

dietz621

Lickfeld

Betriebssystemfamilie XOS

Beschreibung

dietz621

Betriebssystemfamilie XOS

Beschreibung

1.0 - 1	→	1.1.3
2.1 - 1	→	2.2 - 21
2.11 - 1	→	2.11 - 9

Heinrich Dietz
433 Mülheim-Ruhr
Solinger Straße 9
Tel. 0208/485024
Telex 856770

DIETZ Computer
SYSTEME

Zusammenfassung

Diese Dokumentation beschreibt den Betriebssystemkern XOS des Systems DIETZ 600/621, die Möglichkeiten der Dialogsprache, die verfügbaren Betriebssystem-Sonder-Module sowie die möglichen Betriebssystem-Konfigurationen. Neben einer allgemein gehaltenen Einführung in die Aufgaben der verschiedenen Module werden die realisierten Funktionen für die Benutzungs-Praxis beschrieben.

Die Aufgaben "Generieren und Laden des Betriebssystems, Montieren von Geräten und Sprachen" werden in einer eigenen Beschreibung "XOS-Benutzeranleitung" abgehandelt.

Weitere Literatur:

XOS-Benutzeranleitung
XOS-Übersetzungssystem
MULTI-USER-BASEX
FORTRAN-Benutzeranleitung
DIETZ 621 Handbuch

Beachte: Die Beschreibung entspricht dem Betriebssystemstand Release 5 (15.1.1977).
Mit *5 markierte Funktionen sind in Vorgänger-Releases nicht enthalten.

<u>Inhaltsverzeichnis</u>	Seite
1. <u>Übersicht über die Struktur des Gesamtbetriebssystem</u>	1.0-1
1.1 Betriebssystemkern	1.1-1
1.2 Dialog-Modul DIAL	1.2-1
1.3 Konfiguration und Zusatzmodule	1.3-1
1.4 Sprachen des Betriebssystems	1.4-1
1.5 Die Ebenen-Struktur	1.5-1
2. <u>Module des Betriebssystem-Kerns</u>	2.0-1
2.1 Aufrufschema	2.1-1
2.2 Ein/Ausgabe-Kontroll-System	2.2-1
2.3 Betriebsmittel-Verwaltung	2.3-1
2.4 Prozeß-Verwaltung	2.4-1
2.5 Prozeß-Einplanung	2.5-1
2.6 Datei Zugriff	2.6-1
2.7 Arithmetische Operatoren	2.7-1
2.8 Komplexe Ein/Ausgabe COIO	2.8-1
2.9 Sondermodulgruppe	2.9-1
2.10 Systemvariable	2.10-1
2.11 Arbeitsweise der Betriebssystem-Konfiguration	2.11-1
3. <u>Dialog-Kommandos</u>	
3.1 Allgemeines	3.1-1
3.2 Kommandos zur Prozeßverwaltung	3.2-1
3.3 Kommandos zur Betriebsmittelverwaltung	3.3-1
3.4 Kommandos zur Ein/Ausgabe und Dateiverwaltung	3.4-1
3.5 Hilfskommandos	3.5-1
3.6 System-Generierung	3.6-1

	Seite
4. <u>Dienstprogramme</u>	
4.1 Allgemeines	4.1-1
4.2 Dienstprogramme Platteninhalt	4.2-1
4.3 Kopier-Programme	4.3-1
4.4 Dienstprogramm DUMP	4.4-1
4.5 Montieren von Geräten	4.5-1
4.6 Stapelverarbeitung	4.6-1
4.7 Dienstprogramm MODULE	4.7-1
5. <u>Module des erweiterten Betriebssystems</u>	
5.1 Allgemeine Übersicht	5.1-1
5.2 Background-Modul	5.2-1
5.3 Das SPOOLING-Modul	5.3-1
5.4 Programm Wiederanlauf-Modul	5.4-1
5.5 Modul BUS-Switch	5.5-1
Anhang 1: Geräteadressen und Rückmeldeebenen	A1-1
Anhang 2: Einlesen des Urlader-Streifens	A2-1
Anhang 3: XOS-Fehlerliste	A3-1

1. Übersicht über die Struktur des Gesamtbetriebssystems

Nach DIN 44300 besteht ein Betriebssystem aus Programmen, die zusammen mit den Eigenschaften der Rechanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen. Nach derselben DIN-Vorschrift werden ablaufende Programme im folgenden auch als Prozesse (= Job, Task) bezeichnet.

Die Konkretisierung des Begriffes "Betriebssystem" führt zu einer Grobgliederung in 4 Schichten:

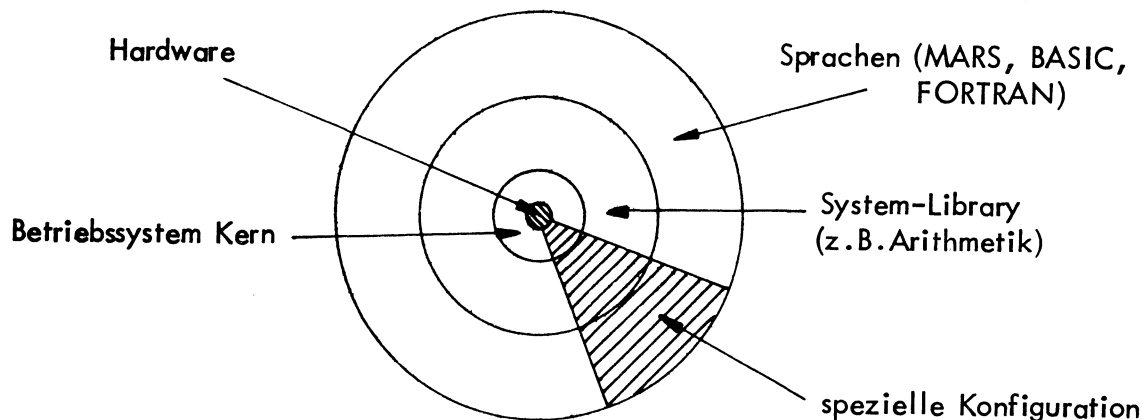


Bild 1: Schichten des Betriebssystems

Dieses Modell ist so zu interpretieren, daß äußere Schichten die Funktionen der inneren Schichten verwenden. In jeder Schicht gibt es eine Reihe von Moduln. Eine bestimmte Auswahl von Moduln in den einzelnen Schichten ergibt ein Programmiersystem (z.B. BASIC, MULTI-USER-BASIC, MARS), was oft auch als Betriebssystem im erweiterten Sinne (z.B. im kommerziellen Bereich) bezeichnet wird.

Die einzelnen Schichten oberhalb der Hardware haben folgende Inhalte:

✱ Betriebssystemkern

Er ist immer vorhanden, speicherresident und für die allgemeine Programmabwicklung unter optimaler Ausnutzung der Betriebsmittel des Rechners bei vorgegebener Strategie zuständig.

* System-Library

Hier sind Prozeduren enthalten, die in eine spezielle System-Konfiguration zur Durchführung verschiedener Standardaufgaben eingebunden werden können.

Beispiele sind die 2-Byte-Integerarithmetik, die 4-Byte- bzw. 8-Byte-Gleitkommaarithmetik, die Standardfunktionspakete, aber auch alle Konversionen. Weiter gehört dazu das Modul für Ein/Ausgabe nach dem SPOOLING-(indirekter Transfer über Platte) Prinzip.

* Sprachen

Hierzu zählen die Implementationen der Programmiersprachen (MARS, BASIC, FORTRAN), die interpretierend verarbeitet oder compiliert werden. Wesentlich ist, daß jede dieser Sprachen in einer speziellen Konfiguration des Betriebssystemkerns laufen kann, im Mono- oder Multiprogrammierungs- aber auch im Timesharing-Betrieb.

* Programm-Bibliothek

Jede Betriebssystemschicht ist gekennzeichnet durch eine eigene Sprache, in der die Funktionen aufgerufen werden können. Dem Anwenderprogrammierer begegnen sie in Form der Sprachelemente der höheren Programmiersprache (z.B. PRINT, INPUT in BASIC). Die Bibliotheksprogramme bauen auf den Programmiersprachen auf. Sie können als zusätzliche Schicht zum Anwender hin interpretiert werden.

Inhalt dieser Beschreibung ist zunächst der Betriebssystemkern, bestehend aus 8 Modulen. Hinzu kommt eine Dialogsprache, mit der ein Benutzer des Betriebssystems per Dialog alle für den praktischen Umgang notwendigen Funktionen ansprechen kann. Weiter gehören eine Reihe abfragbarer Systemvariablen dazu. Für plattenunterstützte Systeme ist das Modul "Dateizugriff" wichtig.

Inhalt spezieller Beschreibungen sind die Programmiersprachen und Programmbibliotheken.

1.1 Betriebssystemkern

Der Betriebssystemkern ist wie die gesamte Struktur des Betriebssystems stark modular aufgebaut. Er besteht aus folgenden Bausteingruppen:

* Input/Output Control-System (IOCS)

Diese Gruppe leistet den Transfer von Daten und Programmen zwischen Haupt- und Externspeicher bzw. peripheren Geräten. Im IOCS sind eine Reihe von Treibern enthalten, von denen die wichtigsten der SLI-Schnittstellen-Treiber und der Plattentreiber sind. Hierzu können nach Bedarf weitere aufgenommen werden. Beispiele sind ein Treiber für Rechner-Rechner-Kopplung und vor allem die Prozeßperipherie-Treiber bei Prozeßanwendung.

* Resource-Management (REMA)

Hier wird die Benutzung eines Betriebsmittels (Resource) koordiniert. Solche Betriebsmittel sind Geräte, Systemprogramme, Speicherpuffer. Die Grundstufe des Resource-Management bildet das Semaphoren-Konzept.

* Prozeß-Management (PRMA)

Dieses Modul führt Buch über den Zustand von Prozessen (Jobs, Tasks), die auf dem Rechner laufen und vergibt die Resource "Rechenzeit" an rechenwillige Prozesse. Die Vergabereihenfolge ist abhängig von einer vorgewählten Strategie (Mono-, Multiprogramming, Timesharing).

* Prozeß-Scheduler (PRSC)

Dieses Modul ist dann vorhanden, wenn Realzeitprozesse gesteuert werden sollen. Es übernimmt die Einplanung von Interrupt-, Zeit- und Start-Aufträgen für Vordergrundprogramme und meldet diese dem Prozeß-Management an, wenn die Interrupt- und Zeitbedingungen eintreffen.

Bild 2 soll den Zusammenhang der Module illustrieren:

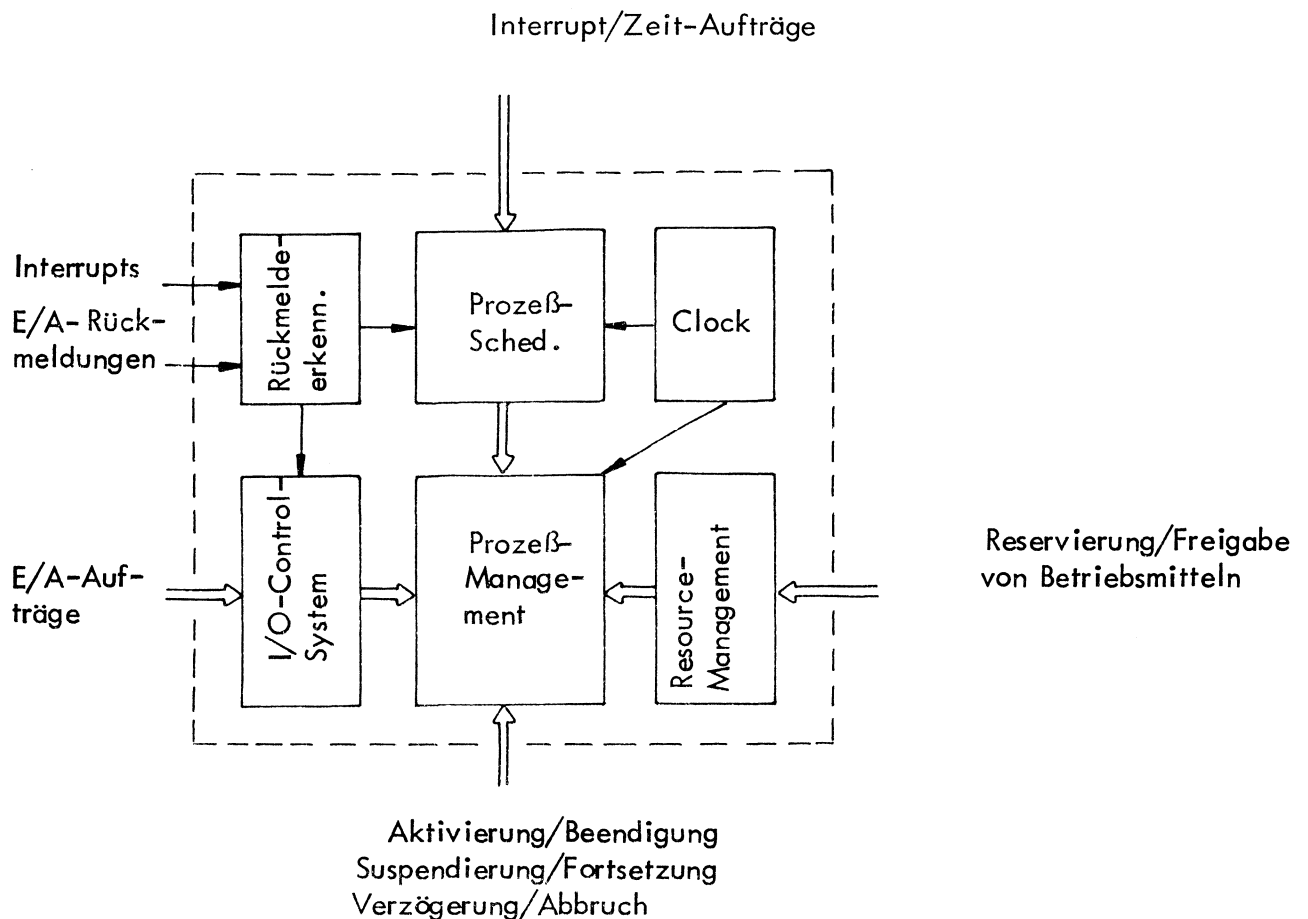


Bild 2: Module des Betriebssystem-Kerns

✱ Datei-Zugriff

Plattenunterstützte Systeme enthalten neben dem Plattentreiber das Modul Dateizugriff FIAC. Es ermöglicht den Aufbau und die Verwaltung eines Dateiverzeichnisses und einen einfacheren Zugriff zu den Daten einer Datei, als es unter ausschließlicher Verwendung des Plattentreibers möglich wäre. Der Datei-Zugriff verwendet außerdem das Modul "Resource-Management".

* arithmetische Operatoren

Im Betriebssystemkern ist eine Integerarithmetik mit Konversionen fest eingebaut. Darüber hinaus lassen sich weitere Arithmetiken wie eine 4-Byte-Real-Arithmetik, eine 8-Byte-Real und BCD-Arithmetik hinzugenerieren.

* komplexe Ein/Ausgabe-Funktionen

Hierunter sind beispielsweise Dateiverwaltungsfunktionen zu verstehen, die über die Möglichkeiten des einfachen blockorientierten Zugriffs in FIAC hinausgehen.

* Sonder-Module

Alle Kernfunktionen, die nicht in obige Bausteingruppen passen, sind in diesem Sondermodul zusammengefaßt.

* Systemvariable

Neben den 7 Modulen, zu denen über spezielle Befehle zugegriffen werden kann, kennt der Betriebssystemkern eine Reihe von Systemvariablen (wie Freispeicherende, -anfang), die im Benutzerprogramm abgefragt werden können.

Alle Betriebssystemkern-Funktionen werden ausführlich in Kapitel 2 dieser Dokumentation beschrieben.

1.2 Dialog-Modul DIAL

Zum Betriebssystem der Anlage DIETZ 600/621 gehört ein Dialogbaustein. Dieser Baustein interpretiert Kommandos, die von der Konsole her eingegeben oder von einer Kommando-Datei (Batch-Datei) beschafft werden. Solche Kommandos existieren:

- für An/Abmeldung von Prozessen
- für Anforderung/Freigabe von Betriebsmitteln
- für Prozeßzustandsabfrage
- für Nachrichtenübermittlung
- für Ein/Ausgabe von Daten und Programmen
- für Dateiverwaltung

Daneben enthält DIAL eine Reihe von Dienstprogrammen für Ausgabe Dateiverzeichnis, Datenaustausch zwischen peripheren Geräten, Montieren von Geräten. Dienstprogramme und einige Kommandos werden im Overlay betrieben.

Im Teilnehmerbetrieb können Kommandos für Teilnehmer gesperrt werden und nur an einer speziellen Konsole zugelassen sein.

DIAL wird ausführlich in Kapitel 3 und 4 beschrieben.

1.3 Konfigurationen und Zusatzmodule

Das Gesamtbetriebssystem XOS umfaßt verständlicherweise eine große Anzahl von Funktionen, die in einer speziellen Implementierung nicht benötigt werden. Da das System andererseits streng modular aufgebaut ist, kann eine spezielle Konfiguration leicht generiert werden.

Als Konfigurationen, die das Modul Prozeßmanagement, also die Vergabestrategie des Systems für den Rechnerkern betreffen, werden als DIETZ-Standard angeboten:

MPOS-t	für Multi-Programmierung
RTOS-t	für Realtime-Programmierung
TSOS-t	für Timesharing-Betrieb
TROS-t	für Timesharing-Betrieb mit Realzeitanwendung

*5) t steht für die Art der Unterstützung des Betriebssystems durch Hintergrundspeicher.

Es existieren folgende Versionen:

t = L	ohne Plattenunterstützung
t = F	für 60 MB-Platte
t = E	für 4.8 MB-Platte
t = D	für 2.4 MB-Platte
t = C	für 0.25 MB-Kleinplatte (DIETZ-Disk)

(Eine Sonderversion t = # hat als Plattenspeicher 2.4 MB-Platten für 24 Slot-Controller)

Im Grundausbau belegt das Betriebssystem im Hauptspeicher 8 KB resident und 40 KB auf der Systemplatte (8 KB TSOS-X und 32 KB Dienstprogrammdatei). Darin sind enthalten:

*5) Die in früheren Releases übliche Unterscheidung zwischen Anlagen mit 8 oder 16 Ebenen entfällt in Rel. 5.

- Treiber für alle Zeichenseriellen Geräte
- Treiber für Systemplatte
- Magnetbandtreiber im Overlay
- Betriebsmittelverwaltung
- Prozeßverwaltung in beliebiger Konfiguration
- Dateiverwaltung FIAC
- Integerarithmetik

sowie alle Möglichkeiten des Dialogmoduls.

In speziellen Systemausstattungen können darüber hinaus weitere Betriebssystem-Module eingebaut werden. Solche sind beispielsweise:

Zusatztreiber (z.B. residenter Magnetbandtreiber, Lochkartentreiber)

Arithmetiken (z.B. 4 Byte Real, 8 Byte Real, BCD)

Dateizugriffsverfahren (z.B. DFMS, DBMS)

komplexe Ein/Ausgabe-Treiber (z.B. Foreground-Background)

Das Dienstprogramm MODULE gestattet den Einbau solcher Module ins Gesamtsystem. Seine Arbeitsweise wird in Kapitel 4 erläutert. Daneben existiert ein Generierungs-dialogprogramm.

Aufgabe und Arbeitsweise von Modulen werden in Kapitel 2 und 6 näher beschrieben.

Das System DIETZ 600/621 kann in einer Mehrrechnerkonfiguration eingesetzt werden, wo 2 oder mehr Rechner zusammengeschaltet werden, um Betriebsmittel an anderen Rechnern verfügbar zu machen. Solche sind vor allem Geräte wie Drucker und Plattenspeicher aber auch das Betriebsmittel Rechenzeit. Grundlage der Kopplung ist im allgemeinen das Master-Slave-Prinzip.

Die Kopplung erfolgt auf 3 Ebenen:

1. Prozedurebene
2. Betriebssystemebene
3. Anwenderprogramm

Auf der Prozedurebene werden Nachrichten von Rechner zu Rechner übertragen ohne Berücksichtigung des Inhaltes. Ebene 2 wertet spezielle Nachrichten, nämlich die Betriebssystembefehle aus. Ebene 3 übergibt Nachrichten (Telegramme) an Anwenderprogramme, die in beliebigen Sprachen geschrieben sind, weiter.

1.4 Sprachen des Betriebssystems

Eine Betriebssystemkonfiguration kann als Sondermodule Interpreter für die Sprachen:

BASIC

BASEX Erweiterung von BASIC für Prozeßverarbeitung

C-BASIC Erweiterung von BASIC für kommerzielle Verarbeitung

MAGICS Erweiterung von C-BASIC für Informationssysteme

enthalten. Diese Verarbeitungsart ermöglicht den sogenannten Mehrbenutzer (Multi-User)-betrieb. Mehrere Teilnehmer können gleichzeitig mit diesem Interpreter arbeiten.

Zum Betriebssystem gehören weiter folgende Systeme:

MARS Assembler

EDIREL Editor

XOSFOR FORTRAN-IV-Compiler

XOSBAS BASEX-BASIC-Compiler

XOSPAS PASCAL-Compiler

XOSLAD Bindelader

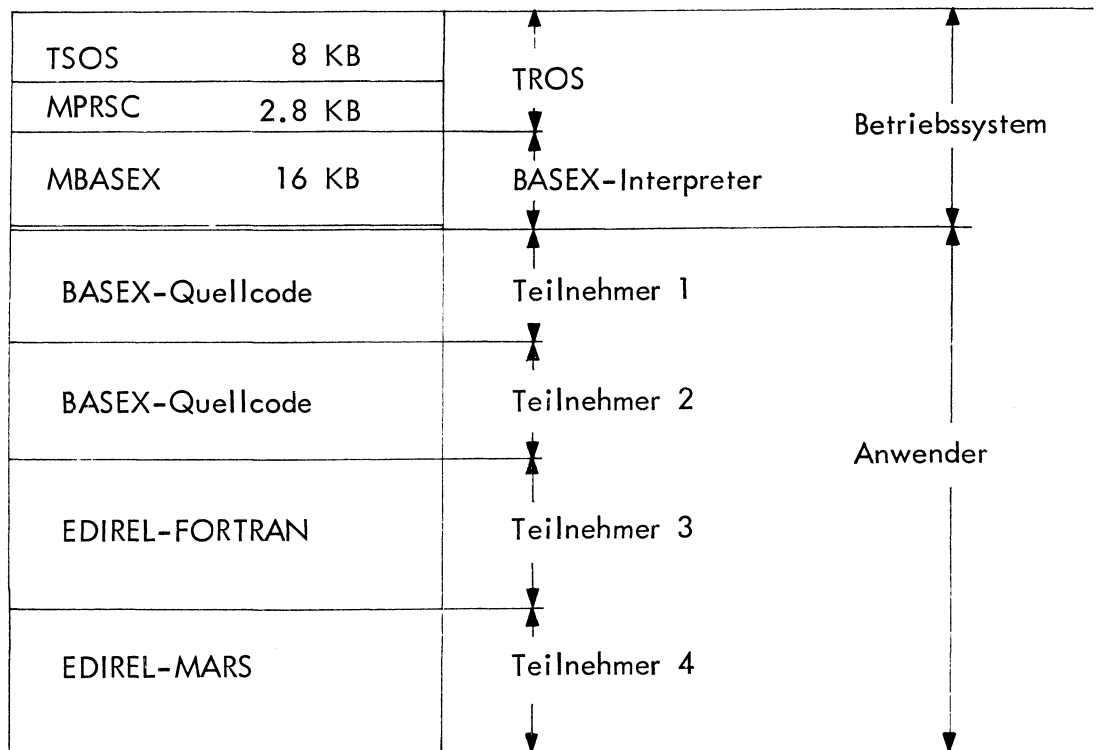
XOSST2 Stage 2-Makrogenerator

DEBUG Interaktives Fehlertestprogramm für Assembler

An jedem System ist Mehrsprachenbetrieb (Multilanguage) möglich, d.h. verschiedene Teilnehmer können nacheinander alle Übersetzer sowie die Unterstützungsprogramme rufen oder Programme durchführen.

Bei Sprachen mit Prozeßanwendung (BASEX, Prozeß-FORTRAN) muß das Betriebssystem um das Modul MPRSC (Prozeßeinplanung) erweitert sein.

Beispiel für eine Speicherbelegung für 4 Benutzer:



1.5 Die Ebenen-Struktur

Der Rechner DIETZ 600/621 besitzt als besonderes Merkmal die Mehr-ebenenstruktur, die das Betriebssystem in seinen Multiprogramming- und Timesharingaufgaben unterstützt.

Dies bedeutet, daß 8 oder 16 Registersätze zu je 128 Byte existieren und daß gleichzeitig 8 oder 16 Prozesse laufen können, zwischen denen das Betriebssystem ohne Zeitverzögerung umschalten kann.

Für Systemaufgaben benötigt das Betriebssystem 4 Ebenen:

CNP Prozesse zur Erkennung von Paritäts (P)- und Netzausfallfehlern (N). Außerdem arbeitet auf dieser Ebene das Clock-Modul, also ein Programm, das durch die Rechner-Uhr (C = Clock) angestoßen wird und zu verschiedenen Aufgaben benötigt wird.

- Führen der Tageszeit
- Start der Zeitauftragsebene bei Wirksamwerden eines Zeitauftrages in Realtimekonfigurationen
- Durchführung eines "Delay-Auftrages"
- Kontrollieren der Zeitscheibe im Timesharingbetrieb

(Für Durchführung von weiteren zeitkritischen " $< 100 \text{ usec}$!" Aufträgen kann in Register '10 - '13 ein UP-Aufruf - CSA,6,aaaa - eingebaut werden.)

IOCS Prozesse des Input-Output-Control-Systems. IOCS belegt 2 Ebenen. Auf der höheren Ebene CNP-1 laufen zeitkritische E/A-Vorgänge, auf der unteren CNP-2 weniger kritische Vorgänge. Die entsprechenden Geräte-Rückmeldungen starten diese Ebenen.

PRMA Prozesse des Prozeß-Management. Sie laufen auf Ebene CNP-3 ab. Die Ebene wird nach Ablauf einer Zeitscheibe von dem Clock-Prozeß gestartet und sorgt für die weitere Vergabe des Rechnerkerns.

Die übrigen Ebenen sind in der Regel den Benutzerprogrammen zugeordnet. In Systemen mit speziellen Betriebssystemprozessen werden jedoch weitere Ebenen belegt.

Solche sind:

PRSC	Realzeit-Interrupt-Auftragsverwaltung
SPOOL	Abwicklung I-O-Aufträge in Spooling-Technik
BACKG	Abwicklung von Vorder-Hintergrund-Betrieb
KOPP	Abwicklung von Aufträgen bei Rechnerkopplung

2. Module des Betriebssystem-Kerns

In diesem Kapitel werden die in Teil 1 umrissenen Aufgaben des Betriebssystem-Kerns näher spezifiziert und die Schnittstellen definiert.

2.1-1

2.1 Aufrufschema

Alle Aufrufe von Betriebssystemfunktionen und Prozeduren erfolgen einheitlich nach folgendem Befehl:

CSA, XKER, KERN
H, 'XY

Dabei ist KERN die Einsprungsadresse ('4020) und XKER das Rückkehrregister ('1E). Die Hexakonstante 'XY gibt im ersten Halbbyte "X" die gerufene Bausteingruppe (IOCS, REMA, PRMA, PRSC), im zweiten Halbbyte "Y" die Funktion innerhalb dieser Gruppe an.

Das generelle Verarbeitungsschema eines Aufrufes ist aus Bild 3 entnehmbar. Jeder Aufruf führt zu einem Gruppenverteiler. Dieser prüft, ob die Angabe 'XY zulässig ist und verzweigt über eine Sprungtabelle (Adresse '405 A + 2 * X) zu den Betriebs-Moduln. Jedes Modul enthält im Kopf eine weitere Tabelle. Deren erster Eintrag "n" gibt die Anzahl der eingebauten Funktionen an, der zweite Eintrag die Adresse eines Grundstellungsprogramms im Modul. Die weiteren Eintragungen verweisen auf die Programme zur Durchführung der Funktionen.

Alle Programme kehren nach Durchführung der Funktion zur Fehlerprüfung an das Fehlerprüfprogramm ('406A) zurück, von wo aus der Rücksprung zum rufenden Programm oder zu einem Fehlerprogramm erfolgt.

Das Schema erlaubt den einfachen Ausbau des Betriebssystems, aber auch die leichte Auslagerung von Funktionen auf einen anderen Rechner bei Rechner-Rechner-Kopplung.

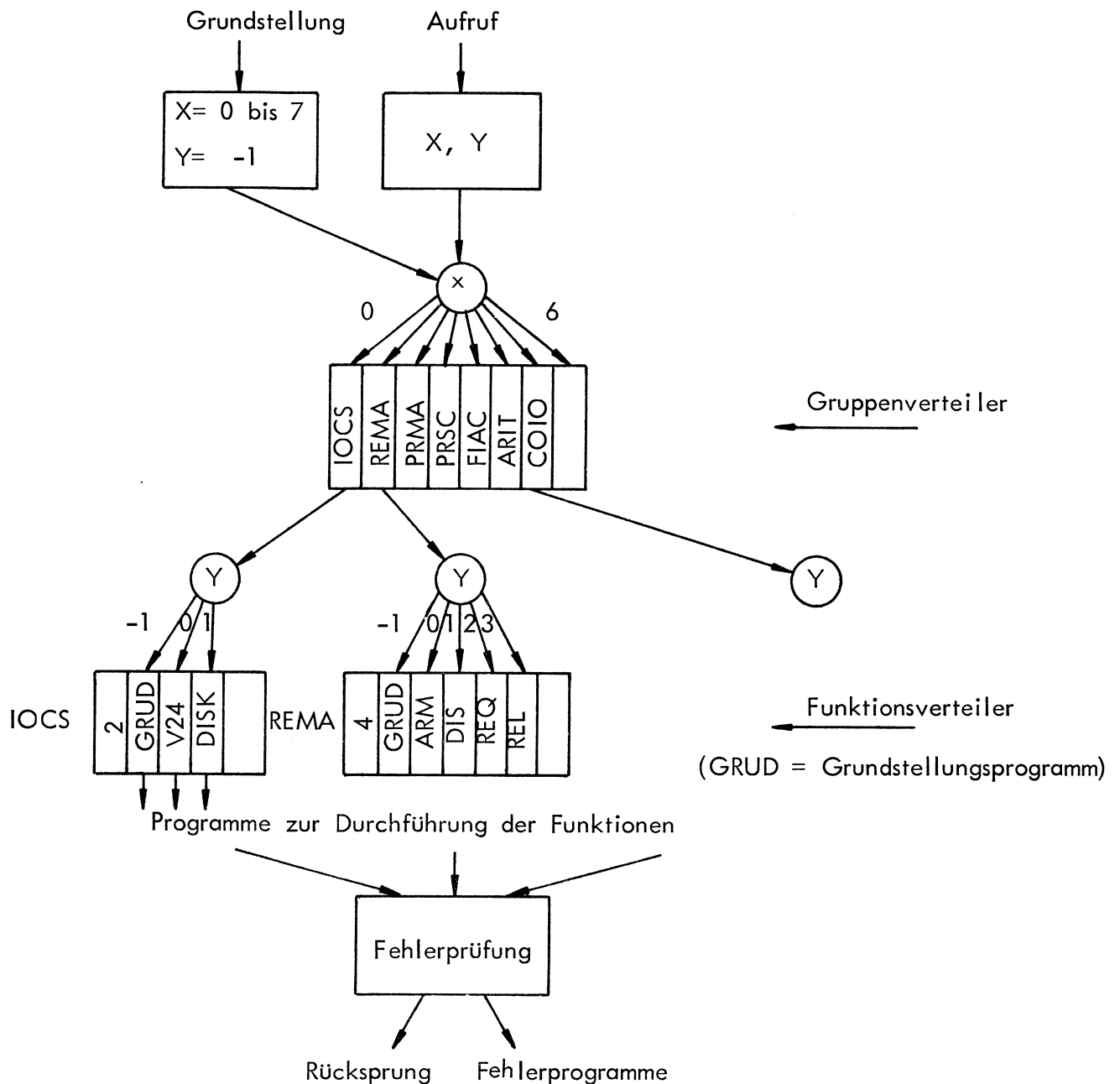


Bild 3: Aufrufschema von Funktionen

Die Parameter, die zur Durchführung einer Funktion nötig sind, werden in den Registern \textcircled{a} und folgenden übergeben. Falls die Funktion jedoch nicht durchführbar ist, wird im allgemeinen dem Fehlerregister ('7E) eine Fehlernummer übergeben. Sie gibt im ersten Halbbyte die zugehörige Fehlergruppe, im zweiten die Fehlerart an, bei 'FF-Inhalt ist die angewählte Funktion nicht vorhanden. Modulgruppe 0 - 3 benutzt generell Register R \textcircled{a} bis '1F, Plattenprogramme bis '3F.

2.1-3

Beim Einbau zusätzlicher Funktionen in das Betriebssystem sind folgende Bedingungen zu beachten:

- a) Nach Einordnung in die Modulgruppe erreicht man über die Adresse '405A des Gruppenverteilers den entsprechenden Funktionsverteiler.
- b) Dort erfolgt der Eintrag der Funktionsadresse und die Erhöhung der Funktionszahl.
- c) Rückkehrpunkt nach Funktionsdurchführung ist '406A.
- d) Es müssen die üblichen Konventionen bezüglich Registerbelegung (R@ - '1F) und der Fehlernummern beachtet werden.
- e) Der Einbau eines Treibers für Geräte mit Rückmelde-Einrichtung verlangt die Verwendung der IOCS-Rückmelde-Erkennung, wie sie in Kapitel 2.2 beschrieben ist.

2.2 Ein/Ausgabe-Kontroll-System

2.2.1 Allgemeines

Das Input-Output-Control-System (IOCS) ist zuständig für den Datenaustausch zwischen Rechnern und peripheren Geräten, welche mit Rückmeldungen arbeiten. Dazu zählen:

- Geräte mit zeichenseriellen Schnittstellen (SLI-Geräte) , wie Bildschirm, Drucker usw.
- Geräte, welche über den direkten Speicherzugriffs-Kanal angeschlossen sind, wie Platten, Analogmeßsystem .

Zur Durchführung des Datenaustauschs besitzt das IOCS-Modul sogenannte Treiber-Programme, die einer speziellen Hardware-Schnittstelle (Interface) zugeordnet sind.

XOS kennt im Standardausbau den SLI-Treiber für zeichenserielle Schnittstellen sowie einen Plattentreiber. Darüber hinaus werden weitere Treiber nach Bedarf eingebaut. Sie existieren als Module für

- Lochkartenleser
- Magnetbandeinheit
- Floppy-Disk
- DIETZ-Disk

Bei der Benutzung von Treibern ist folgendes zu beachten:

Treiber sollten vom Anwender im allgemeinen nicht angesprochen werden. Über die Treiberschicht des Betriebssystems sind komplexere Funktionen gelagert, die auf die Treiber Bezug nehmen und für den Anwender einmal höhere Benutzungssicherheit, aber auch höheren Komfort bringen.

Beispielsweise erlaubt der Plattentreiber die direkte Anwahl von Sektoren auf der Platte. Das Modul FIAC (siehe 2.6) erlaubt diese Möglichkeit nicht mehr. Ein Benutzer kann hier nur auf eine Datei, also einen bestimmten Plattenbereich, zugreifen und so nicht unbeabsichtigt Informationen zerstören. Gleichzeitig braucht er sich nicht um Plattenadressen zu kümmern. FIAC übernimmt vollständig die Speicherverwaltung.

Das Ansprechen des SLI-Treibers über Modulgruppe COIO (Complex IO) ermöglicht die Geräte-Sonderbehandlung sowie Vorder/Hintergrund-Betrieb an einem Gerät.

2.2.2 Arbeitsweise eines Treibers

Ein ablaufendes Treiberprogramm ist ein spezieller Betriebssystemprozeß (Task), der parallel zu anderen Prozessen, z.B. Benutzerprogramm mit hoher Priorität, abgearbeitet wird.

Das Programm besteht aus 3 Teilen:

- Behandlung der Anmeldung
- Reaktion auf Rückmeldung
- Schlußbearbeitung

Der Treiberprozeß wird durch das Anwenderprogramm über einen der in den folgenden Kapiteln beschriebenen Befehle gestartet. Nach Prüfung der Parameter wird der Ein/Ausgabe-Vorgang initialisiert. Danach ruht der Prozeß, er wartet auf eine Geräterückmeldung (Interrupt). Sobald diese eintrifft, führt der Prozeß den Auftrag fort bis schließlich in der Schlußbearbeitung die Fertigstellung des Auftrags signalisiert wird.

Der Rechner DIETZ 600/621 ist ein Mehrebenenrechner. Für die IOCS-Prozesse sind 2 Ebenen unterhalb der CNP (Ebene höchster Priorität) mit der Bezeichnung CNP-1 (zweithöchste Priorität) und CNP-2 (dritthöchste Priorität) reserviert. Eine Geräterückmeldung startet eine von diesen beiden Ebenen. Auf ihnen wird der Teil 2 und 3 des IOCS-Treibers abgewickelt, während der Treiberaufruf auf der rufenden Programmebene behandelt wird.

Folgendes Bild zeigt die Belegung des Rechners im Falle eines Ein/Ausgabe-Auftrags durch einen Benutzer, wobei dieser bis zur Fertigstellung des Auftrages wartet.

2.2-4

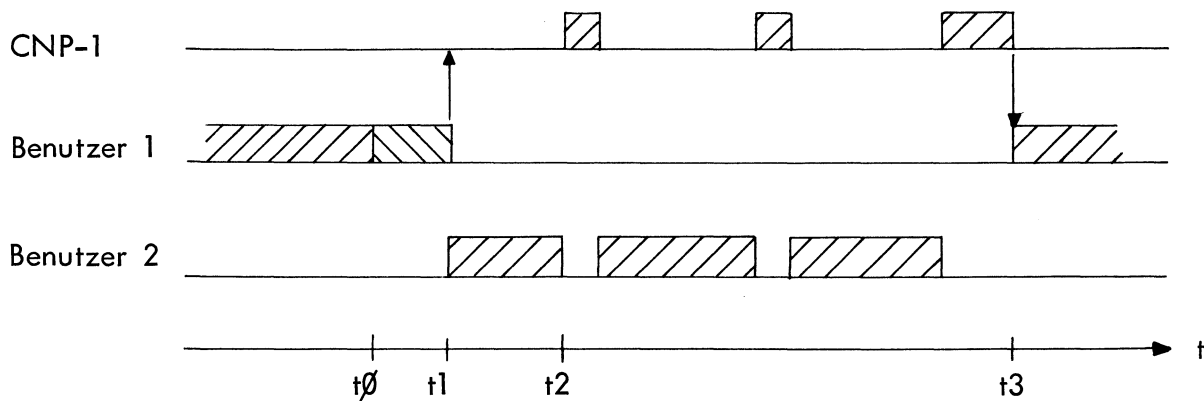


Bild 4 : Behandlung Ein/Ausgabe

- t_0 = Anmeldung Auftrag
- t_1 = Beenden der Anmeldung
- t_2 = Eintreffen Rückmeldung
- t_3 = Ende des Auftrags

In der Zeit, in der der Treiber auf Rückmeldung wartet, wird der Prozessor an andere rechenwillige Programme vergeben, die ihrerseits wieder Aufträge anmelden können.

2.2.3 Steuertabellen des IOCS

Der IOCS-Prozeß wird durch die sogenannte IOCS-Tabelle gesteuert. Sie enthält Angaben über gerätespezifische Eigenheiten, aber auch den aktuellen Durchführungsstand eines IO-Auftrages.

Die IOCS-Tabelle ist geordnet nach Gerätenummern. Pro Nummer sind 6 Byte reserviert mit folgender Bedeutung:

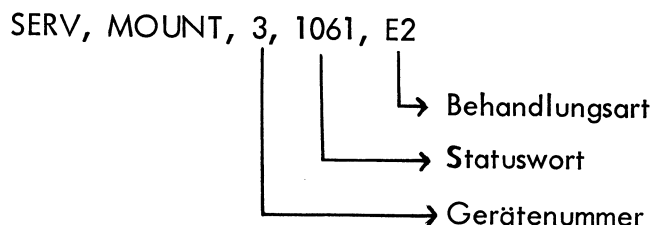
- | | | | |
|-------|---|--|----------------------|
| Ø - 1 | : | Physikalische Adresse des Statuswortes | } <i>serv. Monat</i> |
| 2 | : | Behandlungsart eines Gerätes | |
| 3 | : | Aktueller Zeichenzähler | |
| 4 - 5 | : | Adresse der Treiberaufrufparameter | |

Insgesamt hat die Tabelle Platz für 31 Eintragungen, d.h. es sind 31 Geräte anschließbar.

Als Standard gilt:

- | | |
|------------------------------|-----------------------------|
| 1 | Operator-Master-Konsole |
| Gruppe 2-9 : Pool-Geräte | |
| 2 | Lochstreifenleser, -stanzer |
| 3 | Drucker 1 |
| 4 | Drucker 2 |
| 5 | ADI |
| 6 | ADM |
| 7 | Plotter |
| 8 | Lochkartenleser |
| 9 | Lochkartenstanzer |
| Gruppe 10-20: Massenspeicher | |
| 10 | Systemplatte |
| 11-14 | Weitere Platten-Einheiten |
| 15 | Magnetband |
| Gruppe 21...: Dialoggeräte | |
| 21 | Dialoggerät Teilnehmer 1 |
| 22 | Dialoggerät Teilnehmer 2 |
| . | |
| . | |
| . | |

Bei Anlagen mit Rechnerkopplungsstrecken werden Gerätenummern ab 31 abwärts belegt.
Die physikalische Adresse des Statuswortes und das Byte "Behandlungsart" werden über das Dienstprogramm SERV, MOUNT (siehe Kapitel 4) in die IOCS-Tablle eingetragen, zum Beispiel:



Das Byte "Behandlungsart" enthält folgende Informationen:

Bit 7 Gerät armiert (1) oder nicht armiert (0)

Bit 6 Rückmeldung CNP-1 (1) oder CNP-2 (0)

Bit 5 SLI-Schnittstelle (0) oder nicht (1)

Bit 4 Break erlaubt (1) oder nicht (0)

Bit 3-0: Gerätemerkmal

a: bei SLI-Schnittstellen

4 Interfacetyp 602.050

3 Geräte, welche Rückmeldeverzögerungen bei Steuerzeichen erfordern
(0D und alle Zeichen mit Bit 7 = 1)

b: Sonstige Schnittstellen

F READY-Scanner

E

D

C 600-Zeilen-Drucker

B CAMAC

9 Magnetband

8 Platte

7

6

5

4 DIETZ-Disk

3 Bildschirm im Batch-Betrieb

2 Rechnerkopplung

1 Kartenleser

0 Keine Besonderheit

(Bei Terminierung von Programmen mit Geräteaufträgen in Gruppe F-8 wird der letzte noch beendet).

Anhang 1 stellt die Standarddaten der häufigsten Geräte zusammen.

Bit 7 des Behandlungsart-Bytes kann durch einen speziellen Befehl (ARM/DISARM) gesetzt/gelöscht werden. Eine Rückmeldung eines nichtarmierten Gerätes wird durch das IOCS verriegelt.

Bit 4 des Bytes spezifiziert, ob neben der normalen Geräterückmeldung, die nur aufgrund eines Auftrags entstehen kann, auch eine spontane Meldung (BREAK) erfolgen kann. In der Regel ist dies nur bei Dialoggeräten möglich.

Zur Behandlung solcher BREAK's benutzt IOCS eine zweite Tabelle, die BREAK-Tabelle. Sie enthält geordnet nach Benutzern (max. 12) die Eintragungen:

Byte 1 : Nummer des dem Benutzer zugeordneten Dialoggerätes

2-3 : Adresse des Programms, das bei BREAK gestartet werden soll.

Bei Eintreffen eines BREAK's an einem Dialoggerät wird das zugehörige Benutzerprogramm beendet und an der in der BREAK-Tabelle vermerkten Adresse gestartet. Ist dort die Eintragung \emptyset , so bleibt der BREAK-Interrupt unwirksam.

2.2.4 Einbau von Treibern

In den folgenden Kapiteln werden alle in XOS vorrätigen Treiber beschrieben.

Kann ein einzubauendes Interface mit diesen Treibern nicht angesprochen werden oder ist die Leistung des allgemeinen SLI Treibers nicht ausreichend, so kann der Benutzer einen eigenen Treiber schreiben und einbauen.

Zunächst sind folgende Festlegungen zu treffen:

- Schnittstelle zum Benutzer festlegen
Parameter werden in Register R(ⓐ) und folgende übergeben
Rücksprungregister ist '1E
Der Treiber darf nur Register bis '1E benutzen
- Ansprechennung ØX im IOCS-Sprungverteiler festlegen
- Festlegung der Rückmeldeebene
bei prioritären Geräten ist CNP-1 zuständig
- Festlegung der Gerätenummer

Beim Schreiben des Treibers ist folgendes zu beachten:

Der Treiber besteht aus 2 Teilen

- Initialisierung
- Rückmelde- und Schlußbearbeitung

Initialisierung

- Prüfen der Parameter auf Plausibilität
- Eintrag der Adresse des Rückmeldebearbeitungsprogrammes in die IOCS-Tabelle (Byte 4 - 5 der festgelegten Gerätenummer)
- Erstmaliges Ansprechen des Interfaces
- Endbehandlung des Auftrages
An dieser Stelle steht der Befehl WAIT
(Warten auf Fertigmeldung) und der Befehl JPA,, KERR (Sprung zur Betriebssystemschlußbehandlung)

Rückmeldebearbeitung

Während die Initialisierung auf der rufende Benutzerebene läuft, wird die Rückmeldebearbeitung auf der IOCS-Ebene abgewickelt.

Bei Eintreffen eines Interruptes, dessen Statusadresse in der dem Treiber zugeordneten IOCS-Spalte steht, verzweigt die generelle im IOCS vorhandene Rückmeldeerkennungs-Routine an die bei der Initialisierung eingetragene Fortsetzungsadresse. Dabei werden folgende Parameter übergeben:

R @	=	Inhalt Statuswort
RØC	=	Nummer Gerät damit mehrere Geräte durch einen Treiber behandelt werden können
RØD	=	Adresse IOCS-Eintrag
RØE-R13	=	Eintrag IO Tabelle

Mit diesen Parametern wird der Auftrag fortgesetzt, wobei alle Register bis R21 ^{*)} verwendet werden können. Die Fortsetzung enthält eine Prüfung, ob der Auftrag beendet ist. Ist dies der Fall, wird die Adresse des Fortsetzungsprogrammes gelöscht und mit "SIGNAL, event" signalisiert, daß der Auftrag abgeschlossen ist. In jedem Fall ist über R24 zum Betriebssystem zurückzuspringen.

Das Statusregister des Geräte-Interfaces wird durch die Rückmeldeerkennung nicht gelöscht. Die Rückstellung des externen Startes ist Aufgabe des Rückmeldebearbeitungsprogrammes. In diesem Programm dürfen keine Funktionen wie TER, SUS, CON, DELY angesprochen werden.

Beispiel: Sondertreiber Gerät 7; Treibergruppe 4

Aufruf:	CSA, XKER, KERN H, 'Ø4
A 1	Parameterprüfung
A 2	Adresse B1 nach IOCS [7 * 6 + 4]
A 3	Ansprechen Interface
A 4	Warten auf internes Signal " 7 " (Fertigstellung Auftrag) (LDC, @, 7 ; CSA, XKER, KERN; H' 22)
A 5	Sprung zur Betriebssystemendbehandlung JPA, KERR

^{*)} Das Betriebssystem rettet bei Aufruf Register R20-R25 und speichert sie bei Rücksprung zurück.

2.2.-10

B1	Start rückstellen Abfrage: Bei Ende verzweige zu C1
B2	Fortsetzung Auftrag
B3	JPX,,, '24
C1	Löschen Inhalt IOCS [7 * 6 + 4]
C2	Setze internes Signal 7 (LDC, @, 7 ; CSA, XKER, KERN; H, '23)
C3	JPX,,, '24

In Sonderfällen, wo eine spezielle Fortsetzungsbehandlung nicht notwendig ist, kann man den Programmteil A2, A4, B1 - C3 durch folgenden speziellen Wartebefehl ersetzen:

LDC, @ , 7	/ Sonder-Wait auf
CSA, XKER, KERN	/ Signal 7
H, 'Ø7	

Beispielsweise ist dies bei einem Plotter der Fall, wo nach Ausgabe eines Koordinatenpaares der Plotter signalisiert, daß die Position erreicht ist. Danach ist der Treiberauftrag abgeschlossen.

Beim Schreiben eines Treibers beachte man folgende Randbedingungen:

- Ein Treiber muß relativ geschrieben sein oder beim Einbau relativiert werden.
- Ein Treibereinbau geschieht über das Dienstprogramm SERV, MODULE (siehe dort).
- Ein Treiber darf nur Register bis '1F, in Ausnahmefällen bis '3F, benutzen.

2.2.5 Einbau von Treibern und Geräte-Montage

Das IOCS kann ein Gerät nur betreiben, wenn 2 Voraussetzungen gegeben sind:

1. Der für das Gerät zuständige Treiber muß im Speicher existieren und in der Sprungtabelle "Modul ØX" verankert sein. Nicht-Standard-Treiber werden mit SERV, MODULE (siehe 4.8) eingebaut.
2. Die Adresse des Statuswortes und das Kennungsbyte muß montiert sein. Dies leistet das Dienstprogramm:

SERV, MOUNT (siehe 4.6)

Bei Platten-Einheiten muß außerdem eine Montage mit

SERV, DSKMNT (siehe 4.6)

erfolgen.

Danach kann ein Treiber mit den Befehlen

"Parameteraufbereitung
in R@ und folgende "

CSA, XKER, KERN

H, 'ØX

aufgerufen werden.

2.2.6 SLI-Treiber

Aufruf: H, 'ØØ

Parameter: R(Ⓐ)/R3 = Adresse eines Parameterblocks, dessen einzelne Byte folgende Bedeutung haben:

- | | |
|---------------------------------------|----------|
| 1. Betriebsart, Betriebsablaufanzeige | (1 byte) |
| 2. Gerätenummer | (1 byte) |
| 3. Stopcode | (1 Byte) |
| 4. Pufferlänge | (1 Byte) |
| 5. Pufferadresse | (2 byte) |
| 6. Fehleranzeige | (1 byte) |

Fehler: 'Ø1 Gerät defekt oder nicht vorhanden
'Ø2 Pufferadresse nicht zulässig

Nach Aufruf beschafft der Treiber die Adresse des Parameterblocks und rettet diese in die I/O-Tabelle. Die Parameter des Blocks werden jedoch nicht gerettet. Sie dürfen deshalb während des Transfers nicht verändert werden.

- Das erste Byte des Blocks gibt die Betriebsart an. Bit 0 bis 3 haben dabei folgende Bedeutung:

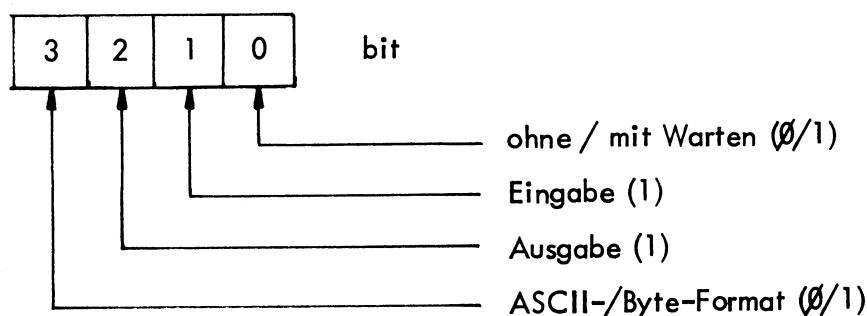


Bild 4a: Betriebsart-Spezifikation

Bei einem Ein/Ausgabe-Auftrag ohne Warten wird der Transfer angestoßen, während das rufende Programm weiterläuft. Dieses muß sich im allgemeinen über den Ausführungsstand des Auftrages erkundigen können, um zu wissen, ob beispielsweise der Pufferinhalt verfügbar ist. Dies ist möglich über Bit 4 des Betriebsartbytes.

Soll ein Transfer im ASCII-Format erfolgen, so werden bei Eingabe die Zeichen ohne Paritätsbit abgelegt, bei Ausgabe wird ein Paritätsbit erzeugt. Außerdem hat das Zeichen '7F (Rubout) die Sonderfunktion "Löschen der Eingabezeile", das Zeichen '5F (←) und '08 (BS) die Funktion "Löschen des letzten Eingabe-Zeichens". Im Byteformat werden die Zeichen direkt ohne Paritätserzeugung und -prüfung transferiert.

Der zweite Parameter des Aufrufblocks ist die Geräte-Nummer. Dem IOCS ist die zu dieser logischen Nummer gehörende Busadresse aus der IOCS-Tabelle bekannt.

Im dritten Parameter gibt der Benutzer einen "Stopcode" an. Wird während des Ein/Ausgabe-Vorganges ein Zeichen angetroffen, das diesem Parameter entspricht, so gilt der Vorgang als beendet. Soll keine Prüfung auf Stopcode erfolgen, so wird als Parameter '80 angegeben. Bei der Anwahl des ASCII-Formats darf der angegebene Stopcode das Paritätsbit nicht enthalten.

Der vierte Parameter spezifiziert die maximale Anzahl von Zeichen, die aus- oder eingegeben werden sollen. Diese Anzahl kann zwischen 1 ('01) und 256 ('00) liegen. Wird der Transfer durch Erreichen dieser Anzahl und nicht aufgrund eines Stopcodes abgeschlossen, ist bit 6 des Betriebsartparameters auf 1 gesetzt.

Die Pufferadresse ist die Anfangsadresse eines Speicherbereiches, aus dem ausgegeben oder in den eingegeben wird.

Tritt während der Ein/Ausgabeoperation ein Gerätefehler auf, so werden die Fehleranzeigen im letzten Parameter abgelegt. Die Bedeutung der Fehlerbits ist geräteabhängig und hardwarespezifisch.

Nach Beendigung eines Transfers kann über Bit 7 - 5 des ersten Parameters eine Erkundigung über den Ablauf eingeholt werden.

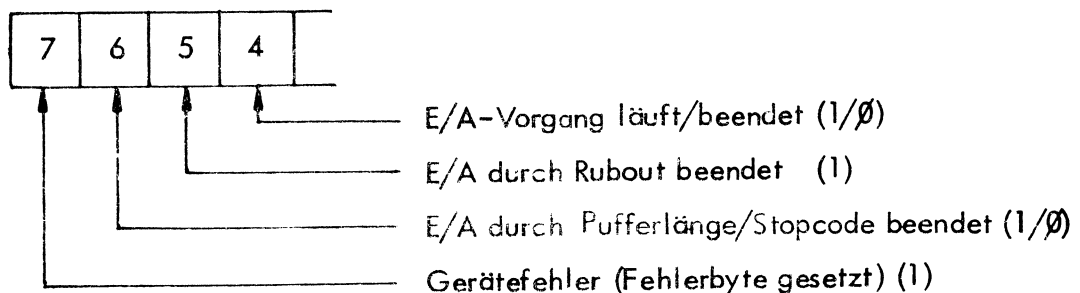


Bild 4b: Ablaufanzeige

Beachte: Der SLI-Treiber sollte nur in Ausnahmefällen direkt angesprochen werden. Statt dessen ist der Aufruf H, '61 zu wählen, wo neben den hier beschriebenen Funktionen einige weitere Leistungen erbracht werden (siehe 2.8)

Modifikation SLI-Treiber für Spezial-Interface:

Bei den Interface-Typen > 602.050 (Stand 10/76) könnten bestimmte Sonderfunktionen durch Ausgabe der Zeichen 'FEXX angewählt werden.

XX = 00 80	Setze Rückwärtslesebit (Lochstreifenleser) Rückschalten auf Vorwärtslesen
XX = 02 82	Wähle zweiten Traktor an (Mannesmann-Drucker) Rückschalten auf ersten Traktor
XX = 03 83	Umschalten gesperrter Code (diverse Drucker) Rückschalten auf Normal-Code
XX = 05 85	Drucken nur von links nach rechts (Mannesmann-Drucker) Drucken in beiden Richtungen

Achtung: Der SLI-Treiber unterstützt entweder nur alte Interfaces oder nach Einbau der Modifikation nur neue. Die Geräte für neue Interfaces sind mit Kennung X4 zu montieren. Die Modifikation wird mit

SERV, MODULE / Modifiziere Standard SLI-Treiber

MTSMOD,, 4

eingebaut.

2.2.7 Platten-Treiber

Aufruf : H, 'Ø1

Parameter : R @ Betriebsart

2 = Lesen

4 = Schreiben

Ø = Abfrage höchster Plattensektor

5 = Modifizieren 1 Sektor

R3/4 Plattensektor

R5 bit Ø - 3 Platteneinheit

bit 4 - 7 Plattensektoradresse - Erweiterung

R6/7 Sektoranzahl zu je 128 byte

R8/9 Hauptspeicheradresse

Fehler : 'Ø3 Platten-Einheit nicht betriebsbereit

'Ø4 Platte schreibgeschützt

'Ø5 Lesefehler (CRC-Prüfung spricht an)
oder Spur nicht gefunden

'Ø6 Parameter unzulässig (z.B. falsche Sektoradresse)

'ØE Fehler des Laufwerks

'ØF Zeitüberschreitung (z.B. zu starke Busbelegung)

Belegung : R @ - R1F

Der Plattentreiber überträgt in Betriebsart 2, 4 eine ganze Anzahl von Sektoren in Einheiten zu 128 Byte in den oder aus dem angegebenen Bereich des Hauptspeichers. Während des Transfers wird das rufende Programm in den Wartezustand versetzt.

Die Plattensektoradresse kann 20 bit lang sein. Dies entspricht etwa 10^6 Sektoren oder 128 MByte pro Einheit.

Die vordersten 4 bit der Adresse müssen in Register 5 (bit 4 - 7) angegeben werden.

Die Betriebsart 0 erlaubt die Abfrage des höchsten (ersten nicht vorhandenen) Plattensektors. Dieser wird in R3-5 zurückgeben.

2.4 MByte :	19 196	512 Byte pro Hardware-Sektor (12 Slot-Platte ^{*)})
4.8 MByte :	38 400	256 Byte pro Hardware-Sektor
Floppy :	2 000	128 Byte pro Hardware-Sektor
DIETZ-Disk :	2 048	128 Byte pro Hardware-Sektor

Die Betriebsart 5 erlaubt die Modifikation eines Plattensektors im Betriebssystem-Arbeitspuffer.

Man erreicht dabei eine automatische Sperre und vor allem höheren Geschwindigkeiten (bei Floppy nicht erlaubt).

Es gilt hier die Sonderregelung:

R6/7	=	1
R8/9		Adresse-Modifikationsprogramm

Die Adresse R8/9 zeigt auf ein Programm außerhalb des Betriebssystemkerns, wo der Plattensektorinhalt (Adresse in R0C) modifiziert wird. Dieses Programm darf nur Register R0 - R9 sowie R0C-R0D verwenden und muß über R0A zurückspringen, wonach das Betriebssystem den Pufferinhalt zurückschreibt.

Beispiel: LDC,@ , 5

LD CD, 3, SEKTOR

LDC, 5, 0

LD CD, 6, 1

LD CD, 8, ROUTIN

CSA, XKER, KERN

H, '01

ROUTIN: LD XD,@ , , '0C

AD CD,@ , 1

ST XD,@ , , '0C

JPX,,, '0A

^{*)} 12 Slot bedeutet: 12 Sektoren pro Spur. Daneben existiert die 2.4 MByte-Platte für 24 Slot. Ein Hardwaresektor umfaßt hier 256 Byte.

Beachte:

1. Der Plattentreiber kann nur Platteneinheiten derselben physikalischen Schnittstelle betreiben. Anschluß mehrerer Plattentypen, zum Beispiel 2.4 MB, DIETZ-Disk, erfordert den Einbau eines zusätzlichen Treibers.
2. Von der logischen Schnittstelle her sind alle Platten-Treiber identisch. Unabhängig von der Zahl der Bytes pro Hardwaresektor, rechnet der Treiber immer in Einheiten von 128 Byte.

Bei Übertragung einer Sektoranzahl, die nicht ein ganzzahliges Vielfaches eines Hardwaresektors ist, ist ein indirekter Transfer über einen Treiber-internen Puffer nötig. Dies hat längere reale Übertragungszeiten zur Folge. Man beachte deshalb die oben angegebene Hardware-Sektoranzahl.

3. Der Platten-Treiber sollte nur in Ausnahmefällen direkt angesprochen werden. Stattdessen ist der Plattenzugriff über das Modul FIAC (siehe 2.6) abzuwickeln, das neben höherem Komfort wesentlich mehr Sicherheit bietet.
4. Der Platten-Treiber des Betriebssystems erlaubt das Ansprechen von Einheit 0 und 1. Sollten darüber hinaus weitere Einheiten adressiert werden können, müssen diese mit folgendem Dienstprogramm in die Unit-Tabelle eingetragen werden:

SERV, DSKMNT Disk-Montage (4.5)

u, w [, a]

u = neue Einheit

w = w1 & w2

w1 = Hardware-Einheit

w2 = Treiber-Nummer

a = Treiber-Adresse (nur bei nicht System-Treiber)

2.2.8 Nachbindbare Platten-Treiber

Bei Rechnerausstattungen mit unterschiedlichen Plattentypen muß das Betriebssystem um die entsprechenden Platten-Treiber erweitert werden. Sie liegen als Modul unter folgender Bezeichnung vor:

MEDISC, t < Ø	2. Platten-Treiber 4.8 MB
ME DISC, t > Ø	2. Platten-Treiber für 2.4 MB (24 Slot)
MCDISC	Platten-Treiber für DIETZ-Disk
MFLOPP	Platten-Treiber für Floppy-Disk

Alle Treiber sind über die in 2.2.7 beschriebene Schnittstelle anzusprechen (Betriebsart 5 : nur Systemresidenz). Sie belegen im Hauptspeicher ca. 1 kByte Platz.

Der Einbau eines Treibers erfolgt mit dem Dienstprogramm MODULE. Die BUS-Adresse ist über das Programm MOUNT in die IOCS-Tabelle einzutragen, mit DSKMNT erfolgt der Eintrag in die Platten-Unit-Tabelle.

Beispiel: Montage einer Floppy-Disk
Treiber-Adresse ' 6ØØØ

SERV, MODULE, MFLOPP, 6ØØØ

SERV, MOUNT, 14, 1AØ2, E4

SERV, DSKMNT, 5, 1, 6ØØØ

In diesem Beispiel erfolgt der Eintrag in Spalte 14 der IOCS-Tabelle, die Nummer der Platteneinheit ist 5. Der zugehörige Treiber hat die Nummer 1 und liegt auf Adresse 6ØØØ.

Der Treiber für eine 4.8 MByte und 2.4 MByte - 24 Slot - Platte ist bis auf die Behandlung der Plattenendgrenze identisch. Der Parameter t spezifiziert den entsprechenden Typ.

Beim Einbau der Platten-Treiber werden als Parameter spezifiziert

n = Treibernummer (1, 2, 3)
t = > Ø 2.4 MByte - MEDISC
 < Ø 4.8 MByte - MEDISC

d = Devicenummer Platte (11, 12, 13, 14)

z = Zusatzsemaphore bei Betrieb mit BUS-Switch (1Ø6)

2.2.9 Magnetband-Treiber

Der Treiber erlaubt die Ansteuerung des Band-Controllers für 7-Spur und 9-Spur 800 und 1600 bpi-Bänder.

Aufruf : H, 'Ø4

Parameter : R (a) Betriebsart
02 Lesen (READ)
04 Schreiben (WRITE)
10 Bandmarke (FILEMARK) schreiben
80 Rückspulen Bandanfang (REWIND)
60 Rückspulen zur Blockanzahl bzw. zur Bandmarke
22 Vorspulen Bandmarke

R5 Bandeinheit (15, 16, ...)

R6/7 Anzahl Byte bei READ-WRITE
Anzahl Blocks bei BACKSPACE-FORWARD

Die Bytezahl (R6/7) pro Block (= ein Lese/Schreibbefehl) kann zwischen 16 und 4095 liegen. Bei BACKSPACE-FORWARD können Ø bis 4095 Blöcke angegeben werden. Die Angabe Ø bedeutet ein Rück- bzw. Vorspulen zur nächsten Bandmarke.

R8/9 Adresse Übertragungspuffer bei Lesen/Schreiben.

In Sprachen (BASEX, CBASIC, FORTRAN) existieren spezielle Bandbefehle zum Ansteuern der Funktionen.

Der Magnetband-Treiber liegt in der Dienstprogramm-Datei SERVIC des Betriebssystem. Bei Aufruf wird er in der Overlay-Bereich des Systems geladen und gestartet.

Bei häufiger Bandbenutzung ist diese Arbeitsweise nicht sinnvoll. Man bindet hier den Treiber resident ein.

Beispiel: Residenter Magnetband-Treiber-Einbau

SERV, MODULE, MTAPE, ~~6000~~

SERV, MOUNT, 15, 1706, E9/ 9-Spur-Magnetband

Die Gerätenummer-Montage MOUNT ist auch bei nicht residentem Band-Treiber erforderlich.

Ein Magnetband wird binär gelesen. Für die Konversion von EBCDIC nach ASCII und umgekehrt stehen Konversions-Routinen zur Verfügung.

Das Dienstprogramm TCOPY (siehe 4.4) erlaubt das Kopieren eines Platten-Volumens auf Band und umgekehrt.

2.2.10 Lochkarten-Leser-Treiber

Der Lochkarten-Leser-Treiber ist von der Benutzerschnittstelle her bis auf das Betriebsart-Byte SLI-compatibel. Dieses kann hier sein:

Ø2	ASCII-Modus (IBM Ø26)
ØA	Byte-Modus
ØØ	Doppelbyte-Modus

Im ASCII Modus wird eine Lochkartenspalte, welche aus Zeile Ø bis 12 besteht, in das ASCII-Format umgewandelt.

Im Byte-Modus erfolgt eine direkte Übertragung von Zeile Ø bis 7 in ein Byte. Im Doppelbyte-Modus wird Zeile 12 - 5 in ein erstes, Zeile 1Ø - 6 in ein zweites Byte übertragen.

Byte	Ø							7	
1	5	4	3	2	1	Ø	12	11	Zeile
2	-	-	-	10	9	8	7	6	

Kann in Betriebsart 2 ein Zeichen auf der Lochkarte nicht nach ASCII konvertiert werden, wird es als ØØ angegeben.

Beachte: 1. In Sprachen ist standardmäßig der Modus Ø2 angewählt. In diesem Modus setzt der Treiber hinter das letzte Zeichen ≠ Blank bzw. in Zelle 81 des Puffers das Zeichen 'ØD. Auf diese Weise läßt sich der Lochkartenleser wie ein gewöhnliches SLI-Gerät mit Warten betreiben.

2. Der Lochkarten-Treiber prüft, ob Devicenummer 8 montiert ist.

Einbau Lochkarten-Treiber : SERV, MODULE, MCARD, Ø

Eintrag IOCS-Tabelle : SERV, MOUNT, 8, 1111, E1

Die Länge des Treibers ist 256 Byte.

2.3-1

2.3 Betriebsmittel-Verwaltung2.3.1 Allgemeines

Eine Rechanlage besitzt Betriebsmittel, die nicht gleichzeitig von mehreren Prozessen, die ein solches benötigen, belegt werden können oder nur von einer beschränkten Zahl belegt werden dürfen. Beispiele dafür sind E/A-Geräte, Speicherbereiche (Puffer), Programme. Im folgenden werden sie auch mit Ressourcen bezeichnet.

E/A-Geräte können nur von einem Prozeß gleichzeitig belegt werden. Dasselbe gilt für Programme, wenn sie nicht "reentrant" sind. Bei der Anlage DIETZ 600/621 kommt dies im allgemeinen nicht vor, da das "Multi-General-Purpose-Register"-Konzept es erlaubt, Programmkontext ohne spezielle "Keller-Technik" zu führen. In Ausnahmefällen ist es jedoch notwendig, den Mehrfachzugriff zu einem Programmstück zu sperren. Das Betriebsmittel "Speicherpuffer" kann prinzipiell durch denselben Mechanismus wie die genannten behandelt werden. Eine verfeinerte Betrachtung, die zu einer Erhöhung des Durchsatzes führt, unterscheidet jedoch zwischen einem Lese- und Schreibzugriff zu dem Puffer. Eine Zuteilung zum Zweck des Lesens kann erfolgen, wenn der Puffer nicht gerade durch einen Schreibvorgang belegt ist, eine Zuteilung zum Zweck des Schreibens kann jedoch nur geschehen, wenn kein anderer Prozeß den Bereich weder liest noch beschreibt.

Jedes Betriebsmittel ist dem System durch eine Nummer bekannt, die sich auf eine Systemvariable bezieht. Der Wert dieser Systemvariablen gibt an, ob ein Betriebsmittel vergeben werden kann. Ist dies nicht der Fall, wird der rufende Prozeß in eine Warteschlange zu diesem Betriebsmittel eingereiht.

In der einfachsten Stufe ist die Systemvariable die binäre Semaphore. Der Wert ist \emptyset , wenn das zugehörige Betriebsmittel belegt ist und 1, wenn es frei ist.

Die Einführung der n-wertigen Semaphore würde die Organisation der Programmbenutzung, die bis zur Stufe n "reentrant" sind, in das Betriebssystem verlagern. Dies hat aber aus obigen Gründen bei der Anlage DIETZ 600/621 keine große Bedeutung.

In einer erweiterten Stufe kann das Konzept "Boltvariable", wie es in der Prozeßrechnersprache PEARL gegeben ist, zur effektiveren Benutzung von Speicherpuffern eingebaut werden.

Bei der Anforderung (Request) zweier oder mehrerer Betriebsmittel durch verschiedene Prozesse können sich sogenannte Betriebsmittelverriegelungen ("Deadlocks") ergeben, wo die Benutzung von Betriebsmitteln völlig blockiert wird. Um dies zu vermeiden, sollte die Anforderung mehrerer Betriebsmittel, die zu einem Verarbeitungsvorgang gehören, "gleichzeitig" erfolgen. Dies geschieht dadurch, daß der Anforderungsprozeß als weiteres Betriebsmittel definiert wird (siehe Bild 7). Eine andere Möglichkeit besteht darin, daß alle Prozesse die Betriebsmittel in derselben Reihenfolge (z.B. aufsteigende Nummer) belegen.

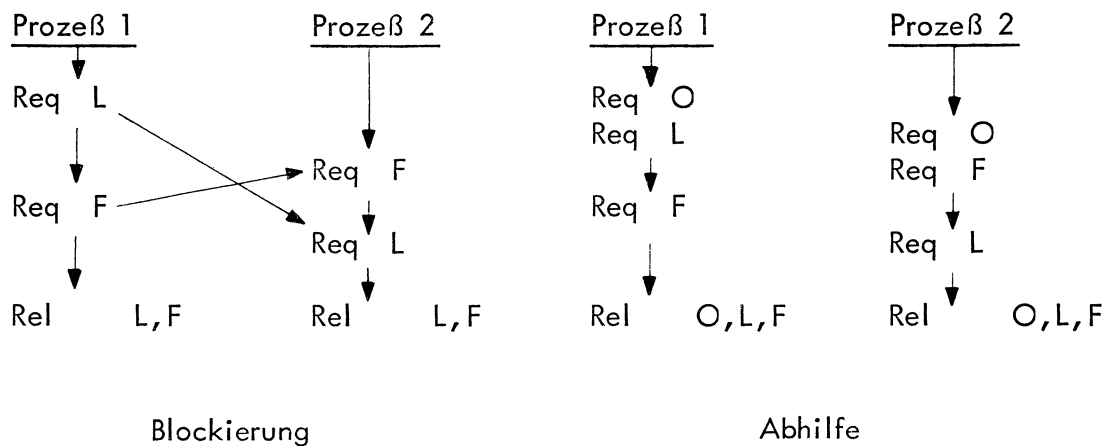


Bild 7: Betriebsmittelvergabe

(Req = Request, Rel = Release)

L = Lochstreifengerät; F = Fernschreiber;

O = allgemeine Semaphore)

Im Betriebssystem selbst ist keine Automatik vorgesehen, Verriegelungen zu erkennen und zu behandeln. Ein Operator kann diese aber durch Abfrage feststellen und Betriebsmittel einem Prozeß entziehen.

2.3-3

2.3.2 Zulassen und Sperren von Geräten

Aufruf : H, '10 ARM (Zulassen eines Gerätes)
H, '11 DISARM (Sperren eines Gerätes)

Parameter : R@ Gerätenummer (031)
0 - 31: Semaphoren für Geräte

Fehler : '11 Gerät nicht existent
'12 Gerät kann nicht gesperrt werden, da
nicht armiert.

Mit dem Befehl "ARM" wird ein Gerät angemeldet und steht ab jetzt der Benutzung zur Verfügung. Mit "DISARM" wird das Gerät der Benutzung entzogen. Alle Prozesse, die es gerade belegen, werden in den Wartezustand versetzt.

Eventuell eintreffende Geräte-Rückmeldungen bleiben unberücksichtigt, so daß Prozesse, die das Gerät belegen, solange warten, bis der Befehl ARM das Gerät wieder zuläßt.

2.3.3 Anforderung und Freigabe von Betriebsmitteln

Aufruf: H,'12 Request "semaphore"
 H,'13 Release "semaphore"

Parameter: R@ Betriebsmittelnummer r
 $\emptyset \dots 127$ oder -1 für alle

 R3 Prozeßnummer p
 $\emptyset \dots 11$ oder -1 für rufenden Prozeß

Ausgang:

Request R @ \emptyset Betriebsmittel konnte belegt werden
p \neq -1 q Betriebsmittel durch Prozeß q belegt

Release R @ \emptyset Betriebsmittel konnte freigegeben werden
p \neq -1 $\neq \emptyset$ keine Freigabe möglich, da nicht belegt

Fehler: '11 Betriebsmittel nicht existent

Belegung: R@- R1 F

Mit dem Befehl "Request" kann ein Betriebsmittel für den laufenden (R3 = -1) Prozeß oder einen anderen angefordert werden. Mit "Release" wird dieses entzogen oder freigegeben (R3 = -1). Falls ein Betriebsmittel bei Anforderung nicht frei ist, kommt der rufende Prozeß in eine zum Betriebsmittel gehörende Warteschlange. Dies geschieht nur bei R3 = -1.

Semaphoren-Nummern dürfen nur unter folgenden Randbedingungen benutzt werden:

-1 bei "Release" bedeutet : Freigabe aller durch den rufenden Prozeß belegten Nummern

\emptyset Resource-Nummer des dem rufenden Prozeß zugeordneten Dialoggerätes, d.h.

 Master/Operator : 1

 Teilnehmer 1 : 21

 usw.

Damit kann beispielsweise erreicht werden, daß das dem Prozeß zugeordnete Dialoggerät für bestimmte Zeit exklusiv reserviert ist.

- 1-31 Langzeitreservierung der Geräte mit den entsprechenden
Gerätenummern durch Benutzerprogramme.
Will ein Programm beispielsweise den Drucker für längere
Zeit exklusiv belegen, so wird er mit "Request, 3" ange-
fordert.
Spezielle Bedeutung haben folgende Plattensemaphore:
- 10 XOS-Overlay-Dienstprogramme
 - 12 Inhaltsverzeichnis der Platte
 - 11 XOS-Overlay-Kommandos
 - 13 Plattentransfer-Puffer
- 32-48 Semaphore für Sonderzwecke in Systemen oberhalb des
Betriebssystemkerns.
Diese dürfen vom Benutzer in keinem Fall verwendet
werden.
- 35 Rechnerkopplungs-Task
 - 36 DFMS-Puffer
 - 37 SPOOLING-Task
 - 38-45 Ein/Ausgabe-Puffer für Prozeßsprachen
 BASEX-FORTRAN für Benutzer Ø bis 8
- 49-95 Freie Verfügung für Benutzer.
Soll beispielsweise ein Programmstück gegen Doppel-
Benutzung abgesichert werden, wird eine Semaphore ver-
einbart, z.B. 5Ø und vor Aufruf der Befehl "Request, 50",
nach Verlassen "Release, 50" eingebaut.
- 96-127 Betriebssystem-interne Semaphore für die Kurzzeit-Reservierung
eines Treibers. Das IOCS sorgt damit für die exklusive Be-
nutzung der Treiberelektronik innerhalb eines Auftrages. Es
ist deshalb nicht erforderlich, für einen IO-Auftrag die Be-
legung durch "Request" anzumelden.

2.4 Prozeß-Verwaltung

2.4.1 Prozeß-Zustände und Übergänge

Das Modul PRMA (Process-Management) verwaltet das Betriebsmittel "Rechnerzeit". Nach verschiedenen Strategien (Mono-, Multiprogramming, Timesharing, Multi-tasking) wird dieses Betriebsmittel den rechenwilligen Prozessen (= Programme, Jobs, Tasks) zugeteilt.

Das Modul führt Buch über den Zustand aller angemeldeten Prozesse und leistet die Überführung zwischen diesen Zuständen nach entsprechenden Anrufen.

Ganz allgemein kann ein Prozeß in einem von 6 Zuständen sein, die in jeder Prozeß-Verwaltung in irgendeiner Realisierung vorkommen (Bild 8).

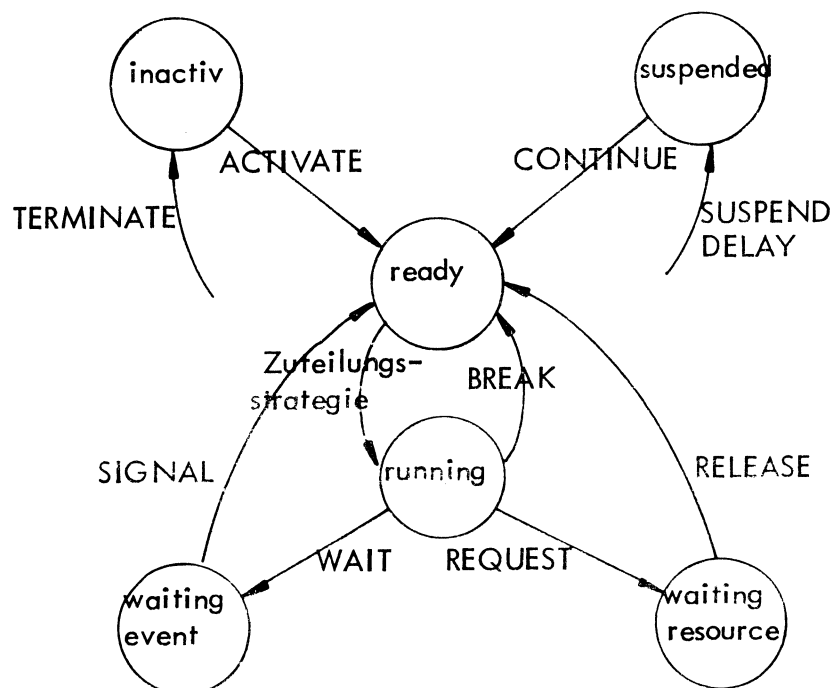
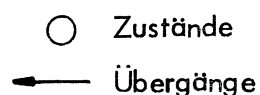


Bild 8: Zustandsschema von Prozessen



Vom Zustand "inaktiv" gelangt ein Prozeß durch den Befehl ACTIVATE in den Zustand "ready". Er wird bei der Neuvergabe des Rechnerkerns berücksichtigt und in den Zustand "running" versetzt, falls die Zuteilungsstrategie ihn dazu bestimmt. Vom Zustand "running" kann er sich selbst durch BREAK wieder in den Zustand "ready" versetzen. Dieser Übergang ist beispielsweise im Timesharing-System sinnvoll, wenn eine Prozeß-Umschaltung an besonders günstigen Programmstellen erfolgen soll. Der Übergang von "running" nach "ready" findet auch nach Ablauf einer Zeitscheibe statt. Durch den Befehl WAIT, der explizit oder in einem E/A-Befehl erteilt werden kann, wird der Prozeß in den Zustand "waiting event" versetzt. Von diesem Zustand gelangt er durch SIGNAL in den Zustand "ready". Dieser Zustand wird vom IOCS nach Beendigung eines E/A-Vorganges, aber auch explizit von einem laufenden Prozeß erteilt.

In den Zustand "waiting resource" geht der Prozeß bei Anforderung eines belegten Betriebsmittels. Sobald der belegende Prozeß im Zustand "running" dieses freigibt und kein anderer Prozeß vorher einen Belegungswunsch angemeldet hat, kommt der Prozeß wieder in den Zustand "ready". Ein laufender Prozeß kann sich selbst oder einen anderen beenden (TERMINATE) oder vorübergehend stilllegen (SUSPEND, DELAY). Fortgesetzt wird er im letzteren Fall durch CONTINUE oder bei DELAY nach Ablauf der angegebenen Zeit.

Das genannte Schema liegt allen DIETZ-Betriebssystemen zugrunde. Es kommen allerdings mehrere Realisierungsformen der Zustände und Zustandsübergänge je nach Zuteilungsstrategie vor. Jedoch sind die folgenden Schnittstellen nach außen hin immer identisch.

2.4.2 Funktionen des Prozeß-Management

Aufruf	:	H, '20	ACTIVATE (Anmelden eines Prozesses)
		H, '21	TERMINATE (Abmelden eines Prozesses)
Parameter	:	R@	Prozeß-Nummer (oder -1)
		R3	Priorität des Prozesses (nur bei ACTIVATE)
		R4/5	Startadresse des Prozesses (nur bei ACTIVATE)

Die Prozeß-Nummer ist eine Zahl zwischen 0 und 11 (16 Ebenen) bzw. 0 und 3 (8 Ebenen). Bei der Konfiguration MPOS, RTOS ist sie zwischen 0 und 12. Die Angabe "-1" (nur bei TERMINATE sinnvoll) beendet den laufenden Prozeß.

Die Priorität eines Prozesses liegt zwischen 0 und 63, wobei 0 die niedrigste Stufe ist. Die Priorität 60 ist im allgemeinen dem Operator-Prozeß vorbehalten. In Betriebssystemen mit Realzeitanwendung ist den Realzeitebenen die Priorität 62 - 59 zugewiesen. Vordergrundebenen haben die Priorität 58 (siehe PRSC).

Wird in Register 3 (ACTIVATE) -1 angegeben, so bekommt der Prozeß diejenige Priorität, die er vor dem TERMINATE (oder 0 zu Beginn) hatte.

Während in einer Multiprogramm-Konfiguration alle Prozesse unterschiedliche Priorität haben, sind sie im reinen Timesharing, abgesehen vom Operator-Prozeß, gleich.

Mit TERMINATE werden alle belegten Betriebsmittel automatisch freigegeben und alle eventuell laufenden Ein-/Ausgaben beendet.

Aufruf	:	H, '22	WAIT	(Warten auf Ereignis)
		H, '23	SIGNAL	(Setzen und Löschen von Ereignissen)
Parameter	:	$R(\textcircled{a}) = e$	Ereignisnummer (Event)	
		$e = -1$	Warte auf irgendein Signal	
Bedeutung e:		\emptyset bis 31	Fertigstellung EA-Transfer	
		36 bis 63	freie benutzbare Ereignisse	

Bei WAIT prüft das Betriebssystem, ob das angegebene Ereignis bereits eingetroffen ist. Ist dies der Fall, so kann der rufende Prozeß weiterlaufen, andernfalls kommt er in den Wartezustand. Ereignisse können Rückmeldungen vom IOCS sein, wenn der Auftrag durchgeführt ist (implizites Signal), aber auch Signale, die durch den Befehl SIGNAL von Prozessen erzeugt werden und so eine Koordination zwischen den Prozessen ermöglichen. Soll nur festgestellt werden, ob ein Ereignis eingetroffen ist, muß bei WAIT das höchstwertigste Bit vom $R(\textcircled{a}) = 1$ sein. Ist nach Rückkehr aus dem Betriebssystem $R(\textcircled{a}) \neq 0$, so ist das Ereignis bereits eingetroffen, andernfalls nicht. Für EA-Meldungen sind die Ereignisnummern $\emptyset - 31$ reserviert. Sie entsprechen den jeweiligen Geräten. Die übrigen Nummern können frei vereinbart werden. In Sprachsystemen sind die Event-Nummern 32 - 36 für spezielle Systemzwecke reserviert.

Ereignisse werden gelöscht, wenn beim Befehl SIGNAL das höchstwertigste Bit von $R(\textcircled{a}) = 1$ ist. Zuvor wird allerdings das Signal gesetzt. Ist dies unerwünscht, so kann zusätzlich bit 6 gesetzt werden.

Aufruf	:	H, '24	SUSPEND	(Zurückstellen eines Prozesses)
		H, '25	CONTINUE	(Fortsetzen eines Prozesses)
Parameter	:	$R(\textcircled{a})$	Prozeß-Nummer	
		R3	Priorität des Prozesses	
			(nur bei CONTINUE)	
			(-1: alte Priorität)	

Bei SUSPEND kann anstelle der Prozeß-Nummer "-1" stehen. In diesem Fall wird der laufende Prozeß suspendiert. Bei CONTINUE eines laufenden Prozesses wird nur die Priorität verändert.

Aufruf : H, '26 BREAK (Abbruch eines Prozesses)
Parameter : Keine

Die Funktion existiert nur bei TSOS und TROS.

Aufruf : H, '27 DELAY (Verzögerung eines Prozesses)
Parameter : R(ⓐ) Prozeß-Nummer (-1: laufender Prozeß)
R3 Zeit (Einheit 100 msec)
(= ∅ : Suspendiere Prozeß)

Der Prozeß wird durch diesen Befehl solange zurückgestellt, bis die angegebene Zeit (0.1 bis 25.5 sec) abgelaufen ist. Dadurch ist beispielsweise eine einfache Zeitüberwachung von E/A-Aufträgen möglich. Falls vor Ablauf der Zeit der Befehl "CONTINUE Prozeß" kommt, wird das Warten abgebrochen.

Aufruf : H, '28 STATUS (Statusübergabe)
Parameter : R(ⓐ) Prozeß-Nummer (-1 laufender Prozeß)

Mit diesem Befehl ist es möglich, den Status eines Prozesses abzufragen. In Register 3 und folgenden werden die entsprechenden Daten übermittelt.

R(ⓐ) : Prozeßnummer
R3 : Zustand des Prozesses
(∅ = inaktiv, 1 = aktiv, 2 = suspendend,
4 = waiting event, 8 = waiting resource und sinnvolle
Kombinationen)
R4 : Priorität
R5 : Resource-Nummer oder Event, worauf gewartet wird
R6/7 : Zeit, die der Prozeß verbraucht hat
R8 : Ebene, die der Prozeß belegt
R9/A : Adresse, wo der Prozeß steht

2.4-6

Aufruf : H, '29 Programmsperre setzen/aufheben
R@ : ≠ ∅ setzen, ∅ aufheben

Dieser Befehl bewirkt, daß bei der Vergabe des Rechnerkerns nur das laufende Programm berücksichtigt wird. Ausgenommen sind die Betriebssystem-Prozesse, die weiterlaufen können. Die Sperre wird bei Befehlen, die in den Wartezustand führen, aufgehoben.

Der Befehl stoppt also im Gegensatz zum Maschinenbefehl DCL/ECL nicht die Betriebssystem-Prozesse. Der Befehl belegt die Register R@ - R5 und R1E, R1F

2.4.3 Fehlermeldungen

- '21 Überführung in verlangten Zustand bei dem angegebenen Prozeß nicht möglich oder Prozeß bereits in diesem Zustand (außer bei TER, CON)
- '22 Startadresse nicht zulässig
- '23 Event ist nicht vorhanden

Achtung ! Die Maschinenbefehle HLT, HSL, SEL sind in TROS und TSOS nicht zugelassen. Sie führen zum Fehlverhalten des Systems, da die Buchführung des Prozeß-Managements über Prozeßzustände umgangen wird. Befehle wie DCL/ECL sind sparsam zu gebrauchen und nur für zeitlich kurze Befehlsfolgen anwendbar.

2.5 Prozeß-Einplanung

2.5.1 Allgemeines

Das Modul "Prozeß-Management (2.4)" hat die Aufgabe, den Rechnerkern an rechenbereite Programme zu vergeben. Die Vergabe-Strategie ist dabei wählbar zwischen:

- Multiprogramming
- Timesharing = Multiprogramming + Timesharing

Im Zustand "running" kann ein Programm sequentiell ablaufen, dabei aber auch parallele Betriebssystem-Prozesse (z.B. IOCS) starten. Ein Programm kann jedoch nicht eigene parallele Programme starten.

Wird der Betriebssystem-Kern jedoch um das Modul "Prozeß-Einplaner (process scheduler)" erweitert, so entsteht ein Multi-Prozeß (= multi-task)-Betriebssystem.

Das Modul PRSC hat mehrere Aufgaben.

Die wichtigste ist die Verwaltung des Betriebsmittels "Rechenebene". Prozesse (Task) können nur dann in den Zustand "ready" gehen, wenn neben anderen Betriebsmitteln eine freie Rechenebene zur Verfügung steht.

Ein Prozeß kann weitere parallele Prozesse mit folgenden Bedingungen anmelden:

- Prozeß soll unmittelbar ohne zeitliche Verzögerung gestartet werden
- Prozeß soll erst nach Eintreffen eines Ereignisses (Event, Interrupt) gestartet werden
- Prozeß soll erst nach Ablauf einer Zeitspanne gestartet werden

Ob der Prozeß tatsächlich unmittelbar nach Eintreffen der Bedingung gestartet werden kann, hängt noch davon ab, ob eine Ebene frei ist. Ist dies nicht der Fall, wird der Prozeß in eine Warteschlange zurückgestellt. Diese hat Platz für je 40 Zeit- und sonstige Aufträge. Es ist jedoch zu beachten, daß eine Zurückstellung im allgemeinen bedeutet, daß die Rechnerlast schlecht kalkuliert wurde. Bei zu häufigem Zurückstellen können sogar Signale bzw. Zeitaufträge verlorengehen. Ein Rechner muß im Multi-Prozeß-Betrieb in der Regel mit weniger als 80 % seiner vollen Belastbarkeit gefahren werden, wenn Spitzenbelastungen aufgefangen werden sollen.

2.5-2

PRSC existiert in 2 die Vergabe-Strategie der Ebenen betreffenden Ausbaustufen.

- Zuordnung einer Ebenennummer zur Priorität des Auftrages
(Strategie: BASEX)
- Freie Vergabe von Ebenen nach Verfügbarkeit
(Strategie: Prozeß-FORTRAN)

Nach außen drückt sich die zweite Strategie jedoch nur in der besseren Ebenen- ausnutzung und im etwas aufwendigeren Verwaltungsalgorithmus aus. Die verfügbaren Funktionen sind gleich.

Bild 9 zeigt, wie die Ebenenbelegung nach Strategie 1 und 2 ist. Es wird dabei angenommen, daß ein System für 2-Benutzerbetrieb gewählt wurde. Dies bedeutet, daß zwei Programme, bestehend aus mehreren Prozessen, den Rechner benutzen.

Ebene	Bedeutung	
F	CNP	
E	IOCS	prioritäre Prozesse
D	IOCS	
C	PRMA	
B	PRSC -	Interrupt-Priorität 62
A	PRSC -	Interrupt-Priorität 61
9	PRSC -	Interrupt-Priorität 60
8	PRSC -	Zeit-Aufträge 59
7	Benutzer 2 -	Aufträge-Priorität 3
6	Benutzer 1	
5	Benutzer 2	2
4	Benutzer 1	
3	Benutzer 2	1
2	Benutzer 1	
1	Benutzer 2	∅
∅	Benutzer 1	

Bild 9a : Ebenenbelegung Strategie BASEX

Ebene	Bedeutung
F	CNP
E	IOCS prioritäre Prozesse
D	IOCS
C	PRMA
B	PRSC
A - 2	durch PRSC vergebbare Ebene
1	Benutzerprogramm 1
Ø	Benutzerprogramm Ø

Bild 9b : Ebenenbelegung Strategie FORTRAN

In der BASEX-Strategie werden Ebenen B - 8 reserviert für die Sonderaufträge Interrupt- und Zeitbehandlung. Die übrigen Ebenen sind den Benutzern fest zugeordnet, sie entsprechen der Auftragspriorität. In der FORTRAN-Strategie belegt PRSC die Ebene B. Die Ebenen A-2 werden frei vergeben. Interruptsignale starten im ersten Fall die zugeordneten Ebenen B-9, Zeitsignale Ebene 8. Im zweiten Fall wird nur Ebene B gestartet.

Die im folgenden beschriebenen Funktionen für Assembleraufruf haben ihre entsprechende Ausdrucksform in den Sprachen:

BASEX = Realtime-Erweiterung von BASIC
Prozeß-FORTRAN = Realtime-Erweiterung von FORTRAN
PEARL

2.5.2 Funktionen des Multiprozeß-Betriebes

Aufruf : H, '30 STOP Prozeß
Parameter : keine

Durch STOP meldet sich ein Prozeß bei dem Modul PRSC ab, so daß die durch ihn belegte Ebene frei wird.

Aufruf : H, '31 START Prozeß unmittelbar
Parameter : R@/R3 Startadresse Prozeß
R4 Systemebene bzw. Priorität
R6/R7 freier Übergabeparameter

Belegung : R@-R1F
Rückgabe : R@ Übergabeparameter

Durch START wird ein Prozeß für die Belegung einer Ebene eingeplant. Der Parameter R4 spezifiziert die Ebene bzw. die Priorität, mit der der Prozeß laufen soll. Er wird jedoch erst dann laufen, wenn die gewählte Ebene (BASEX) bzw. eine Ebene frei ist. Der Parameterwert in R6/7 wird von PRSC bei Wirksamwerden des Startes in R@/3 übergeben.

Aufruf : H, '32 ON INT Start
Parameter : R@/R3 Startadresse Prozeß R4 Eventnummer
(R6 Priorität)
Rückgabe : R@ Interruptnummer
Belegung : R@-R1F

Mit ON INT wird ein sogenannter Interrupt-Auftrag angemeldet. Dies bedeutet, daß PRSC den zugehörigen Prozeß nach Eintreffen des angegebenen Interrupt-Signals starten soll.

Aufruf	:	H, '33	ENABLE Interrupt
		H, '34	DISABLE Interrupt
Parameter	:	R@	Interruptnummer (0 ... 127)
Belegung	:	R@ - R1F	

Damit ein Interrupt überhaupt von PRSC akzeptiert wird, muß er mit ENABLE entriegelt werden. In der Ausgangsstellung sind alle Interrupts verriegelt. Ein Interrupt kann explizit durch DISABLE verriegelt werden.

Weitere Einzelheiten zu Interrupts siehe 2.5.3.

Aufruf	:	H, '35	AFTER Start
Parameter	:	R@/R3	Startadresse Prozeß
		R4-R6	Zeit in Einheiten von msec
		(R8	Priorität)
		R9/10	Parameterwert
Rückgabe	:	R@/3	Parameterwert
Belegung	:	R@-R1F	

Durch AFTER wird ein Prozeß angemeldet, der erst nach Ablauf einer Zeit wirksam werden soll. Ist zu diesem Zeitpunkt eine Ebene frei, so wird der Auftrag gestartet, andernfalls in eine Warteschlange eingereiht und seine Durchführung zurückgestellt.

Beachte: Der Parameter R8 ist nur bei FORTRAN-Strategie zugelassen. In der BASEX-Strategie laufen Zeitaufträge immer auf Ebene 8. Der Benutzer muß hier beachten, daß Zeitaufträge nicht zu lange die Zeitebene blockieren.

2.5-6

Aufruf	: H, '36	CANCEL	Löschen von Aufträgen
Parameter	: R@	Art des zu löschenden Auftrags	
		- 1	= alle Aufträge pro Benutzer
		- 4	= lösche alle Aufträge, welche die Adresse in R3/R4 haben
		- 128	= Lösche alle Aufträge, welche mit einer Adresse > = R3/R4 beginnen
		x	= Interruptauftrag x (0 ... 127)
	R3/R4		Adresse

Mit CANCEL werden die von einem Programm aufgesetzten und vorgemerkten Aufträge gelöscht. Dies kann von jedem Prozeß aus geschehen.

Bei Beendigung eines Programms, welches aus mehreren parallel laufenden Programmen besteht, müssen diese gelöscht werden. Dazu wird der Befehl CANCEL, -1 verwendet. PRSC startet, unabhängig davon, woher der Befehl CANCEL, -1 erteilt wurde, das Programm auf der Hauptebene hinter dem Auftrag.

Aufruf	: H, '38	EVERY	iterierte Zeitaufträge
Parameter	: R@/R3	Startadresse Prozeß	
	R4/R6	Zeitintervall in msec	
	R8	Priorität	
	R9/R10	Parameterwert	

Mit EVERY wird ein Zeitauftrag aufgesetzt, der wiederholt in den angegebenen Zeitintervallen wirksam werden soll. Er muß mit CANCEL beendet werden.

An der Position H, '37 der PRSC-Sprungtabelle stehen folgende Parameter:

1. Byte bit 7 - 1 Useranzahl
 bit 0 = 0 : 32 bit Interface
 1 : 16 bit Interfaces
2. Byte Level, auf dem PRSC läuft

2.5.3 Interrupt-Aufträge

Der Rechner DIETZ 621 besitzt 128 Standard-Interrupteingänge, welche in Gruppen zu 8 bzw. 16 auf Interruptkarten untergebracht sind. Sie haben die BUS-Adressen '38YX (X = Interruptnummer auf der Karte, Y = Kartennummer). Jede Karte kann einer Rechnerebene zugeordnet werden. Dies geschieht bei Montage des Moduls PRSC.

Neben den Standard-Interrupts läßt PRSC auch Sonderinterrupts zu, beispielsweise:

- Zähler-Überlauf-Signal
- CAMAC-Interrupt (LAM)

In BASEX-Strategie werden Interruptkarten der Ebene B, A, 9 zugeordnet. Die entsprechenden Aufträge werden mit der Ebenen-Priorität abgewickelt. In FORTRAN-Strategie entfällt diese explizite Zuordnung. Interrupts starten die Ebene B.

Für Systeme mit der Anforderung optimaler Reaktionszeit wird für die Interrupterkennung eine spezielle Ebene reserviert. Dort läuft ein vom Benutzer zu schreibendes Assemblerprogramm. Nach Erkennen und Bearbeiten des Interrupts kann dieses mit dem Befehl

Aufruf	:	H, '39	Sonder-Interrupt-Erkennung
Parameter	:	R(Ⓐ)	Event-Nummer
		R19	2 Sonderkennung

in den üblichen Ablauf von PRSC verzweigen.

Diese letzte Möglichkeit kann insbesondere dann vorteilhaft verwendet werden, wenn ein Interruptsignal ein Sammel-Interrupt ist (z.B. LAM-Behandlung in CAMAC), wo der eigentliche Interrupt noch identifiziert werden muß.

Beachte: Ein Interrupt wird bei Eintreffen registriert und zurückgestellt, wenn die Ebene nicht frei ist. Kommen vor Fertigstellung eines Interruptauftrages weitere Signale, so wird nur der erste Interrupt registriert; alle anderen gehen verloren.

2.5-8

2.5.4 Fehlermeldungen

- '31 Falsche Interrupt- oder Ebenen-Nummer
- '32 Startadresse nicht zugelassen
- '33 Auftragstabellen-Überlauf
(mehr als 40 Startaufträge oder Zeitaufträge sind angemeldet)

2.5.5 Einbau von PRSC

Das Modul Prozeßeinplanung PRSC ist im Grundbetriebssystem TSOS bzw. MPOS nicht vorhanden.

Mit dem Dienstprogramm SERV, MODULE wird PRSC hinzugeneriert und ergänzt TSOS/MPOS zu TROS/RTOS.

Dabei sind folgende Modul-Parameter anzugeben:

1. Modul BASEX-Strategie

SERV, MODULE, MPRSC, a, i_B , i_A , i_9 k, n, z

- a Moduladresse
- i_B Erste Interrupt-Karte Ebene B
- i_A Erste Interrupt-Karte Ebene A
- i_9 Erste Interrupt-Karte Ebene 9
- k Letzte Interrupt-Karte
- n 8 oder 16 bit Interfacetyp
- z Zahl der Benutzer 1, 2, 4

2. Modul FORTRAN-Strategie

SERV, MODULE, MPRSCF, a, l, e, n, z

- a Moduladresse
- l Ebene, die die Task belegt
- e Zahl der Interrupt (= Event-)Nummern
- n 8 oder 16 bit Interfacetyp
- z Zahl der Benutzer 1, 2, 4

Bei Einbau des Moduls erwartet es nach Ausgabe der Eventnummer ee die Eingabe der zugehörigen Bus-Adresse aaaa, des Startbits im Statuswort bb und eines Eventtypes cc.

Event-Nr.	Ext-ad	Startbit	Typ
ee:	aaaa	bb 1,2,...80	cc 00 für Standard-Interrupt 01 für Zähler 02 für V24 03 für Sonder-Interrupt

Die Eingabe schließt ab, wenn bei aaaa eine 0000 eingegeben wird.

2.6 Datei-Zugriff

2.6.1 Allgemeines

Das Modul "Datei-Zugriff" (FIAC = file access) ist bei allen plattenunterstützten Betriebssystemen vorhanden. Es stützt sich auf den Plattentreiber.

FIAC verwaltet den Platz auf Platteneinheiten (Volumina). Eine solche kennzeichnet eine Platte oder einen Plattenstapel.

Ein Volumen wird durch FIAC in Dateien strukturiert. Alle Dateien werden in einem Inhaltsverzeichnis, welches dem Volumen zugeordnet ist, verwaltet. Dieses enthält folgende Eintragungen:

6 Byte	Dateiname
2 Byte	Sektoradresse Byte 0, 1
2 Byte	Dateilänge Byte 0, 1
1 Byte	Dateiart
1 Byte	Adresse + Längenfortsetzung (Byte 2)
1 Byte	Protectioncode
3 Byte	Autorcode

Der Speicherplatz wird unabhängig von der zugrundeliegenden Plattenart (Sektorgröße 128, 256, 512 Byte) immer in Einheiten von sogenannten Blöcken (logische Sektoren) zu 128 Byte verwaltet. FIAC erlaubt Volumina bis 2^{20} Sektoren = 128 MByte und Dateigrößen von 128 MByte.

Das Byte "Dateiart" kennzeichnet den Inhalt einer Datei. Es ist folgender Schlüssel zugrundegelegt:

Schlüssel Bedeutung

'1X	Quellsprache
12	MARS
14	BASEX/BASIC
15	C-BASIC
16	FORTRAN
17	PEARL
18	PASCAL
19	STAGE-II-Makros
'2X	Intermediärer Sprachcode
21	Relozierbarer Code (Lader)
24	BASEX/BASIC-Interpreter Code
25	C-BASIC-Interpreter Code
'3X	Maschinencode
30	Absoluter Code
31	Relativer Code
32	System Code (SERVIC, TSOS)
'4X	Nichtausführbarer Code (Daten)
40	FIAC-Code
41	DFMS-Code (C-BASIC)
42	DBMS-Code (MAGICS)
43	RFMS-Code (FORTRAN)
44	Formulardateien (MAGICS)

Die Kennzeichen werden von den einzelnen Sprachen geprüft. Sie können über den Befehl "FILE" gesetzt werden.

Das Byte "Protectioncode" enthält den sogenannten Zugriffsschlüssel für Dateisicherung und Dateischutz. Es wird durch den Befehl "PROTECT" gesetzt. Ein Dateischutz ist nur in Verbindung mit dem Autorcode wirksam.

Während das Dateiverzeichnis vom oberen Ende der Platte her ausgebaut wird, wird der Dateibereich vom unteren Ende her zugewiesen. Auf diese Weise hat man nur eine Überlaufgrenze.

In FIAC ist der Dateizugriff blockorientiert. Der Zugriff zu den Blöcken erfolgt direkt über die Sektornummer relativ zum Anfang der Datei. Eine Erweiterung der Dateistruktur und die damit verbundene Verbesserung der Zugriffsform bietet die Systemprozedur DFMS (data file management System). Ihr liegt die Satz- (Record) struktur zugrunde, Sätze sind direkt über Satznummer oder indirekt über Schlüssel, die in Schlüsselverzeichnissen (Index) untergebracht sind, adressierbar. Beide Formen erlauben auch den sequentiellen Zugriff.

Die dritte Ausbaustufe des Dateizugriffs ist DBMS (data based management system), das die Möglichkeiten von DFMS umfaßt, wobei aber als Erweiterung zunächst eine Dynamik in der Satzlänge zugelassen ist und Sätze außerdem in direkt adressierbare Felder unterteilt sind. Dann besteht die Möglichkeit zur Verzeigerung von Dateien.

DFMS ist Bestandteil des kommerziellen BASIC, DBMS der Sprache MAGICS (multi access general information and communication system), die für Informationssysteme entwickelt wird.

Beim Löschen einer Datei tritt das Problem der "Garbage Collection" auf. Es gibt 3 Möglichkeiten für den Zeitpunkt der damit verbundenen Reorganisation der Platte:

- bei jedem Löschen automatisch

- bei Überlauf des Plattenspeichers automatisch

- durch ein spezielles Dienstprogramm.

Hier ist die dritte Möglichkeit realisiert. Allerdings ist es so, daß bei Neueröffnung einer Datei das System prüft, ob ein vorher gelöschter Bereich belegt werden kann.

2.6.2 Funktionen zum Dateiverzeichnis

Mit dem Befehl CREATE beauftragt der Benutzer das Modul "Datei-Zugriff", Platz für eine neue Datei zu reservieren und einen entsprechenden Parameterblock (Name, Länge, Protectionscode,...) im Verzeichnis anzulegen. Durch die Befehle ALTER, LENGTH, PROT, FILE können die Parameter abgeändert werden. Mit KILL wird eine Datei gelöscht, d.h. ihr Parameterblock aus dem Verzeichnis entfernt. Bei der Eröffnung einer Datei erhält sie den Status "privat" und wird mit dem Code des Autors der Datei versehen. Um diesen Code anzumelden, steht der Befehl CODE zur Verfügung.

Aufruf	:	H, '44	CREATE (Neuanlegen einer Datei)
Parameter:		R@	Platteneinheit
		R3 - R8	Name (6 Zeichen)
		R9 - RA	Länge (in Sektoren)
			(Bit 19-16 der Länge steht im linken Halbbyte von R@)

Beim Anlegen einer Datei sucht das System den kleinsten derjenigen Bereiche auf der Platte, die für die Vergabe der in R9/RA spezifizierten Länge ausreicht. Manchmal kommt es allerdings vor, daß von vornherein nicht bekannt ist, wie lang eine Datei wird (z.B. Programmgenerierung). In diesem Falle wird anstelle der Länge die Zahl "-1" angegeben. Das System reserviert daraufhin einen Maximalbereich von 64 KByte (= 512 Sektoren). Außerdem wird bei einem Generierungsversuch kein Fehler gemeldet, wenn die Datei zum angegebenen Namen bereits existiert. Stattdessen steht in R@ : -1 bei Aussprung.

Aufruf	:	H, '45	LENGTH (Längenänderung einer Datei)
Parameter:		R@	Platteneinheit
		R3 - R8	Name der Datei
		R9 - RA	Neue Länge der Datei

Eine solche Längenreduzierung ist beispielsweise dann erforderlich, wenn bei Eröffnung noch nicht genau bekannt war, wieviel Platz die Datei belegen wird.

Aufruf	:	H, '46	PROTECT (Zugriffsschlüssel eingeben)
Parameter:		R@	Platteneinheit
		R3 - R8	Name
		R9	Zugriffsschlüssel

Die einzelnen Bit des Zugriffsschlüssels haben folgende Bedeutung:

Bit 0 : Datei ist beschreibbar (0) oder nur lesbar (1)

Bit 1 : Zur Zeit frei

Bit 2 : Datei enthält ausführbare Programme (0) oder Daten (1)

Bit 3 : Zur Zeit frei

Bit 4 : Lesezugriff zur Datei ist privat (1) oder öffentlich (0)

Bit 5 : Schreibzugriff zur Datei ist privat (1) oder öffentlich (0)

Bit 6,7 : Zur Zeit frei

Bei Neuanlegen einer Datei durch CREATE erhält sie den Zugriffsschlüssel "privat" ('30). Gleichzeitig trägt das System den Berechtigungscod des Autors ein. Mit dem expliziten Befehl PROTECT kann der Autor den Zugriffsschlüssel ändern und die Datei z.B. der Öffentlichkeit zugänglich machen ('00) oder auch nur dafür sorgen, daß sie gegen Schreiben geschützt wird ('21). Auch der Fall, daß nur einer lesen darf, aber alle anderen schreiben können, ist in der Praxis sinnvoll und hier durch Bit 4 realisierbar.

Aufruf	:	H, '47	ALTER Ändern des Dateinamens
Parameter:		R(0)	Platteneinheit
		R3 - R8	alter Name
		R9 - RE	neuer Name

Falls der neue Name gleich dem alten ist, wird kein Fehler gemeldet.

Aufruf	:	H, '48	KILL Löschen einer Datei
Parameter:		R(0)	Platteneinheit
		R3 - R8	Name

Aufruf	:	H, '49	CODE Eingabe eines Codes
Parameter:		R(0) - R4	Code des Autors (3 byte)

Aufruf : H, '4A VOLUME - Name Abfrage

Parameter: R(ⓐ) -R7 Name der Platte

Vor Aufruf ist mit Req, 12 der Zugriff zum Plattenverzeichnis anzufordern, nachher mit Rel, 12 wieder freizugeben.

Aufruf : H, '43 FILE-Kennung

Parameter: R(ⓐ) Unit
R3 - R8 Datei-Namen
R9 Datei-Kennzeichen

Diese Funktion trägt ins Dateiinhaltsverzeichnis die Kennung über die Art der Datei ein. Dieses Kennzeichen ist bei OPEN und durch das FILE-Kommando abfragbar.

Alle Befehle, die in diesem Abschnitt aufgeführt sind, stehen auch als Kommando zur Verfügung.

2.6.3 Funktionen zum Lesen/Beschreiben einer Datei

Bevor zu einer Datei lesend oder schreibend zugegriffen werden kann, ist diese zu öffnen. Das System übergibt daraufhin eine Arbeitsnummer. Bei allen Zugriffen ist diese Arbeitsnummer anzugeben. Auf diese Weise braucht der Zugriff später nicht mehr über das Dateiverzeichnis erfolgen, da das System im Hauptspeicher eine Tabelle mit den physikalischen Daten aufbaut. Nach dem Zugriff (nach einer Folge von Lese/Schreibzugriffen) ist die Datei wieder zu schließen.

Aufruf : H, '40 OPEN Öffnen einer Datei für Lesen/Schreiben
und Beschaffung der Dateiparameter

Parameter: R@ Platteneinheit
R3-8 Name (6 Zeichen)

Rückgaben: R@ Arbeitsnummer (0 - 31)
R3/4 Dateiadresse
R5/6 Dateilänge
R7 Dateiadresse hoch - rechtsbündig
R8 Einheit
R9 Zugriffsberechtigung für rufendes Programm
(2=READ, 4= WRITE, 6=READ/WRITE)
R10 Benutzernummer
R12 Protectionscode
R13 Dateikennzeichen

Ist Bit 7 vor R@ = 1, so wird keine Arbeitsnummer vergeben. Es erfolgt jedoch obige Parameter-Rückgabe.

Aufruf : H, '41 CLOSE Schließen einer Datei

Parameter: R@ Arbeitsnummer (0 - 31) (-1: alle Arbeitsnummern)

Die angegebene Arbeitsnummer steht wieder für weitere Vergabe zur Verfügung. Sind bereits 32 Dateien geöffnet, so wird bei erneutem Versuch Fehler gemeldet.

Bei R@ = -1 werden alle Dateien des rufenden Benutzers geschlossen.

Aufruf : H, '42 GET/PUT = Lesen und Schreiben von Blöcken

Parameter: R@ Arbeitsnummer
R3/4 Kernspeicheradresse
R5/6 Sektoradresse relativ zum Anfang der eröffneten
Datei (Ø = erster Sektor der Datei)
R7/8 Anzahl der zu übertragenden Sektoren
R9 Betriebsart (2 = GET, 4 = PUT)

Bei der Übertragung beachte man, daß die logische Sektorgröße von 128 Byte nicht mit der physikalischen Sektorgröße übereinzustimmen braucht (z.B. 128, 256, 512 byte). Damit der Transfer direkt, d.h. ohne über den Zwischenpuffer des Betriebssystems erfolgen kann, ist in R7/8 möglichst ein Vielfaches der physikalischen Größe anzustreben.

Die Funktion OPEN belegt die Register R@- '21.
CLOSE, GET/PUT belegen die Register R@- '1F.

2.6.4 Fehlermeldungen

- ' 41 Falsche Namensangabe oder Name nicht gefunden oder Datei nicht geöffnet.
- ' 42 Startadresse nicht erlaubt ($\leq '40000$)
- ' 43 Dateilänge falsch ($\leq '0$) oder $L1 < L0$ bei LENGTH
- ' 44 Bei dem gegebenen Zugriffsschlüssel ist verlangte Zugriffsart nicht möglich (Protectioncode Violation)
- ' 45 Überlauf des Plattenspeichers
- ' 46 Bei GET/PUT-Zugriff wird Dateiende erreicht (End of File)
- ' 47 Alle Arbeitsnummern sind bereits belegt
- ' 48 falscher Dateityp

2.7-1

2.7 Arithmetische Operatoren2.7.1 Allgemeines

Die Gruppe 5 des Betriebssystems ist für Module zur arithmetischen Verarbeitung von Daten vorgesehen.

Dabei gilt folgende Zuordnung:

H, '50	2 Byte Integer-Arithmetik
H, '51	4 Byte Binär Floating-Point-Arithmetik
H, '52	8 Byte Binär Floating-Point-Arithmetik
H, '53	8 Byte BCD Floating-Point-Arithmetik

Im Grundausbau ist nur die Gruppe H, '50 vorhanden. Andere Arithmetiken werden bei Bedarf dazugebunden. Insbesondere werden die Arithmetiken in den Sprachen

51 BASIC, BASEX, FORTRAN

52 FORTRAN

53 CBASIC

benötigt.

Das generelle Aufrufschema für die Arithmetik ist:

LDC, '19, n	/ n = Operations-Funktion
CSA, XKER, KERN	/ Betriebssystemaufruf
H, '5X	/X = Arithmetik

Die Operation bzw. Funktion wird einheitlich folgendermaßen angegeben:

- | | | |
|----|--|------------|
| 0 | Addition | + |
| 1 | Subtraktion | - |
| 2 | Multiplikation | \times |
| 3 | Division | / |
| 4 | Negation | Unit - |
| 5 | Absolutbetrag | |
| 6 | Ganzzahliger Teil | |
| 7 | Signum (0 für Zahl = 0, 1 für Zahl > 0, -1 für Zahl < 0) | |
| 8 | Normalisieren | |
| 9 | Konversion ASCII-Darstellung in Zahlendarstellung | |
| 10 | Konversion Zahlendarstellung in ASCII-Darstellung | |
| 11 | Konversion Festkomma in Zahlendarstellung | |
| 12 | Konversion Zahlendarstellung in Festkomma | |
| 13 | | |
| 14 | | |
| 15 | Potenzierung | \uparrow |
| 16 | Wurzel | |
| 17 | Sinus | |
| 18 | Cosinus | |
| 19 | Tangens | |
| 20 | Arcustangens | |
| 21 | Logarithmus | |
| 22 | Exponentialfunktion | |
| 23 | Gleich verteilte Zufallszahlen im Intervall [0,1 [| |

Gruppe 15 - 23 existiert nur für binäre Floating-Point-Arithmetik

2.7.2 2 Byte Festkomma-Arithmetik (Modul '50)

Zahlendarstellung : 2 Byte-Zweierkomplement

Operationen : 2 *
3 /
9 ASCII-Binär
10 Binär ASCII

Registerbelegung : R@ - R1F

Parameterübergabe bei * und / :

I : R@/3 : 1. Operand
RA/B : 2. Operand
0 : R@/3 : Resultat
R4/5 : Divisionsrest

Parameterübergabe bei Konversion ASCII-Binär

I : R@ : Vorzeichen (kann entfallen)
R3-R7 : Ziffern (Abschluß '00')
RA : Konversionsbasis (10 = Dezimal, 16 = Hexa)
0 : R@/3 : Resultat im Zweierkomplement

Parameterübergabe bei Konversion Binär - ASCII

I : R@/3 : Zahl
RA : Konversionsbasis (10 = Dezimal, 16 = Hexa)
0 : R@-R7 : Resultat der Konversion, 6 Zeichen mit führenden Blanks,
Vorzeichen vor erster signifikanter Stelle.

Beachte: Das Modul bietet keine Überlaufprüfung. In der Datei XOSBIB
(siehe XOS-Übersetzungssystem) liegt eine allgemeine Integer-
Verarbeitungs-Routine mit Überlaufprüfung vor.

2.7.3 4 Byte-Real-Arithmetik (Modul '51)

Zahlendarstellung :	Byte 0 - 3	Mantisse binär
	Byte 4	Exponent binär

Stellenwert	:	Bit 0 von Byte 0	1
		Bit 6 von Byte 3	2 ↑ 23

Genauigkeit : 6 - 7 Dezimalstellen

Größenbereich : 10 ↑ - 32 bis 10 ↑ 45

Negative Zahlen : Zweierkomplement

Die Arithmetik verarbeitet nicht - normalisierte Zahlen.
Nach Operationsdurchführung ist die Zahl nicht normalisiert.

Registerbelegung : R(a) - R1F

Operationen : + - * /

Konversionen

viele Funktionen in XOSBIB

Die weiteren Einzelheiten entnehme man dem Handbuch XOS-Übersetzungssystem.

Die 4 Byte-Real-Arithmetik wird mit

SERV, MODULE, MARFB4, a, k

eingebaut.

k gibt an, welche Funktionen eingebaut werden sollen.

Steht ein Gleitkommaprozessor zur Verfügung, so erfolgt der Einbau mit

SERV, MODULE, MGRFB4, a, k

2.7.4 8 Byte Real-Arithmetik (Modul '52)

Zahlendarstellung : Byte 0 - 6 Mantisse binär
 Byte 7 Exponent binär

Stellenwert : Bit 0 von Byte 0 1
 Bit 6 von Byte 3 2 ↑ 55

Genauigkeit : 15 - 16 Dezimalstellen

Größenbereich : 10 ↑ -32 bis 10 ↑ 53

Negative Zahlen : Zweierkomplement

Die Arithmetik verarbeitet nicht - normalisierte Zahlen.
Nach Operationsdurchführung ist die Zahl nicht normalisiert.

Registerbelegung : R0 - R2F

Operationen : + - * /
 Konversionen
 viele Funktionen in XOSBIB

Die weiteren Einzelheiten entnehme man dem Handbuch XOS-Übersetzungssystem.

Die 8 Byte-Real-Arithmetik wird mit

SERV, MODULE, MARFB8, a, k
eingebaut.

2.7.5 BCD - Floating-Point-Arithmetik (Modul '53)

Größe des Paketes	: '700 = 1.8 KByte
Zahlendarstellung	: Byte 0 = LSB Bit 1 - 7 Exponent (Binär-Zweierkomplement) <div style="text-align: right;">Bit 8 Vorzeichen Mantissee</div> <div style="text-align: right;">Byte 1 BCD-Ziffer 10^0, 10^1</div> <div style="text-align: right;">Byte 7 = MSB BCD-Ziffer 10^{12}, 10^{13}</div>
Genauigkeit	: 14 Stellen
Größenbereich	: $1 - 10^{-64}$ bis $9999999999999999 \cdot 10^{64}$
Negative Zahlen	: Betrag und Vorzeichen (in Bit 8 LSB)

Die Darstellung muß nicht normalisiert sein. Nach Operationsdurchführung ist die Zahl nicht normalisiert. Bei Notwendigkeit wird sie vor Operationsdurchführung normalisiert.

Registerbelegung	: R0 - R2F
Operation	: 0 + 1 - 2 * 3 / 4 Negation 5 Absolutbetrag 6 Ganzzahliger Tag 7 Signum 8 Normalisieren 9 Konversion ASCII-Darstellung-Zahlendarstellung 10 Konversion Zahlendarstellung -ASCII-Darstellung 11 Konversion Integer-BCD-Floatingpoint 12 Konversion BCD-Floatingpoint-Integer

2.7-7

Parameterübergaben:

<u>Operation 0 - 3</u>	1 : R@ /R9 RA/R10 0 : R@ /R9	1. Operand 2. Operand Resultat
<u>Funktionen 4 - 8</u>	1 : R@ /R9 0 : R@ /R9	Operand Resultat
<u>Konversion 9</u>	1 : R@ /R19 0 : R@ /R9	ASCII Zeichenstring mit folgenden Zeichen: Ø - 9, ., E, +, - Der Zeichenstring endet mit 'ØØ' BCD-Darstellung
<u>Konversion 10</u>	1 : R@ /R19	Resultat R a Vorzeichen Mantisse (blank/-) R3 Stelle 10 ¹³ ... R15 Stelle 10 ^Ø R16E R17 Vorzeichen Exponent (blank/-) R18 Exponentenstelle 10 ¹ R19 Exponentenstelle 10 ^Ø
<u>Konversion 11</u>	1 : R@ /3 0 : R@ /R9	Integeroperand BCD-Resultat
<u>Konversion 12</u>	1 : R@ /R9 0 : R@ /3	BCD-Zahl Integer-Operand
<u>Registerbelegung :</u>	R@ - R1F R@ - R2F	Operation + - alle übrigen

2.7.6 Hardware-Arithmetik

Der Rechner DIETZ 621 kann mit einem Festkomma- oder Gleitkommaprozessor für 4 Byte Real-Zahlen ausgerüstet werden.

Die Version DIETZ X2 enthält darüber hinaus auch die Möglichkeit, die 8 Byte Binär und BCD-Arithmetik in Hardware zu realisieren.

Das Hilfsprogramm MTSMOD, welches mit
SERV, MODULE

gerufen wird, führt die Einsprünge für Integer- und Real-Arithmetik auf Hardware zurück.

SERV, MODULE, MTSMOD, 2	Festkommaprozessor
, 1	Gleitkommaprozessor
(, 4	neues Interface)

2.8-1

2.8 Komplexe Ein/Ausgabe COIO

2.8.1 Allgemeines

In der Modulgruppe COIO sind alle nicht elementaren Ein-Ausgabefunktionen zusammengefaßt, die über den normalen Treibervorgang hinausgehen. Im Standardbetriebssystem sind nur die Funktion CODC (= Kommandodekodierung '60) und der Gerätetreiber ('61) vorhanden. Weitere Module (DFMS) müssen bei Bedarf über das Dienstprogramm SERV, MODULE nachgerüstet werden.

2.8.2 Kommandodekodierung CODC

Aufruf : H, '6Ø

Parameter: R@/3 I : Adresse des zu analysierenden Strings
 R4/ 5 P : Anfangsadresse der Parameterbeschreibung
 R6/ 7 O : Adresse des Ausgabepuffers

Die Kommandodekodierung dient der Analyse von Kommandos einfacher Syntax anhand einer Steuertabelle. Ein erkennbares Kommando ist
 <Kommando> := <Schlüsselwort> [, <Par >]ⁿ [/<Kommentar>] cr

Ein Schlüsselwort besteht aus mindestens 3 Zeichen:

Parameter können sein

Integerzahlen

Hexazahlen

Textstrings

Beispiel:

MES, Ø, "AUSGABE AUF GERAET"

Die Syntaxanalyse erfolgt anhand einer Syntax-Liste, welche die verschlüsselte Beschreibung möglicher Kommandos enthält. Es ist möglich, die Syntax-Liste im Benutzersystem fortzusetzen.

Jedes Listenelement besteht aus

Adresse der Beschreibungsfortsetzung (2 Byte)

Anzahl der folgenden Kommando-Beschreibungen (1 Byte)

Beschreibung der Kommandos (4 - 260 Bytes)

Jedes Kommando wird beschrieben durch:

Byte: Ø - 2 Schlüsselwort (Zeichen 1-3)
 3 Ausgabecode für das Kommando
 4 Anzahl der zugelassenen Parameter
 5 Parameter 1
 6 Parameter 2
 usw.

Ein Parameter wird folgendermaßen verschlüsselt:

Bit: 0 - 2	Maximalzahl der zu analysierenden Eingabezeichen
3 = 1	Defaultwert bei num. Parametern (-1)
4 = 1	Parameter ist 2 Bytezahl sonst 1 Bytezahl
5 = 1	Parameter in Hexaformat bei bit 6 = 0 Parameter langerString (> 7 Zeichen) bei bit 6 = 1
6 = 1	Textstring sonst entsprechend bit 5-Konvention
7 = 1	Default ist zugelassen d.h. Parameter kann fehlen

Nach Erkennung eines Kommandos wird im Ausgabe-Puffer 0 abgelegt.

Byte 0 :	Code für das Kommandowort
Byte 1 - 7:	Parameterwerte bzw. -adressen (Typ in Klammern)
Dezimalzahl (0) :	2 Byte (bit 4 = 1) 1 Byte (bit 4 = 0)
Hexazahl (01) :	2 Byte (bit 4 = 1) 1 Byte (bit 4 = 0)
Kurz-String (10) :	Anzahl Bytes entsprechend Angabe in bit 0-2 Beschreibung
Lang-String (11) :	Adresse erstes Byte des Strings Adresse letztes Byte des Strings

Bei fehlerhaftem Kommando wird gemeldet

'60 :	Schlüsselwort wurde nicht gefunden
'61 :	Fehler bei Parameteranalyse

Beispiel:

BESCH 1 : 2 * A, BESCH 2	/ Zeiger zur Fortsetzung
D, 3	
T, "END"	/ Kommando END
H, '00	/ Code
D, 0	/ kein Parameter
T, "LOA"	/ Kommando LOAD
H, '02	
D, 2	
H, '82	/ 1 Byte Dezimal-Default 0
H, '47	/ kurzer String

BESCH 2 : 2 * A, 'ØØØØ / keine Fortsetzung
D, 3

Aufruf : LOAD, 3, DATEI

Ausgabe : Ø2 Ø3 444154 4549

2.8.3 Ein/Ausgabe-Treiber

Aufruf : H, '61

Parameter: wie SLI-Treiber H, '00

Der Ein/Ausgabe-Treiber hat im Gegensatz zum SLI-Treiber '00 erweiterte Aufgaben. Solche sind

- a) Belegung des angesprochenen Gerätes über Request
- b) Umsetzen der Gerätenummer 0 über die Break-Tabelle in die eigentliche Teilnehmer-Gerätenummer.
- c) Prüfung der Parameter auf Zulässigkeit
- d) Behandlung bei Foreground-Background-Betrieb
- e) Simulation von Sondergeräten als SLI-Schnittstelle
- f) Auswertung Fehler Byte

Im Normalfall sollte der SLI-Treiber nur über H, '61 angesprochen werden.

Der über den direkten Speicherzugriffs-Kanal angeschlossene 600-Zeilen-Drucker läßt sich vom Benutzer über die übliche Schnittstelle H, '61 ansprechen. Es ist jedoch zu beachten, daß dazu der Einbau eines Sondermoduls MLP600 erforderlich ist. Dieser Einbau geschieht mit:

SERV, MODULE, MLP 600, a, d

d = Devicenummer (3,)

2.8.4 Modul DFMS

1. Allgemeines

DFMS (data file management system) ist ein Betriebssystemmodul, mit dem es möglich ist, satzstrukturierte Dateien zu verwalten und auf sie direkt über Satznummer oder indirekt über Schlüsselverzeichnis zuzugreifen.

DFMS baut auf dem Grund-Modul FIAC (file access) auf, welches bei allen plattenunterstützten Betriebssystemen des Rechners DIETZ 600/621 vorhanden ist.

DFMS wird in Gruppe '60 (komplexe IO-Programme) des Betriebssystems eingebaut und wie üblich über den Befehl:

LDC, '19, n	/ n = Funktionsnummer
CSA, XKER, KERN	/ Betriebssystemaufruf
H, '68	/ Modulgruppe

angesprochen. Folgende Funktionen sind vorhanden:

n = 0	CREATE	Arbeitsdatei erzeugen
1	CRIND	Schlüsseldatei erzeugen
2	OPEN	Datei eröffnen
3	EXOP	Datei exklusiv öffnen
4	CLOSE	Datei schließen
5	UPDATE	Lesen zum Zweck des Veränderns
6	MODIFY	Modifizieren
7	READ	Lesen
8	WRITE	Schreiben
9	SREC	Selektiere Satz direkt oder über Schlüssel
10	GEKEY	Lese Schlüssel
11	SEKEY	Suche Schlüssel
12	ENKEY	Eintragen Schlüssel

13	COKEY	Verknüpfen Schlüssel
14	UNKEY	Löschen Schlüssel
15	REKEY	Umbenennen Schlüssel

Fehlermeldungen in DKER ('7E)

100	End of File
101	End of Record
102	Falscher Dateityp
103	Nicht existenter Key, Key = 0, Sortierung nicht eingehalten doppelter Key
104	Zu große Dateiparameter
105	Arbeitsnummer verbraucht oder falsche Arbeitsnummer

Parameterübergabe in : R@ und folgende

Registerbelegungen : R@ - R4F

Modulgröße : ca. 4 KByte

Belegungssemaphore : '24 , Signale: '22, '23

Im folgenden werden nur die Schnittstellen beschrieben. Eingehende Beschreibung der Funktionen siehe C-BASIC-Beschreibung.

Das Modul DFMS wird eingebaut mit

SERV, MODUL

MDFMS, a, p, l, z, m

a = Adresse, an der das Modul beginnt

p = Arbeitspuffergröße (512, 1024, 2048, 4096)

l = Zahl der Arbeitsnummern (1 bis 255)

z = Zahl der Arbeitspuffer (0,1 ...12)

m > Ø Maskenzeichen * zugelassen

Die Arbeitspuffergröße bestimmt die Anzahl von Sektoren, die im Speicher bearbeitet werden können. Jede geöffnete Datei benötigt eine Arbeitsnummer. Pro Nummer reserviert das System 20 Byte. Wird bei z eine Ø angegeben, so existiert 1 Arbeitspuffer, der bei Mehrbenutzerbetrieb verwaltet wird. Bei mehreren Puffern läßt sich der Durchsatz erhöhen.

Mit 16 Arbeitsnummern und 512 Byte-Puffer benötigt das Modul etwa 4 KByte.

2. Funktionen Dateiverzeichnis

Funktion :	R19 : Ø	CREATE
Parameter :	R@ :	Unit
	R3-8:	Dateiname (6 byte)
	R9-A:	Satzzahl (< 32767)
	RB-C:	Satzlänge (< 32767)

Bei dieser Funktion wird eine DFMS-Arbeitsdatei angelegt. Die ersten 20 Byte enthalten den Dateibeschreibungsvektor, die übrigen werden auf Ø gesetzt.

Funktion :	R19 : 1	CRIND
Parameter:	R@ :	Unit
	R3-8:	Dateiname
	R9-A:	Schlüsselzahl (< 32767)
	RB-C:	Schlüssellänge (< 12Ø)
	RD :	Hashverfahren + Direktbereichsgröße

Diese Funktion erzeugt und formatiert eine Schlüsseldatei. Sie enthält ebenfalls 20 Byte Dateibeschreibung.

RD enthält im rechten Halbbyte eine Angabe über die Größe des Direktbereiches, im linken eine Angabe über die Art des Zugriffverfahrens (s. C-BASIC-Handbuch)

2.8-9

Der Dateibeschreibungsvektor hat folgenden Aufbau:

Byte	Direktdatei	Schlüsseldatei
Ø	'Ø X (X 3 Entwicklungskennung)	'CX
1 2	Satzanzahl	Schlüsselanzahl
3 4	Satzlänge + 2	Satzlänge + 8
5 6	Ø	logisch erster Schlüsselsatz
7 8	Ø	Hash-Maske
9 A	2	Blockzahl + Dateikennung
B C		erste freie Nummer im Indirektbereich
D E	nächster sequentiell belegbarer Satz	Ø

3. Funktionen OPEN-Tabelle

Funktion : R19 : 2 OPEN, 3 EXOP

Parameter: R@ : Unit (oder -1)

 R3/8 : Datei / Index-Name

 R9 : Index-Unit oder FF, falls eine Datei geöffnet

 RA/F : Index-Name

Rückgabe R@/3 : Arbeitsnummer

Mit OPEN wird eine Arbeitsdatei oder Schlüsseldatei oder eine Arbeitsdatei über Schlüsseldatei geöffnet. Im ersten Fall enthält R9 : 'FF. Damit ergeben sich 2 Adressierungsmöglichkeiten für die DFMS-Sätze:

- a) direkt
- b) indirekt (Öffnung zweier Dateien)

EXOP prüft, ob die Datei bereits geöffnet ist. Falls ja, wird der rufende Teilnehmer in den Wartezustand versetzt (Signal '21). Falls die Datei nicht geöffnet ist, wird der Zugang zu ihr bis CLOSE gesperrt.

Bei OPEN/EXOP werden die zugriffsrelevanten Dateiparameter (16) in der OPEN-Tabelle vermerkt. Doppelöffnungen belegen 2 Arbeitsnummern. Bei R@ = -1 wird der Pufferinhalt gelöscht.

Funktion : R19 : 4 CLOSE, CLOSE ALL

Parameter: R@/ 3 : Arbeitsnummer oder FFFF

CLOSE löscht den OPEN-Vektor für die unter der Arbeitsnummer eröffnete Datei oder für beide Dateien, gibt eine Satzbelegung (UPDATE) und eine Dateibelegung (EXOP) frei.

CLOSE ALL löscht alle Eintragungen des rufenden Benutzers.

4. Lesen und Beschreiben von Sätzen

Funktion : R19 : 5 UPDATE, 6 MODIFY
7 READ, 8 WRITE

Parameter: R@ /3 : Arbeitsnummer
R 4 /5 : Adresse Variablenliste
R 6 : Anzahl Variabler
R 7 : < 0 falls mit Next Record

Mit READ, UPDATE werden Daten aus dem gerade angewählten Satz (Stand und Satzzeiger in Opentabelle) gelesen und den Variablen der in R4/5 angegebenen Variablenliste zugewiesen. Die Anzahl der Variablen steht in R6.

Mit WRITE, MODIFY werden Daten in den gerade angewählten Satz (Stand Satzzeiger in Opentabelle) geschrieben. Die Daten werden ebenfalls über die Variablenliste beschafft.

Aufbau der Variablenliste:

Länge }
Adresse } Variable 1 (4 Byte)

Länge }
Adresse } Variable 2 (4 Byte)

usw.

Die Länge spezifiziert die Zahl der zu übertragenden Zeichen, die Adresse weist auf das erste Zeichen der Folge.

Besonderheiten bei den einzelnen Funktionen:

- READ : Übertragung ab Stand des Satzoffsetzeigers im Satz
- UPDATE :
1. Falls Satz nicht belegt, gegen Zugriff sperren
2. Satzoffsetzeiger in Opentabelle (Byte E,F) retten
3. Übertragung ab Stand des Satzoffsetzeigers im Satz
- WRITE : Übertragen der Daten ab bisherigen Stand des Satzendezeigers. Nach Übertragung Satzendezeiger auf den Stand des erreichten Satzoffsetzeigers stellen.
- MODIFY :
1. Satzoffsetzeiger mit dem vorher geretteten Stand (Byte E,F der Opentabelle) laden.
2. Prüfen, ob Satz frei, falls nicht, warten.
3. Übertragen
4. Satzoffsetzeiger wie bei UPDATE retten

In Byte R7 < 0 , wird bei allen Funktionen der Satzzeiger fortgeschaltet und der Satzoffsetzeiger auf 0 gestellt. Außerdem wird die Satzbelegung aufgehoben. Bei Byte R7 > 0 bleibt der Satzzeiger, der Satzoffsetzeiger und die Satzbelegung erhalten.

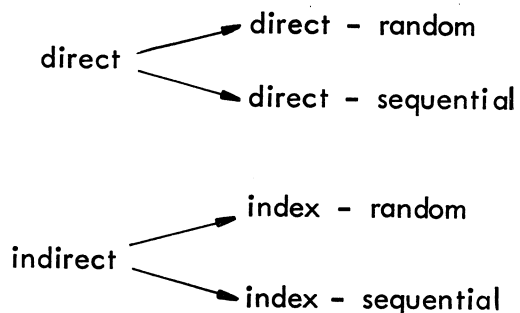
Fehler 101 entsteht dann, wenn bei Fortschalten des Satzoffsetzeigers der Satzendezeiger erreicht wird.

5. Anwahl eines Satzes

Funktion : R 19 : 9 Select Record
Parameter: R@ / 3 : Arbeitsnummer
 R 4 / 5 : Schlüssellänge oder \emptyset
 R 6 / 7 : Schlüsseladresse oder Satznummer

Die Funktion wählt einen Satz direkt über Satznummer ($R4 + 5 = 0$) oder über Schlüssel an. Eine eventuelle Satzreservierung durch die betreffende Arbeitsnummer wird aufgehoben. Die Satzzeigereintragung der Direkt- und Schlüsseldatei werden entsprechend den Parametern R4 - R7 gesetzt, der Satzoffsetzeiger auf 0 gestellt.

Zugriffsarten:



6. Schlüssel lesen

Funktion : R 19 : 10 GEKEY
Parameter: R@ / 3 : Arbeitsnummer
 4 / 7 : Zielvariable für Schlüssel (Länge u. Adresse)
 8 / 9 : Zielvariable für Satznummer
 RA : < 0, falls neue Schlüsselwahl

Der durch den Satzzeiger (Schlüssel OPEN-Tabelle) angewählte Schlüsselinhalt wird durch GEKEY den angegebenen Variablen zugeordnet.

Eintragungen der Schlüsseldatei:

- 2 Byte: Zeiger auf Nachfolger im Indirektbereich
- 2 Byte: Zeiger auf Direktsatz (nach <R8>)
- n Byte: Schlüssel (nach < R6>)

Falls RA < 0 ist, wird anschließend zum nächsten eingetragenen Schlüssel weitergeschaltet.

7. Suchen eines Satzes

Funktion :	R 19	:	11 SEKEY
Parameter:	R 0 /3	:	Arbeitsnummer
	R 4 /7	:	Schlüssellänge + -adresse
	R 8 /B	:	Zielvariable
	RC	:	Operator (Bit 1 = , Bit 2 < , Bit >)

Mit SEKEY wird derjenige Schlüssel bestimmt, der die Bedingung:

$$= , > , < , \quad > = , < = \text{Schlüssel R4/7}$$

erfüllt. Im Schlüssel R4/7 können die Maskenzeichen * angegeben sein, welche nicht zum Schlüsselvergleich herangezogen werden. * ist nur bei sortiertem Hash anstelle 2, 3, 4 ... sinnvoll. Die Operatoren meinen die Reihenfolge der Eintragungen im Schlüsselverzeichnis. Der gefundene Schlüssel wird in die Zielvariable übertragen. Durch SEKEY wird der Satzzeiger der Indirektdatei und zugehörigen Direktdatei gesetzt.

Der erste in einem Schlüsselverzeichnis eingetragene Satz wird bereits durch OPEN angewählt.

8. Eintragung eines Schlüssels

Funktion :	R 19	:	12 ENKEY
Parameter:	R 0 /3	:	Arbeitsnummer
	R 4 /7	:	Schlüssel
	R 8 /9	:	Satznummer oder FFFF
	RA	:	Ø Eintrag nach Dateityp, 1 Eintrag sortiert, 2 Eintrag aufsteigend sortiert

Mit ENKEY wird der durch R4/7 angegebene Schlüssel in das Schlüsselverzeichnis eingebracht. Gleichzeitig wird ihm die Nummer des in R8/9 angegebenen Satzes zugeordnet. Falls dort FFFF steht, wird die Satznummer aus der zugehörigen Direktdatei bestimmt und diese anschließend inkrementiert.

Zur Durchführung der Operation wird ein Arbeitsbereich von 128 Byte benötigt.

Der Schlüsseleintrag geschieht über die bei CRIND angegebene Hashfunktion.

Bei Mehrdeutigkeiten und numerischen Schlüsseln wird Fehler gemeldet.

Funktion :	R 19	:	13 COKEY
Parameter:	R@ /3	:	Arbeitsnummer 1
	R 4 /7	:	Schlüssellänge und Adresse 1
	R 8 /9	:	Arbeitsnummer 2
	RA/ D	:	Schlüssellänge und Adresse 2

Der Befehl COKEY wird wie ENKEY, außer daß die einzutragende Satznummer zuerst über den Schlüssel 2 der zweiten Datei beschafft wird.

COKEY ist also:

SEKEY W2, A \mathcal{S} = Schlüssel 2

GEKEY W2, A \mathcal{S} , A

ENKEY W1, Schlüssel 1, A

9. Löschen, Umbenennen Schlüssel

Funktion	R 19	:	14 UNKEY
Parameter:	R@ /3	:	Arbeitsnummer
	R 4 /7	:	Schlüssel

2.8-15

UNKEY löscht den angegebenen Schlüssel, d.h. das erste Schlüsselzeichen wird auf \emptyset gesetzt. Der Schlüssel wird aus dem Zeigerverband gelöst. Der freigewordene Platz ist jedoch nur im Direktbereich ohne Reorganisation wieder besetzbar.

Funktion :	R 19	:	15 REKEY
Parameter:	R@/3	:	Arbeitsnummer
	R4 /7	:	Schlüssel 1
	R8 /B	:	Schlüssel 2

REKEY = UNKEY Schlüssel 1 + ENKEY Schlüssel 2

2.8.5 Interbus-Treiber

Bei Kopplung zweier Rechner über die sog. Interbus-Schnittstelle benötigt das System als Ausbaustufe den

Interbus-Treiber MINBUS.

Dieser Treiber regelt den Datenverkehr zwischen beiden Rechnern. Er wird mit SERV, MODULE eingebaut und über

H, '62

angesprochen. Die Parameterübergabe ist wie beim Treiber H, '61. Bei einem Eingabeauftrag schreibt der Treiber in den Längenparameter die tatsächliche Länge der übertragenen Daten.

Montagebeispiel:

SERV, MOUNT, 5, 3 F80 , A0

SERV, MODULE, MINBUS, 0, 5

Wie üblich ist für den Interbus eine Devicenummer (hier 5) festzulegen. Diese muß im Modul als Parameter angegeben werden.

2.9-1

2.9 Sondermodulgruppe

Die Sondermodul-Gruppe enthält Funktionen, die zu speziellen Zwecken dienen, die nicht in Gruppe 0 bis 6 einzuordnen sind. Diese werden mit dem Befehl:

CSA, XKER, KERN

H, '7 X

aufgerufen.

Alle Kommandos der Gruppe 7 X belegen die Register R 0 - R4F sowie R70 bis R7F.

Zur Zeit sind folgende Funktionen verfügbar:

Aufruf	:	H, '70	<u>Kommandoanalyse</u>
Parameter:		R 0 / R3 =	Textadresse
		R 4 / R5 =	0000
		R 6 / R7 =	Adresse Parameterpuffer

Das Kommando erlaubt die Analyse und Ausführung von Zeichenstrings nach vorgegebener Verschlüsselung der Betriebssystem-Kommandotabelle. Die Parameter werden im angegebenen Parameterpuffer abgelegt.

Aufruf		H, '72	<u>Rückkehr Dialogbetrieb</u>
Parameter:		keine	

Mit diesem Befehl kann in den Dialogteil des Betriebssystems gesprungen werden. Das laufende Programm wird mit CLOSE ALL, RELEASE ALL terminiert.

Aufruf	:	H, '71	<u>Overlay-Aufruf</u>
Parameter:		R 0 / R11	Versorgungsparameter des Overlayprogramms
		R16 / R17	Nummer des Overlayprogramms

Die Overlayfunktion lädt das durch R16/R17 adressierte Programm von der Dienstprogrammdatei SERVIC in den Overlaybereich (Adresse in R3E / R3F) des Betriebssystems und startet es. Das Programm kann die

Register R@/R11 zur Parameterspeicherung verwenden. Es kehrt am Ende über JPA,, '406A zum normalen Betriebssystemaufruf zurück. Durch die Overlayfunktion werden alle Register bis R21 zerstört.

Aufruf : H, '76 Link Overlay

Parameter: R 5/R 6 Sektor des nachzuladenden Programmes

Mit LINK können Overlayteile (4 Sektoren) nachgezogen werden. Die Overlaypufferadresse ist stets

5C00	schneller Overlay
5E00	langsamer Overlay
5BFE	enthält den aktuellen Inhalt des schnellen Puffers (Sektornummer in SERVIC)

2.10-1

2.10 Systemvariable

Systemvariable sind Parameter, die vom Betriebssystem je nach Konfiguration gesetzt und von Benutzerprogrammen zur speziellen Verwendung abgefragt werden können. Die Variablen sind in festen Speicherplätzen im Betriebssystem untergebracht und haben folgende Bedeutung:

Adresse	Name	Bedeutung
' 4008	CCNP	Nummer der CNP-Ebene ('0700 oder '0F00)
' 400A	CFSA	Freispeicher-Anfang (erstes belegbares byte des Speichers)
* 5 ' 400C	CFSE	Freispeicher-Ende erste freie Adresse modulo 256 Diese beiden Adressen markieren den benutz- baren Speicherbereich

Beispiele:

48 K : 0100
96 K : 01C0
128 K : 0200

' 400E	CTIM	Zeiger auf Uhrzeit und Datum Millisekunde (3 Byte) 0...3599 999 Sekunde (1 Byte) 0...59 Minute (1 Byte) 0...59 Stunde (1 Byte) 0...23 Tag (1 Byte) Monat (1 Byte) Jahr (1 Byte)
--------	------	--

Uhrzeit und Datum werden vom System geführt.

' 4010	CCLO	Zeiger auf Uhrtakt (3 Byte) + Zeitscheibe (1 byte, neg., Einheit 100 msec)
--------	------	---

' 4012	CNBP	Zeiger auf Tabelle zur Behandlung von Systemfehler Netzfehler Programm N Busfehler Programm B Parität Programm P
--------	------	--

Bei Netzwiederkehr wird der Rechner bei der Adresse '4000 gestartet. Dort sind 3 byte reserviert für die Aufnahme des absoluten Sprungbefehls zu einem Programmteil, der das Programm an definierter Stelle wieder aufsetzt.

Sind in der Systemtabelle keine Adressen eingetragen, so erfolgt bei BUS- und Paritätsfehlern die Ausgabe der Speicheradresse und Partitionsnummer, wo der Fehler aufgetreten war. Eine solche kann bei Netzausfall nicht mehr erfolgen, so daß hier zum Zweck irgendwelcher besonderer Maßnahmen (max. zur Verfügung stehende Zeit: 1 msec) in jedem Fall eine Adresse eingetragen sein sollte.

'4014	CIOT	Adresse der Input/Output-Tabelle Der Inhalt der Tabelle ist in Kapitel 2.2 beschreiben.
'4016	CDIA	Fortsetzen nach Systemgrundstellung
'4018	CBRE	Adresse einer Tabelle, wo die logischen Nummern derjenigen Geräte stehen, die den Prozessen zugeordnet sind, sowie die Adressen der Reaktionsprogramme bei Betätigung der BREAK-Tasten an diesen Geräten.

Neben dem Readybit (bit 0) des Statuswortes kann auch das Bit 6 einen Start erzeugen. In diesem Fall wird in der BREAK-Tabelle die Geräte-Nummer gesucht und der zugehörige Prozeß terminiert, sowie an der eingetragenen Adresse wieder aktiviert. Wird die Geräte-Nummer nicht gefunden, oder ist die Startadresse nicht vorhanden, so ist der BREAK unwirksam. Die BREAK-Tabelle ist nach Prozeßnummern geordnet und enthält im 1. Byte die Geräte-Nummer, im 2.-3. Byte die BREAK-Startadresse.

'401A	CESY	Adresse einer Fehlertabelle, welche für die einzelnen Benutzer angibt, wie die Fehler, die im Betriebssystem auftreten, behandelt werden sollen. Enthält die Tabelle für einen Benutzer die Eintragung '0000, so kehrt das Betriebssystem direkt zum rufenden Programm zurück, das durch die Abfrage von '7E hinter
-------	------	---

dem Aufruf den Fehler behandeln kann. Bei Eintragung einer Adresse wird zu dem angegebenen Punkt verzweigt. Dort erwartet das System ein Programm, das mit '7E den Fehler generell behandelt.

'401C	CRES	Adresse der Resource-Tabelle. Sie enthält pro Resource 1 byte, wobei das rechte Halbbyte die belegende Prozeßnummer, das linke Halbbyte die Zahl der auf die Resource wartenden Prozesse enthält.
'401E	CUCO	Adresse Usercode-Tabelle
'4057	CRESI	Nummer der Systemplatte
'4058	KERE	Rückkehradresse bei nicht vorhandener Funktion
'405A	KE00	Modul-Gruppen-Beginn
'406A	KERR	Rückkehradresse Modul
* 5 '406C	CPAR	Adresse der Tabelle für die Benutzerbereiche
'406E	CBAS	Adresse der Tabelle für die Basisregister der Ebenen
'4070	COM	COMMON-Speichergrenze
'4072	CUSE	Benutzernummer für Kommando USE
'4074	CINT	Interpretersprache 1. Byte: 0 keine, 1 BASIC, 2 BASEX, 3 C-BASIC
'4075-'4077		Module-Kenn-Bit
'4078	CNAM	Adresse des Betriebssystem-Namens
'407C	CSPA	Anfangsadresse des Interpretersystems

Die Register '7E und '7F haben auf allen Rechnerebenen folgende feste Bedeutung und dürfen anderweitig nicht verwendet werden:

'7E VUSN Benutzernummer

Beim Mehrbenutzerbetrieb in einer Programmiersprache, die interpretierend abgearbeitet wird (BASEX), ist es notwendig, abfragen zu können, welcher Benutzer gerade auf den allen gemeinsamen Zeit- und Interruptebenen aktiv ist. Beispielsweise müssen bei der Interpretation benutzerspezifische Parameter angesprochen werden. Dasselbe gilt für das Ansprechen von Geräten, die benutzerspezifisch sind. Zu diesem Zweck führt das System eine Zelle VUSN ('7F) pro Ebene, wo die Nummer des Benutzers (= identisch mit der Nummer des Arbeitsplatzes) enthalten ist, der die betreffende Ebene belegt. Von der Benutzernummer ist die Ebenen (= Prozeß)-Nummer zu unterscheiden. Ohne PRSC sind beide identisch.

'7E DKER Fehlerregister

Das Betriebssystem verwendet diese Zelle generell zum Eintragen der Fehlernummer. Dasselbe sollte auch bei allen Anwenderprogrammen geschehen.

Beim Einsprung ins Betriebssystem wird DKER gelöscht.

Es wird empfohlen, bei allen Assembler-Programmen folgende Konvention einzuhalten, die für Systemprogramme(Interpreter, Compiler) gilt:

'7C XREL Relativierungsregister

'7A XØ Rückkehrregister UP-Stufe Ø

'78 X1 Rückkehrregister Stufe 1

usw.

Stufe Ø darf keine weiteren UP's aufrufen, Stufe 1 nur solche der Stufe Ø, usw.

'7Ø-'4Ø Register zur Speicherung von Werten, die globalen Charakter haben

'2Ø-'3E Register zur Speicherung von Werten mit lokalem Charakter

'3E Rückkehrregister für Operatoren der Bibliotheksdatei

'1E Rückkehrregister für Betriebssystemaufrufe

2.11-1

2.11 Arbeitsweise der Betriebssystem-Konfiguration

2.11.1 Allgemeines

Am praktischen Bedarf orientiert, haben sich 4 Konfigurationen herausgebildet, die als DIETZ-Standard angeboten werden:

MPOS-Operating-System für Multi-programming

RTOS -Operating-System für Realtime-Anwendung

TSOS -Operating-System für Time-Sharing

TROS -Operating-System für Time-Sharing in Realtime-Anwendung

Alle 4 Konfigurationen laufen mit und ohne Plattenunterstützung. Sie unterscheiden sich durch :

- Unterschiedlichen Ausbau des IOCS
- Unterschiedliche Zuteilungs-Strategie in PRMA
- Vorhandensein des Moduls PRCS.

Alle Betriebssysteme haben das in Kapitel 3 beschriebene Dialogmodul , wobei bei einer Nichtplatten-Version die Plattenbefehle auf einen Fehlerausgang führen. Alle plattenunterstützten Systeme haben automatisch die Module Platten-Treiber, FIAC und die Plattendialog-Befehle.

2.11.2 MPOS

Das Betriebssystem MPOS (Multi-Programming Operating-System) erlaubt den "gleichzeitigen" Ablauf mehrerer Programme, die gemeinsame Betriebsmittel benutzen. Die Programme stehen jedoch nicht in gegenseitigem Kontakt wie beim Multitasking. Die Betriebsmittel-Verwaltung erfolgt auf der Basis der Semaphore. Im IOCS ist der SLI-Treiber und eventuell der Platten-Treiber vorhanden.

Die Programmabwicklung wird stark von der Ebenenstruktur des DIETZ 600/621 unterstützt. Deshalb benötigt das Prozeß-Management keine eigene Ebene. Die Umschaltung von einer Ebene auf die andere erfolgt bei Ein-/Ausgabe- und Wartebefehlen, bei nicht erfüllbarer Betriebsmittel-Anforderung und bei Aktivieren und Beenden von Programmen höherer Priorität.

Bild 12 zeigt die Ebenenverteilung in MPOS

Ebenenbelegung

MPOS läuft entweder auf der 8- oder 16-Ebenen-Konfiguration.

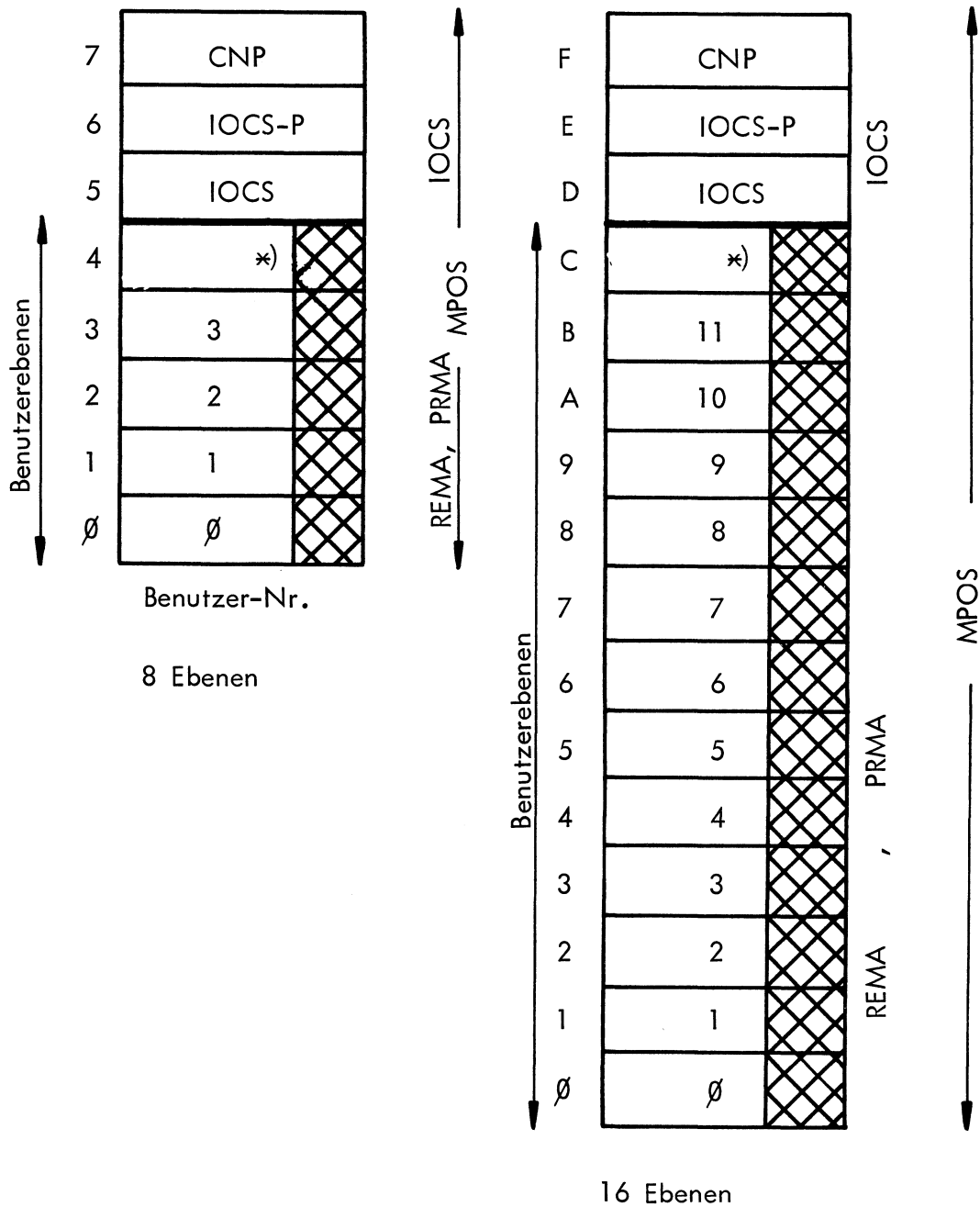


Bild 12: Ebenenverteilung im Multiprogramming

*) frei für Betriebssystem-Sonder-Prozesse

Das Bild verdeutlicht, daß die Betriebssystem-Funktionen in den Modulen REMA und PRMA auf den Benutzerprogrammebenen ablaufen. Eine dieser Benutzerebenen ist (im allgemeinen Ebene \emptyset) für die Abwicklung des Operator-Dialogs zuständig. Er hat höchste Priorität. Alle anderen Programme besitzen die Priorität, die der Ebenennummer entspricht, auf der sie laufen.

Beispiel für Programmablauf

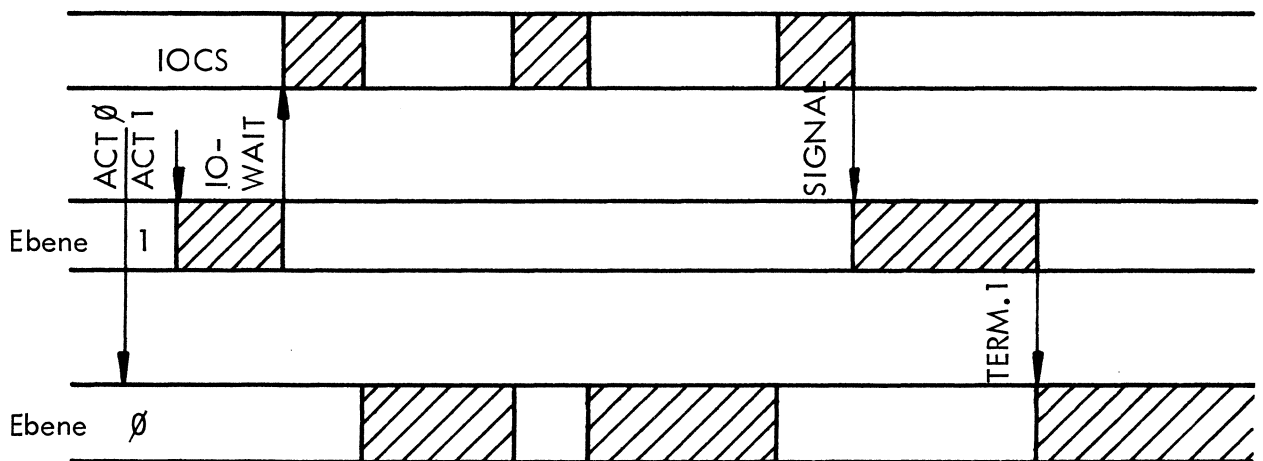
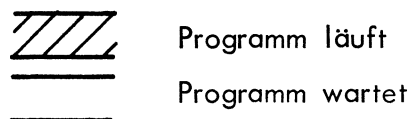


Bild 13: Arbeitsweise MPOS



Zu Beginn sei durch den Operator ein Programm auf Ebene \emptyset bzw. 1 aktiviert worden. Ebene 1 hat höhere Priorität und kann arbeiten. Sobald das Programm auf einen Ein-/Ausgabeauftrag mit Warten stößt, wird das IOCS aktiv und initialisiert den Vorgang. Danach kann das Programm auf Ebene \emptyset laufen. Immer dann, wenn das IOCS durch die Geräterückmeldung gestartet wird, wird das Programm \emptyset unterbrochen. Nach Fertigstellung des IO-Auftrages läuft das Programm auf Ebene 1 weiter (SIGNAL durch Betriebssystem). Der Befehl TERMINATE, 1 beendet das Programm auf Ebene 1. Ebene 0 läuft weiter.

2.11.3 RTOS

RTOS (Realtime Operating-System) ist ein Echtzeit-Betriebssystem, das Prozeßanwendungen des DIETZ-Systems 600/621 unterstützt und in Verbindung mit den Programmiersprachen MARS und BASEX und Prozeß-FORTRAN arbeitet.

Es bietet neben MPOS folgende Funktionen:

- Einplanung von Zeitaufträgen
- Einplanung von Interruptaufträgen, Erkennung der Interrupts
- Verwaltung der Ebenenbelegung
- Behandlung der Prozeßperipherie

RTOS ist deshalb gegenüber MPOS um die Module "Prozeß-Treiber", "PRSC" erweitert.

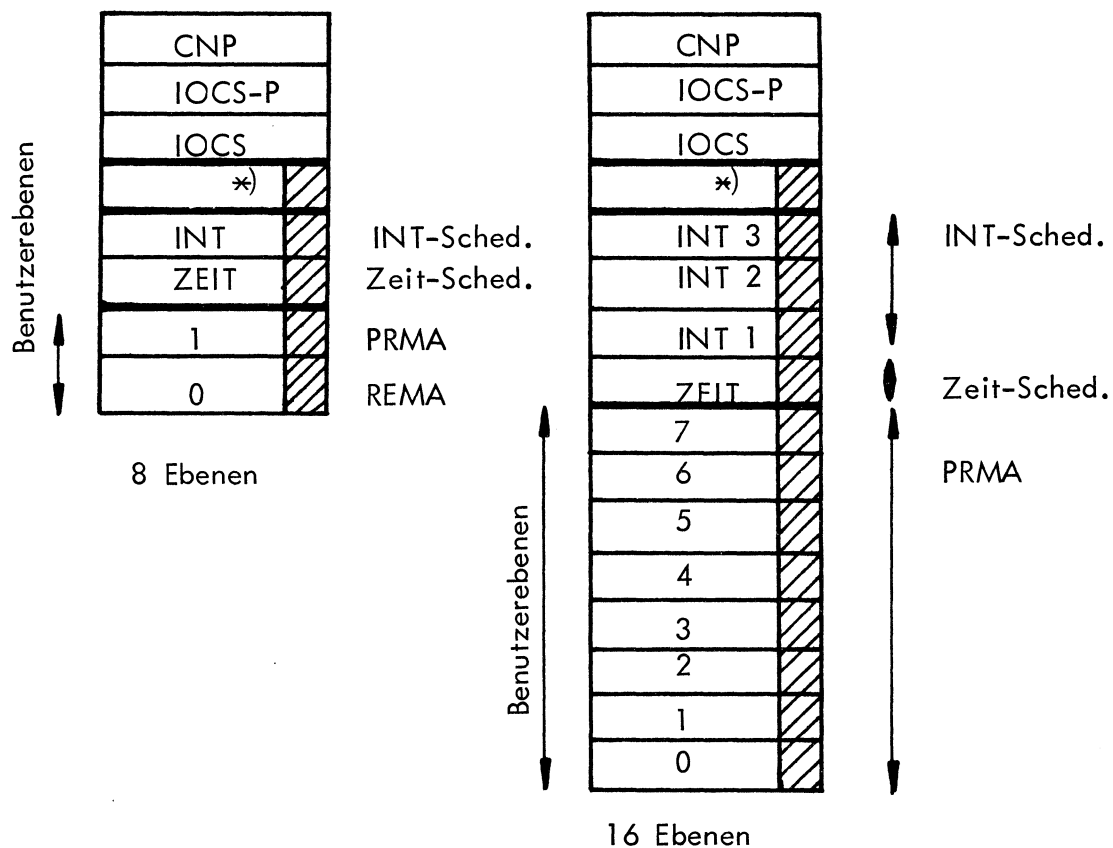


Bild 14: Ebenenverteilung bei RTOS

*) frei für Betriebssystem-Sonder-Prozesse

Gegenüber MPOS besteht der Unterschied, daß einige Ebenen für die Durchführung der Interrupt- und Zeitbehandlung vorbehalten sind. Alle Interrupt- und Zeitauftragsprogramme laufen auf den betreffenden Ebenen ab. Sie haben Vorrang vor den "normalen" Benutzer-Programmen und vor dem Operator-Dialog.

Da die einzelnen Systemebenen für alle derartigen Aufträge gemeinsam verwendet werden, ist es notwendig, Warteschlangen zur Abarbeitung aufzubauen. Auch dies leistet RTOS.

Beispiel:

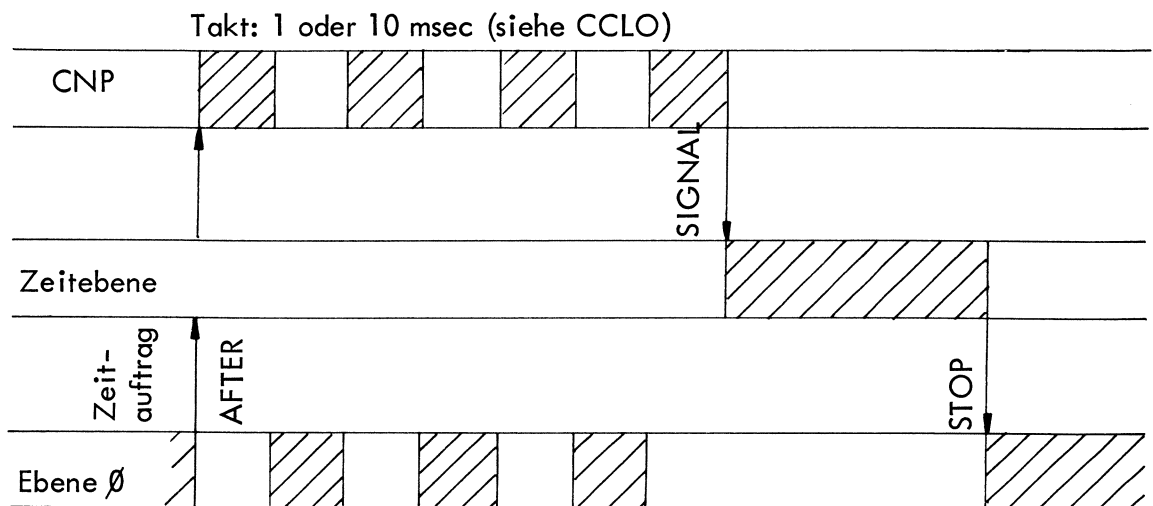


Bild 15: RTOS-Arbeitsweise

Ebene 0 erteilt einen Zeitauftrag. Die Verwaltung für Zeitaufträge setzt daraufhin einen Zeitzähler auf der CNP-Ebene. Die Realzeituhr dekrementiert diesen Zähler laufend. Sobald er 0 ist, wird die Zeitebene gestartet und der Auftrag durchgeführt. Bei STOP läuft Ebene 0 weiter.

RTOS kennt ein umfangreiches Spektrum von Prozeßperipherie-Treibern. Sie sind in der Dokumentation BASEX bzw. MARS ausführlich beschrieben.

2.11-7

2.11.4 TSOS

TSOS (Timesharing Operating-System) ist ein Betriebssystem, das Multi-User-Anwendungen (BASIC, EDITOR) zugrunde liegt. Bei diesem Betriebssystem muß die Verwaltung der Programme (Jobs) nach einer anderen Strategie als im Multiprogramming erfolgen, da hier im allgemeinen alle Jobs gleichberechtigt sind und ihre Aufträge nicht nacheinander, sondern quasi gleichzeitig (nach Timeslice-Methode) abgearbeitet werden müssen. Während bei MPOS die Hardware des DIETZ 600/621 für die "Programmumschaltung" allein ausreicht, geht dies bei TSOS nicht mehr. Wie dort erfolgt jedoch auch hier eine Umschaltung, wenn im Prozeß-Zustandsdiagramm eine Änderung auftritt, zusätzlich aber noch nach Ablauf einer Zeitscheibe. Dies erfordert eine zusätzliche Verwaltungsebene.

Ebenen-Belegung

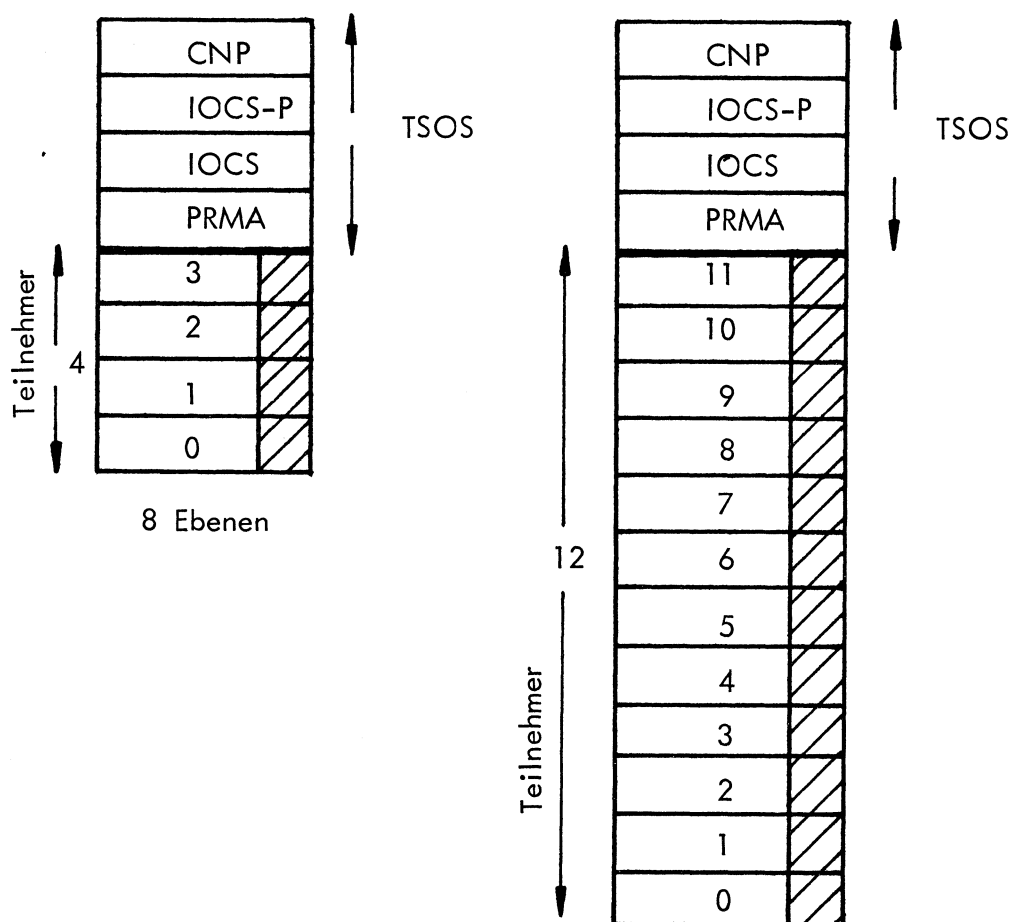


Bild 16.: Ebenenbelegung TSOS 16 Ebenen

Die Ebene PRMA (oder Job-Scheduler) ist zuständig für die Zuteilung von Rechenzeit an die Teilnehmer nach dem Zeitscheibenverfahren. Die unteren 4 bzw. 12 Ebenen werden durch deren Programm belegt. Da alle Teilnehmer gleichberechtigt sind, darf die Ebenenzuteilung hier nicht im Sinne von Prioritäten verstanden werden. Sie ermöglicht einen schnellen Job-Wechsel ohne Retten von Kontext.

TSOS kann jedoch auch als Version mit 64 Teilnehmern geliefert werden. In diesem Fall belegen mehrere Teilnehmer eine Ebene. Bei jedem Umschalten wird ein begrenzter Kontext (Register) gerettet.

Die Zeitscheibe beträgt im allgemeinen 100 msec, kann jedoch auch verändert werden (siehe Systemvariable CCLO).

2.11-5 TROS

Das Betriebssystem TROS (Timesharing-Realtime-Operating-System) ist eine Kombination von TSOS und RTOS und unterstützt z.B. Multi-User-BASEX. 1 2 oder 4 Teilnehmer können "gleichzeitig mit BASEX arbeiten und werden gleichberechtigt bedient. Ihre Realzeitaufträge und Startaufträge haben jedoch Vorrang. Im übrigen ist es auch möglich, einem Teilnehmer eine höhere Priorität als den anderen zu geben.

Hauptsächlich ist TROS bei Systemen mit edukativem Charakter und für Systeme, wo parallel zum laufenden Prozeß Programme erstellt werden sollen, interessant. Allerdings darf vom Rechner im Falle des gleichzeitigen Laufens von Prozessen nur ein beschränktes Realzeitverhalten erwartet werden.

Ebenen-Belegung

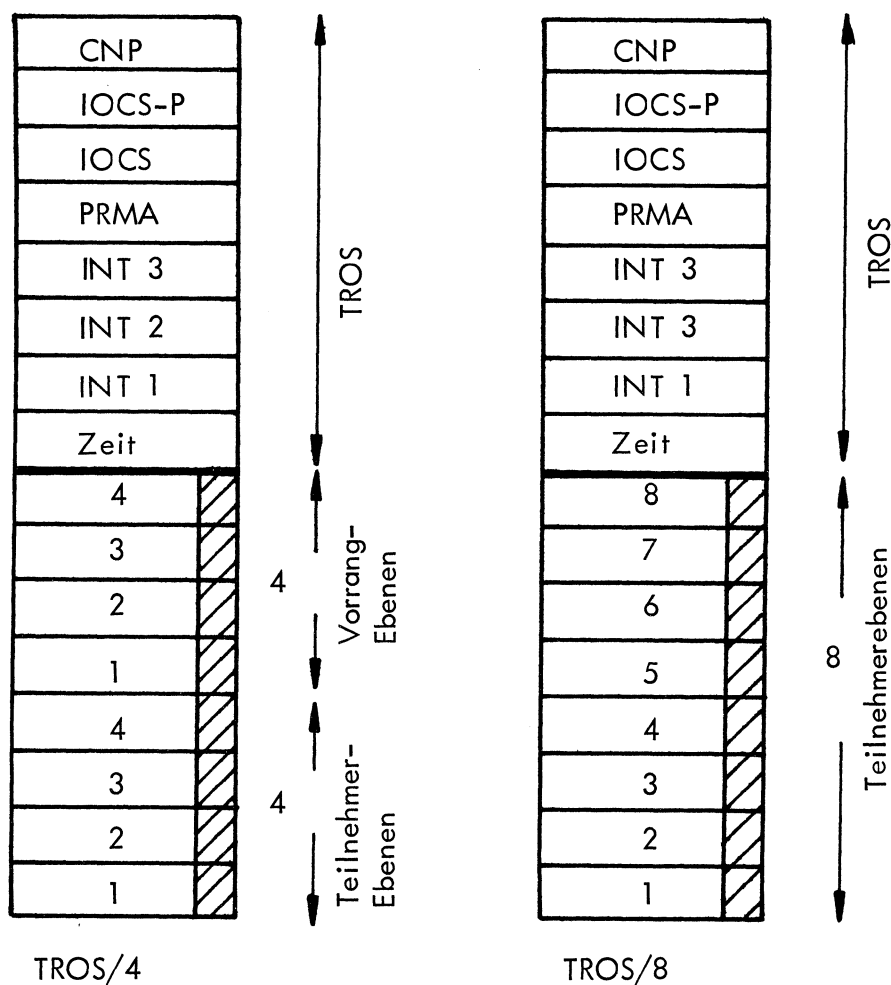


Bild 17: Ebenenbelegung TROS.

3. Dialog-Kommandos

3.1 Allgemeines

In den Beschreibungen von Kapitel 2 wurden die Schnittstellen des Betriebssystems zum System- bzw. Assemblerprogrammierer hin aufgezeigt. Die Syntax der Befehle richtet sich nach der Maschinensprache des Rechners. Jede Betriebssystemkonfiguration enthält darüber hinaus ein Dialog-Modul. Dieses Modul kennt

- Kommandos
- Dienstprogramme,

die von der Benutzerkonsole her in benutzerfreundlicher Syntax eingegeben werden und unmittelbar zur Durchführung kommen.

Das Betriebssystem meldet sich grundsätzlich mit

DIAL

*

zur Eingabe von Befehlen für das Dialog-Modul bereit. Voraussetzung dafür ist, daß der Teilnehmerkonsole eine Mindestpartition von 2 Sektoren zugewiesen ist. Nach Eingabe des Kommandos (Abschluß durch cr) wird dieses unmittelbar bearbeitet.

Bei Fehleingaben dienen die Zeichen "Rubout" bzw. "←" oder H^C oder Backspace" zum Löschen der gesamten Zeile bzw. des letzten Zeichens. Zahlen können, wo sinnvoll, sowohl dezimal als auch in Hexa eingegeben werden. Alle Zeichen nach " / " vor cr sind Kommentar.

Beachte:

Die Durchführung eines Kommandos erfolgt in einem Overlay-Bereich des Betriebssystems. Im Mehrbenutzerbetrieb muß ein Teilnehmer deshalb gelegentlich auf die Freigabe des Bereiches warten, so daß es bei seinem Kommando zu Wartezeiten kommen kann.

Einige Kommandos sind ausschließlich am Operator/Master-Platz (Gerät 1) zugelassen. Sie sind im folgenden mit !M bezeichnet.

3.2 Kommandos zur Prozeßverwaltung

ACT, t [, [p] [, aaaa]] !M

Aktiviere den Teilnehmer t ($t = 1, 2, \dots, 11$) mit Priorität p ($\emptyset, 1, 2, \dots, 63$) an Adresse aaaa. Die Angabe p kann bei $p = \emptyset$ entfallen. Entfällt aaaa, so meldet sich das System an Teilnehmerkonsole t mit DIAL und ist zur Eingabe von Dialogbefehlen bereit. Entfällt aaaa nicht, so wird ein Programm in der Teilnehmerpartition vorausgesetzt und dieses bei aaaa gestartet.

GOTO [, [p] [, aaaa]]

Mit diesem Kommando kann der Master oder ein Teilnehmer zur Durchführung eines Programmes nach Adresse aaaa gehen. Die Prioritätsangabe p kann bei $p = \emptyset$ wieder entfallen, ebenfalls aaaa, wenn an den Anfang der Partition verzweigt werden soll.

TER [, t] !M

Beende das Programme des Teilnehmers t und sperre den Zugang zu diesem Arbeitsplatz.

SUS [, t] !M

Suspendiere das Programm des Teilnehmers t. Falls t entfällt, werden alle laufenden Programme suspendiert.

CON [, t [, p]] !M

Das vorher suspendierte Programm des Teilnehmers t wird mit Priorität p fortgesetzt. Falls p entfällt, gilt die bisherige Priorität, falls t entfällt, werden alle Teilnehmerprogramme, die suspendiert sind, fortgesetzt.

STAT [, t] !M

Ausgabe der Zustandstabelle der Prozeßverwaltung. "t" ist dabei die Nummer des Teilnehmers, dessen Zustandsvektor ausgegeben werden soll. Entfällt t, erfolgt Ausgabe der Gesamttabelle.

Bei der Ausgabe wird angezeigt:

Zustand, belegte Rechnerebene, Betriebsmittel oder Signal auf das gewartet wird oder das zuletzt belegt wurde, belegte Rechnerebene, verbrauchte Zeit in Einheiten von 100 msec, Priorität, aktuelle Programmadresse.

3.3 Kommandos zur Betriebsmittelverwaltung

ARM, r !M

Armiere das Gerät r (1,31)

DIS, r !M

Sperre das Gerät r

REQ, r [, t]

Fordere das Betriebsmittel r (im allgemeinen : Gerät) für Teilnehmer t an. Falls t entfällt, für eigenen Bedarf.

REL, n [, t]

Gebe Betriebsmittel n frei.

CHG, r1, r2 !M

Vertausche die physikalische Adresse von Gerät r1 und r2. Damit kann sowohl der Master als auch ein Teilnehmerplatz verlagert werden. Zu beachten ist, daß das Kommando nur für Geräte auf gleicher Ebene und vom gleichen Typ erlaubt ist.

RES, r !M

Abfrage des Zustandes des Betriebsmittels r. Das System meldet, ob das Betriebsmittel frei ist bzw. wer es gerade besetzt und wieviele Teilnehmer in der Betriebsmittelwarteschlange stehen.

ASG, t, r [,aaaa] !M

Ordne dem Teilnehmer t das Dialoggerät r zu und setze die BREAK-Adresse auf den Wert aaaa. Falls letztere Angabe entfällt, meldet sich das System bei BREAK mit DIAL.

SIG, e

Signalisiere Ereignis e (= 1,64) bzw. lösche Ereignis e* bei Eingabe von e = e* + 192.

3.3-2

MES, d, "text"

Ausgabe der Nachricht "text" auf Gerät d ($d = 1, 2, 3, 21, 22, \dots, 31$).
"text" kann bis zu 48 Zeichen umfassen.

TIME, h, m, s

Mit diesem Kommando wird die Uhrzeit gesetzt (! nur Master) und abgefragt (an beliebigen Plätzen). Die Angaben h, m, s bedeuten Stunde, Minute, Sekunde.

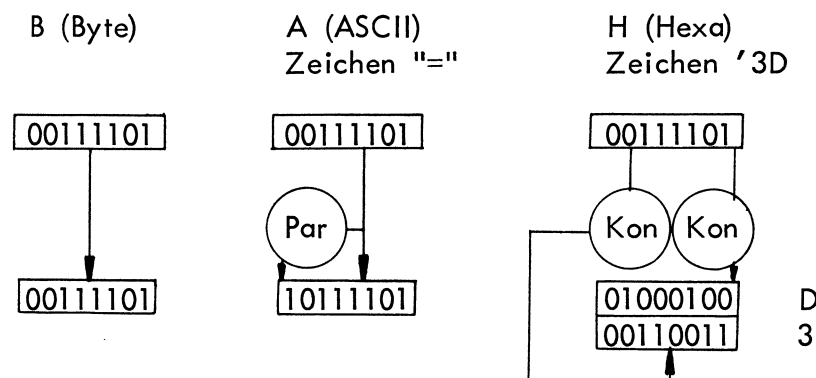
DATE, d, m, y

Es wird das Tagesdatum gesetzt (nur Master) oder abgefragt. Uhrzeit und Datum werden fortlaufend geführt.

3.4 Kommandos zur Ein-/Ausgabe und Dateiverwaltung

LIST, [d], aaaa, bbbb [, f]

Ausgabe des Speicherbereiches aaaa, bbbb auf Gerät d (DUMP).
Diese Ausgabe ist in 3 Formaten möglich: B, A oder H. Falls eine
Formatangabe entfällt, ist H gemeint.



Im Byte-Format wird ein Speicherbyte direkt ausgegeben, im ASCII-Format wird bit 7 durch das Paritätsbit ersetzt.

Im Hexa-Format werden 2 Zeichen ausgegeben. Sie entsprechen den beiden Hexaziffern der Speicherzelle. Außerdem erfolgt hier die Ausgabe von crlf nach je 16 Byte.

READ, [d], aaaa, bbbb [,f]

Eingabe in den angegebenen Speicherbereich von Gerät d im Byte-, ASCII- oder Hexa-Format.

Im Byte-Format findet keine Paritätsprüfung statt. Im Hexa-Format werden Blanks und crlf überlesen. Mit "XOFF" wird die Eingabe bei H und A beendet.

Alle folgenden Kommandos sind nur vorhanden, wenn das Betriebssystem plattenunterstützt arbeitet. Dabei bedeutet stets:

u = Platteneinheit (0, 1, 2). Sie kann bei "0" entfallen.

file = Dateiname (1...6 ASCII-Zeichen)

LOAD [, u] , file [, aaaa] [,s [,e]]

Die Datei mit dem Namen "file" auf Einheit u wird in den Speicherbereich aaaa bis (aaaa + Dateilänge) eingegeben. Entfällt die Angabe aaaa, wird ab Partitionsanfang eingespeichert. Die Zusatzangabe s, e erlaubt es, Teile einer Datei beginnend mit Sektor "s" und der Länge "e" zu laden.

SAVE [, u] , file [, aaaa] [,s [,e]]

Die Datei "file" wird mit dem Speicherinhalt ab aaaa (oder Partitionsanfang) bis zur eröffneten Dateilänge beschrieben.

RUN [, u] , file [, aaaa] [,s [,e]]

Dieses Kommando ist die Abkürzung von LOAD und GOTO. Gestartet wird das Programm auf derjenigen Rechnerebene, wo das Dialogprogramm läuft.

CREA [, u] , file , l

Eröffnen einer Datei unter dem Namen "file" mit der Länge l
(1,....32767)

KILL [, u] , file

Löschen einer Datei.

ALTER [, u], file 1, file 2

Ändern des Namens einer Datei "file 1" in "file 2".

PROT [, u] , file [, p]

Abändern des Protections-Codes p der Datei "file".

FILE [, u] , file , z

Eingabe Dateikennzeichen z (2.6.1)

LENGTH [, u] , file, l

Verkürzen der Datei "file" auf Länge l.

CODE [, ccc]

Eingabe des Benutzercodes zum Schutz privater Dateien. Falls ccc entfällt, wird '000' angenommen. Alle damit bezeichneten Dateien sind öffentlich.

3.5 Hilfskommandos

LOP, aaaa

Dieses Kommando erlaubt den Einbau von Adreßstops an die Stelle aaaa. Das System gibt den Inhalt von dieser und der folgenden Adresse aus und baut den Befehl JPL, -1 (F8FF) ein.

LOP, aaaa, R, bbbb

Mit diesem Kommando wird an die Stelle aaaa der vorher ausgegebene Inhalt bbbb wieder eingebaut.

Im Adreßstop erlaubt das System, die Ausgabe von Registerinhalten, die Veränderung von Programmteilen oder die Abfrage der Statusinformation und ermöglicht so den Programm-DEBUG.

Für den Programm-DEBUG mit komfortablen Möglichkeiten steht das Hilfsprogramm DEBUG zur Verfügung.

CV, z1 [, z2, op]

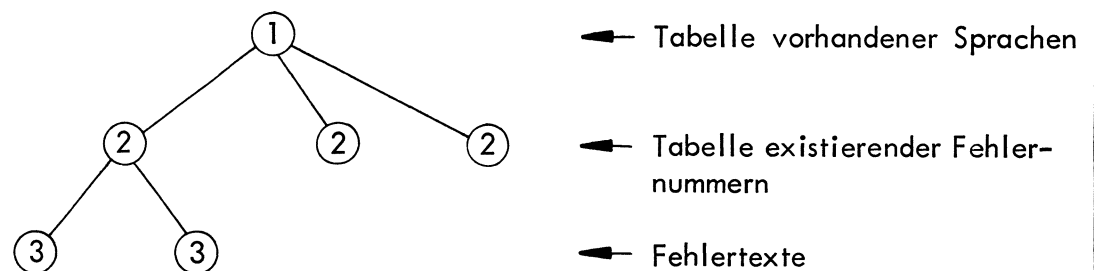
Dieses Kommando erlaubt die Durchführung der 4 arithmetischen Operationen (z1 op z2), wobei op = + - * : ist, sowie die Konversion Hexaformat nach Dezimalformat und umgekehrt, falls der geklammerte Ausdruck entfällt. Zahlen können sowohl im Hexa- als auch im Dezimalformat eingegeben werden.

z.B.: CV, 10, '0A, *
CV, '4000

ERR, n

Ausgabe eines Kurztextes zur Fehlernummer n. Voraussetzung ist, daß die Datei ERRGES eingerichtet ist.

Die Gesamtfehlerdatei enthält die Texte der Fehlermeldungen des Betriebssystems XOS und aller unter XOS lauffähigen Sprachen. Durch bedingtes Assemblieren läßt sich aus einer Gesamtfehlerdatei AERGES eine spezielle Fehlerdatei bilden, die nur die tatsächlich benötigten Fehlertexte enthält (z.B. kommerzielles System). Die Fehlerdatei hat eine dreistufige Baumstruktur:



Es gibt folgende Zuordnung zu den Systemen:

ØXXX	XOS
1XXX	Bibliothekssdatei XOSBIB
2XXX	MARS
3XXX	Editor
4XXX	BASEX
5XXX	C-BASIC
6XXX	FORTTRAN
7XXX	PEARL
8XXX	Stage II
9XXX	PASCAL

Ist auf der Systemplatte die Datei ERRGES nicht vorhanden, erfolgt die Meldung:

Kein Zugriff auf Datei ERRGES.

Wird eine Fehlernummer aus einer nicht eingebundenen Sprache angewählt, erfolgt die Meldung:

Fehler nicht in Error-Datei.

3.6 System-Generierung

3.6.1 Erzeugen neues Betriebssystem

Das Betriebssystem XOS besteht immer aus den beiden Dateien

SERVIC 256 Sek / Datei der Betriebssystemüberlagerungsteile

TSOS-X 64 Sek / Grundbetriebssystem

Es ist zu beachten, daß SERVIC und TSOS-X immer die erste bzw. zweite Datei auf der Platte sein müssen und nicht verändert werden dürfen. (Laden auf Platte: Anhang 2)

Für die praktische Arbeit an einer speziellen Rechnerkonfiguration konfiguriert man sich aus TSOS-X eine spezielle Modifikation durch

- Einbau von Treibern
- Eintrag von Adressen in Betriebssystemtabellen
- Einbau von Moduln.

Die so entstehende Konfiguration kann nun als neues Betriebssystem unter die ersten 32 Dateien auf der Platte abgespeichert werden. Dies leistet folgendes Kommando:

NEW, u, f Ablegen neuer Konfigurationen
 u Platteneinheit
 f Zieldateiname

Das neue Betriebssystem kann dann wie TSOS-X unter dem Namen f gerufen werden.

Beachte:

1. Das Betriebssystem enthält keine residente Grundstellung. Diese wird bei NEW aus TSOS-X entnommen. TSOS-X muß (!) deshalb vorhanden sein.
2. Ein Betriebssystem muß unter den ersten 32 Dateien liegen.
3. Die Datei muß vorher in ausreichender Größe creiert sein.

Beispiel:

Nach Betätigen von "Reset" an der Rechnerkonsole, meldet sich ein Platten-Bootstrap mit "?".

- ? TSOS-X / Eingabe Namen Ur-System
DIAL
- * SERV, MOUNT, 3, 1061, C4 / Montage Drucker Interface >602 . 050
DIAL
- * SERV, MODULE, MCARD / Einbau Kartentreiber
MODULE MCARD... / Meldung Einbaubereich
DIAL
- * SERV, MOUNT, 8, 1111, E0 / Montage Kartenleser
DIAL
- * LOP, 400C, R, 0001 / 48 K
DIAL
- * CREATE, TSOS-Y, 66 / Anlegen Datei
DIAL
- * NEW, 0, TSOS-Y / Ablegen auf Platte

In diesem Beispiel wurde vorausgesetzt, daß das Urbetriebssystem bereits auf Platte existiert. Anhang 2 beschreibt, wie es über Lochstreifen eingelesen wird.

3.6.2 Verteilung von Partitionen (* 5)

Im Grundzustand eines Betriebssystems besitzt der Operator/Master 2 Sektoren zur Abwicklung des Systemdialoges. Soll am Masterplatz mit Programmsystemen (z.B. EDITOR) gearbeitet werden oder sollen andere Teilnehmer zugelassen werden, so erfolgt jetzt eine Partitionsverteilung.

PART [\ast oder \S] , P1, P2, P3,P12

\ast COMMON-Grenze ist 80000 (sonst C0000)

\S Ausschalten Memory-Manager

Pi Partitionen in Einheiten von 128 Byte

Die Partitionsvergabe erfolgt in Blöcken zu 128 Byte vom vorgegebenen Speicherende (siehe CFSE) aus nach vorne für Teilnehmer \emptyset (= Operator) 1, 2, usw. Soll ein Teilnehmer keine Partition erhalten, wird $P_i = \emptyset$ eingegeben. Nach Durchführung des Kommandos melden sich die Teilnehmerkonsolen mit DIAL zur Eingabe von Systemkommandos bereit.

Für das Verständnis der Angaben " \ast bis \S " ist die sogenannte "COMMON-Grenze" von Bedeutung. Diese ist eine Adresse des Hauptspeichers (80000 bzw. C0000), oberhalb derer die sog. Speicher-Segmentierungseinheit ("Memory-Manager") zu der in einem Programm angegebenen Adresse den Inhalt eines Basisregisters hinzuaddiert. Dieses Basisregister wird beim Kommando PART gesetzt und bleiben solange unverändert, bis ein neues PART-Kommando eingegeben wird.

Unterhalb der COMMON-Grenze liegt das Betriebssystem mit den zusätzlich eingebauten Modulen, sowie eventuell ein Interpretersystem.

Die COMMON-Grenze existiert nur bei Anlagen mit einer Ausstattung von mehr als 48 K-Hauptspeicher. Sie wird mit " \ast " auf den Wert 80000 festgelegt, mit " \S " wird der Memory-Manager ausgeschaltet.

Beachte:

1. Bei Systemen mit COMMON-Grenze = '80000' ist die maximale Partitionsgröße 32 K, sonst 16 K. Die letzte zugewiesene Partition kann jedoch bis zum Freispeicheranfang hinaus verlängert sein und deshalb max. 40 bzw. 24 K umfassen. Diese Einschränkungen sind im Rechner DIETZ X2 aufgehoben.

2. Bei Neuvergabe der Partitionen werden alle Teilnehmerprogramme terminiert. Eine automatische dynamische Speichervergabe ist für die Betriebssystemversion Rel 6 in Arbeit.

Bei Eingabe von PART ohne Angabe wird die aktuelle Partitionsverteilung ausgegeben.

Für die verschiedenen Sprachsysteme werden folgende Mindestpartitionsgrößen (in Einheiten von Blöcken zu 128 Byte) benötigt:

40 KB (48 KB - 8 KB Betriebssystem)	: 320 Sektoren
<u>56 KB (64 KB - 8 KB Betriebssystem)</u>	<u>: 448 Sektoren</u>
EDIREL (Editor rel)	: 106 Sektoren
MARS (Assembler)	: 128 Sektoren
XOSLAD (Bindelader)	: 100 Sektoren
XOSFOR (FORTRAN-Compiler)	: 138 Sektoren
XOSBAS (BASEX-Compiler)	: 100 Sektoren
BASEX (BASEX-Interpreter)	: 10 Sektoren
BASIC (BASIC-Interpreter)	: 4 Sektoren
C-BASIC (C-BASIC-Interpreter)	: 40 Sektoren

3.6-5

3.6.3 Überwachung von Partitionen

MEM

Abfrage Speicherraum

Jeder Benutzer kann mit MEM den ihm zur Verfügung stehenden Adreßraum abfragen.

USER, t

!M

Umschalten zu Partition t

Alle Adressen, die ein Teilnehmer in einem Kommando spezifiziert, beziehen sich auf seinen eigenen Adreßraum. An Adressen anderer Partitionen kann nur der Master herankommen, indem er mit USER, t auf eine bestimmte Partition umschaltet. Alle von ihm eingegebenen Adressen oberhalb der COMMON-Grenze beziehen sich jetzt auf die angewählte Partition.

3.6.4 Wechseln Betriebssystem

Eine Systemplatte kann mehrere Betriebssystemkonfigurationen enthalten. Jede Konfiguration kann in gleicher Weise gerufen werden.

Nach Betätigung der "Reset-Taste" am Rechner meldet sich der Platten-Bootstrap mit "?". Nach Eingabe des Namens des gewünschten Betriebssystems gefolgt von "cr" oder von der Platten-Nummer, meldet sich das System mit einem Release-Text. Falls das System nicht existiert, wird erneut "?" ausgegeben.

Nachdem ein Betriebssystem einmal im Speicher ist, kann es ohne Betätigung von Reset mit dem Dienstprogramm:

SERV, CHANGE, u u = Ø, 1

gewechselt werden. u spezifiziert die Platteneinheit, von der der Bootstrap genommen werden soll. Nach Betätigen von cr, meldet sich dieser wieder mit "?".

Beispiel:

"Reset"

? TSOS-X cr

- Systemmeldung -

SERV, CHANGE, Ø

? TSOSXX1

- Systemmeldung -

4.1-1

4. Dienstprogramme

4.1 Allgemeines

Dienstprogramme sind System-Prozeduren, die in der Datei `SERVIC` gesammelt sind und mit dem Kommando

SERV, Name

in den Systemoverlay geladen und durchgeführt werden. Nach dem Laden meldet sich das Dienstprogramm mit SYS und wartet auf die Eingabe einer Parameterspezifikation. Statt dessen können die Parameter hinter dem Namen direkt eingegeben werden.

Beispiel:

- a) SERV, LIST / Ausgabe Inhaltsverzeichnis
SYS
* A, 1 / Alle von Unit 1
- b) SERV, LIST, A, 1 / identisch mit a)

Die Eingabe kann wie üblich korrigiert werden. Sie läßt sich im Falle a) auch durch E^c abbrechen.

Beachte:

Ein Benutzer muß u.U. warten, bis ein anderer den Dienstprogramm-Overlay-Bereich freigibt.

4.2 Dienstprogramme Platteninhalt

4.2.1 Platten-Volumen

SERV, VOLUME
u [, f]

Definition des Namens "f" bzw. Abfrage des Namens (Angabe ohne f) der Platteneinheit u
Die Systemplatte erhält beim Laden den Namen SYSTEM

4.2.2 Ausgabe Dateiverzeichnis

SERV, LIST

Listen Platteninhaltsverzeichnis

s, u, d = Ausgabe des Freibereiches der Platteneinheit u auf Gerät d
(0 = Konsole, 2 = Stanzer, 3 = Drucker).
LIST gibt 3 Zahlen aus
X Y Z

Dabei ist X die Summe der Lücken innerhalb der Dateien,
Y der Freibereich am Ende und Z = X + Y.

A, u, d = Ausgabe des Gesamtverzeichnisses.

F, u, d, f = Ausgabe des Verzeichnisses ab Name f.

M, u, d, m = Ausgabe aller Namen, welche von Maske m durchgelassen werden (spezielles Maskenzeichen für beliebig: *)
zum Beispiel: m = DATEIX nur Datei-DATEIX ausgeben
m = B ***** alle BASEX-Dateien ausgeben
m = A B**5 alle Assembler-Dateien mit B am Anfang und 5 am Ende ausgeben

I, u, d, , Z1, Z2 = Alle Namen von Zeile Z1 bis Z2 ausgeben.

Ausgegeben werden jeweils alle Dateien mit Name, Sektoradresse, Länge, File-code, Protectionscode, Autorcode, zu denen der rufende Prozeß zugriffsberechtigt ist, also nicht in die privaten anderer Teilnehmer. Ausnahme ist der Operator, der Zugang zu allen hat.

4.2-2

4.2.3 Initialisieren Namensverzeichnis

SERV, INIT
u

Lösche Namensverzeichnis der Platte u. Bei einem Versuch, die Systemresidenz zu initialisieren, meldet das System ERR 142

4.2.4 Initialisieren Dateiinhalt

SERV, SCRAT
u, f, [,hh]

Setzen des Dateiinhaltes der Datei f auf Einheit u auf den Wert 'hh (oder 'ØØ, falls hh nicht angegeben ist).

4.2.5 Inhaltsverzeichnis SERVIC

SERV, SERLIS
u

Ausgabe des Inhaltsverzeichnisses der Datei SERVIC auf Einheit u.
Ausgegeben wird der Name und die Anfangsadresse in SERVIC.

4.2.6 Formatieren einer Platte

SERV, FORMAT

u

Beheben von "Header- und Paritätsfehlern"
auf Platte u bei Controller 640 (9.6 MB
bzw. 2.4 MB 24 Slot)

Vor dem Formatieren werden alle Prozesse terminiert und ein Pufferbereich von 6 K ab CFSA auf WF00 der Systemresidenz gesichert. Bei der Formatierung bleiben die Daten auf der zu formatierenden Platte erhalten. Unter bestimmten Fehlerbedingungen kann jedoch nicht die komplette Restaurierung gewährleistet werden.

Enthält die Platte noch keine Header-Information, kann sie nicht formatiert werden. Es kommt die Fehlermeldung: xx Platte nicht formatierbar xx .

Für das erstmalige Beschreiben von Platten existiert ein Dienstprogramm, welches nicht in SERVIC enthalten ist.

4.2.7 Prüfen Platte

SERV, DSKCHE

C, u

Prüfen von Platten auf Platten(Lese)-fehler.

Die Platte auf Einheit u wird in Blöcken zu je 2 Sektoren ab Sektor 0 gelesen. Eventuelle Fehler werden mit

ERR u SEKTOR S (S = Sektor)

gemeldet. Anschließend liest das System weiter.

4.2-4

4.2.8 Reorganisation Platte

SERV, PACK

P, u

Zusammenschieben Inhaltsverzeichnis und
Plattenbereich

L, u

Reorganisation mit Löschen

In dieser Betriebsart wird der Reihe nach jeder
Dateiname ausgegeben und auf die Eingabe
"Y" (für Löschen) bzw. "N" (für Beibehalten)
gewartet.

Anschließend erfolgt die Reorganisation wie
bei P. Mit E^c kann die Funktion L abgebrochen
werden.

Beachte: Das Dienstprogramm benutzt den Arbeitsspeicher von CFSA bis
CFSE und terminiert deshalb vor Ausführung alle Prozesse.

4.2.9 Einbau neues Dienstprogramm

SERV, NEWUT

u, f

Einbau Dienstprogramm in Datei SERVIC unter
dem Namen f.

Das Dienstprogramm kann maximal 4 Sektoren
groß sein. Es kann absolut für Adresse ' 5E00
geschrieben sein und steht nach Einbau unter
dem Namen f zur Verfügung.

4.3 Kopier-Programme

4.3.1 Peripherer Datenaustausch

SERV, PERIPH

C, u1, f1, u2, f2

Copy : Datei f1 von Einheit u1 nach Einheit u2 auf Datei f2.

Beim Kopieren wird die kürzere Datei als Kopie - länge herangezogen.

P, u, f, d, t

Print: Gebe Datei f auf Einheit u auf Gerät d im Format t

(t = A : ASCII, t = B : Byte) aus.

R, u, f, d, t

Read: Lese Datei f vom Gerät d im Format t ein.

T, d1, t1, d2, t2

Transfer : Übertrage Daten vom Gerät d1 im Format t1 auf Gerät d2 im Format t2.

Bei diesem Programm ist zu beachten, daß die Übertragung 1 : 1 erfolgt und keine geräte-spezifische Aufbereitung geschieht.

4.3-2

4.3.2 Mehrfach Kopieren von Dateien

SERV, FCOPY

u1, f1, u2, f2

Kopiere Datei f1 von Einheit u1 auf f2, Einheit u2. Bei diesem Vorgang wird im Gegensatz zur Kopierfunktion in PERIPH eine Datei f2 creiert, falls sie nicht vorhanden ist und auch der Protectionscode und Filecode mit übernommen.

u1, f1, u2

Kopiere alle Dateien auf Einheit u1, die der Spezifikation f1 entsprechen, auf Einheit 2. Die Angabe f1 kann Maskenzeichen " * " enthalten, zum Beispiel:

Ø, BA **** : alle Dateien mit BA am Anfang

4.3.3 Lochstreifen -Kopieren

SERV, KENN

t

Stanzen einer Kennung auf Lochstreifenvorspann. Der Text " t " darf bis zu 80 Zeichen haben.

SERV, PUNCH

u, f

Stanzen der Datei f auf Einheit u mit Datensicherungsbytes.

SERV, READ

u, f [s]

Lesen des Lochstreifens (Einlegen im Rubout-Bereich) auf Datei f. Falls nicht vorhanden, wird die Datei in der Länge s eröffnet.

4.3.4 Platten-Kopieren

SERV, DCOPY
su, du [,1]

Kopieren Gesamtplatte von Platteneinheit su auf Einheit du.
Falls als dritter Parameter eine 1 angegeben ist, bricht das Dienstprogramm bei Kopierfehlern nicht ab.

Beachte: Das Dienstprogramm belegt 8 K-Speicher ab CFSA zum Zwecke des Kopierens. Vor Durchführung werden deshalb eventuell laufende Prozesse terminiert. Der Inhalt des benötigten Speicherplatzes wird vor dem Kopiervorgang auf WFF gerettet und nach dem Kopieren wieder restauriert.

4.3.5 Band-Kopieren

SERV, TCOPY
d, u, t [,n]

Kopiere Platteninhalt auf Band (d = W) bzw. von Band auf Platte (d = R). Die angesprochene Platteneinheit ist u, die Bandeinheit t.
Auf Band können mehrere Platteninhalte kopiert werden, wobei n die Kopienummer angibt. Kopiert wird in Blöcken zu 4 K.

Beachte: Wie das Plattenkopier-Programm, so benötigt auch das Bandkopier-Programm 8 K-Speicher. Es gilt das dort Gesagte.

Beim Kopieren in DCOPY und TCOPY können folgende Fehler auftreten:

Fehler 142: Verbotener Versuch auf die Systemresidenz zu kopieren. Abbruch.

Fehler 141: Platten haben unterschiedliche Sektoranzahlen. Abbruch.

ERR n UNIT u(x) : n = Fehlernummer wie XOS

u = Platten Unit

x = R Lesefehler

W Schreibfehler

C Vergleichsfehler

4.4 Dienstprogramm DUMP

SERV, DUMP

Ausgabe Inhalt Kernspeicher, Datei oder Plattensektor.

MA, d, aaaa, bbbb [, , , l]

Ausgabe Kernspeicherbereich .

Der Bereich aaaa bis bbbb des Kernspeichers wird auf Gerät d ausgegeben. Die Zeilenzahl pro Seite ist l. Die Ausgabe erfolgt in Hexa und ASCII. Letzteres kann unterdrückt werden, falls statt MA nur M angegeben ist.

FA, d, sa, sb, u, f [, l]

Ausgabe Datei

Der Sektor sa bis sb der Datei f auf Einheit u wird auf Gerät d ausgegeben. Die Ausgabe geschieht wie bei MA. Statt FA kann auch nur F stehen, wenn die ASCII-Ausgabe unterdrückt werden soll.

DA, d, sa, sb, u [, l]

Ausgabe Sektor

Der Platteninhalt von Sektor sa bis sb auf Einheit u wird auf Gerät d ausgegeben. DA ist wieder durch D ersetzbar.

Sektornummern sind in Hexa anzugeben.

4.5-1

4.5 Montieren von Geräten4.5.1 Eintrag IOCS-Tabelle

SERV, MOUNT

Montage Gerät

d, b, t

Die Statuswortadresse b und der Gerätetyp t wird in der IOCS-Tabelle unter der Gerätenummer d vermerkt. Es erfolgt eine Prüfung der Betriebsbereitschaft und Ausgabe der Gerätenummer auf dem angesprochenen Gerät.

d,, t

Der Gerätetyp von d wird geändert.

d

Der Eintrag wird entfernt.

Bezüglich der Gerätenummer und des Gerätetyps wird auf das Kapitel IOCS verwiesen.

Beispiel: SERV, MOUNT, 3, 1Ø61, C2 / Montage Drucker (altes Interface)

4.5.2 Eintrag READY-Scanner-Tabelle

SERV, RSCMNT

Montage READY-Scanner

r, l, d1, d2,

Sollen Geräterückmeldungen über den READY-Scanner abgefragt werden, was bei mehr als 10 Geräten immer erforderlich ist, so muß eine Eintragung in die betriebsinterne READY-Scanner-Tabelle erfolgen. Mit r (= 0,1) wird die Nummer des READY-Scanners angegeben, mit l die Rückmeldeebene (l = CNP - 1, Ø = CNP-2). di ist die Folge der über den READY-Scanner laufenden Geräte.

Beispiel: SERV, RSCMNT, Ø, Ø, 21, 22, 23, 24, 26

Beachte: Nach Montage des READY-Scanners müssen die Geräte wie oben über SERV, MOUNT eingetragen werden.

4.5.3 Eintrag Platten-Unit-Tabelle

SERV, DSKMNT

Montieren Platteneinheit

u, w [, a]

Die Hardware-Einheit w1 mit der Treiber-Nummer w2 (w = w1 & w2, je ein Halbbyte) bekommt die logische Einheit u zugewiesen. Der Treiber liegt bei a. Beim systemresidenten Treiber Ø muß keine Adresse angegeben werden.

Die Plattenstatus-Adresse wird wie oben in die IOCS-Tabelle eingetragen. Jedoch ist beim Anschluß mehrerer Plattentypen noch eine zusätzliche Eintragung erforderlich.

Alle Plattentypen (2.4 MByte, 9.6 MByte, Floppy) werden von der Benutzer-Schnittstelle her alle gleich angesprochen.

Ihre Ansteuerung erfordert jedoch unterschiedliche Treiber. Nach dem Einbau eines solchen an der Adresse a muß eine Eintragung der Treiber-Adresse sowie die Zuordnung einer logischen Unit-Nummer in eine spezielle Tabelle erfolgen. Dies leistet das Programm DSKMNT.

4.6 Stapelverarbeitung

Kommandos, aber auch Daten, mit denen ein Programm versorgt wird, werden im Dialogbetrieb gewöhnlich von der Benutzerkonsole (Device 0) eingegeben. Diese Betriebsart hat bei "planbaren" Eingabe-Folgen den Nachteil, daß eine Bedienungsperson dauernd an der Konsole sein muß, um keine Rechenzeit zu verschenken. In diesem Anwendungsfall ist die Stapel- (Batch) Verarbeitung sinnvoller.

Ein Prozeß ist als "Batch-Prozeß" gekennzeichnet, wenn in der ihm zugeordneten Konsolnummer (siehe BREAK-Tabelle) bit 6 gesetzt ist. In diesem Fall beschafft das IOCS anstelle der Konsoleingaben die Kommandos und Daten von einer sogenannten Batch-Datei.

Der Benutzer hat die Möglichkeit, über das Dienstprogramm

SERV, BATCH

u

eine Batch-Datei zu füllen und sie nach Senden von S^C (XOFF)-CR zur Durchführung zu bringen. Voraussetzung ist, daß eine öffentliche Datei BATi (i = User-Nummer) existiert.

Stattdessen kann auch eine über den Editor oder sonstwie erstellte Datei f mit dem Befehl

SERV, BATCH

u, file

zur Durchführung gebracht werden.

Häufig wird die Möglichkeit zur automatischen Generierung eines Betriebssystems verwendet. Sie ist aber auch für alle sonstigen Eingabefolgen erlaubt.

Beispiele:

Der Operator hat das Sprachsystem BASIC geladen, wobei ein Teilnehmer in "Batch" arbeiten will. Er gibt nach Aktivierung durch den Operator ein:

SERV, BATCH

Ø

LOAD, 1, PRG 1 CR / Lade Programm 1 von Platte 1

RUN CR / Starte Programm 1

LOAD, Ø, PRG 2 CR / Lade Programm 2 von Platte Ø

LIST (3) CR / Gebe Programm auf Schnelldrucker

REQ, 2 CR / Reserviere Lochstreifengerät

RUN CR / Starte Programm 2

REL, 2 CR / Gebe Lochstreifengerät frei

XOFF CR / Abbruch der Eingabe

Nach Eingabe von XOFF-CR beschafft das Betriebssystem bei allen BASIC-Eingaben von "Gerät Ø" die Informationen von der Batch-Datei.

Fehlermeldung

'88 Batch-Datei nicht vorhanden / nicht frei

4.7-1

4.7 Dienstprogramm MODULE

4.7.1 Erstellen eines Moduls

Ein Modul ist eine eigenständige Prozedur des Betriebssystems (Treiber) oder einer Sprache (Arithmetik), welche nach Bedarf in den Speicher geladen und an das Betriebssystem oder die Sprache angeklint wird.

Der Aufbau eines Moduls muß nach folgendem Schema erfolgen :

Byte 0 - 255 : Modul-Einbau-Routine

Byte 256 - Ende : Modul-Prozedur

Die Modul-Einbau-Routine hat die Aufgaben :

- Verarbeiten eventueller Modifikationsparameter
- Relativierung des Moduls
- Anbau des Moduls an das Betriebssystem

Ein Modul kann über 6 Parameter konfiguriert werden. Diese werden vom Betriebssystem in den Registern $\langle '7A \rangle + 8$ als 2 Byte-Binärwerte übergeben. Mit diesen können Teile des Moduls, dessen Speicheradresse in $'40$ ausgewiesen wird, modifiziert werden. Ein Modul kann relativ geschrieben sein. Ist dies jedoch zu aufwendig, kann es unter Verwendung einer Betriebssystem-Routine, welche in $'5E20$ beginnt, relativiert werden. Diese Routine ermöglicht auch den Anbau an das Betriebssystem.

Aufruf : CSA, XKER, $'5E20$

Parameter: R0 / R3	0 oder Adresse des Modulanfangs
R4 / R5	0 oder Adresse der Modulgrundstellung
R6 / R7	0 oder Rücksprung aus der Grundstellung
R8 / R9	0 oder Adresse der Relativierungstabelle
RA	$'80$ oder Nummer des Moduls im Betriebssystem

Der Parameter R0/R3 muß nur angegeben werden, wenn der Modulaufruf über die Sprungtabelle des Betriebssystems erfolgen soll. Die Modulgruppe spezifiziert in diesem Fall RA (z.B. '53 BCD).

Die Parameter R4 bis R7 sind dann notwendig, wenn vor Inbetriebnahme des Moduls eine Grundstellung erfolgen muß. In diesem Fall wird dem Betriebssystem angegeben, wo die Grundstellungsroutine beginnt und wo der Rücksprung eingetragen werden kann.

ANFANG : Grundstellung
(R4)

ENDE : JPA,, XYZ
(R6) : ENDE + 1

R8/R9 spezifiziert eine Tabelle, welche alle absoluten Adressen relativ zur Adresse 0 enthält, die durch die tatsächliche Adresse zu ersetzen sind. Ihr Abschluß ist 2 * V.

Beispiel: Einbau BCD-Arithmetik

ANFANG : LDRD, 0, '40 / Anfangsadresse
4 < & LDC, 4, 0 / keine Grundstellung
LDCD, 8, RELT / Relativierungstabelle
LDC, '0A, '53 / Kennung BCD
CSA, '1E, '5E20
JPX,,, '7C

RELT : 2 * A, MARKE 1 + 1
:
:
2 * V
V0, '0100 / Auffüllen mit Null
BCD : / BCD-Arithmetik
MARKE1 : JPA,, ENDE
ENDE : JPX,,, XKER

Der Rücksprung der Moduleinbau-Routine in das Betriebssystem erfolgt über Register '7C. Es ist zu beachten, daß die beiden letzten Byte der Einbau-Routine die Länge des einzubauenden Moduls oder Ø - Inhalt haben müssen. Im letzteren Fall wird als Modullänge die Anzahl der Sektoren der Moduldatei benutzt. Der Inhalt von Register 7A darf nicht verändert werden.

Fehlermeldung n der Routine 5E2Ø :

230 Nummer ist belegt
 Grundstellung zeigt schon auf die angegebene Adresse

4.7.2 Einbau Modul

SERV, MODULE	Das unter dem Namen f auf Einheit u gespeicherte
u, f, a, P1, P2...P6	Modul wird geladen.

Das Dienstprogramm lädt die Modul-Einbau-Routine nach '5FØØ des Betriebssystems Overlay und den Rest der Datei f (Sektor 2 bis Ende) ab Adresse a (bzw. CFSA bei Rel. 5). Danach werden die Parameter P1 bis P6 aufbereitet und ab Register < '7A > + 8 übergeben. Die Einbau-Routine wird gestartet. Diese modifiziert das Modul (Adresse in '40) und relativiert es. Danach meldet das System die Einbaugrenzen des Moduls.

5. Module des erweiterten Betriebssystems

5.1 Allgemeine Übersicht

Das Betriebssystem XOS kann zur Zeit um die folgenden Module erweitert werden:

Name	Funktion	Kapitel
1. MCARD	Treiber Kartenleser	2.2.10
MTAPE	Treiber Magnetband	2.2. 8
MEDISC	Treiber 4.8 MByte + 2.4 MByte (24 Slot)	2.2. 8
MCDISC	Treiber DIETZ-DISK	2.2. 8
MFLOPP	Treiber Floppy-DISK	2.2. 8
MINBUS	Treiber Interbus	2.8. 5
MLP600	Treiber Data-Products-Drucker	2.8. 3
2. MBACKG	Background-Verarbeitung	5.1
MSPOOL	Spool-Ausgabe	5.2
MAUTO	Automatischer Wiederanlauf	5.3
MBUSSW	BUS-Umschalter 20 bit-BUS	5.4
MRESGR	Residente Grundstellung	5.5
3. MPRSC	Prozeß-Scheduler	2.5
MPRSCF	Prozeß-Scheduler erweitert	2.5
MBCD	BCD-Arithmetik	2.7. 5
MARFB4	4 Byte-Gleitkomma-Arithmetik	2.7. 3
MARFB8	8 Byte Gleitkomma-Arithmetik	2.7. 4
MDFMS	Data File Management System	2.8. 4

5.2 Background-Modul

Im Mehrbenutzerbetrieb ist jedem Teilnehmer eine Speicherpartition und ein Dialoggerät zugeordnet. Es ist jedoch auch möglich, mehrere Programme in verschiedenen Speicherpartitionen über ein Dialoggerät zu betreiben. Alle Ein/Ausgabe-Aufträge dieser Programme über das zugeordnete Dialoggerät werden über das Background-Modul verwaltet und gesteuert. Die Zugehörigkeit der Aufträge zu den Programmen wird durch Sonderzeichen vor den einzelnen Textzeichen deutlich gemacht. Diese Sonderzeichen sind bei Programmpartition X das Hochzeichen über der Taste mit Ziffer X.

Partition 1 : !
2 : "
3 : #
4 : §
usw.

Eine Eingabezeile muß mit einem der zugelassenen Sonderzeichen beginnen, damit das Background-Modul die Eingabe richtig weiterleiten kann. Eine Ausgabezeile wird durch das Modul mit denselben Zeichen versehen, damit der Benutzer die entsprechende Zuordnung vornehmen kann. Es ist jedoch zu beachten, daß es nicht sinnvoll ist, mehrere dialogintensive Programme über ein Dialoggerät zu betreiben. Der Normalfall ist, daß ein Programm dialogintensiv ist, alle anderen aber das Dialoggerät nur bei Start und Systemmeldungen belegen.

Einrichtung Backgroundbetrieb:

SERV, MODULE, u, MBACKG, a, l, d

u = Platteneinheit

a = Moduladresse (entfällt bei Rel 5)

l = Programmebene, die MBACKG belegen kann (Regel: 11)

d = Nummer des zugeordneten Dialoggerätes (Regel: 1)

ASG, t, 128 + d

t = Programmpartition

d = Dialoggerät

Der Background-Betrieb ist nur an einem Gerät d möglich. Mit SERV, MODULE wird der Verwaltungsprozeß ins Betriebssystem eingebaut. Dieser Prozeß belegt die Rechnerebene 1. Mit ASG werden diejenigen Programmpartitionen angegeben, die dem Dialoggerät zugeordnet werden.

Beispiel:

SERV, MODULE, Ø, MBACKG, Ø, 11, 1

ASG, 1, 129

ASG, 2, 129

ASG,, 129

Hier wird durch MBACKG Ebene 11 belegt. Die Partitionen Ø, 1, 2 arbeiten mit Dialoggerät 1.

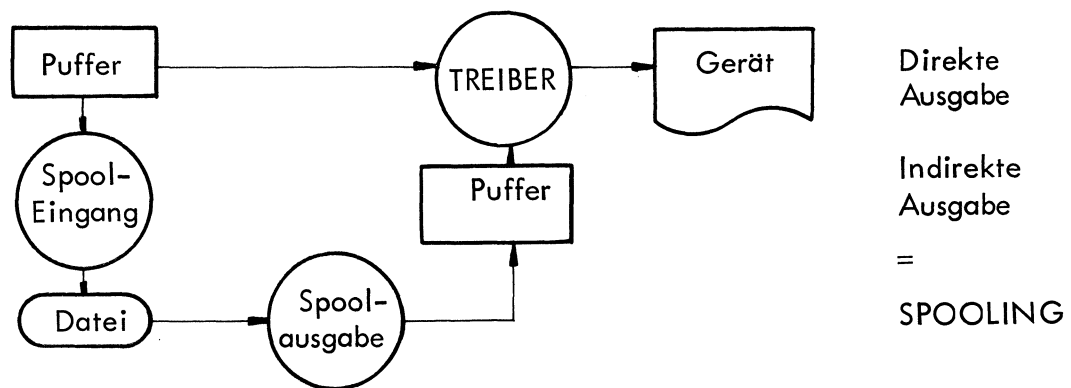
Beachte:

1. MBACKG benötigt eine Verwaltungsebene. Sie kann nicht an normale Teilnehmer vergeben werden.
2. Die Break-Taste wirkt nur auf das Programm, welches in der Partition läuft, die dem Dialoggerät üblicherweise zugeordnet ist (Hauptebene).

5.3 Das SPOOLING-Modul

SPOOLING ist ein spezielles Ausgabeverfahren auf "langsame" Geräte (Drucker).

Statt bei Ausgabeaufträgen die Pufferinhalte direkt auf das Gerät zu geben, werden diese bei SPOOLING zunächst in eine Datei (SPOOL-Datei) auf der Platte abgelegt. Auf einen speziellen Befehl hin wird dann, wenn das gewünschte Gerät frei ist, der Inhalt der SPOOL-Datei ausgegeben.



Neben den Nachteilen, daß das SPOOLING-Verfahren Platz im Hauptspeicher und auf der Platte belegt und natürlich den Prozessor stärker belastet, hat es folgende Vorteile:

1. Ein Programm mit Ausgabeaufträgen muß nicht auf die Fertigstellung dieser Vorgänge warten.
2. Ein Programm muß nicht auf Zuteilung eines Gerätes warten, wenn dies im Multi-Programming gerade belegt ist.
3. Das Gerät wird besser ausgenutzt, wenn zwischen zwei Druckaufträgen rechenintensive Vorgänge nötig sind.

Das Spooling-Modul MSPPOOL besteht aus 3 Teilen:

- Spooltask (resident)
- Spooldienstprogramme
- Spooloverlay-Datei

Die Spooltask belegt eine Rechnerebene. Sie wickelt Ein/Ausgabe-Aufträge über Standardgeräte-Peripherie über Platte ab. Die Dienstprogramme erlauben die Steuerung der Spoolvorgänge durch den Benutzer.

Zur Aufnahme von "Spooldaten" muß der Benutzer die Spooldatei SPOOLF auf der Systemresidenz mit 1025, 2050, 3075,.....31775 Sektoren einrichten.

Einrichtung Spooltask und Dienstprogramme:

SERV, MODULE, u, MSPOOL, a, l, m, t, d1, d2, d3

u = Platteneinheit

l = Level, auf dem die Spooltask läuft

m = Größe der für Spoolarbeitsnummer reservierten Tabelle
(m * 8 = Platzbedarf)

t = Zahl der Teilnehmer

d1..d3 = Geräte, über die gespoolt werden soll

SERV, NEWUT, u, SPOOL

Mit diesem Dienstprogramm werden die Overlay-Utilities in die Dienstprogrammdatei SERVIC eingebaut.

Utilities werden mit SERV, SPOOL unter folgender Spezifikation aufgerufen:

STA	Abfrage des Spoolzustandes
MES, M, m, "text"	Definition einer Nachricht, die vor Durchführung des Spoolvorganges der Marke m auf Masterkonsole angegeben werden soll.
MES, D, r, "text"	wie oben für Gerät r
REW, D, r	Wiederholen der Ausgabe auf Gerät r mit Abbruch des laufenden Vorganges
REP, D, r, i	Wiederholen der Ausgabe auf Gerät r i-mal
REP, M, m, i	Wiederholen der Ausgabe für Arbeitsmarke m i-mal
ABO, D, r	Lösche Daten für Vorgang auf Gerät r mit Abbruch des laufenden Vorganges
ABO, M, m	Lösche Daten für Arbeitsmarke m
CON, D, r	Fortsetzen der Datenausgabe für Gerät r nach Ausgabe der Spool-Nachricht

In Sprachen wird ein Spoolvorgang mit dem Befehl REQ, d eingeleitet, mit REL, d beendet. Der Benutzer kann Utilities sowohl im Dialog als auch als Statement (DIAL in C-BASIC, BASEX) verwenden.

5.4 Programm-Wiederanlauf-Modul

Das Modul " MAUTO " (automatisches Abarbeiten von bestimmten BATCH-Dateien nach Netzausfall) gewährleistet das automatische Laden und Starten von Programmen.

Das Ereignis zum Start der Abarbeitung kann sein:

- Netzausfall bzw. Netzwiederkehr
- Betätigen der Reset- und Start-Taste .

Die Steuerung des Ablaufs im Falle eines Wiederanlaufs wird mit Kommandos durchgeführt, die auf einer bestimmten Datei (Batch-Verarbeitung) abgelegt sind.

Um über die Wiederanlaufroutine das Programm neu zu starten, sind bestimmte Vorkehrungen zu treffen:

- Betriebssystem im Kernspeicher um Modul MAUTO erweitern
- Auftragsdatei füllen.

Zusätzlich muß der Schalter "Auto-Loader" am Plattencontroller ausgeschaltet sein.

Der Plattenschreibschutz im Drive, der beim Einschalten von Gesamtanlagen mit Wechselplatte einmalig durch manuelles Betätigen der Protect-Taste ausgeschaltet wird, muß durch eine Hardwarebrücke aufgehoben werden.

Ablauf des Wiederstarts

Das im Kernspeicher stehende Betriebssystem durchläuft die Grundstellung, findet in der Systemvariable CDIA (' 4016) die Fortsetzungsadresse für die Wiederanlaufroutine, durchläuft sie und eröffnet daraufhin die speziellen BATCH-Auftragsdateien, die anschließend gestartet werden.

Besitzt ein User keine spezielle Auftragsdatei, so tritt beim Versuch, diese zu eröffnen, ein Fehler auf, der jedoch interpretiert wird. Dieser User wird in den Kommando-Mode versetzt.

Aufbau von Auftragsdateien

Die Auftragsdateien, die unter dem Namen

BAUTOn (n = Usernummer)

auf der Platte liegen, können entweder durch den Text-Editor oder durch einen BATCH-Auftrag erstellt worden sein.

Dabei sind die Kommandos in der Reihenfolge einzugeben, wie sie auch sonst über die Tastatur der Konsole im direkten Dialogbetrieb eingegeben würden.

Beispiel über den Inhalt einer Auftragsdatei :

```
RUN, NBASIC / Neu-BASIC laden
FFFF/ KERNSPEICHER-ENDE
B000/ SPEICHER F. BENUTZER 0
D500/ SPEICHER F. BENUTZER 1
FFFF/ SPEICHER F. BENUTZER 2
# / ENDE SPEICHERZUTEILUNG
GOTO/ MASTERKONSOLE IN BASIC
LOAD, PROG/PROGRAMM LADEN
MES, 1, *** ALLES GESTARTET ***
RUN
```

5.5 Modul BUS-Switch

Die Anlage DIETZ 600/621 kann als Doppelprozessor geliefert werden, wo zwei Rechner über eine Hardware-Einrichtung BUS-Switch die Möglichkeit haben, auf ein Gerät zuzugreifen. Will ein Prozessor auf das Gerät zugreifen, so prüft die Hardware, ob dieses gerade vom anderen Prozessor belegt ist.

Hauptbeispiel für ein BUS-Switch-Gerät ist eine Platteneinheit.

Das Betriebssystem muß bei einer Doppelprozessoranlage um das Modul MBUSSW erweitert werden. Es wird folgendermaßen montiert:

SERV, MODULE, u, MBUSSW, a, d, r1, r2, r3

d = Gerätenummer (17, 18, ...)

r_i = Semaphore (siehe System-Semaphore)

Die Semaphore r1, r2, r3 regeln langfristige Verarbeitungszugriffe zu einem Gerät.

Anhang 1 :
Geräteadressen und Rückmeldeebenen

Device- Nummern	Status- wort	Ebene CNP-	Kennung	Bezeichnung
1	1001	0		Operator/Masterkonsole
2	1011	0		Lochstreifengerät
3	1061	-1		Drucker 1
4	1021	-1		Drucker 2
5	1031			
6	3F01	-1		ADM
7	1121	-1		Plotter
9	1131	-1		Lochkartenstanzer
8	1111	-1		Lochkartenleser
10	1807/1818	-1		Platte WPCE 621 oder WP 9.6 oder Festplatte
14	1A01	-1		Diskette
15	1705	-1		Band 16 bit Controller
	1706			Band 20 bit Controller
21	1151	-2		Dialogkonsole User 1
22	1161	-2		Dialogkonsole User 2
23	1171	-2		Dialogkonsole User 3
24	1181	-2		Dialogkonsole User 4
25	1191	-2		Dialogkonsole User 5
26	11A1	-2		Dialogkonsole User 6
27	11B1	-2		Dialogkonsole User 7
28	11C1	-2		Dialogkonsole User 8
29	11D1	-2		Dialogkonsole User 9

Device- Nummern	Status- wort	Ebene	Kennung	Bezeichnung
30	11E1	-2		Dialogkonsole User 10
31	11F1	-2		Dialogkonsole User 11
	Alternativ zu 31, 3Ø,			
31	1501	-1		DFÜ 1 (1401 = CRC-Prüfung)
30	1511	-1		DFÜ 2 (1411 = CRC-Prüfung)
29	1521			weitere Strecken
28	1531			

Bemerkungen:

- Gerät 1 und 2 (100X, 101X) bleiben auf Ebene Ø. Sie werden vom IOCS jedoch so behandelt, als wären sie auf Ebene CNP-2, d.h. dort laufen die Treiber. Die Rückmelde-Abfrage erfolgt jedoch getaktet durch die Uhr (msec).
- Im Urladerstreifen TSOS X ist standardmäßig nur Gerät 1, 2, 10 montiert. Alle übrigen sind durch die Montierungs-Dienstprogramme einzutragen (siehe 4).
- Mehr als 10 Rückmelde-Interfaces in einem System benötigen einen READY-Scanner. Es ist möglich, auch innerhalb eines Systems Gerät mit und ohne Scanner zu betreiben.

Anhang 2 : Einlesen des Urlader-Streifens

1. Vorbereiten der Platte

Die Schalter des Plattencontrollers müssen richtig stehen.

Controller WPCE 621 (2.4 MB, 12 Slot)

	1	unten	Adreßinkrementierung zugelassen
	2	Mitte	Auto-Load bei Netzausfall
!	3	oben	LOAD nicht zugelassen
!	4	oben	Schreibschutzspur Ø unwirksam
	5	Mitte	Schreibsperre unwirksam

Controller WP 9.6

	1		Schreibschutz unwirksam
!	2	unten	Schreibschutzspur Ø unwirksam
	3		Auto-Load
	4		LOAD-Taste

Der LOAD/RUN-Schalter am Plattenlaufwerk muß auf RUN stehen. Die READY-Lampe muß leuchten, alle anderen Lampen müssen aus sein.

2. Einlesen Lochstreifen

Der Lochstreifen ist im Rubout-Bereich in den schnellen Leser zu legen.
Der BS-Schalter am Rechner muß gesetzt werden, das Konsolgerät ist
betriebsbereit zu machen.

- Taste RS (Reset) und ST (Start) betätigen.
Das Bootstrap-Programm wird eingelesen.
- Bei Halt den BS-Schalter wegnehmen, ST betätigen.
- Das System meldet sich mit 2 Fragen:

? NEW : Antwort Y Inhaltsverzeichnis neu anlegen
N altes Inhaltsverzeichnis beibehalten

Es wird davon ausgegangen, daß die beiden ersten Einträge
auf der Platte SERVIC und TSOS-X sind.

? DISK : Antwort F 60 MB - Platte
E 4.8 MB
H 2.4 MB 24 Slot
D 2.4 MB 12 Slot
C DIETZ-Disk

Nach Eingabe der Antworten liest der Urlader die Initialisierungs-Routine
für die angegebene Platte ein, initialisiert sie und bringt die Datei SERVIC
und TSOS-X auf die Platte.

Anhang 3: XOS-Fehlerliste

- 1 Gerät defekt / nicht vorhanden
- 2 Pufferadresse für Treiber nicht zulässig
- 3 Gerät ist nicht betriebsbereit
- 4 Hardware-Schreibschutz
- 5 Lese (CRC)-Fehler oder Spur nicht gefunden
- 6 Treiberparameter falsch
- 7 Filemark gelesen
- 8 Begin-of-tape erreicht
- 9 End-of-tape erreicht
- 10 Blocklängen-Fehler
- 11 Papier-Schacht leer bzw. voll
- 12 Fehlerhafte Datenübertragung
- 13 Einziehfehler beim Kartenleser
- 14 Geräte-Operation nicht beendet
- 15 DMA-Zugriff nicht möglich (Time-out)
- 16
- 17 Resource nicht existent
- 18 Gerät kann nicht gesperrt werden, da disarmiert
- 19 gew. Speicherbereich zu groß
- 20 doppelter Request
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30

A 3-2

- 31
- 32
- 33 Überführung in verlangten Zustand nicht möglich
- 34 Startadresse nicht zulässig
- 35 Event nicht vorhanden
- 36
- 37
- 38
- 39
- 40 Keine Spool-Marke frei
- 41 Spoolfile-Überlauf
- 42 Spool-Marke falsch
- 43 Spool-Kommando: Parameter nicht plausibel
- 44 Plattenfehler in Spoolfile
- 45 Überlauf Spoolgeräte-Tabelle
- 46
- 47
- 48
- 49 Falsche Interrupt- oder Ebenen-Nummer
- 50 Startadresse nicht zugelassen
- 51 Auftragstabellen-Überlauf
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60

- 61
- 62
- 63
- 64
- 65 Datei nicht oder doppelt vorhanden bzw. nicht offen
- 66 Startadresse nicht erlaubt
- 67 Dateilänge falsch (= 0)
- 68 verlangte Zugriffsart nicht möglich
- 69 Plattenüberlauf
- 70 Dateiende bei Get-Put erreicht
- 71 Alle Arbeitsnummern belegt
- 72
- 73
- 74
- 75
- 76
- 77
- 78
- 79
- 80
- 81 Konversions-Fehler
- 82
- 83
- 84
- 85
- 86
- 87
- 88
- 89
- 90

A 3-4

91	
92	
93	
94	
95	
96	
97	Kommando nicht bekannt
98	Falsche Paramter im Kommando
99	
100	Dateiende bei Zugriff zu DFMS-Datei erreicht
101	Satzende bei Zugriff zu DFMS-Datei erreicht
102	Falscher Dateityp angegeben
103	Schlüssel nicht existent, doppelt (Code 2.6) oder leer
104	Dateigrößenparameter unzulässig
105	Arbeitsnummern alle belegt oder unzulässiger Wert
106	DEV belegt oder andere Prozedur
107	Prozedur bereits eingetragen
108	Prozedur nicht eingetragen
109	Durchführung nicht möglich
110	Überlauf-USER-Tabelle
111	Überlauf Puffer-Tabelle
112	Beendigung zur Zeit nicht möglich
113	alte Release-Nummer
114	Empfangsfehler
115	Sendefehler
116	N Zeitfehler
117	N Sequenzfehler
118	N BCC/CRC-Fehler
119	N Steuerzeichenblockfehler
120	N Block wiederholt gesendet

- 121 Modemfehler
- 122
- 123
- 124
- 125
- 126
- 127
- 128
- 129 Dienstprogramm nicht existent
- 130 falsche Parameter im Dienstprogramm
- 131 falsche Gerätenummer
- 132 Gerät ist belegt beim Demontieren
- 133 Gerät oder BUS-Adresse bei MOUNT schon vorhanden
- 134 bei MOUNT ist BUS-Adresse nicht verdrahtet
- 135 bei MOUNT keine Rückmeldung
- 136 Batchdatei schreibgeschützt
- 137 CRC-Fehler Lochstreifen
- 138 Uhrzeit falsch eingegeben
- 139 falsche Parameter bei List/Read
- 140
- 141 Platten haben unterschiedliche Sektoranzahl
- 142 Platte defekt, kann nicht initialisiert werden
- 143 Länge Utility stimmt nicht
- 144 Funktion auf Systemresidenz verboten
- 145
- 146
- 147
- 148
- 149 Meßbereichsüberschreitung (CALL INAI)
- 150

A 3-6

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181	
182	
183	
184	
185	
186	
187	
188	
189	
190	Speicherüberlauf (Afigraf-Speicher)
191	Speicherüberlauf- Rechner
192	Unterbildname falsch (doppelt oder nicht vorhanden)
193	falsche Bildnummer
194	Text zu lang
195	OPEN fehlt bei CLOSE oder Aufruf Primitive
196	Zahlenbereichsüberschreitung
197	
198	
199	
200	
201	
202	
203	
204	
205	
206	
207	
208	
209	
210	

A 3-8

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230 Nummer des Moduls ist bereits belegt

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

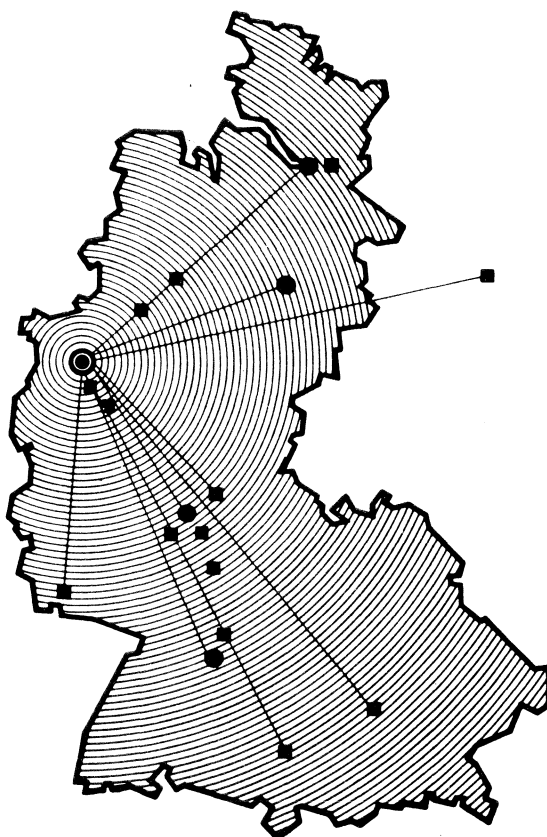
251

252

253

254

255 Funktion nicht definiert



• **dietz**-Hauptsitz und Geschäftsstellen in der Bundesrepublik

Hauptsitz und
Geschäftsstelle Mülheim
Solinger Straße 9
Postfach 130 220
D-4330 Mülheim a.d. Ruhr 13
Tel.: (0208) 48 50 24
Telex 856 770

Geschäftsstelle Hamburg
Grenzkehr 7
D-2100 Hamburg 90
Tel.: (040) 7 68 27 27

Geschäftsstelle Hannover
Alter Flughafen 21
D-3000 Hannover 1
Tel.: (0511) 63 55 42

Geschäftsstelle Frankfurt
Hardtstraße 102
D-6208 Bad Schwalbach
Tel.: (06124) 43 43

Geschäftsstelle Stuttgart
Kaiserstraße 141
D-7410 Reutlingen
Tel.: (07121) 1 70 01;
Telex 729 539

■ **dietz**-Vertriebsorganisation für die kommerziellen Systeme

Ehrig GmbH
Bismarckstraße 45
D-1000 Berlin 12
Tel.: (030) 3 41 70 21

Friedrich Karl Schroeder
Steilshooper Straße 293
D-2000 Hamburg 60
Tel.: (040) 63 86-1

BOG Daten-Kontor
Rather Straße 25
D-4000 Düsseldorf 30
Tel.: (0211) 48 20 61

Computer-Ring Rheinland
Solinger Straße 159
D-4018 Langenfeld 4
Tel.: (02173) 1 78 81

DACO
Computer Vertrieb GmbH
Bult 2
D-4400 Münster
Tel.: (0251) 4 00 73

Vordemfelde & Genck
Großhandelsring 8
D-4500 Osnabrück
Tel.: (0541) 5 70 71

ORG-TEAM
Gesellschaft für Datensysteme mbH
Triftstraße 2-10
6380 Bad-Homburg v.d.H.
Tel.: (06172) 3 50 05-6

BOD Datensysteme GmbH
Nahestraße 2
D-6500 Mainz
Tel.: (06131) 6 20 07

BOD Datensysteme GmbH
Bahnhofplatz 14
6550 Bad-Kreuznach
Tel.: (0671) 3 15 90

BOD Datensysteme GmbH
Trierer Straße 64
6600 Saarbrücken 64
Tel.: (0681) 4 61 65

Daten-Kontor GmbH & Co.
Friedrich-Engelhorn-Straße 7-9
D-6800 Mannheim 1
Tel.: (0621) 33 20 91-93

Günter Seybold KG
Weidacher Straße 26
D-7024 Filderstadt 1
Tel.: (0711) 70 20 69

MDT-Computer-Vertriebs-GmbH
Dorfstraße 21
D-8000 München 60
Tel.: (089) 8 11 20 63

Weiß Büro – Datentechnik
Salzstraße 27
D-8960 Kempten
Tel.: (0831) 2 44 64

9070596

20

7 11 34 35 42 49 40

Heinrich Dietz
Solinger Straße 9
4330 Mülheim-Ruhr
Tel.: (0208) 485024
Telex 856770

DIETZ Computer
SYSTEME