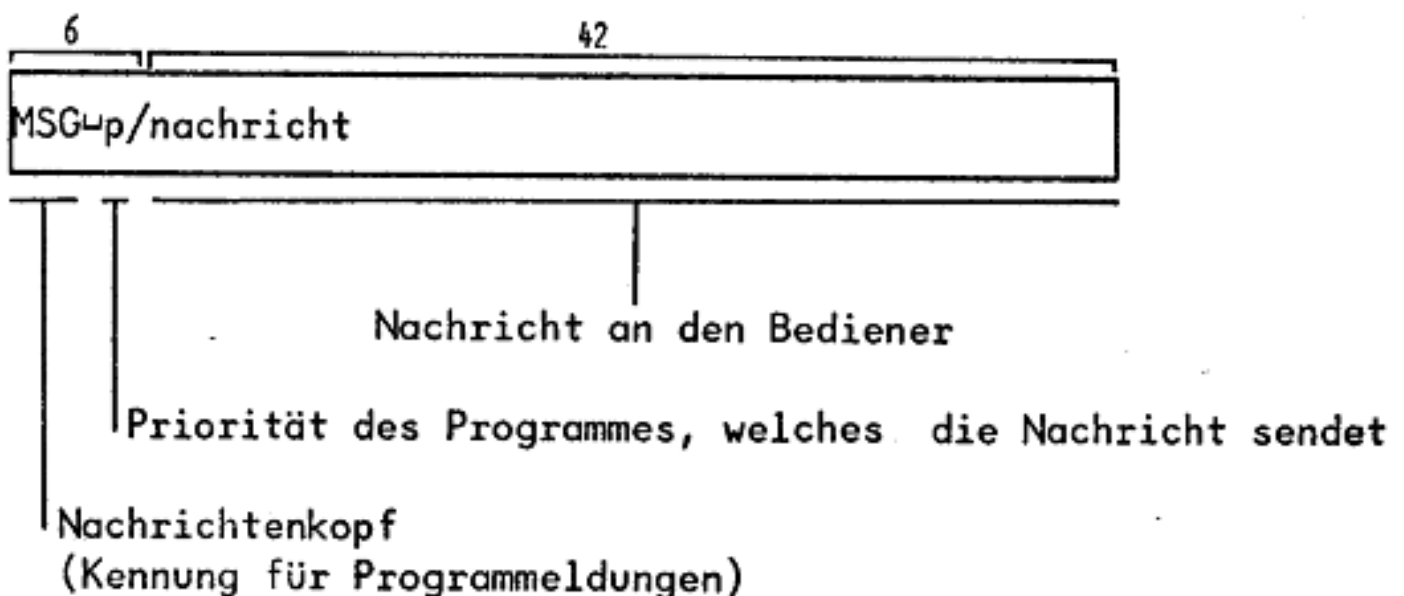
Format:

Folgende Systemmeldungen können auftreten:

Meldungstext	Bedeutung
p/INT-nr-xxxxxxx	<p>Ein Programm mit der Priorität "p" ist aufgrund eines Internfehlers beendet worden.</p> <p>p = Priorität des beendeten Programmes</p> <p>nr = Internfehler-Nummer</p> <p>xxxxxxx = dezimale Speicheradresse desjenigen Befehls, relativ zum Partitionanfang, bei dem der Internfehler auftrat.</p>
p/DVC-gg-RESRV	<p>Ein Programm mit der Priorität "p" wartet auf die Freigabe eines Gerätes "gg", welches bereits von einem anderen Programm reserviert ist.</p> <p>p = Priorität des wartenden Programmes</p> <p>gg = Geräte-Nummer</p>
p/MSG-REJCT	<p>Eine mit dem Monitoraufruf MSG abgesandte Nachricht konnte von dem Empfangsprogramm nicht übernommen werden (siehe Monitoraufruf MSG).</p> <p>p = Priorität des Empfangsprogrammes</p>

Programm Meldungen (MSG)

Anwenderprogramme können beliebige Nachrichten in der Systemzeile anzeigen. Die Programm Meldungen sind im Nachrichtenkopf mit "MSG" gekennzeichnet und müssen stets vom Bediener quittiert werden.

Format:

Bedienungsablauf:

1. MON-Anruftaste drücken.

Vorhergehende Tastatureingaben müssen mit einer Auslösetaste abgeschlossen sein.

2. Wenn die MON-Lampe leuchtet und in der Systemzeile die Kennung MON erscheint, kann der gewünschte Aufruf über die Tastatur eingegeben werden.

3. Alle Eingabeparameter müssen mit mindestens 1 Leerzeichen (= Leertaste = blank) voneinander getrennt eingegeben werden. Mit der Korrekturtaste können im Fehlerfall die eingegebenen Zeichen gelöscht und die Eingabe wiederholt werden.

4. Mit einer beliebigen Auslösetaste wird die Eingabe beendet.

MONITORMonitoraufruf IPL

Mit dem Monitoraufruf IPL (Initial Programme Load) wird das System initialisiert, d.h. in einen definierten ^{Aufgangs-}zustand _{gebracht}. Es wird zwischen dem Standard-IPL und einem speziellen IPL unterschieden. 1)

Beim Standard-IPL wird nur das System in Grundstellung gebracht. Es ist kein Datenträger notwendig.

Beim speziellen IPL werden Systemdaten von einer Magnetplatte (z.B. Floppy Disk) in den Speicher geladen.

Voraussetzung ist, daß sich kein Programm im Zustand "gestartet" oder "unterbrochen" im Speicher befindet, d.h. die Anlage inaktiv ist.

Der Monitoraufruf IPL wird bei der Inbetriebnahme¹⁾ des Systems mit der IPL-Taste zwangsläufig angefordert.

Format:IPL \square [xxx]Bedeutung:

IPL	Name des Aufrufs
xxx	Wahlweise. Kennung für speziellen IPL. Die Systemdaten müssen unter dem Namen "IPLxxx" auf Platte in der Systembibliothek SYSCOR gespeichert sein. Fehlt die Angabe xxx, wird ein Standard-IPL ausgeführt. 2)

Fehlermöglichkeiten:

EC: 1 = formal falsche Parameterangaben.

EC: 2 = Es sind noch aktive Programme im Speicher.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

1) siehe Abschnitt 4.1 Inbetriebnahme

Monitorkaufuf ALC

Mit dem Monitorkaufuf ALC (Allocate Memory) werden die Speicherbereiche (Partitions) für die einzelnen Anwenderprogramme eingerichtet, wobei maximal 6 Bereiche definiert werden können. Für jede Partition kann eine Größe zwischen 128 und 32767 Bytes angegeben werden. Voraussetzung ist, daß sich kein Programm im Zustand "gestartet" oder "unterbrochen" im Speicher befindet, d.h. die Anlage inaktiv ist.

Format:

ALC [*nnnnn] p2 [p3] [p4] [p5] [p6] [p7]

Bedeutung:

ALC	Name des Aufrufs
p2	Definition der Größe des 1. Speicherbereichs (Partition 2). Partition 0 und 1 sind für das Betriebssystem reserviert und können vom Bediener nicht angesprochen werden. Für p2 kann die Speichergröße entweder in Bytes oder Kilobytes angegeben werden. Speichergröße in Bytes : $128 \leq p2 \leq 32767$ Speichergröße in Kilobytes: $1 \leq p2 \leq 32$
p3-p7	Wahlweise. Definition weiterer Speicherbereiche (Partition 3-7) im selben Format wie p2.

nnnnn *wahlweise, Absolute dezimale Adresse, ab der die folgenden Partitionen eingerichtet werden so*

Fehlermöglichkeiten:

EC: 1 = Formal falsche Parameterangaben.

EC: 2 = Es sind noch aktive Programme im Speicher.

EC: 3 = Es steht nicht soviel Speicherplatz zur Verfügung, wie angegeben wurde.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

1) Wird nur für Sonderfälle, bzw. vom Service-Techniker verwendet.

Monitoraufruf DAT

Mit dem Monitoraufruf DAT (Enter Date) wird das Tagesdatum eingegeben, in den ECR-Bereich übernommen und in der Systemzeile angezeigt. Das Datum wird ungeprüft abgespeichert. Der Bediener ist für die Richtigkeit der Angaben selbst verantwortlich. Der Aufruf kann jederzeit wiederholt werden.

Format:

DAT = datum

Bedeutung:

DAT	Name des Aufrufs
datum	Tagesdatum in beliebigem Format, je nach Vorschrift des Anwenders. z.B.: TTMMJJ Es werden ausschließlich die ersten 6 Zeichen übernommen.

Fehlermöglichkeiten:

EC: 1 = Parameter fehlt.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Monitoraufruf LOD

Mit dem Monitoraufruf LOD (Load Phase) kann eine beliebige Programmphase in einen freien Speicherbereich (Partition) geladen werden. Die Programmphase kann aus der Platten-Systemdatei SYSCOR oder über DFÜ geladen werden. Eine Programmphase kann nur dann geladen werden, wenn die angegebene Partition "frei" ist, bzw. ein zu überladendes Programm sich im Zustand "geladen" oder "beendet" befindet.

Format:

LOD \cup P \cup gg \cup name \cup [vadr]

Bedeutung:

LOD	Name des Aufrufs
p	Priorität und gleichzeitig auch Partition (Speicherbereich), in welche die Programmphase geladen werden soll ($2 \leq p \leq 7$).
gg	Geräte-Nr. des Ladegerätes gg = 7ϕ Magnetplatte (z.B. Floppy Disk)
name	Name der zu ladenden Programmphase (1-8 α -num. Zeichen).
vadr	Verschiebeadresse in dezimaler Darstellung ($\emptyset \leq vadr$) Bei Angabe der Verschiebeadresse wird die Programmphase um die Verschiebeadresse erhöht in die Partition geladen. Ist die Verschiebeadresse = \emptyset , oder entfällt die Angabe, wird die Programmphase am Beginn der Partition geladen.

Fehlermöglichkeiten:

EC: 1 = Formal falsche Parameterangaben.

EC: 2 = Die angegebene Programmphase kann nicht geladen werden, weil bereits ein anderes Programm in der Partition aktiv ist, oder kein Speicherplatz eingerichtet ist.

EC: 3 = Falsche Geräte-Nr. bzw. Gerät nicht vorhanden.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Gerätefehler:

Tritt beim Laden ein Gerätefehler auf, wird der Fehlercode 3stellig mit führenden Nullen in der Systemzeile angezeigt.

Fehlercodes (Floppy Disk):

fc	Bedeutung
004	Lesefehler
006	Bibliothek SYSCOR nicht gefunden
007	Gerät defekt oder nicht geladen
008	Platte nicht initialisiert oder Gerät defekt
011	Positionierfehler
012	Gerät defekt
035	Programmname nicht in der Bibliothek vorhanden
041	Plattenüberlauf
053	Speicherüberlauf bei den Ladeadressen

Das angegebene Programm kann zum Teil schon im Speicher stehen. Für diese Partition ist dann nur noch der Monitoraufruf LOD oder EXC zulässig.

Monitoraufruf RUN

Mit dem Monitoraufruf RUN (Run Programme) wird das im Aufruf angegebene Programm gestartet. Ein Programm kann nur dann gestartet werden, wenn es sich im Zustand "geladen" oder "beendet" befindet.

Format:

RUN \leftarrow {^pname}

Bedeutung:

RUN	Name des Aufrufs.
{ ^p name}	Angabe des Programmnamens, bzw. der Programmpriorität. p = Priorität ($2 \leq p \leq 7$) $name$ = Programmname (1-8 α -num. Zeichen)

Fehlermöglichkeiten:

- EC: 1 = Formal falsche Parameterangaben.
- EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Programm geladen oder kein Speicherplatz eingerichtet.
- EC: 3 = Programm vorhanden, bzw. unter der angegebenen Priorität ist ein Programm geladen, aber nicht in einem der Zustände "geladen" oder "beendet".

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Monitoraufruf EXC

Mit dem Monitoraufruf EXC (Execute Programme) kann eine beliebige Programmphase in einen freien Speicherbereich geladen werden. Das geladene Programm wird anschließend automatisch gestartet.

Der Monitoraufruf EXC ist eine Zusammenfassung der Aufrufe LOD und RUN.

Format:

EXC \cup p \cup gg \cup name [\cup vadr]

Bedeutung:

(Siehe LOD-Beschreibung)

Fehlermöglichkeiten:

(Siehe LOD-Beschreibung)

Monitoraufruf STP

Mit dem Monitoraufruf STP (Stop Programme) wird das im Aufruf angegebene Programm angehalten und in einen Wartezustand versetzt. Mit dem Monitoraufruf CNT (Continue Programme) kann der Programmablauf wieder fortgesetzt werden.

Format:

$$\text{STP } \left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$$
Bedeutung:

STP	Name des Aufrufs.
$\left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$	Angabe des Programmnamens, bzw. der Programmpriorität. p = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α -num. Zeichen)

Fehlermöglichkeiten:

- EC: 1 = Formal falsche Parameterangaben.
 EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Programm geladen.
 EC: 3 = Programm ist vorhanden, bzw. unter der angegebenen Priorität ist ein Programm geladen, aber noch nicht gestartet.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Monitoraufruf CNT

Mit dem Monitoraufruf CNT (Continue Programme) kann ein im Wartezustand befindliches Programm wieder aktiviert und der Programmablauf fortgesetzt werden. Der Wartezustand kann entweder durch einen Monitoraufruf STP (Stop Programme) oder durch das Programm selbst (Befehl SUSPEND) verursacht sein. Hat sich das Programm selbst in den Wartezustand versetzt, ist CNT nur dann wirksam, wenn das Programm CNT bei SUSPEND als zulässig kennzeichnet.

Format:

$$\text{CNT} \sim \left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$$
Bedeutung:

CNT	Name des Aufrufs.
$\left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$	Angabe des Programmnamens, bzw. der Programmpriorität. $p = \text{Priorität } (2 \leq p \leq 7)$ $\text{name} = \text{Programmname } (1-8 \leftarrow \text{num. Zeichen})$

Fehlermöglichkeiten:

- EC: 1 = Formal falsche Parameterangaben.
- EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Programm geladen.
- EC: 3 = Programm ist vorhanden, bzw. unter der angegebenen Priorität ist ein Programm geladen, befindet sich aber nicht im Wartezustand.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Monitoraufruf CAN

Mit dem Monitoraufruf CAN (Cancel Programme) wird das im Aufruf angegebene Programm beendet, d.h. der Programmablauf abgebrochen und evtl. noch belegte Geräte freigegeben. *) Das beendete Programm kann bei Bedarf mit dem Monitoraufruf RUN (Run Programme) wieder gestartet werden.

Format:

CAN - {^pname}

Bedeutung:

CAN	Name des Aufrufs.
{ ^p name}	Angabe des Programmnamens, bzw. der Programmpriorität. <p>p = Priorität (2 ≤ p ≤ 7) name = Programmname (1-8 α-num. Zeichen)</p>

Fehlermöglichkeiten:

- EC: 1 = Formal falsche Parameterangaben.
EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Programm geladen.
EC: 3 = Programm ist vorhanden, bzw. unter der angegebenen Priorität ist ein Programm geladen, aber noch nicht gestartet.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

*) Die Geräte werden nicht physikalisch entriegelt;

Monitoraufruf MSG

Mit dem Monitoraufruf MSG (Send Message) kann eine beliebige Nachricht an ein Programm übergeben werden. Voraussetzung ist jedoch, daß das angegebene Programm für den Empfang von externen Nachrichten eingerichtet ist (SUSPEND, RECV).

Ablauf: Der Monitor versucht, die Nachricht an das angegebene Programm weiterzuleiten.

Ist nach einer bestimmten Wartezeit die Nachricht nicht übernommen worden, wird das Senden der Nachricht mit dem Fehlerhinweis "MSG/REJCT" beendet (siehe Systemmeldungen).

Format:

$$\text{MSG} \cup \left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\} \cup \text{text}$$
Bedeutung:

MSG	Name des Aufrufs .
$\left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$	Angabe des Programmnamens, bzw. der Programmpriorität. p = Priorität ($2 \leq p \leq 7$) name= Programmname (1-8 α -num. Zeichen)
text	Beliebiger Text, maximal 38 Zeichen lang.

Fehlermöglichkeiten:

- EC: 1 = Formal falsche Parameterangaben, bzw. ein anderer MSG-Aufruf ist noch aktiv.
- EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Programm geladen oder kein Speicherplatz eingerichtet.
- EC: 3 = Programm ist vorhanden, bzw. unter der angegebenen Partition ist ein Programm geladen, aber noch nicht gestartet.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Monitoraufruf PSC

Mit dem Monitoraufruf PSC (Process State Control) können die aktuellen Programmzustände abgefragt werden. Die Ausgabe der Informationen erfolgt auf der Systemzeile.

Format:

$$\text{PSC} = \left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$$
Bedeutung:

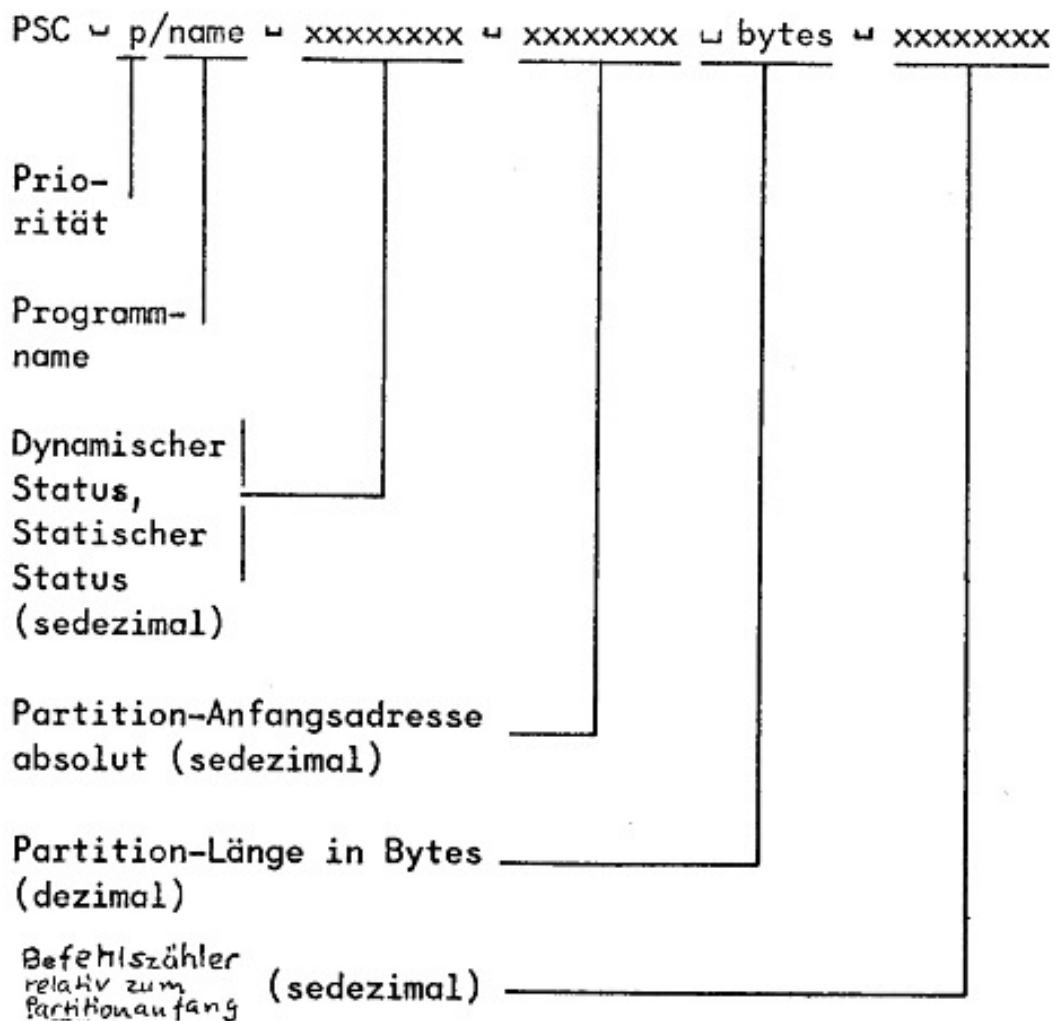
PSC	Name des Aufrufs.
$\left\{ \begin{array}{l} p \\ \text{name} \end{array} \right\}$	Angabe des Programmnamens, bzw. der Programmpriorität. $p = \text{Priorität } (2 \leq p \leq 7)$ $\text{name} = \text{Programmname } (1-8 \text{ \textasciitilde num. Zeichen})$

Fehlermöglichkeiten:

EC: 1 = Formal falsche Parameterangaben.

EC: 2 = Programm nicht vorhanden, bzw. unter der angegebenen Priorität ist kein Speicherplatz eingerichtet.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Ausgabeformat:Dynamischer Status

0000	_____	Grundstellung
0001	_____	Programm befindet sich im "Bereit-Zustand"
0002	_____	Programm befindet sich im "Blockiert-Zustand"
0004	_____	Programm befindet sich im "Aktiv-Zustand"
0008	_____	Programm befindet sich im "Warte-Zustand"

Statischer Status

_____	0000	Grundstellung
_____	0002	Programm ist "geladen" (Monitoraufruf LOD)
_____	0004	Programm ist "gestartet" (Monitoraufruf RUN)
_____	0008	Programm ist "unterbrochen" (Monitoraufruf STP)
_____	0100	Programm ist "beendet" (SVC TERMINATE)
_____	0200	Programm ist "fehlerhaft beendet" (Internfehler)
_____	0400	Programm ist "durch Bedieneringriff beendet" (Monitoraufruf CAN)
_____	0001	Partition ist "eingrichtet" (Monitoraufruf ALC)

Monitoraufruf DVC

Mit dem Monitoraufruf DVC (Device State Control) können die aktuellen Gerätezustände abgefragt werden. Die Ausgabe der Informationen erfolgt auf der Systemzeile.

Format:

DVC \rightarrow gg

Bedeutung:

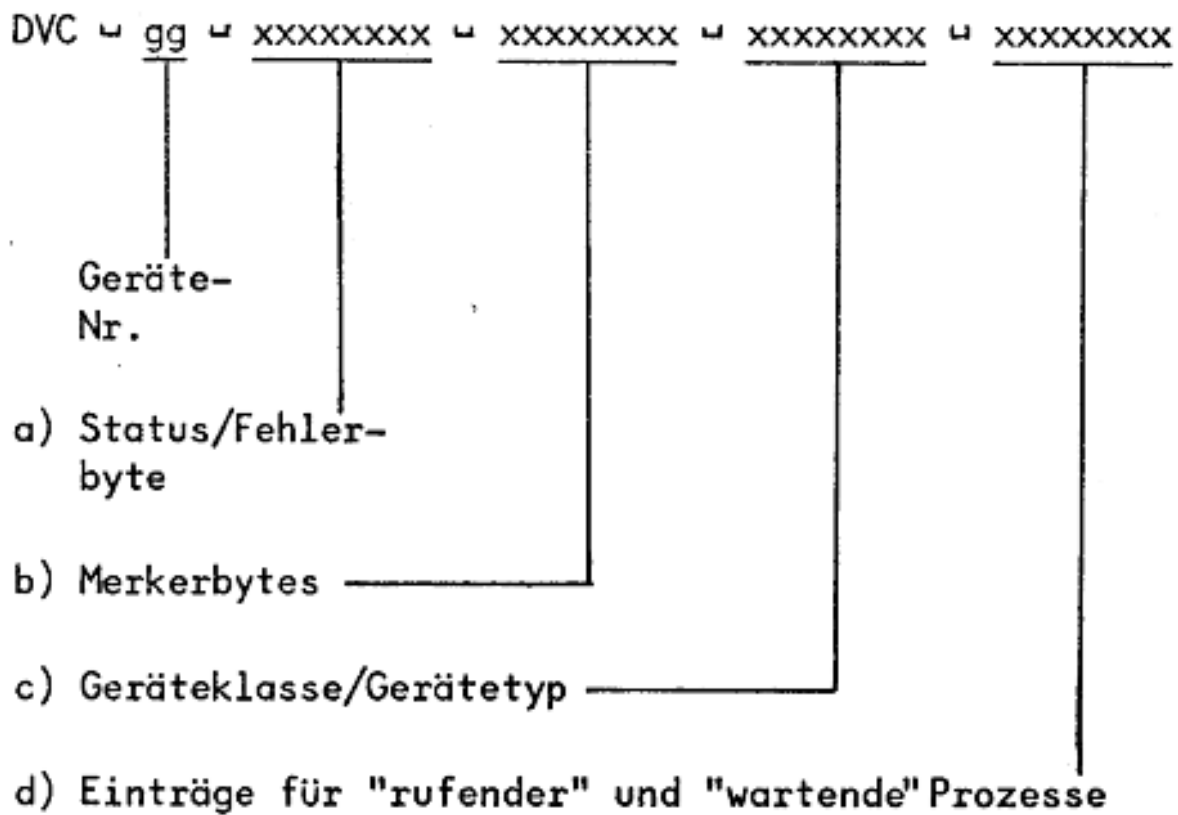
DVC	Name des Aufrufs
gg	Angabe der Gerätenummer in dezimaler Darstellung z.B.: FDE gg = 70 DFÜ gg = 40 Display gg = 30 Drucker gg = 10 Tastatur gg = 01

Fehlermöglichkeiten:

EC: 1 = Parameter fehlt.

EC: 2 = Das angegebene Gerät ist im System nicht bekannt,
 bzw. nicht angeschlossen.

Der Aufruf wird abgewiesen und die Funktion nicht ausgeführt.

Ausgabeformat:

Die Angaben a) bis d) werden in sedezimaler Darstellung je 2 Bytes angezeigt.

Monitoraufruf TST

Mit dem Monitoraufruf TST (Actuate Test System) wird das Testsystem DEBUG aktiviert.

Die erste Phase des Testsystems wird mit der höchsten Priorität (= Ø) in den Speicher geladen und gestartet. Die DEBUG-Routinen müssen auf der Systemplatte in der Systembibliothek "SYSCOR" permanent zur Verfügung stehen.

Eine genaue Beschreibung der Funktionen ist im Abschnitt 'Testsystem' nachzulesen.

Format:

TST~

Bedeutung:

TST	Name des Aufrufs
-----	------------------

Fehlermöglichkeiten:

Beim Laden der Phase können Gerätefehler auftreten (Fehlercodes und Bedeutung siehe LOD-Beschreibung).

30.9.76/EE 41/BS/

5. GENERIERUNG EINES ANWENDERSYSTEMS

Dienstprogramm GUMS

~~(wird ergänzt)~~

1. Allgemeines

Mit dem Dienstprogramm GUAS (= Generate User Application System) kann ein Anwendersystem für das Betriebssystem TA 1069 generiert werden. Da für das Dienstprogramm die MBKE benötigt wird, ist es nur auf der TA 1000 ablauffähig.

Speicherbedarf:

Das Programm benötigt ca. 3,5 K-Bytes.

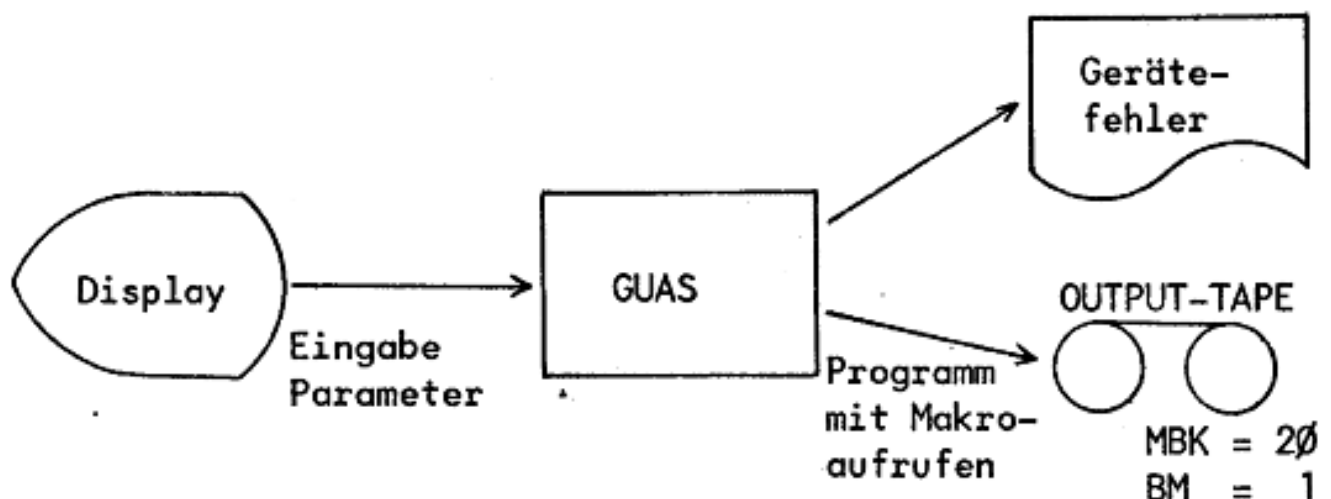
Lademodus:

Das Programm kann von Band oder Platte geladen werden.

Mindestausstattung:

- 1 DISPLAY
- 1 FDE
- 1 MBK
- 1 DRUCKER

Datenfluß:



2. Funktionen

Das Dienstprogramm GUAS generiert mit über DISPLAY einzugebenden Systemparametern ein Programm mit Makroaufrufen (MBKE = 20, BM = 1). Dieses Programm muß anhand einer zum Dienstprogramm gehörigen Makrodatei expandiert und assembliert werden.

Partitions, die im Anwendersystem geladen oder gestartet werden sollen (siehe 3. (a2)), werden im Assemblerlauf - PASS 2 entsprechend der Reihenfolge der PCB-Generierung (siehe 3. (a)) von der entsprechenden Bandmarke kopiert.

Das entstehende Objektprogramm bildet das gewünschte Anwendersystem für die TA 1069. Es muß mit dem Dienstprogramm SYSCON unter dem Namen IPL $\alpha \alpha \alpha$ (α = Alpha-numerisches Zeichen) auf Platte kopiert werden.

Es wird vom BS nach Einschalten der TA 1069 ^{und} Angabe des IPL-Namens $\alpha \alpha \alpha$ von der Platte geladen (siehe auch 6.).

GUAS - Makros: SYBE1, PCB, LINK, SYBE2, ~~ZS~~.

3. Versorgung

Das Programm muß über Bildschirm mit folgenden Systemparametern versorgt werden:

a) Parameter für die PCBs:

- a1) PROGNA = Name der einzurichtenden Partitionen
(max. 6 Zeichen)
- a2) S = Status der Partition; es sind folgende
Eingaben möglich:
E , d.h. Partition eingerichtet
L , d.h. Partition geladen
S , d.h. Partition gestartet
- a3) LAENG = Länge der einzurichtenden Partition
- a4) STRTA = Startadresse des Programms, relativ
zum Partitions-Anfang
- a5) BM = Bandmarke, hinter der das zugehörige
Programm steht (2stellige Dezimalzahl)
- a6) PARAN = Partition-Anfangsadresse, relativ zum
Beginn des Anwenderspeichers

Anmerkung:

1. Wird keine Partition-Anfangsadresse angegeben (PARAN = 000000),
so wird die Partition unmittelbar an die vorherige angehängt.
2. Bei Partitions, die nur eingerichtet sein sollen, sind die
Eingaben für STRTA und BM bedeutungslos.

b) Weitere zu versorgende Systemparameter:

b1) Länge des DFÜ-Puffers (geradzahlig!)

b2) Länge des Zusatzsoftware-Bereichs (geradzahlig!)

b3) Identifikation Datenstation (8stelliges Literal)

b4) Identifikation Zusatzsoftware (3stellige Dezimalzahl)

Bemerkung:

Adress- und Längenangaben erfolgen als 5stellige Dezimalzahl und relativ zum Beginn des Anwenderspeichers!

Programmierte Cursorbewegungstasten:

① = rechts

② = links

⓪ = Alpha-Beendigungstasten

4. Prüfungen und Meldungen

- a) Die über Bildschirm eingegebenen Systemparameter werden auf Plausibilität geprüft:

Bsp. LAENG = 35Ø*Ø wird als syntaktisch falsche Längenangabe erkannt (Sonderzeichen innerhalb Zahl!).

Bei (erkannter) falscher Eingabe erscheint auf dem Bildschirm die Meldung:

'***FALSCH EINGABE: WIEDERHOLEN!'

- b) Die Eingabe einer "PCB-Parameter-Zeile" (3. (a)) kann korrigiert werden durch Beantwortung der Frage:

'KORREKTUR NOETIG?' mit 'J'

- c) Zu große Längen- bzw. Adress-Eingaben können erst bei der Verarbeitung der Parameter festgestellt werden.
Meldung auf Drucker:

'FEHLER: BELEGTER KSP-BEREICH > 64 K'

Das Programm muß neu gestartet werden.

d) Gerätefehler für MBKE oder DISPLAY werden über Drucker in folgender Form ausgegeben:

'DVC/xx ERROR: yy'

Das Programm muß neu gestartet werden.

5. Beispiel für die Eingabe der Systemparameter

Bildinholt

1. EINGABE DER PARAMETER FUER PCBS:

PROGNA 5 LAENG STRTA BM (PARAM)

PR1 L 04000 02058 01 UUUUU

PR2 S 01000 00611 02 UUUUU

PR3 E 05300 UUUUU UU 00000

2. EINGABE SYSTEM-PARAMETER:

DFUE-PUFFER-BEREICH: 00000 BYTES (GERADZÄHLIG!)

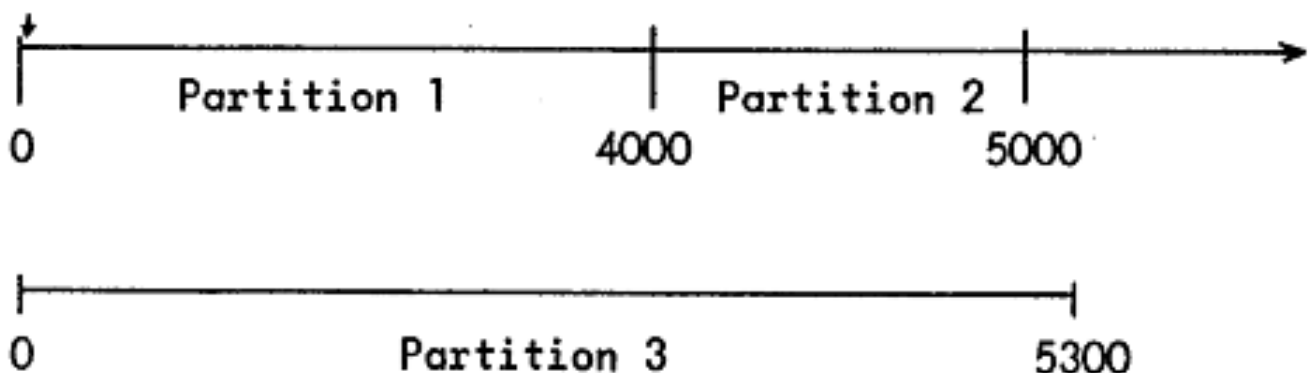
ZUSATZSOFTWARE-BER.: 00000 BYTES (GERADZÄHLIG!)

IDENTIF.-DATENSTAT.: UUUUUUUU

IDENTIF.-ZS : UU

Expandiertes Anwendersystem:

Beginn Anwenderspeicher



6. Vorgehensweise nach Erstellung des Anwendersystems mit GUAS

(1) Das auf Band generierte Anwendersystem mit SYSCON unter dem Namen IPL $\alpha\alpha\alpha$ (α = Alpha-numerisches Zeichen) auf Platte nach SYSCOR bringen.

(2) TA 1069 einschalten:

Auf Display erscheint:

MON-IPL

Eingabe des IPL-Namens $\alpha\alpha\alpha$ und Auslösen mit Taste (A).

6. Testsystem DEBUG

Allgemeines

Das Debug hat im Betriebssystem die Aufgabe, das Austesten der Anwenderprogramme zu erleichtern. Das Debug ist ein selbständiges Programm unter der Priorität \emptyset im Betriebssystem; es muß bei Bedarf geladen werden.

Speicherbedarf und Speicherbelegung

Für das Debug müssen ab der absoluten Adresse 6144 \emptyset (= 15.00.00.00) 4k Bytes im Kernspeicher reserviert werden. Beim Ladevorgang wird das Debug ab dieser Adresse in den Kernspeicher geladen.

Debug laden und starten

Das Debug wird mit dem Monitoraufruf TST geladen und gestartet und ist anschließend aufrufbar.

Voraussetzung:

Das Debug muß in der Systembibliothek "SYSCOR" unter dem Namen "SYSTEM" zur Verfügung stehen.

Das Debug übt im wesentlichen folgende Funktionen aus:

1. Kontrolle und Überwachung der Anwenderprogramme
2. Anzeige von Speicherinhalten
3. Ändern von Speicherinhalten
4. Programmmodifikation
5. Übertragen von Speicherbereichen in die Systembibliothek.

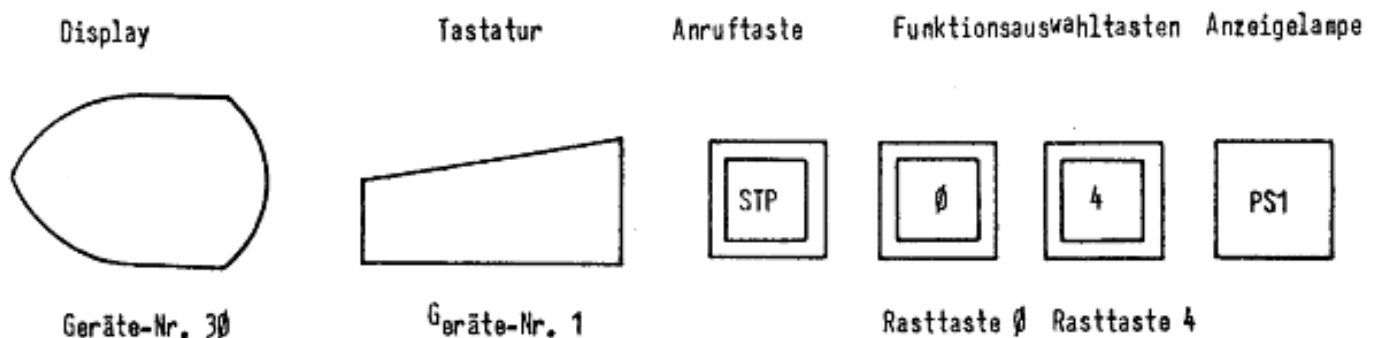
Testen von Programmen

Nicht unterbrechbare Programmabschnitte:

Programmabschnitte, die nach jeder Unterbrechungsverarbeitung (Bit 8 im linken Byte der Programmaske) unbedingt fortgesetzt werden, können in diesen Abschnitten durch keine Debugfunktion überwacht werden.

Bedienungs- und Anzeigeelemente

Für Bedienung und Anzeige des Debugs sind folgende Elemente vorhanden:



Das Display wird vom Debug benutzt für die Debugausgabe und für die Anzeige der Debuganweisungen.

Die Tastatur dient zur Eingabe der Debuganweisungen.

Die Anruftaste hat folgende Funktionen:

1. Debug aktivieren
2. Debuganzeige quittieren bzw. unterbrochene Prozesse fortsetzen (siehe Funktionsauswahl)
3. automatisch ablaufende Debugfunktionen unterbrechen

Die Funktionsauswahltasten haben folgende Bedeutung (siehe Funktionsauswahl):

1. Mit der Rasttaste 4 wird die Tastatur für die Eingabe der Debuganweisungen belegt.

2. Mit Rasttaste \emptyset wird in eine Schleife verzweigt, in der die vom Benutzer betätigten Rasttasten angezeigt werden. In dieser Schleife können für die Anwenderprogramme Rasttasten gesetzt bzw. zurückgesetzt werden. Diese Funktion wird z.B. bei automatisch ablaufenden Debugfunktionen benötigt.

Bei der Bedienung ist zu berücksichtigen, daß Tastatur und Display sowohl von Anwenderprogrammen als auch von Debug belegt wird.

Debug-Anzeige: Werden im Anwenderprogramm Daten in den Bildspeicher über die Tastatur eingegeben (BWO), so muß diese Eingabe mit einer Auslösetaste abgeschlossen sein, bevor die Debug-Anzeige erscheint.

Tastatureingabe: Werden im Anwenderprogramm Daten über die Tastatur eingegeben (ACS, ACPS), so muß diese Eingabe mit einer Auslösetaste abgeschlossen sein, bevor Debuganweisungen eingegeben werden können.

Funktionsauswahl

Die Funktionsauswahl wird über Rasttasten getroffen.

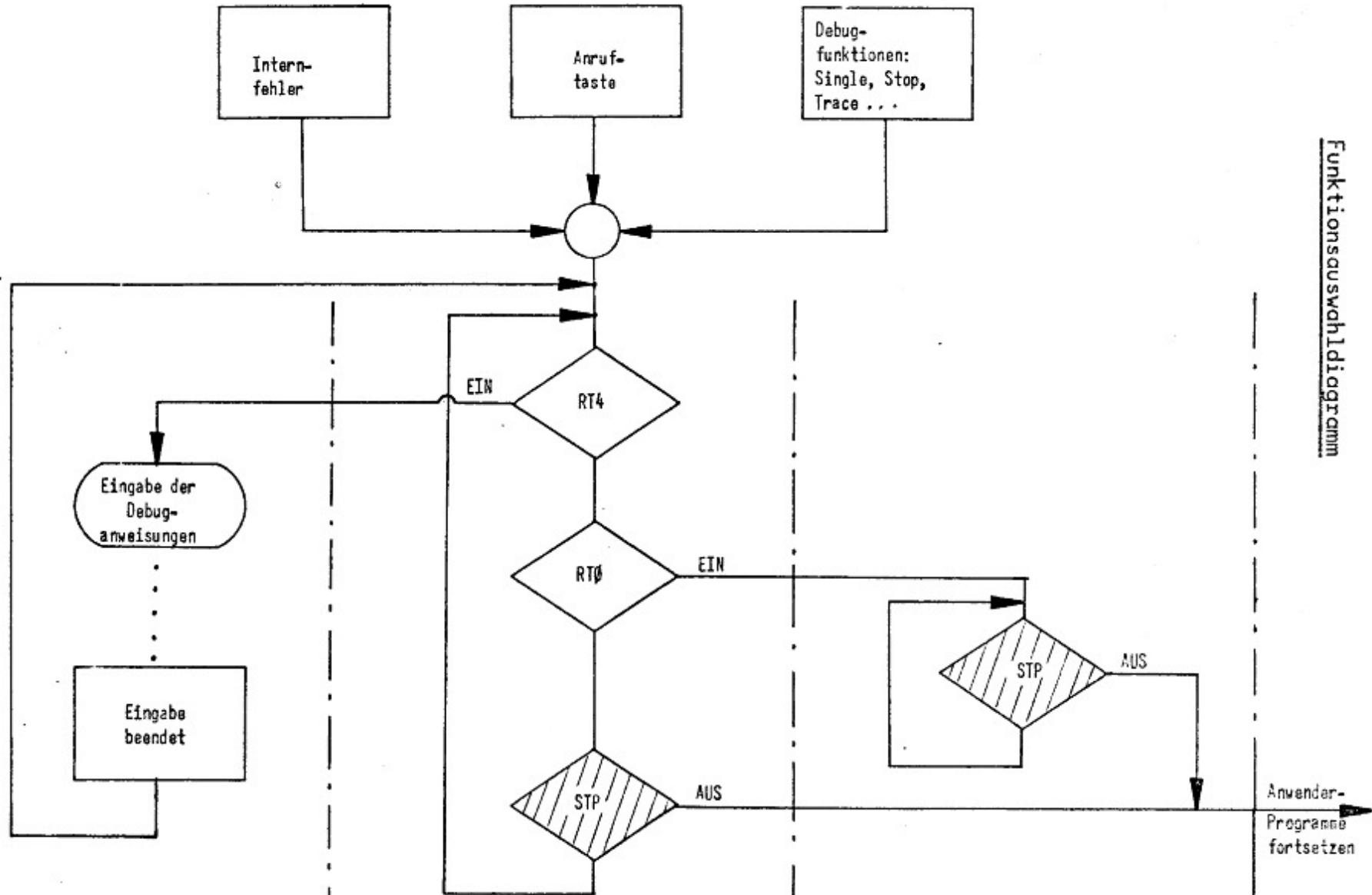
Die Funktionsauswahlschleife wird aktiviert durch:

1. Betätigen der Anruftaste
2. Debug-Funktion z.B. Trace
3. Unterbrechen einer automatisch ablaufenden Debug Funktion (z.B. automatisch Single) mit der Anruftaste
4. Internfehler
5. Nach Beenden der Eingabe einer Debuganweisung

Daß die Funktionsauswahlschleife aktiviert ist, wird angezeigt durch:

Anzeigelampe PS1, Anruftaste (STP) leuchtet

Aktivierung der Funktionsauswahl-Schleife durch:



Funktionsauswahldiagramm

Anzeige:

PS1

PS1, STP-leuchtet

PS1 -blinkt, STP-leuchtet
Anzeige der für die Anwenderprogramme
gesetzten Rasttasten

Ausgabe:

Debuganweisungen über die Tastatur

Funktionsauswahl über Rasttasten

Setzen von Rasttasten für Anwenderprogramme

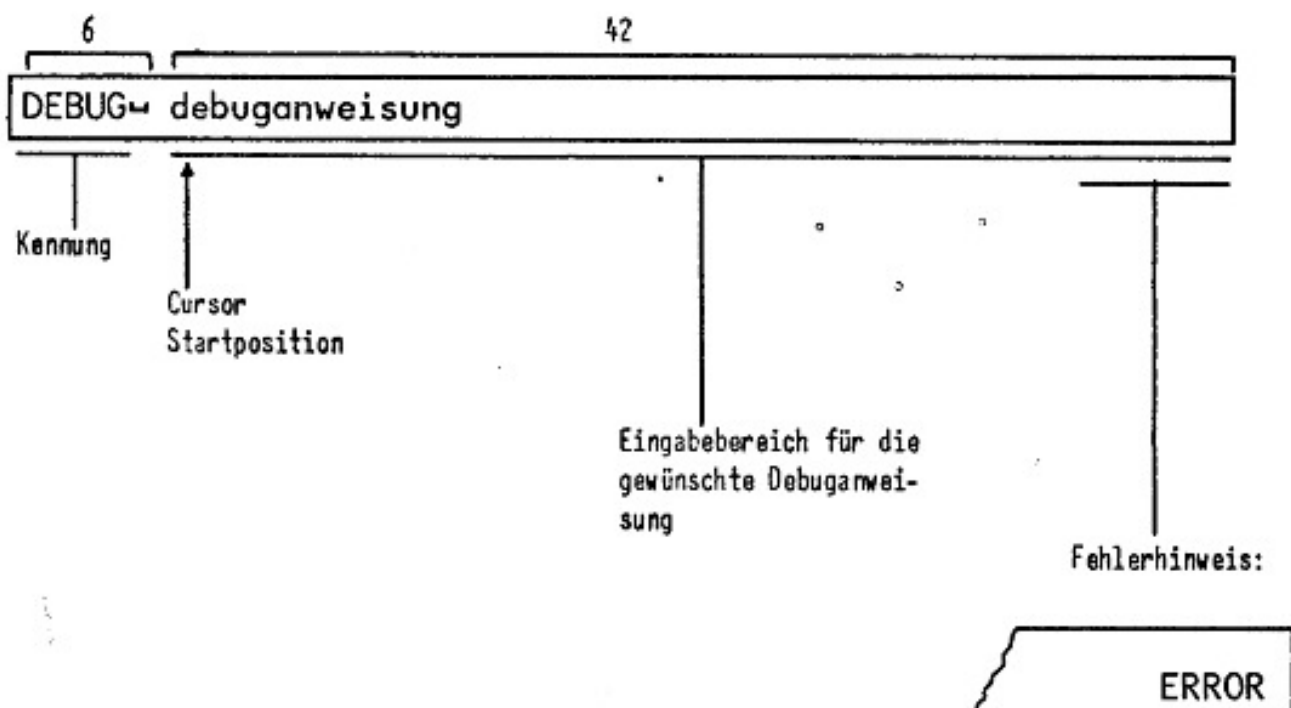
DEBUG

Eingabe der Debuganweisungen (RT4)

Debuganweisungen werden bei Bedarf vom Benutzer über die Tastatur eingegeben und auf dem Display angezeigt. Der Cursor zeigt die Position für die Eingabe an.



Nach Beenden der Eingabe wird zur Funktionsauswahlschleife verzweigt.

Fehlerhaft eingegebene Anweisungen werden angezeigt und können durch erneute Funktionsanwahl (RT4) korrigiert werden.

Format:

Bedienungsablauf:

Anzeige: PS1-Lampe, STP-RT-leuchtet

1. Funktionsauswahltaste (RT4) drücken.
Vorhergehende Tastatureingaben (in Anwenderprogrammen) müssen mit einer Auslösetaste abgeschlossen sein.
2. Wenn auf dem Display neben der Kennung DEBUG der Cursor erscheint, kann die gewünschte Anweisung über die Tastatur eingegeben werden.
3. Alle Eingabeparameter müssen mit mindestens 1 Leerzeichen (= Leertaste = blank) voneinander getrennt eingegeben werden. Mit den Cursorstasten ( , ) kann der Cursor in der Eingabezeile bewegt werden.
4. Mit den Auslösetasten kann die Eingabe beendet werden.

Debuganweisung SL

Mit der Debuganweisung SL (SINGLE) wird das in der Anweisung angegebene Programm im Einzelschritt ausgeführt. Nach jedem ausgeführten Befehl wird das Debug aktiv. Der ausgeführte Befehl wird angezeigt (siehe Debugausgabe). Anschließend kann der Bediener über die Funktionsauswahlschleife Funktionen wählen oder das Programm fortsetzen.

Format:

$$SL \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

SL	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1 - 8 α-num. Zeichen)

Fehlermöglichkeit:

Formal falsche Parameterangabe.

Programm nicht vorhanden.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung TR

Mit der Debuganweisung TR (TRACE) werden alle Programmverzweigungen des angegebenen Programmes kontrolliert. Sobald der lineare Ablauf des Programmes unterbrochen ist, wird das Debug aktiv und zeigt den ausgeführten Befehl, der die Verzweigung verursacht hat, und den Zielbefehl an. Anschließend kann der Bediener über die Funktionsauswahlschleife Funktionen auswählen oder das Programm fortsetzen.

Format:

$$TR \sim \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

TR	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α-num. Zeichen)

Fehlermöglichkeiten:

Formal falsche Parameterangabe.

Programm nicht vorhanden.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung ST

Mit der Debuganweisung ST (STOP) können Stopadressen (Befehlsadressen) des angegebenen Programmes eingegeben werden. Diese Adressen werden vom Debug überwacht.

Sobald eine Stopadresse erreicht ist, wird das Debug aktiv und gibt den zuletzt ausgeführten Befehl aus. Anschließend kann der Bediener über die Funktionsauswahlschleife Funktionen auswählen oder das Programm bis zur nächsten angegebenen Stopadresse fortsetzen.

Die Debuganweisung ST kann für maximal 2 Programme angegeben werden. Für ein Programm können nicht mehr als 4 Stopadressen angegeben werden.

Format:

$$ST \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\} \text{adr1} [\text{adr2}] [\text{adr3}] [\text{adr4}]$$
Bedeutung:

ST	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α-num. Zeichen)
adr1	Angabe der 1. Stopadresse (relativ zum Partition\anfang in sedezimaler Darstellung (4stellige Sedezimalzahl)
adr2- adr4	Wahlweise Angabe weiterer Stopadressen (2, 3 und 4) im selben Format wie adr 1.

Fehlermöglichkeiten:

Formal falsche Parameterangaben.

Programm nicht vorhanden.

Funktion wurde bereits für 2 Programme angegeben.

Angegebene Adresse außerhalb der Partition.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung AS

Mit der Debuganweisung AS (AUTOMATIC SINGLE) wird das in der Anweisung angegebene Programm im Einzelschritt ausgeführt. Nach jedem ausgeführten Befehl wird das Debug, kurzzeitig, nur für die Ausgabe des Befehls aktiv. Anschließend wird sofort der nächste Befehl ausgeführt usw.

Diesen automatischen Ablauf kann der Bediener durch Betätigen der Debuganruftaste (STP-Rasttaste) unterbrechen; das Debug bleibt aktiv und der Bediener kann über die Funktionsauswahlschleife Funktionen auswählen oder das Programm fortsetzen.

Format:

$$AS \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

AS	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1 - 8 α-num. Zeichen)

Fehlermöglichkeiten:

Formal falsche Parameterangabe

Programm nicht vorhanden

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung AP

Mit der Debuganweisung AP (AUTOMATIC STOP) können Stopadressen (Befehlsadressen) angegeben werden. Diese Adressen werden überwacht.

Sobald eine Stopadresse erreicht ist, wird das Debug kurzzeitig für die Anzeige des zuletzt ausgeführten Befehls aktiv und das Programm wird fortgesetzt. Dieser automatische Ablauf kann durch Betätigen der Debuganruftaste (STP-Rasttaste) unterbrochen werden.

Stopadressen können für maximal zwei Programme angegeben werden. Pro Programm können nicht mehr als 4 Stopadressen angegeben werden.

Format:

$$AP \sim \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\} \text{adr1} [\text{adr2}][\text{adr3}][\text{adr4}]$$
Bedeutung:

ST	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 \mathcal{L} -num. Zeichen)
adr1	Angabe der 1. Stopadresse (relativ zum Partitionanfang) in sedezimaler Darstellung (4stellige Sedezimalzahl)
adr2- adr4	Wahlweise Angabe weiterer Stopadressen (2, 3 und 4) im selben Format wie adr1.

29.8.77/ES3/BS

DEBUG

Fehlermöglichkeiten:

Formal falsche Parameterangabe

Programm nicht vorhanden

Funktion wurde bereits für 2 Programme angegeben

Angegebene Adressen außerhalb der Partition

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung AT

Mit der Debuganweisung AT (AUTOMATIC TRACE) werden alle Programmverzweigungen des angegebenen Programmes kontrolliert. Sobald der lineare Ablauf des Programmes unterbrochen ist, wird das Debug kurzzeitig für die Anzeige der Programmverzweigung aktiv. Anschließend wird der nächste Befehl verarbeitet usw.

Diesen automatischen Ablauf kann der Bediener durch Betätigen der Debuganruftaste (STP-Rasttaste) unterbrechen; das Debug wird aktiv und der Bediener kann über die Funktionsauswahlschleife Funktionen auswählen oder das Programm fortsetzen.

Format:

$$AT \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

AT	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α -num. Zeichen)

Fehlermöglichkeiten:

Formal falsche Parametereingaben

Programm nicht vorhanden

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung DU

Mit der Debuganweisung DU (DUMP) können Adressen von Feldern und deren Längen angegeben werden. Bei den Funktionen Single, Trace und Stop werden die Inhalte der Felder angezeigt.

Die Debuganweisung DU kann für maximal 2 Programme angegeben werden; pro Programm sind maximal 2 Adressen zulässig.

Format:

$$DU \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\} \text{adr1-l1} [\text{adr2-l2}]$$
Bedeutung:

DU	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α -num. Zeichen)
adr1	Angabe der 1. Feldadresse (relativ zum Partitionsanfang) in sedezimaler Darstellung (4stellige Sedezimalzahl).
l1	Angabe der Länge des 1. Feldes (2stellige Dezimalzahl) ($1 \leq l \leq 24$)
adr2	Wahlweise. Angabe der 2. Feldadresse im selben Format wie adr1.
l2	Angabe der Länge des 2. Feldes im selben Format wie l1

Fehlermöglichkeiten:

Formal falsche Parameterangabe.

Programm nicht vorhanden.

Funktion wurde bereits für 2 Programme angegeben.

Angegebene Adresse außerhalb der Partition.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung SM

Mit der Debuganweisung SM (SET MARKER) können Internmerker und Merker (MØ - M15) für das genannte Programm gesetzt werden.

Wird in einer Anweisung der Internmerker mehrmals angegeben, so gilt der zuletzt eingegebene.

Format:

$$SM \left\{ \begin{matrix} P \\ \text{name} \end{matrix} \right\} \cup \left\{ \begin{matrix} i \\ m \end{matrix} \right\} \left[\cup \left\{ \begin{matrix} i \\ m \end{matrix} \right\} \right] \dots$$
Bedeutung:

SM	Name der Anweisung
$\left\{ \begin{matrix} P \\ \text{name} \end{matrix} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α-num. Zeichen)
$\left\{ \begin{matrix} i \\ m \end{matrix} \right\}$	Angabe der Merker bzw. Internmerker i = Internmerker (E, H, L, O) m = Merker (MØ, ... M15)

Fehlermöglichkeiten:

Formal falsche Parameterangabe.

Programm nicht vorhanden.

Stehen in einer Anweisung falsche Parameter und sind in der selben Anweisung Internmerker angegeben, so kann der Internmerker anschließend verändert sein. Merker werden bei fehlerhaften Parameterangaben nicht verändert.

Debuganweisung RM

Mit der Debuganweisung RM (RESET MARKER) können Merker (MØ - M15) für das genannte Programm rückgesetzt werden.

Format:

$$RM \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\} \left[m \right] \dots$$
Bedeutung:

RM	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α-num. Zeichen)
m	Angabe der Merker m = Merker (MØ, ... M15)

Fehlermöglichkeiten:

Formal falsche Parameterangabe

Programm nicht vorhanden

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung DE

Mit der Anweisung werden die Debugfunktionen für das angegebene Programm aufgehoben. Die evtl. von dem Programm belegte Parametertabelle ~~im Debug~~, für die Funktionen Dump und Stop wird frei für andere Programme.

Format:

$$DE \sim \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

DE	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität, bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α -num. Zeichen)

Fehlermöglichkeiten:

Formal falsche Parameterangaben.

Programm nicht vorhanden.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Debuganweisung CP

Mit der Debuganweisung CP (COPY) kann ein Speicherbereich auf die Platte in die Systembibliothek "SYSCOR" übernommen werden.

Voraussetzungen:

1. Der Speicherbereich liegt innerhalb der angegebenen Partition bzw. ist nicht größer als die Partition.
2. Soll die Partition unter dem Partitionnamen ausgelagert werden, so muß der Name der Partition bereits in der Systembibliothek "SYSCOR" stehen.
3. Soll ein Programmteil unter einem anderen Namen in die Systembibliothek übertragen werden, so muß dieser angegebene Name bereits in der Systembibliothek stehen.
4. Es muß eine genügende Anzahl von Spuren bzw. Sektoren für das Übertragen des Speicherbereiches vorhanden sein. Die Anzahl benötigter Sektoren (n) ergibt sich aus:

$$n = (l/123) + 1$$

l = Größe des zu übertragenden Speicherbereiches

Hinweis: Bei der Erstaufnahme von Programmen kann zusätzlich Speicherplatz reserviert werden (Floppy-Disk Dienstprogramme SYSCON).

nicht

Format:

$$CP \cup \left\{ \begin{array}{l} P \\ \text{name } 1 \end{array} \right\} \left[\begin{array}{l} \text{Uadr1} \\ \left[\begin{array}{l} \text{Uadr2} \\ \left[\text{Uname2} \right] \end{array} \right] \end{array} \right]$$

Bedeutung:

CP	Name der Anweisung
{P name1}	Angabe der Priorität, bzw. Partitionname. P = Priorität ($2 \leq p \leq 7$) name1 = Partitionname (1-8 α -num. Zeichen)
adr1	Anfangsadresse in sedezimaler Darstellung (4stellig) Anfangsadresse des zu übertragenden Speicherbereiches relativ zum Partitionanfang. Ist die Anfangsadresse nicht angegeben, wird ab Partitionanfang übertragen.
adr2	Endeadresse in sedezimaler Darstellung (4stellig) Endeadresse des zu übertragenden Speicherbereiches relativ zum Partitionanfang. Ist die Endeadresse nicht angegeben, wird bis zum Partitionende übertragen.
name2	Programmname (1-8 α -num. Zeichen) in der Systembibliothek "SYSCOR".

Fehlermöglichkeiten:

Formal falsche Parameterangabe

Programm nicht vorhanden

Angegebene Adressen liegen außerhalb der Partition

Fehlernummer 60: Der für das Bibliothekselement vorhandene Bereich in der Systembibliothek "SYSCOR" ist zu klein.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Gerätefehler: Tritt beim Übertragen ein Gerätefehler auf, wird der Fehlercode in der Anweisungszeile angezeigt.

29.8.77 ES3/BS

DEBUG

Der angegebene Speicherbereich kann bereits zum Teil übertragen sein. Durch Eingabe der Debuganweisung CP kann das Übertragen wiederholt werden.

Debuganweisung EA

Die Debuganweisung EA stellt dem Bediener eine Eingaberoutine zur Verfügung, die ihm gestattet innerhalb der angegebenen Partition folgende Funktion auszuführen:

Speicherinhalte anzeigen

Speicherinhalte ändern

Format:

$$EA \left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$$
Bedeutung:

EA	Name der Anweisung
$\left\{ \begin{array}{l} P \\ \text{name} \end{array} \right\}$	Angabe der Priorität bzw. des Programmnamens P = Priorität ($2 \leq p \leq 7$) name = Programmname (1-8 α -num. Zeichen)

Fehlermöglichkeiten:

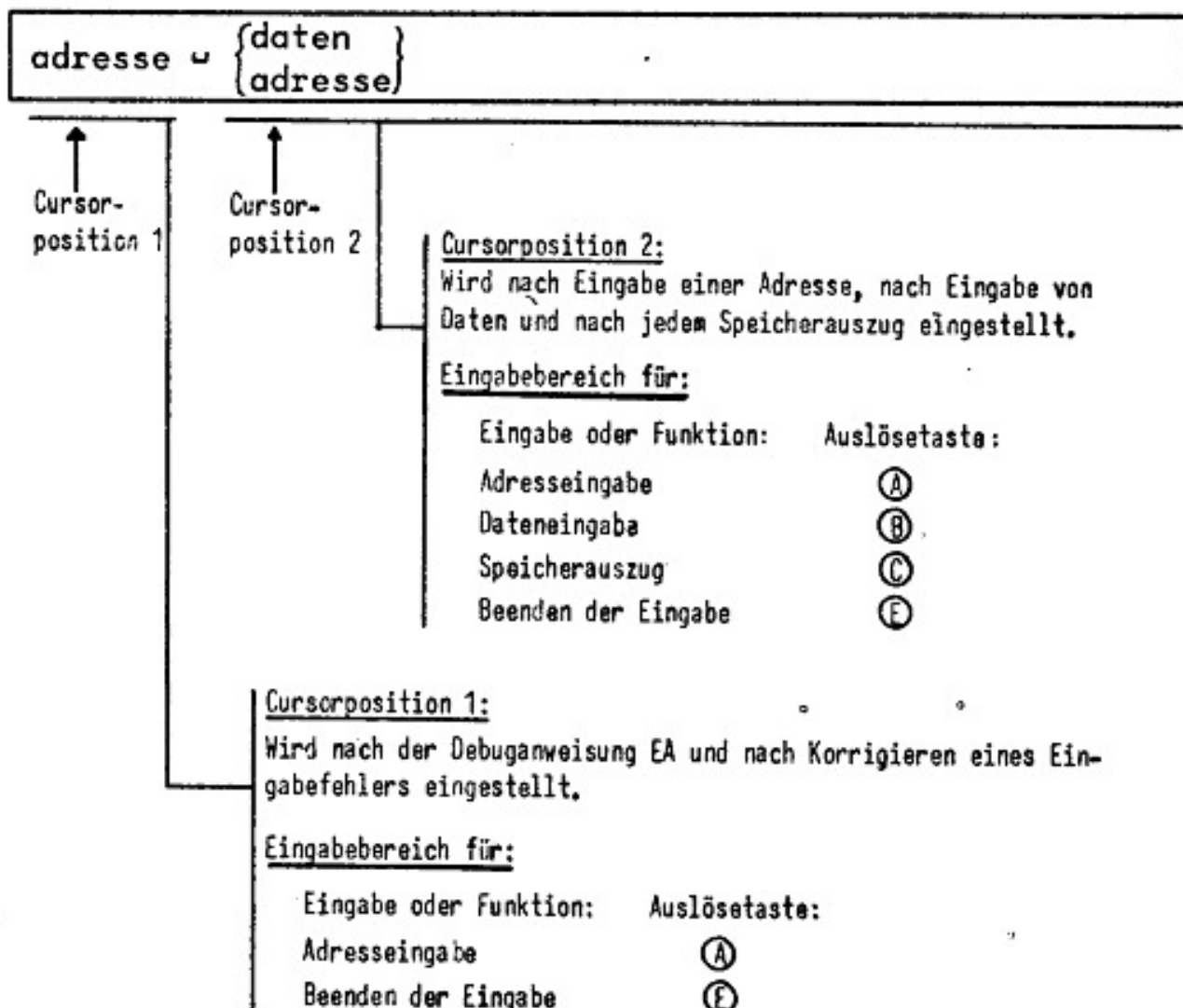
Formal falsche Parameterangabe.

Programm nicht vorhanden.

Die Anweisung wird abgewiesen und die Funktion nicht ausgeführt.

Eingaberoutine:

In der Eingaberoutine können Adressen und Daten eingegeben oder die Funktion Speicherauszug angegeben werden. Die Eingabe der Adressen und Daten sowie die Ausgabe der Daten erfolgt in sedezimaler Darstellung. Die Adressenein- und Ausgabe erfolgen relativ zum Partitionanfang.

Eingabeformat:

Adresseingabe:

Alle Adressen müssen als 4stellige Sedezimalzahlen (8 Ziffern) angegeben werden.

Soll eine Adresse eingegeben werden während ein Speicherauszug angezeigt wird (Cursorposition 2), so ist mit der Adresseingabe die Anzeige zu überschreiben und nach der Eingabe der 4stelligen Sedezimalzahl ein Leerzeichen (blank) einzugeben.

Die Adresseingabe ist mit der Auslösetaste (A) zu beenden.

Dateneingabe:

Die Eingabe der Daten erfolgt byteweise. Für jedes Byte ist eine 2stellige Sedezimalzahl (4 Ziffern) einzugeben. In einer Eingabezeile können die Daten für maximal 8 Bytes eingegeben werden. Nach jeder 2stelligen Sedezimalzahl (Daten für 1 Byte) können wahlweise Leerzeichen (Blank) eingegeben werden.

Sollen Daten eingegeben werden während ein Speicherauszug angezeigt wird (Cursorposition 2), so ist mit der Dateneingabe die Anzeige zu überschreiben.

Die Dateneingabe ist mit der Auslösetaste (B) zu beenden.

Speicherauszug:

Der Speicherauszug wird mit der Auslösetaste \textcircled{C} (Cursor-Position 2) ausgelöst. Es wird der Speicherinhalt von 8 Bytes angezeigt.

Beispiel:

00120201 - 0102-0000-0201-0200-1201-0405-0107-1504

<u>Adresse</u>	Speicherinhalt (8 Bytes)
----------------	--------------------------

Beenden der Eingabe:

Die Eingaberoutine kann an jeder Stelle mit der Auslösetaste \textcircled{E} beendet werden.

Eingabefehler:

Formal falsche Eingabe.

Unzulässige Auslösetaste.

Partitionüberschreitung bei der Dateneingabe.

Fehlerhafte Eingaben werden mit der Lampe NUM angezeigt. Mit der Korrekturtaste 'C' muß die Eingabe gelöscht werden.

Debuganweisung PO

Die Anweisung PO (PRINTER ON) gestattet es, zusätzlich zur Display-Ausgabe ein Druckprotokoll zu erstellen. Das Druckprotokoll wird im selben Format wie auf dem Display ausgegeben (siehe Debugausgabe: Zeile 1 bis Zeile n).

Debuganweisungen selbst werden nur auf dem Display angezeigt.

Bei der Debuganweisung EA und der Funktion Speicherauszug (Auslösetaste (C)) wird ein Druckprotokoll erzeugt.

Die Parameter der Anweisung geben die Formularbreite und die Weichenstellung an.

Format:

PO \downarrow par1 \downarrow par2

Bedeutung:

PO	Name der Anweisung
par1	Angabe der Formularbreite (3stellige Dezimalzahl max. 255)
par2	Angabe der Weichenstellung (1stellige Dezimalzahl) 0 = Auswurf nach oben 1 = Auswurf nach hinten 2 = Auswurf nach hinten mit Schneiden

Debuganweisung PF

Die Debuganweisung PF (PRINTER OFF) schaltet die zusätzliche Protokollierung über den Drucker wieder ab.

Diese Anweisung kann zur Abschaltung des Debugprotokolles eingegeben werden, wenn z. B. der Drucker auch in einem Anwenderprogramm benutzt wird oder Störungen am Drucker auftreten.

Format:

PF

Bedeutung:

PF	Name der Anweisung
----	--------------------

Fehlermöglichkeiten:

keine

Debuganweisung EN

Mit der Debuganweisung EN (END) werden alle Debugfunktionen gelöscht. Anschließend ist das Debug nicht mehr anrufbar.

Format:

EN

Bedeutung:

EN	Name der Anweisung
----	--------------------

Fehlermöglichkeiten:

Keine

Format Zeile 3:

xxxxxxxxxcodexxxxx....

sedezimale
Speicher-
adresse des
Befehls,
relativ zum
Partition-
anfang

Mnemonic

Maschinenbefehl
sedezimal in der
Befehlslänge (byte-
weise Darstellung)

Zeile 4:

Die Anzeige der Zeile 4 erscheint nur bei der Funktion Trace. Bei der Funktion Trace wird auf der Zeile 3 der Verzweigungs-
befehl und auf der Zeile 4 der Zielbefehl angezeigt.

Format Zeile 4:

wie Format Zeile 3

7. DIENSTPROGRAMME

(wird ergänzt)