

RUHR - UNIVERSITÄT BOCHUM

Arbeitsberichte  
des  
Rechenzentrums

Direktor: Prof. Dr. H. Ehlich

Nr. 7204

Einführung in die Benutzung des  
Teilnehmer-Rechensystems TR 440  
in der Ruhr-Universität Bochum

von  
R. Mannshardt / P. Pottinger

Bochum, Mai 1972

~~Load / Unload~~

## Inhaltsverzeichnis

	<u>Seite</u>
Vorwort, Abkürzungen, Organisatorisches	4
<u>1. Allgemeines</u>	
11. Grundsätzliches über den Verkehr mit der Rechenanlage	6
12. Bausteine der Kommandos	13
13. Form eines Tätigkeitskommandos	17
14. Vermittlerkommandos	22
<u>2. Programme</u>	
21. Allgemeines	26
22. Ein-und Ausgabe von Programmen	28
23. Ein-und Ausgabe von Daten	30
24. Testhilfen	33
25. Speicherplanung	36
26. Kommandos für TAS-Programme	37
<u>3. Benutzerstationen</u>	
31. Grundbegriffe	38
32. Eingabe	41
33. Ausgabe	43
34. Operateurverkehrseingaben	46
<u>4. Gespräche</u>	
41. Grundform eines Gesprächs	50
42. Reaktionen auf Fehlermeldungen	54
43. Anweisungen	56
44. Vorrangige Kommandos	58
45. Testhilfen	61
<u>5. Dateien</u>	
51. Grundbegriffe	67
52. Dateiverkehr allgemein	71
53. Texthaltung	77
54. Bibliotheken	79
55. Magnetbänder	81

## Inhaltsverzeichnis (Fortsetzung)

	<u>Seite</u>
<u>6. Sonstiges</u>	
61. Informationsdienste	86
62. Steuergrößen und Sprünge	87
63. Speicherung von Kommandos	91
64. Graphische Ausgabe	93
65. Codewandlung	95
66. Analog-Digital-Wandlung	96
<u>Anhang</u>	
I. Voreinstellung und Maximalwerte	97
II. Speicherbedarfsgruppen	98
III. Anlagenspezifische Voreinstellungen (Bochum)	99
IV. Karten- und Lochstreifencodes	100
V. Zentralcode	101
VI. Beispiele für Lochkartenabschnitte	102

Einführung in die Benutzung des Teilnehmer-RechensystemsTR 440in der Ruhr-Universität BochumVorwort

Diese Einführung soll dem Benutzer der Rechenanlage helfen, die Leistungen der Anlage kennenzulernen und in Anspruch zu nehmen. Das wichtigste Hilfsmittel dazu ist die Kommandosprache, die in einem Handbuch (kurz: HKS) ausführlich beschrieben ist. Diese Einführung soll den Gebrauch des HKS erleichtern, indem sie die wichtigsten Begriffe erläutert und die für bestimmte Zwecke benötigten Kommandos angibt; die ausführliche Beschreibung der Kommandos muß im HKS nachgelesen werden. Daneben wird auch auf spezielle Bochumer Kommandos und Unterprogramme hingewiesen. Besonders berücksichtigt wird die Verwendung von ALGOL und FORTRAN; Näheres darüber steht in den zugehörigen TR 440 - Handbüchern. Benutzer der Dialogsprache AIDA seien vorwiegend auf die AIDA-Literatur verwiesen; eine besondere Kenntnis der Kommandosprache wird dort nicht benötigt.

Weiterführende Literatur über das System TR 440 (Begriffe, Befehle, Assemblersprache TAS, Bedienung, Hardware) kann bei Bedarf im Rechenzentrum ausgeliehen werden.

Laufende Änderungen der Software werden am Schwarzen Brett des Rechenzentrums mitgeteilt; ausführliche Beschreibungen von Neuerungen sind in der Programmbibliothek (NA 03/54) erhältlich. Benutzer mit einer Problemnummer werden regelmäßig durch die "Hauspostille" COMPuTerPOST informiert.

#### Abkürzungen:

Der Hinweis "HKS 4" bedeutet "Teil 4 im HKS". Bei den Kommandos wird meistens der Teil des HKS angegeben, in dem das Kommando beschrieben wird, z.B. XBA (7). Spezielle Bochumer Kommandos, die nicht im HKS stehen, werden durch BO gekennzeichnet, z.B. BAND (BO). Auf rechenzentrums-spezifische Literatur wird gelegentlich in der Form (BO.C2.03) hingewiesen.

#### Weitere Abkürzungen:

BKZ	=	Benutzerkennzeichen
Kdo	=	Kommando
Spz	=	Spezifikation
STDDDB	=	Standarddatenbasis
OEFDB	=	Öffentliche Datenbasis
LFD	=	Langfristige Datenhaltung
SAS	=	Satellitensystem
RJE	=	Remote Job Exit

#### Zur Organisation:

Bevor ein Benutzer den TR 440 in Anspruch nimmt, muß er eine "Problemnummer" beantragen. Außerdem ist bei Bedarf zu beantragen:

- a) Speicherplatz in der LFD  
(wird unter einem BKZ reserviert)
- b) Magnetbänder  
(Kennzeichen werden dem Benutzer mitgeteilt).

Weiteres wird am Schwarzen Brett und in den Räumen für die Benutzer mitgeteilt.

## 1. Allgemeines

### 11. Grundsätzliches über den Verkehr mit der Rechenanlage

#### 111. Aufbau der Anlage

Die Rechenanlage TR 440 besteht aus dem Digitalrechner RD 441 als Zentraleinheit, den angeschlossenen Peripheriegeräten (Ein-/Ausgabegeräten sowie Trommel-, Platten- und Magnetbandspeichern) und dem Satellitenrechner RD 186 mit angeschlossenen Benutzerstationen (Konsolen und Sichtgeräten). Von den Benutzerstationen aus haben die Benutzer direkten Zugriff zu den Leistungen des Rechners. Die Zuweisung der Leistungen und der Betriebsmittel der Anlage an die Benutzer wird vom Betriebssystem (BS3) organisiert. Das Satellitensystem (SAS) steuert speziell den Verkehr der Benutzerstationen.

Für die Eingabe von Daten und Programmen stehen im Rechenzentrum folgende Geräte zur Verfügung:

Lochkartenleser,

Lochstreifenleser (für 5- und 8-Kanal-Streifen).

Die Ausgabe der Ergebnisse erfolgt über Schnelldrucker, Karten- und Streifenstanzer und Plotter.

Die Konsolen dienen gleichzeitig der Eingabe (über Tastatur oder Lochstreifen) und der Ausgabe (auf Papier oder Lochstreifen).

Die Sichtgeräte arbeiten im wesentlichen wie Konsolen; die Eingabe erfolgt nur über Tastatur, die Ausgabe auf dem Bildschirm. Was im folgenden von Konsolen gesagt wird, gilt i.a. auch für Sichtgeräte. Man kann außerdem auch Bilder, wie sie auf dem Plotter gezeichnet werden können, auf dem Bildschirm ausgeben.

#### 112. Kommandos und Spezifikationen

Die Eingabe erfolgt in Form von Kommandos (kurz: Kdos); auch Programme und Daten sind Bestandteile von Kdos. (Ausnahme: Bei Gesprächen können auch "Anweisungen" und Daten direkt eingegeben werden, s.4.). Die meisten (im HKS beschriebenen) Kdos besitzen eine oder mehrere Spezifikationen (kurz: Spzn), denen bei der Eingabe bestimmte Spz-Werte zugewiesen werden müssen, z.B. Angaben für die Übersetzung eines Quellprogramms oder für die Protokollierung eines Datentransports. Dabei kommt auch der Wert "-" ("undefiniert") vor. Eine Spz heißt obligat, wenn sie einen von - verschiedenen Wert haben muß, andernfalls optional. Im HKS sind die Spzn zu jedem Kdo in einer bestimmten Reihenfolge aufgeführt, zuerst die obligaten, dann die optionalen Spzn. Für jede Spz ist auf der Maschine ein Wert voreingestellt. (vgl. Anhang III.)

Diese Voreinstellung kann vom Benutzer vorübergehend verändert werden durch Deklarationskdos (HKS 2, Abschn. 4); durch ein solches Kdo lassen sich außerdem Spz-Werte (die z.B. ein ganzes Programm umfassen können) unter internen Namen speichern (s. 124., 131.). Alle anderen Kdos heißen Tätigkeitskdos (s. 13.). Eine besondere Klasse hiervon bilden die Vermittlerkdos (s. 14.); diese richten sich an gewisse "Vermittler", nämlich an Teile des Betriebssystemes zur Steuerung der Eingabevorgänge.

### 113. Abschnitte und Gespräche

Der Auftrag, den ein Benutzer durch eine Folge von Kdos der Maschine erteilen will, kann als "Abschnitt" oder als "Gespräch" formuliert sein. Der Auftrag wird dann im "Abschnittsmodus" bzw. im "Gesprächsmodus" bearbeitet. Er erhält von der Maschine sofort eine Auftragsnummer.

Ein Abschnitt beginnt mit dem Kdo XBA (7 oder 8) und endet mit XEN (7 oder 8). Die dazwischenliegenden Kdos werden nacheinander abgearbeitet, sofern keine Sprünge eingeplant sind (s. 62).

Wird ein Abschnitt per Lochkarten eingegeben, so spricht man von einem Lochkartenabschnitt. Die erste Karte (auf der das Kdo XBA (7) beginnt) muß eine rote Karte ohne Eckenabschnitt sein und muß mit der Problemnummer des Benutzers beschriftet sein; außerdem muß sie gut sichtbar



gekennzeichnet werden, wenn bei der Bearbeitung des Abschnitts ein Operateureingriff verlangt wird:

S wenn Karten oder Streifen zu stanzen sind

F wenn ein Plotter betrieben wird

A wenn Magnetbänder verwendet werden.

Eine solche Kennzeichnung ist auch nötig für Lochstreifenabschnitte, die in einem Streifenleser im Rechenzentrum eingegeben werden sollen. Wird ein Abschnitt auf einer Konsole eingegeben (per Tastatur oder Lochstreifen), so heißt er Konsolabschnitt. Nach Eingabe eines Abschnitts hat der Benutzer keinen Einfluß auf die Bearbeitung eines Auftrages mehr.

Beispiele für Abschnitte sind im Anhang VI zu finden.

Ein Gespräch wird auf einer Konsole geführt, es beginnt mit XBC(S) und endet mit XEN(S) oder XEG(S) (wenn der Auftrag im Abschnittsmodus fertig bearbeitet werden soll).

Ein Gespräch sollte nur geführt werden, wenn der direkte Kontakt mit der Maschine gewünscht ist; Aufträge mit langen Bearbeitungszeiten (z.B. Übersetzung sehr großer Programme) sollten als Abschnitte eingegeben werden.

Magnetbänder können im Gespräch nicht verwendet werden, höchstens in dem nach XEG folgenden Abschnitt.

#### 114. Ablaufprotokolle

Nach der Bearbeitung eines Abschnitts wird auf einem Schnelldrucker ein Ablaufprotokoll erstellt. Dieses enthält Meldungen über die Ausführungen der einzelnen Kdos, einschließlich Fehlermeldungen, und es ist evtl. durchsetzt von Ergebnissen, die von ALGOL- oder FORTRAN-Programmen auf dem Schnelldrucker ausgegeben werden sollen.

Bei einem Gespräch erhält man stattdessen ein Konsolprotokoll. Es enthält die protokollierten Eingaben, Meldungen (wie beim Abschnitt) und Ergebnisse, die für die Konsole bestimmt sind.

Mit Hilfe des Kdos DRPROTOKOLL (3) kann das Ablaufprotokoll auf einem gewünschten Gerät (z.B. auf einer Konsole) ausgegeben werden; auch für Gespräche können Ablaufprotokolle angefordert werden. So wird z.B. durch das Kdo

◇ DRPROTOKOLL, EIN, FS(08,1),2

das Ablaufprotokoll auf der Konsole 8 zweimal ausgedruckt. Mit dem Kdo NEUSEITE (3) wird im Ablaufprotokoll eine neue Seite begonnen.

115. Protokollierung von Operatorläufen

Die meisten Kdos bewirken den Start eines "Operators".

Beispiele:

- a) Das Kdo UEBERSETZE (3) startet einen Übersetzer, z.B. für die Übersetzung eines FORTRAN-Programms den Operator PS&FTNCOMP.
  
- b) Das Kdo STARTE (3) startet einen Operator, der aus einem Quellprogramm durch Übersetzen und Montieren entstanden ist; der Name des Operators wird (in der Spz PROGRAMM) vom Benutzer festgelegt (Standardbezeichnung: STDHF). Soll der Operator mehrmals gestartet werden, so können die verschiedenen "Operatorläufe" noch durch besondere Namen unterschieden werden (Spz LAUF).
  
- c) Einige Kdos für Dateienverkehr (z.B. EINTRAGE (4), DRUCKE (4)) bewirken den Start des Operators PS&DATTRANSP.
  
- d) Kdos für Texthaltung starten gleichnamige Operatoren, z.B. startet das Kdo TEINTRAGE (4) den Operator TEINTRAGE.

Im Ablaufprotokoll wird jeweils Start und Ende eines Operatorlaufs mit Angabe des Operator(lauf)namens und der Nettorechenzeit für den Operatorlauf protokolliert.

116. Teilaufträge

Bei der Bearbeitung eines Auftrages können außer dem Ablaufprotokoll noch weitere Ausgabeaufträge für Schnelldrucker, Stanzer und Plotter vorkommen; diese werden sofort nach der Eingabe der betreffenden Kdos bearbeitet, so daß nun die Ausgaben in "Teilaufträgen" getrennt erfolgen. Die Kennzeichnung der einzelnen Teilaufträge erfolgt in folgender Form:

<Auftragsnummer> <Teilauftragsnummer> [A] <Stationsnummer>

A bedeutet, falls angegeben, "Ablaufprotokoll". Stationsnummer = 0 bzw. 1 bedeutet: Auftrag wurde im Rechenzentrum bzw. von einer Konsole gestartet.

## 12. Bausteine der Kommandos

### 121. Codes

Die Eingabegeräte verarbeiten Aufträge wahlweise in folgenden Codes (s. Anhang IV).

Lochkarten: KC1 (normal), KC2 (IBM 029), KC3 (IBM 026,  
DFG-Code),

BIN (Binärcode des TR 440, vgl. 222., 223.,  
523.),

Lochstreifen (5 Kanäle): SC1 (normal), SC2 (enthält tiefgestellte 10),

Lochstreifen (8 Kanäle): BIN (Binärcode).

Bei Eingabe auf Konsoltastatur sind die Zeichen in SC1 verschlüsselt, bei Eingabe per Sichtgerät in TC1.

Sollen Lochkarten bzw. Lochstreifen nicht im Code KC1 bzw. SC1 eingelesen werden, so muß mit dem Kdo XUM (7 bzw. 8) die Voreinstellung geändert werden (s. 144).

Für die Ausgabe können die Stanzer in den Kdos STANZE (4), TKOPIERE (4) und BINAERAUS (4) auf die verschiedenen Codes eingestellt werden.

Intern sind die Zeichen im Zentralcode verschlüsselt; jedes Zeichen wird durch eine Oktade = 8 Bit dargestellt (s. Anhang V).

Zur Verwendung von fremden Codes s. 65.

122. Fluchtsymbole

"Fluchtsymbole" haben vor allem die Aufgabe, Kdos einzuleiten.

Das codeunabhängige Fluchtsymbol (symbolisch:  $\emptyset$ ):

- a) steht auf Lochkarten vor jedem "Vermittlerkdo" (s.14.) in Spalte 1, gefolgt von 1, 2 bzw. 3 in Spalte 2, wenn dieses Kdo im Code KC1, KC2 bzw. KC3 gelocht ist:

$\emptyset$  1X...

Lochung: 12-11-5-8,

Anschlag auf einem Locher im Code  $\begin{matrix} \text{KC1} & \text{oder} & \text{KC2:} & \emptyset \\ \text{KC3:} & & & \square \end{matrix}$

(Mehrfachlochung)

- b) steht auf Lochstreifen, die vom Streifenleser verarbeitet werden sollen, ebenfalls vor jedem Vermittlerkdo, gefolgt von einer Zusatzcode-Angabe (die nur für dieses Vermittlerkdo gilt).

Fluchtsymbol:  $\blacksquare$  Bu Bu Bu Bu oder & Bu Bu Bu Bu Bu

Zusatzcode:  $\begin{matrix} \text{SC1} & & \text{Lochung in Tetraden:} & \text{01} & \text{Abdruck:} & \text{T} \\ \text{SC2} & & & \text{03} & & \emptyset \end{matrix}$

- c) wird im Konsolbetrieb nur benötigt, wenn vorher die Fluchtsymbolerkennung abgeschaltet oder der Binärkode eingeschaltet war; Darstellung wie unter b). Für Sichtgeräte ist ein codeunabhängiges Fluchtsymbol nicht definiert.

Das codeabhängige Fluchtsymbol (symbolisch:  $\diamond$ , bei Konsoleingabe auch  $\square$ ) steht vor jedem Kdo, das nicht

vom codeunabhängigen Fluchtsymbol eingeleitet wird. Außerdem muß jedes Vermittlerkdo und jede Konsoleingabe in einem Gespräch mit "◇." abgeschlossen werden. Eine Folge "◇" gilt wie ein einziges.

Lochkarten:			Konsole/Fernschreiber		
<u>Code</u>	<u>Lochung</u>	<u>Anschlag</u>	<u>Code</u>	<u>Tetraden</u>	<u>Anschlag</u>
KC1	0-2-8	■	SC1	00	■
KC2	0-2-8		SC2	05	!
KC3	11-3-8	§	Sichtgerät(Code TC1):Anschlag ■		

Mit Hilfe des Kdos XUM (7 bzw. 8) kann das codeabhängige Fluchtsymbol abgeschaltet werden; im Code BINAER wird es sowieso nicht erkannt.


Ein am Eingabegerät nicht darstellbares Zeichen des Zentralcodes kann in der Form ◇n eingegeben werden, wobei n eine dreistellige Dezimalzahl ( $64 \leq n \leq 255$ ) ist, evtl. mit führender Null.

Durch ◇(wird eine Zeichenkette eingeleitet, die nicht von der Maschine verarbeitet, sondern nur protokolliert werden soll; die Zeichenkette wird durch den nächsten Zeilenwechsel oder durch ein Vermittlerkdo beendet (vgl. auch 135).

In einem Gespräch wird eine Eingabe des Benutzers mit ■: angefordert.

Weitere Anwendungen des codeabhängigen Fluchtsymbols :  
s. 123. und 322.

123. Fremdstrings

Programme und Daten werden innerhalb von Kdos in Form von Fremdstrings eingegeben. Ein Fremdstring ist eine Kette von beliebigen Zeichen, die kein Fluchtsymbol sind. Der Beginn eines Fremdstrings wird durch "/" angekündigt; der Fremdstring endet beim nächsten Fluchtsymbol. Falls dieses kein neues Kdo einleitet, folgt ihm ein "/"; d.h. der Fremdstring f tritt dann so auf: /f 

Leerstellen in einem Fremdstring gelten als Zeichen, außerhalb von Fremdstrings werden sie ignoriert. Wann eine Zeichenkette als Fremdstring zu betrachten ist, geht aus dem Zusammenhang hervor.

124. Namen

Zur Bezeichnung von Kdos und ihren Spzn werden Standardnamen verwendet. Ein Standardname ist eine Kette von höchstens 12 alphanumer. Zeichen (d.h. Buchstaben und Ziffern), die mit einem Buchstaben beginnt. Ein Name von Standardlänge darf außerdem das Zeichen "&" enthalten; solche Namen dienen z.B. zur Bezeichnung von Dateien. Ein FORTRAN-Name ist im Gegensatz zum Standardnamen auf 6 Zeichen begrenzt; er kann außerhalb eines FORTRAN-Programms eine Datenbasis bezeichnen. Ein ALGOL-Name kann beliebig lang sein, aber nur die ersten 35 Zeichen sind signifikant.

Ein interner Name (zur Speicherung eines Spz-Wertes) ist eine höchstens 6stellige Dezimalzahl, die nicht mit 0 beginnt.



13. Form eines Tätigkeitskommandos131. Primäres Tätigkeitskommando

Eine "primäres Tätigkeitskdo" (vgl. 112.) beginnt mit dem "Tätigkeitsnamen", evtl. gefolgt von Wertzuweisungen (für Spzn), die durch Kommata voneinander getrennt sind; Beispiel (mit vorangesetztem Fluchtsymbol, s. 122.):

◇ UEBERSETZE, QUELLE = PRØG2, SPRACHE = FTN

Tätigkeitsname: UEBERSETZE; unter diesem Namen ist das Kdo im HKS zu finden. Ein Tätigkeitsname hat stets die Form eines Standardnamens (s. 124.); ist der erste Buchstabe ein X, so ist das Kdo ein Vermittlerkdo (s. 141.).

Eine Wertzuweisung hat die Form

<Spzname> = <Spzwert>

Im obigen Beispiel, 1. Wertzuweisung:

Spzname : QUELLE,            Spzwert : PRØG2

Ein Spzname ist stets ein Standardname und bezeichnet im HKS innerhalb eines Kdos eine bestimmte Spz.

Spzwerte können sehr unterschiedlich aussehen; ihre Form wird im HKS in der Beschreibung der jeweiligen Spz festgelegt.

Die Wertzuweisung bewirkt, daß die Spz mit dem angegebenen Namen den angegebenen Wert erhält. Ist als Spzwert ein in-

terner Name angegeben (s. 124.), so ist der wirkliche Wert unter diesen Namen gespeichert (HKS2, Abschn. 1.5). Wird für eine Spz eines einzugebenden Kdos keine Wertzuweisung angegeben, so erhält die Spz den voreingestellten Wert (s. 112.); im obigen Beispiel:

Als Quellprogramm für die Übersetzung wird der Inhalt der Datei PRØG2 verwendet, Quellsprache ist FORTRAN. Für die nicht explizit aufgeführte Spz NUMERIERUNG wird der voreingestellte Wert - STD - verwendet, d.h. die Zeilen des Programms erhalten bei der Bearbeitung dieselben Nummern wie in der Datei. Die übrigen Spzn des Kdos erhalten ebenfalls voreingestellte Werte.

Bemerkung (s. 112.): Die Wertzuweisung für eine

- a) obligate Spz darf höchstens dann fehlen, wenn ein von " - " verschiedener Wert voreingestellt ist;
- b) optionale Spz muß nur dann angegeben werden, wenn sie einen vom voreingestellten Wert verschiedenen Wert erhalten soll.

Abkürzungen (mit "." dahinter):

- a) Ein Tätigkeitsname kann abgekürzt werden, soweit keine Verwechslung mit einem anderen Tätigkeitsnamen möglich ist, z.B. "UEB."

- b) Ein Spzname kann abgekürzt werden, soweit keine Verwechslung mit einem anderen Spznamen desselben Kdos möglich ist, z.B. S. (Innerhalb des Kdos UEBERSETZE gibt es nur eine Spz, deren Namen mit S anfängt.).

### 132. Weitere Vereinfachungen

Falls die Wertzuweisung für die im HKS an 1. Stelle aufgeführte Spz die erste Wertzuweisung im einzugebenden Kdo ist, kann der Spzname und das Zeichen "=" fehlen; dieselbe Vereinfachung gilt für weitere Spzn, falls die Wertzuweisungen in derselben Reihenfolge wie im HKS angegeben sind, z.B.

◇ UEB., PRØG2, FTN

Läßt man mehrere Kommata aufeinander folgen, entspricht dies der Zuweisung der voreingestellten Werte an die im HKS folgenden Spzn; z.B.

◇ UEB., SP.=FTN,,,GR,Q.=PRØG2

Hier wurden den der Spz SPRACHE folgenden Spzn NUM. und MØ die voreingestellten Werte zugewiesen, die folgende Spz VARIANTE erhält den Wert GR; bei der nächsten Wertzuweisung spielt die Reihenfolge im HKS keine Rolle, da der Spzname Q. explizit angegeben wird. Ausnahmen: s. 141.

### 133. Sekundäres Tätigkeitskmando

Will man ein Tätigkeitskdo mehrmals unmittelbar hintereinander ausführen, wobei nur die Werte für gewisse Spzn abgeändert werden, so braucht der Tätigkeitsname nicht mehr angegeben zu werden; Beispiel: dem Kdo in 131. folge

Q. = PRØG4

Dann wird das UEBERSETZE-Kdo nochmals ausgeführt; für das Quellprogramm wird dann aber der Inhalt von Datei PRØG4 verwendet.

Ein solches verkürztes Kdo heißt "sekundäres Tätigkeitskdo."; es besteht aus einer oder mehreren Wertzuweisungen (durch Kommata getrennt); die erste Wertzuweisung muß mit  $\langle$ Spzname $\rangle$  = beginnen, d.h. die Vereinfachungen in 132. sind erst von der zweiten Wertzuweisung ab erlaubt.

Ein primäres Tätigkeitskdo kann von beliebig vielen sekundären Tätigkeitskdos gefolgt werden; alle diese Kdos zusammen bilden im Sinn der Kdosprache ein einziges Kdo. Sie können in einem Gespräch nur zusammen eingegeben werden, d.h. eine Eingabe in einem Gespräch darf nicht mit einem sekundären Tätigkeitskdo beginnen. Eine Wertzuweisung in einem Kdo bleibt solange gültig, bis derselben Spz in einem sekundären Tätigkeitskdo (an späterer Stelle des vollständigen Kdos) ein neuer Wert zugewiesen wird.

#### 134. Kommandoprozeduren

Soll eine bestimmte Folge von Kdos mehrmals durchlaufen werden, so kann man sie zu einer "Prozedur" zusammenfassen, die mit einem einzigen Kdo aufzurufen ist. Näheres s. HKS2, Abschn. 5.2. Man kann eine Prozedurvereinbarung als "Voreinstellung" für das vereinbarte Kdo betrachten.

#### 135. Kommentare

Ein Kommentar wird eingeleitet durch  $\diamond =$ . Alle folgenden Zeichen bis zum nächsten Fluchtsymbol werden nicht ausgewertet, sondern nur protokolliert. Ein Kommentar entspricht einem Kdo und kann daher nicht innerhalb eines Kdos stehen.

## 14. Vermittlerkommandos

### 141. Allgemeine Kennzeichen

Mit Hilfe der "Vermittlerkdos" wird die Eingabe der übrigen Tätigkeitskdos gesteuert. Sie richten sich an den "Papiervermittler" (HKS 7) oder an das "Satellitensystem" (HKS 8). Sie sind wie primäre Tätigkeitskdos aufgebaut, jedoch mit folgenden Besonderheiten:

- a) Der Tätigkeitsname beginnt mit "X".
- b) In den Wertzuweisungen darf der Spzname nicht fehlen und nicht abgekürzt werden.
- c) Das Kdo wird mit "◇." bzw. "▣." abgeschlossen.

Soll das Kdo in den Karten- oder Streifenleser eingegeben werden, so muß es mit dem codeunabhängigen Fluchtsymbol und Codeangabe eingeleitet werden; dasselbe ist evtl. auch bei Konsoleingabe nötig (vgl. 122.). Weitere Vorschriften s. HKS 7 (Abweichungen von der allgemeinen Kommando-Syntax). Im folgenden werden die Kdos XBA, XBG, XEN, XEG und XUM behandelt; weitere Vermittlerkdos werden in 3. und 4. besprochen.

### 142. Einleitung eines Auftrages

Das Kdo XBA leitet einen Abschnitt ein, das Kdo XBG ein Gespräch (s. 113.). Sie haben dieselben Spzn. Die erste Wertzuweisung muß mit "BEN=" beginnen; es folgen maximal

30 Zeichen einschließlich Leerstellen zur Kennzeichnung des Benutzers, und zwar zunächst die 6stellige Problemnummer (muß im Rechenzentrum beantragt werden), dann (nur bei Konsoleingabe) "/n/" (n = Nr. der Konsole), anschließend der Name des Benutzers, evtl. noch seine Telefonnummer. In vielen Fällen braucht keine weitere Spz angegeben zu werden. Ein Kartenabschnitt beginnt also z.B. mit

Ø 1 XBA, BEN=290045 MUELLER - SCHMIDT ◇.

(Dieses Kdo soll auf einer roten Karte stehen.) Ein Gespräch wird eröffnet z.B. mit

▣ XBG, BEN = 520304/4/MAYER TEL. 5982 ▣.

Das erste von "Blank" verschiedene Zeichen nach der Problemnummer bzw. nach /n/ soll ein Buchstabe sein. Dieser Buchstabe und die nächsten 5 von "Blank" verschiedenen Zeichen, jedoch höchstens bis zu einem Punkt (ausschließlich), bilden das Benutzerkennzeichen (BKZ) des Auftrags. Das BKZ ist wichtig für die LFD. Man kann als BKZ z.B. den Namen einer Benutzergruppe angeben und hinter einem Punkt den Namen des individuellen Benutzers anfügen:

Ø 1XBA, BEN=120208 PHY/1.SCHULZE ◇. (Das BKZ lautet hier "PHY/1".)

Weitere Spzn: FKZ dient zur Kennzeichnung des Auftrages (z.B. TEST5), maximal 6 Zeichen zulässig.

Jeder Auftrag sollte ein auftragsspezifisches FKZ tragen, so daß auch für die Operateure mehrere Aufträge desselben Benutzers unterscheidbar sind; sonst sind Verwechslungen bei der Rückgabe unvermeidbar. -

Die Voreinstellungen und Maximalwerte für die übrigen Spzn sind im Anhang I angegeben. Jede "Speicherbedarfsgruppe" (Anhang II) legt bestimmte Werte für KSB, TSB, DRS, PSB fest, sofern diese nicht explizit im Kdo angegeben werden. Es wird empfohlen (vgl. 252.):

	TAS	5
Für mittlere Programme in	ALGOL	: SBG = 7
	FORTRAN	9 (voreingestellt)

Für große Programme ist ein erhöhter KSB vorzusehen; sollen Dateien angelegt werden, muß evtl. ein erhöhter TSB oder PSB angemeldet werden, jedoch ist der Bedarf für LF-Dateien nicht zu berücksichtigen. Beim PSB ist außerdem der Umfang (DRS) der Druckerausgabe zu berücksichtigen, da die auf Schnelldrucker auszugebende Information auf Platte zwischengespeichert wird; es muß gelten:  $PSB \geq DRS+1$ . Die Spz RZS bezieht sich auf die Zeit, in der der Auftrag den Rechnerkern belegt (Nettorechenzeit).

Möglichst exakte Angaben betr. Speicherbedarf, Druckseitenanzahl und Rechenzeit durch alle Benutzer tragen wesentlich zur Erhöhung des Durchsatzes (Verkürzung der Wartezeit) bei. Für die Bearbeitung von Abschnitten ist folgende



Einteilung zu beachten:

Kurzläufe: RZS  $\leq$  1 und DRS  $\leq$  50 (werden bevorzugt bearbeitet)

Normalläufe: RZS  $\leq$  10 und DRS  $\leq$  150

Langläufe: RZS  $\leq$  60 und DRS  $\leq$  300 (vorwiegend nachts bearbeitet)

Sonderläufe: RZS  $>$  60 oder DRS  $>$  300 oder KSB  $>$  60 oder  
TSB  $>$  400 oder PSB  $>$  2000 (nur mit Genehmigung)

#### 143. Beendigung eines Auftrags

Sowohl Abschnitte als auch Gespräche werden mit dem Kdo XEN beendet. Wird ein Gespräch durch XEG beendet, so wird die noch laufende Bearbeitung des Auftrags im Abschnittsmodus fortgesetzt.

#### 144. Das Kommando XUM

Das Kdo XUM wird benötigt:

- a) wenn die Eingabe nicht im voreingestellten Code erfolgt,
- b) bei Wechsel des Eingabe-Codes,
- c) wenn das codeabhängige Fluchtsymbol (vorübergehend) nicht erkannt werden soll,
- d) wenn von Lochkarten nicht alle Spalten gelesen werden sollen,
- e) wenn die Eingabe von der Konsole nicht protokolliert werden soll, obwohl sie weitergeleitet wird.

## 2. Programme

### 21. Allgemeines

Der TR 440 verarbeitet Programme in TAS, FÖRTRAN, ALGÖL60, BCPL, CØBØL (Näheres s. UEBERSETZE (3), Spz SPRACHE). Um mit einem solchen Programm Ergebnisse zu erhalten, sind 3 Schritte nötig, die durch je ein Kdo bewirkt werden:

- a) UEBERSETZE (3): Das Quellprogramm wird übersetzt, es wird ein Montageobjekt erzeugt. Mit einem einzigen Kdo kann man ein Hauptprogramm und mehrere Unterprogramme übersetzen, jedoch nicht mehrere Hauptprogramme. Jedes übersetzte Programm ergibt ein Montageobjekt.
- b) MØNTIERE (3): Aus einem Montageobjekt oder aus zusammengehörigen Montageobjekten wird ein Operatorkörper erzeugt. Die Montageobjekte bleiben erhalten. (Statt "Operatorkörper" sagt man oft "Operator" oder auch "Programm".).
- c) STARTE (3): Ein Operatorkörper wird (evtl. zusammen mit Eingabedaten) gestartet, es entsteht ein Operatorlauf (vgl. 115. b). Montageobjekte und Operatorkörper bleiben erhalten.

### Bemerkungen:

- 1) Einige Standardfälle im Lochkartenbetrieb sind im HKS 1C beschrieben. Man beachte aber: Für Spz SPRACHE ist ALG 60 voreingestellt. Weitere Beispiele s. Anhang VI.

- 2) Die obigen 3 Kdos sind in der Kdoprozedur RECHNE (3) zusammengefaßt. Das Kdo RECHNE bewirkt, daß jene 3 Kdos hintereinander ausgeführt werden. Für Wertzuweisungen steht nur eine Auswahl der ursprünglichen Spzn explizit zur Verfügung; für die übrigen Spzn der 3 Kdos gelten die Voreinstellungen. Falls die Übersetzung oder Montage nicht fehlerfrei ausgeführt werden kann, wird das Kdo RECHNE vorzeitig beendet, d.h. es unterbleibt ein Montage- bzw. Startversuch.
- 3) Programme in ALGOL 60 können im Dialogsystem AIDA auch ohne Übersetzung und Montage gefahren und geändert werden; AIDA umfaßt außerdem noch APL und LISP.
- 4) In einem FORTRAN-Programm kann der Name des Operatorlaufs (s. STARTE, Spz LAUF) mit CALL OLNAME (**S**) als FTN-String S geliefert werden. Der Programmablauf kann so allein durch Änderung des Laufnamens gesteuert werden.

## 22. Ein- und Ausgabe von Programmen

### 221. Quellprogramme

Das zu übersetzende Quellprogramm wird im Kdo UEBERSETZE (3) unter der Spz QUELLE angegeben. Es wird entweder als Fremdstring innerhalb des Kdos eingegeben (also auf Lochkarten/-streifen oder von der Konsole) oder aus einer Datei geholt. Die Eingabe aus einer Datei empfiehlt sich dann, wenn das Programm evtl. noch korrigiert werden muß (Bedingungen für eine solche Datei: s. Spz QUELLE.). Das Programm wird auf Wunsch auf Schnelldrucker oder Konsole ausgedruckt (Spz. PROTOKOLL), evtl. durchsetzt mit Meldungen über syntaktische Fehler. (In der Spz PROTOKOLL kann auch zusätzliches Ausdrucken des übersetzten Programms im Objektcode verlangt werden.) Mit VARIANTE = FF ist auch die formatfreie Eingabe eines FORTRAN-Programms über Konsole möglich (s. TR 440-FORTRAN-Handbuch).

### 222. Montageobjekte

Die von einem UEBERSETZE-Kdo erzeugten Montageobjekte (Benennung: Spz MØ) bleiben in der STDDB (s. 513) während eines Abschnitts bzw. Gesprächs erhalten, bis sie entweder durch ein neues Montageobjekt desselben Namens überschrieben oder mit dem Kdo LØESCHE (3) gelöscht werden. Mit dem Kdo DUMPE (3) lassen sich Montageobjekte ausdrucken, mit dem Kdo BINAERAUS (4) in verschlüsseltem

Binärkode (VBC) ausstanzen oder in eine Datei eintragen.  
Über die Speicherung in einer Benutzerbibliothek s. 54.

Die vom MONTIERE-Kdo angeforderten Montageobjekte (Benennung: Spz MØ) können als Ergebnis einer Übersetzung (in der STDDB) oder als Bibliotheksprogramm (in der ØEFDB, s. 513., oder in einer angemeldeten Benutzerbibliothek) vorliegen, oder sie können mit BINAEREIN (4) in die STDDB geholt werden. Letzteres ist nur möglich, wenn das Objekt auf Karten oder Streifen oder in einer Datei mit Hilfe des Kdos BINAERAUS (4) gespeichert wurde. Das Anmontieren von Bibliotheksprogrammen kann in MONTIERE-Kdo durch ZUSATZ = FBI verhindert werden.

Liegen mehrere gleichnamige Montageobjekte vor, so entscheidet die "Hierarchie", welches montiert wird: Zuerst wird in der STDDB, zuletzt in der ØEFDB gesucht, dazwischen ggf. in Benutzerbibliotheken.

### 223. Operatorkörper

Die von einem MONTIERE-Kdo erzeugten Operatorkörper (Benennung: Spz PRØGRAMM) bleiben während eines Abschnitts bzw. Gesprächs erhalten, bis sie entweder durch einen neuen Operatorkörper desselben Namens ersetzt oder mit dem Kdo LØESCHE (3) gelöscht werden. Die Kdos DUMPE (5) und BINAERAUS (4) lassen sich auch auf Operatorkörper

anwenden (vgl. 222.); außerdem können Operatorkörper in einer Benutzerbibliothek gespeichert werden (s. 54.). Außerdem können mit dem Kdo ~~FK~~ORRIGIERE (3) Korrekturen unmittelbar am Operatorkörper vorgenommen werden. Der vom STARTE-Kdo benötigte Operatorkörper (Benennung: Spz ~~PR~~PROGRAMM) liegt entweder als Ergebnis eines ~~MO~~NTIERE-Kdos vor (in der STDDDB), oder er steht in einer angemeldeten Benutzerbibliothek, oder er kann mit BINAEREIN (4) geholt werden (vgl. 222.). Liegen mehrere gleichnamige Operatorkörper vor, so entscheidet die Hierarchie (vgl. 222.), welcher gestartet wird.

### 23. Ein- und Ausgabe von Daten

#### 231. Standardeingabedatei

Daten, die von einem Programm auszuwerten sind, können innerhalb des Kdos STARTE (3) als Fremdstring eingegeben werden (Spz DATEN). Sie werden in der Standardeingabedatei abgelegt und können von dort aus vom Operatorlauf sequentiell aufgerufen werden. Diese Datei hat die symbolische Gerätenummer (=Dateiparameter) 5 in FØRTRAN und ALGØL. (Diese Nr. bezeichnet also den Lochkarten- oder den Lochstreifenleser oder die Konsole, je nachdem, von welchem Eingabemedium das STARTE-Kdo eingegeben wird.)

FORTRAN: einlesen mit READ-Anweisung, formatgebunden, oder mit NAMELIST. Jede Lochkarte ergibt einen Satz, d.h. Satzlänge beträgt 80 Zeichen, falls alle 80 Spalten gelesen werden, vgl. 144.; bei Eingabe per Konsole wird der Beginn eines Satzes durch Wagenrücklauf mit Zeilenvorschub definiert.

ALGOL: einlesen mit READ-Prozedur oder mit den IFIP-Prozeduren INREAL, INARRAY (formatfrei; Trennzeichen zwischen zwei Daten: jedes Zeichen, das nicht "Ziffernzeichen" ist, d.h. keine Ziffer und nicht ', 10, +, -, ., E, D) oder mit den Knuth-Prozeduren INPUT, INLIST (formatgebunden) oder zeichenweise mit der IFIP-Prozedur INSYMBOL oder mit IN.

Falls in der Lese-Anweisung eine von 5 verschiedene Gerätenr. steht, muß im STARTE-Kdo in der Spz DNUMMER eine Zuordnung zur 5 erfolgen.

### 232. Ausgabe von Ergebnissen

Ergebnisse können ausgedruckt werden ins Ablaufprotokoll (Gerätenr. 6) oder (in einem Gespräch) ins Konsolprotokoll (Gerätenr. 9). Im Gespräch erfolgt die Ausgabe erst, wenn die letzte Eingabe in der Maschine soweit verarbeitet ist, daß die Maschine eine neue Eingabe erwartet. Will man die

Ausgabe schon erhalten, wenn die betreffende Ausgabe-Anweisung aufgerufen wurde, so verwende man die Prozedur VERDRG (BO.CO.02).

FORTRAN: ausdrucken mit WRITE (mit Gerätenr.) oder PRINT (nur ins Ablaufprotokoll); stets formatgebunden.

Vorschubzeichen	Blank	1	+	0
Wirkung auf Schnelldrucker:	neue Zeile	neue Seite	kein Vorschub	eine freie Zeile
Wirkung auf Konsole:	neue Zeile	neue Zeile	kein Vorschub	eine freie Zeile

ALGOL: formatfrei ausdrucken mit PRINT, TYPE, WRITE (ins Ablauf- oder Konsolprotokoll, kann mit SYSUNIT gesteuert werden) oder mit IFIP-Prozeduren (OUTREAL, OUTARRAY) oder formatgebunden mit Knuth-Prozeduren (ØUTPUT, ØUTLIST) oder zeichenweise mit ØUTSYMBOL (IFIP) oder ØUT.

Notfalls muß in der Spz DNUMMER eine Unnumerierung der Gerätenummer erfolgen (vgl. 231.).



233. Austausch mit anderen Dateien

Man kann in einem FÖRTRAN- oder ALGÖL-Programm auch Daten aus anderen Dateien einlesen und Ergebnisse in Dateien eintragen, etwa um sie dann auszustanzen (Näheres s. 52.). Als eine Eingabedatei (mit Gerätenr. 8) kann die Konsole betrachtet werden, wenn von dort Daten auf Anforderung eingegeben werden (s. 413.).

24. Testhilfen241. Dumps

Falls bei der Ausführung des Kdos STARTE (3) ein Fehler auftritt, wird i.a. der Operatorlauf abgebrochen, und es wird (falls nicht im Kdo MONTIERE (3) stand: ZUSATZ = RVI) mit Hilfe des "Rückverfolgers" ausgedruckt, an welcher Stelle des Quellprogramms der Fehler aufgetreten ist. Auf Wunsch werden Variablen und Felder ausgedumt, d.h. ihre Namen und momentanen Werte werden ausgedruckt; undefinierte Werte von INTEGER- bzw. REAL-Größen erscheinen als 0 bzw. ÄÄ...Ä (auf Schnelldrucker) oder !!!...! (auf Konsole), undefinierte logische Größen als FALSE. Voraussetzungen für einen Dump:

- a) Im Kdo UEBERSETZE (3) muß die Spz VARIANTE den WERT D haben, und das übersetzte Montageobjekt muß beim Operatorlauf noch vorhanden sein.

- b) Im Kdo STARTE (3) muß der Dump unter der Spz DUMP spezifiziert werden.

Weitere Maßnahmen in einem Gespräch s. 452.

### 242. Dynamische Kontrollen

Mit Hilfe der Spz DYNKØN im Kdo UEBERSETZE (3) lassen sich (in das Montageobjekt) dynamische Kontrollen einbauen; diese sorgen dafür, daß beim Lauf des zugehörigen Operatorkörpers (Kdo STARTE (3)) folgende Punkte geprüft werden:

- a) Einhalten von Indexgrenzen (bei Feldern),
- b) Verträglichkeit von aktuellem Parameter und formalem Parametertyp (beim Aufruf von Unterprogrammen oder Prozeduren),
- c) Zulässigkeit von Schleifenparametern (in DØ-Schleifen).

Die Prüfung kann auf einzelne Teile des Programms beschränkt werden.

Wird bei der Prüfung ein Fehler festgestellt, so wird der Lauf abgebrochen (evtl. mit Dump, vgl. 241.).

Bei Testläufen eines Programms sollten grundsätzlich dynamische Kontrollen verwendet werden.

### 243. Überwacher

Der "Überwacher" protokolliert während eines Operatorlaufs auf Wunsch die Ausführung gewisser Anweisungen; Voraussetzung ist eine entsprechende Wertzuweisung an die Spz TRACE im Kdo UEBERSETZE (3) (Maßnahmen in einem Gespräch s. 453.). In der Spz UEBWS des Kdos STARTE (3) wird festgelegt, wieviel Seiten der Überwacher auf dem Schnelldrucker höchstens ausgeben soll. (Eine andere Art der Überwachung gestattet die Subroutine TESUEB: Die nach dem Aufruf CALL TESUEB (n) durchzuführenden nächsten n Maschinenbefehle werden überwacht. Dadurch läßt sich z.B. überprüfen, wie ein FTN-Statement übersetzt wurde.)

Setzt man in UEBERSETZE:TRACE= -STD-, VAR. = GS oder KV, und in STARTE: = BTR, so werden im Fehlerfall oder auf Anforderung (s. 453.) die letzten 20 überwachten Schritte ausgegeben (Backtracing).

### 244. Weitere Kontrollen

Die Spz PROTOKOLL der Kdos UEBERSETZE (3) und MONTIERE (3) bietet weitere Möglichkeiten, etwaige Fehler zu finden. Für den Benutzer sind dabei vor allem die Referenzlisten interessant, in denen verzeichnet ist, welche Variablen und Felder an welchen Stellen des Quellprogramms vorkommen (UEB., PROT.=R).

Eine Kontrolle in Gesprächen ist mit Hilfe der Kontrollereignisse möglich (s. 451.).

In einem FORTRAN - oder ALGOL-Programm kann mit Hilfe der Subroutine bzw. Prozedur TIME die "Zeit" während des Operatorlaufs abgefragt werden; die Differenz zwischen zwei Abfragen ergibt die dazwischenliegende Nettorechenzeit in  $10^{-5}$  Sekunden. Diese Abfrage kann für Verzweigungen im Programm benutzt werden, insbesondere zur Beendigung des Operatorlaufs nach einer vorgegebenen Zeit.

25. Speicherplanung251. Speicherbedarf

Beim Kdo XBA bzw. XBG (s. 142.) muß der voraussichtliche Kernspeicherbedarf berücksichtigt werden, den ein mit UEBERSETZE (3) aufgerufener Compiler benötigt (TAS: 17 K, ALGOL: 26 K, FORTRAN: 31 K, für kleine Programme 20 K, COBOL: 17 K). Diese Angaben sind bei den Speicherbedarfsgruppen berücksichtigt (s. 142.). Sollen in einem Auftrag Programme verarbeitet werden, die mehr Speicherplatz beanspruchen als der verwendete Compiler, so ist für den Kernspeicherbedarf das "größte" Programm ausschlaggebend; der Trommelspeicherbedarf erhöht sich jedoch mit jedem weiteren Programm (da Montageobjekte und Operatorkörper auf der Trommel gespeichert werden, von wo sie nur bei Bedarf in den Kernspeicher geholt werden.).

252. Besondere Maßnahmen bei großen Programmen

Für "große" Programme ist nicht nur ein erhöhter Speicherbedarf anzumelden, sondern es können (oder müssen) noch folgende Maßnahmen getroffen werden:

- a) im Kdo UEBERSETZE (3) : setze VARIANTE = GR (nur bei einem FORTRAN-Programm sinnvoll), DYNKØN = -, TRACE = -; in der Spz TRANSFER können Programmteile als zuladbar erklärt werden (d.h. sie werden im Operatorlauf erst bei Bedarf in den Kernspeicher geholt; hierbei sind in ALGØL und FORTRAN die Prozeduren LØAD und UNLØAD zu verwenden).

- b) im Kdo MONTIERE (3): in der Spz TRANSFER müssen die Montageobjekte der im Kdo UEBERSETZE als zuladbar erklärten Programmteile angegeben werden. Die Spz OVERLAY erlaubt eine noch bessere Ausnutzung der Speicherkapazität.
- c) Es ist dafür zu sorgen, daß Montageobjekte und Operatorkörper so bald wie möglich ersetzt oder gelöscht werden (Ersetzen durch ein gleichartiges Objekt desselben Namens, Löschen mit dem Kdo LÖESCHE (3)). Diese Maßnahme betrifft nur den Trommelspeicher.

#### 26. Kommandos für TAS-Programme

Für TAS-Programme stehen außer den in 21. bis 25. besprochenen Kdos noch zur Verfügung:

- a) KOMPRIMIERE (4) (für Ausgabe von TAS-Quellen).
- b) Die Kdos für Makrodienste: MEINTRAGE, MLÖESCHE,  
MAUSGABE (4).
- c) Im Kdo UEBERSETZE die Spz VERSION.
- d) Das Kdo DEFINIERE (HKS2, Abschn. 5.3), das es gestattet, ein TAS-Programm in der Form eines Kdos aufzurufen.

### 3. Benutzerstationen

#### 31. Grundbegriffe

##### 311. Die Benutzerstation

Eine Benutzerstation ist ein Ein/Ausgabe-Gerät, von dem aus ein Benutzer direkt mit der Rechenanlage verkehren kann. Als Benutzerstationen sind Konsolen (On-line-Fernschreiber) und Sichtgeräte (SIG 100) vorhanden. Wir bezeichnen im folgenden der Einfachheit halber jede Benutzerstation als Konsole.

Eine Konsole kann sich, wenn sie in Betrieb ist und keine Störung vorliegt, in einem der folgenden Zustände befinden:

Eingabezustand: Der Benutzer hat die Eingabe eines Abschnitts begonnen und noch nicht beendet, oder er ist in einem Gespräch eingabeberechtigt (s. 412.).

Ausgabestatus: Die Maschine hat eine Ausgabe begonnen (s. 33.).

Grundzustand: falls weder Ein- noch Ausgabestatus besteht; d.h. der Benutzer kann einen Auftrag beginnen oder die Maschine kann mit der Ausgabe eines Ablaufprotokolls oder Teilauftrags beginnen.

Ist nach dem Einschalten der Konsole keine Eingabe möglich, so liegt eine Störung vor, oder die Konsole ist gesperrt.

312. Das Satellitensystem

Das Satellitensystem (SAS) stellt die Verbindung zwischen Rechenanlage und Benutzerstationen dar. Es meldet sich auf der Konsole mit

□ □ SAS\* <Text> [ <Spez> ]

Als Text können einige bestimmte Meldungen erscheinen, die im folgenden besprochen werden; es kann aber auch ein beliebiger, vom TR 86 S-Operator eingegebener Text auftreten (s. 335.). Dahinter kann als Spezifikation eine zweistellige Sedezimalzahl stehen; ihre Bedeutung ergibt sich aus folgender Liste:

- 40 Systemauslauf
- 41 Konsolverkehr beendet
- 42 Engpass TR 440
- 43 UMF zu groß
- 44 E/A-Fehler, Fehler bei Hintergrundtransport
- 45 UMF zu klein
- 47 Abbruch durch System
- 48 Fluchtsymbolgebiet-Überlauf
- 49 Systemfehler im Satellitenvermittler
- 51 Geraet nicht vorhanden, belegt, blockiert oder nicht betreibbar
- 60 Übergabeverbot, Rechnerkoppelungsengpass
- 61 Rechnerkoppelungsengpass
- 62 Pufferengpass im TR 86
- 63 Fehler in der Klammerstruktur einer Datei
- 64 Fehlerhafter oder unzulässiger Satz, z.B. Graphiksatz im Textmodus

Meldungen des SAS, soweit nicht später besprochen:

□□ SAS\*GRUNDZUSTAND KONSOLE GESPERRT

Das SAS (nicht die Konsole!) befindet sich im Grundzustand; auf der Konsole ist keine Eingabe möglich.

□□ SAS\*KONSOLE FREI [Spez]

Die Konsole befindet sich im Grundzustand.

□□ SAS\*UNZUL. KOMMANDO

Eingegebenes Vermittlerkdo darf im momentanen Zustand der Konsole nicht eingegeben werden.

□□ SAS\*FALSCHES KOMMANDO

Eingegebenes Kdo sieht aus wie ein Vermittlerkdo (beginnt mit □X), wird aber nicht verstanden (z.B. wegen falscher Spz).

□□ SAS\*ABBR. [Spez]

Abbruch einer Ausgabe.



## 32. Eingabe

### 321. Normale Eingabe von Aufträgen

Im Grundzustand der Konsole kann mit dem Kdo XBA (8) ein Abschnitt, mit XBG (8) ein Gespräch eröffnet werden. Der weitere Verlauf eines Gesprächs wird in 41. beschrieben. Die Eingabe eines Abschnitts schließt sich unmittelbar an das XBA-Kdo an; die Konsole befindet sich im Eingabezustand, bis mit XEN (8) die Eingabe beendet wird. Anschließend kommt die Meldung:

□□ SAS\*KONSOLE FREI

Während des Eingabezustandes können alle Vermittlerkdos eingegeben werden außer XBA und XBG; XEG ist nur in Gesprächen zulässig. Man kann z.B. mit XUM (8), Spz EPR die Protokollierung der eingegebenen Zeichen (außerhalb von Vermittlerkdos) aus- und einschalten.

### 322. Korrekturen bei der Eingabe

Hat man bei der Eingabe an der Konsole einen Fehler gemacht und bemerkt ihn vor Beendigung der Eingabe, so kann er rückgängig gemacht werden mit '□' oder mit Vermittlerkdos:

▣' Letztes eingegebenes Zeichen wird gelöscht (Zwischenraum, Wagenrücklauf, Zeilenvorschub werden ignoriert); nur möglich, falls letztes Zeichen kein "▣". Mehrfache Eingabe von ▣' löscht entsprechend viele Zeichen von rechts nach links.

▣ XLZ ▣. Löschen einer Zeile (falls der folgende Zeilenwechsel noch nicht eingetastet ist), jedoch nur bis zum letzten ▣ (außer "▣n" (s. 122) und ▣'), ▣. oder ▣:. Bei der erneuten Eingabe der Zeile vergesse man nicht etwaige führende Leerstellen (z.B. in einer FORTRAN-Anweisung)!

Die neue Eingabe kann sofort nach dem Kdo erfolgen; die Maschine meldet sich nicht.

▣ XLE ▣. a) Ein eingegebener Abschnitt wird ganz gelöscht, nach der Meldung

▣ ▣ SAS\*EING. GEL., KONS. FREI

befindet sich die Konsole im Grundzustand.

b) In einem Gespräch wird die Eingabeinformation, die noch nicht mit "▣." abgeschlossen ist, bis zum letzten "▣:" gelöscht. Vermittlerkdos, die in die Eingabeinformation eingestreut und bereits ausgeführt sind, können

jedoch nicht mehr rückgängig gemacht werden. Nach der Meldung

**⌘⌘ SAS\*EINGABE GELOESCHT**

kann der Benutzer neue Kdos eingeben.

### 33. Ausgabe

#### 331. Mögliche Ausgaben auf der Konsole

Auf der Konsole können folgende Ausgaben erscheinen:

- a) im Eingabezustand Protokollierung der eingegebenen Zeichen (kann mit XUM (8) z.T. unterdrückt werden),
- b) in einem Gespräch die Antworten und Anfragen der Maschine,
- c) RJE - Ausgaben (s. 333.),
- d) Meldungen vom SAS (s. 312.) und von Operateuren (s. 334., 335.).

Auf dem Sichtgerät sind außerdem graphische Ausgaben möglich (s. 64.).

#### 332. Abbrechen einer Ausgabe

Jede Ausgabe, die nicht mit **⌘⌘ SAS\*** beginnt, kann mit XAB (8) abgebrochen werden. Wegen Abbruch der Ausgabe in

einem Gespräch s. 414., wegen Abbruch einer RJE-Ausgabe s. 333.

### 333. Remote Job Exit (RJE)

Eine RJE-Ausgabe (Ablaufprotokoll, s. 114., oder anderer Teilauftrag, s. 116.) läßt sich auf eine Konsole bringen, wenn man im Kdo DR~~PRO~~~~T~~~~K~~~~O~~~~L~~L (3), DRUCKE (4), K~~O~~MPRIMI~~E~~RE (4), TK~~O~~PIERE (4) für GERAET bzw. ZIEL die Konsole angibt; hat die Konsole etwa die Nummer 18, so gebe man an:

FS (18, 1).

Eine RJE-Ausgabe unterbricht weder eine andere RJE-Ausgabe noch ein Gespräch. Umgekehrt läßt sich in eine RJE-Ausgabe auch keine Abschnittseingabe und kein Gespräch einschachteln.

Von einer Konsole aus können innerhalb einer RJE-Ausgabe die Kdos XAB (8), XM~~O~~ (8) und XAS (8) eingegeben werden; am Sichtgerät lassen sich die Funktionstasten FO-F3 bedienen.

Die RJE-Ausgabe wird eingerahmt durch je einen einzeiligen Text:

<Auftr.-Nr.> / <Teilauftr.-Nr.> <BEN-String> <FKZ><Datum>

Sie wird zusätzlich beendet mit:

▣▣ SAS\*ENDE AUSGABE

Die Schlußzeilen werden auch bei Abbruch durch ▣XAB▣. ausgeschrieben.

Danach befindet sich die Konsole im Grundzustand.

#### 334. Meldungen vom TR 440 - Operateur

Der TR 440 - Operateur hat die Möglichkeit, einen Text an eine oder alle Benutzerstationen zu senden. Der Text wird sofort ausgegeben, wenn sich die Benutzerstation im Grundzustand befindet oder auf Ausgabe wartet. Wird gerade eine Ausgabe durchgeführt, so wird vor dem Text noch die letzte Ausgabe-Teilmenge ausgegeben. Für Benutzerstationen, die sich gerade im Eingabezustand befinden, wird der Text aufbewahrt, bis der Benutzer seine Eingabe abgeschlossen hat.

Beispiel:

▣ ØP▣

HEUTE WIRD DER RECHNER UM 22 UHR ABGESCHALTET

▣▣ SAS\*ENDE AUSGABE

Die Texte können mit dem Kommando `␣XAB␣`. abgebrochen werden. Ein im jeweiligen Zustand zulässiges `␣XEN␣`. bzw. `␣XEG␣`. bewirkt ebenfalls den Abbruch der Textausgabe.

### 335. Meldungen vom TR 86 S-Operator

Für den Fall, daß das im TR 86 S liegende Satellitenprogramm (Teil des Satellitensystems) nicht an den TR 440 gekoppelt ist (oder gekoppelt werden kann), wurde eine Benachrichtigung der in dieser Situation aktiv werdenden Benutzerstationen vorgesehen. Die Eingabe eines Vermittlerkommandos führt zu der Standardausgabe:

```
␣␣SAS*KEIN KONSOLVERKEHR
```

Der Operator des TR 86 S hat die Möglichkeit, diesen Text zu variieren und somit auch den Grund für die Standardausgabe anzugeben. Allgemeiner ist also die Form der Ausgabe für diesen Zustand anzugeben mit:

```
␣␣SAS* <text vom operateur>
```

## 34. Operateurverkehrseingaben

### 341. Allgemeines

Die Kdos `XM` (8) und `XAS` (8) für "Operateurverkehrseingaben" sind in jedem Zustand der Konsole möglich. Im Aus-

gabezustand wird noch die letzte Ausgabe-Teilmenge ausgegeben, ehe dem Benutzer mit "ØP␣:" Eingabeberechtigung erteilt wird. Die Eingabe ist mit ␣. abzuschließen. Innerhalb der Eingabe sind Fremstrings zugelassen, ebenso die unter 322. genannten Korrekturmöglichkeiten und die Kdos XEN (8) und XEG (8); die letzten bewirken zusätzlich den Abbruch des Operateurverkehrs. Falls die Eingabe wegen Engpass nicht weitergegeben werden kann, erfolgt Rückmeldung:

␣␣ SAS\*ØP-KDØ NICHT AUSFUEHRBAR

#### 342. Eingabe für die Kontrollschreibmaschine

Das Kdo XMØ (8) erlaubt es, Texte von der Konsole auf die Kontrollschreibmaschine des TR 440 zu leiten (maximal 90 Zeichen); Beispiel:

␣ XMØ ␣.

ØP␣: BITTE WARTUNG MELDEN, KONSOLE F5-1 DEFECT ␣.

Anschließend wird der alte Zustand der Konsole wiederhergestellt. - Das Kdo XMØ darf nur in sinnvollen Fällen verwendet werden, z.B. um eine Operateuranfrage zu beantworten.

343. Verkehr mit der Kontrollfunktion

Nach dem Kdo XAS (8) kann der Benutzer auf der Konsole einen Teil der Kdos eingeben, die sonst nur der Operateur auf der Kontrollschreibmaschine eingeben kann. Ein solches Kdo beginnt nicht mit einem Fluchtsymbol, sondern mit dem Namen des Kdos, evtl. gefolgt von einer oder mehreren Spzn, durch Kommata getrennt. Es kommen folgende Kdos in Frage:

SIT ("Situation")  
SITAN ("Situation mit Auftragsnummern")

mit den Spezifikationen ALL, G ("Geräte"), GSP ("Gespräche"), AB ("Abschnitte"), P ("Prioritäten"),  
<einzelne Auftragsnummer>, <Gerätesymbol>, <Zustandssymbol> sowie die Kommandos

ENG ("Engpässe")  
ENGAN ("Aufträge in Engpässen")

ohne Spezifikationen,

ferner das Kommando PARAUS ("Systemlaufparameter ausgeben") mit den Spezifikationen

SBTAB (Speicherbedarfsgruppen-Tabelle)  
PFREIG } Platten- bzw. Trommelspeicher, der  
PFREIA } noch für Gespräche bzw. Abschnitte  
TFREIG } zur Verfügung steht  
TFREIA }  
ANR nächste zu vergebende Auftragsnummer  
RRZ Summe der Restrechenzeiten aller noch in der Maschine befindlichen Aufträge (in Stunden)  
SIG Liste der gesetzten Signale

und das Kommando SEND. Mit diesem Kommando machen die Operateure den Konsolbenutzern Mitteilungen.



Durch SEND in Verbindung mit XAS ist es also möglich, von einer Konsole eine Nachricht an eine andere zu leiten. Das Kommando hat den Aufbau

SEND, < Text > , < Konsoladresse >

Eine Liste aller angeschlossenen Konsolen folgt.  
Achtung: Wird dieses Kommando - es könnte sicherlich in bestimmten Fällen von großem Nutzen sein - mißbraucht, so muß es wieder entfernt werden.

Es muß immer eine komplette Kommandofolge auf einmal eingegeben werden.

Beispiel:

□□ØPA:SIT,GSP,,ENG,SITAN,AB,,PARAUS,TFREIG,SBTAB,,  
SEND,PROGRAMM FERTIG,F17-1,F18-1,,BITTE DATEI  
ABMELDEN,F19-1 □ .

Hierdurch wird aufgelistet, wieviele Gespräche sich in welchem Zustand befinden und ob Engpässe bestehen. (Durch das doppelte Komma wird von der Spezifikations-ebene wieder auf Kommandoebene heruntergeschaltet, ENG hat keine Spezifikationen, man bleibt also in Kommandoebene.) Weiter wird aufgezählt, welche Abschnitte sich in welchem Zustand befinden. Der Trommelspeicher, der künftigen Gesprächen noch zur Verfügung steht, und die Speicherbedarfsgruppen-Tabelle wird mitgeteilt und schließlich wird eine Botschaft an zwei, eine andere an einen Empfänger gesandt.

Bei SIT, SITAN sind folgende Zustandssymbole möglich:

DRU (Druckend), PLO (Plottend), SST (Streifenstanzend), KST (Kartenstanzend), BLO (Blockiert), KEI (Konsol-Eingabe), FSR (Ausgabe auf Konsole). Zulässig als Gerätesymbol sind außer den Konsol-adressen die folgenden:

D1-0, D2-0 (Drucker), Z1-0 (Plotter),  
B1,...,B5,B7 (Bandgeräte), A5-0, A8-0 (5-bzw.  
8-Kanal-Streifenstanzer), S1-0 (Kartenstanzer).

4. Gespräche41. Grundform eines Gesprächs411. Gesprächseröffnung

Nach fehlerfreier Eingabe des Kdos XBG (8) (s. 113., 142.) meldet die Maschine die von ihr erzeugte Auftragsnummer (mit der die Ausgabeprotokolle versehen werden), Datum und Uhrzeit der Anmeldung des Gesprächs, Benutzerkennzeichen.

Dann kommt die Aufforderung

GIB KOMMANDOS □ :

Bei einem Fehler im Kdo XBG wird der Auftrag evtl. abgebrochen (z.B. bei falscher Problemnummer) mit Freigabe der Konsole, oder es kommt (z.B. bei syntaktischem Fehler) die Meldung (vgl. 311.):

□□ SAS\*FALSCHES KOMMANDO

Danach muß das (richtige) Kdo neu eingegeben werden.

412. Eingabe von Kommandos

Nach der Aufforderung "GIB KOMMANDOS □:" besteht Eingabeberechtigung, d.h. der Benutzer kann Kdos eingeben. Die Eingabe muß mit "□" beginnen. Soll nur ein einzelnes Kdo eingegeben werden, so ist dieses mit "□." abzuschließen. Der Benutzer muß dann die Meldung der Maschine abwarten

und darf erst nach Aufforderung neue Kdos eingeben. Er kann statt eines einzelnen Kdos auch eine Kdofolge eingeben, in der die Kdos durch "⌘" getrennt sind; das letzte Kdo wird mit "⌘." abgeschlossen. Eine Kdofolge kann auch sekundäre Teilkdos enthalten, jedoch muß dann das zugehörige primäre Teilkdo mit eingegeben werden.

#### 413. Eingabe von Daten in ein Programm

In 231. wurde gezeigt, wie Daten, die im Kdo STARTE als Fremdstring stehen, in FORTRAN oder ALGOL eingelesen werden. In 523. wird die Übergabe von Daten aus einer beliebigen Datei beschrieben. Wählt man in einer Lese-Anweisung als Gerätenummer die "8", ohne einen Dateinamen zuzuordnen, so bewirkt die Anweisung auf der Konsole den Ausdruck "⌘: "; es werden dann die für die Anweisung benötigten Daten von der Konsole erwartet. Die Eingabe der Daten muß mit "⌘." abgeschlossen werden. Statt Daten kann man auch vorrangige Kdos (s. 44.), aber keine Anweisungen (s. 43.) einschieben.

Damit man weiß, was nach "⌘:" erwartet wird, sollte man vor die Lese-Anweisung eine Schreib-Anweisung einfügen (mit Gerätenummer 9), die einen erläuternden Text ausdrückt, z.B. "GIB X EIN".

#### 444. Eingriffe in ein Gespräch

Man kann zu jedem Zeitpunkt eines Gesprächs (also nicht nur während der Eingabeberechtigung) mit einem der folgenden Vermittlerkdos eingreifen (s. HKS 8):

- a)  $\square$  XAB  $\square$ . Eine laufende Ausgabe wird abgebrochen, die Maschine meldet sich nach Abarbeiten der schon eingegebenen Kdos wieder mit " $\square$ :". Man weiß dann evtl. nicht, ob eine Fehlermeldung ausgegeben werden sollte, ob vorrangige Kdos oder nur Anweisungen einzugeben sind. Das Kdo ist dann zu verwenden, wenn eine Ausgabe abgebrochen werden soll, ohne daß die Ausführung der zuletzt eingegebenen Kdos unterbrochen wird. (Meldungen vom SAS werden durch das Kdo XAB nicht unterdrückt.)
- b)  $\square$  XAN  $\square$ . Eine laufende Ausgabe wird unterbrochen; der Abwickler meldet sich nach gewissen Auslaufvorgängen (s. HKS 8) mit " $\square \square$  ABW\* $\square$ :". Man kann jetzt Anweisungen (s. 433.) oder vorrangige Kdos (s. 441.) eingeben. Gibt man " $\square$ ." ein, so wird an der Unterbrechungsstelle fortgefahren. Das Kdo XAN wird benötigt, wenn in die Ausführung von bereits eingegebenen Kdos eingegriffen werden soll.

c)  $\square$  XEN  $\square$ . Hiermit wird das Gespräch beendet, d.h. es sind keine Eingaben mehr möglich; es kommt die Meldung

$\square$   $\square$  SAS\*KONSOLE FREI

Eine noch laufende Auftragsbearbeitung wird abgebrochen.

d)  $\square$  XEG  $\square$ . Im Unterschied zu c) wird eine laufende Auftragsbearbeitung nicht abgebrochen, sondern im Abschnittsmodus zu Ende geführt.

#### 415. Zeichenweise Korrekturen in einem Kapitel

Das Kdo TZKØRRIGIERE (6) erlaubt in einem Gespräch zeichenweise (und zeilenweise) Korrekturen in einem Kapitel. Zeilennummer und einzutragende Information werden dazu nicht als Spz-werte, sondern erst auf Anfrage eingegeben. Führende Leerstellen in der Information lassen die entsprechenden Zeichen im alten Text unverändert. Mit " $\square$ ." wird die Korrektur einer Zeile beendet; will man den Rest der Zeile löschen, so muß man vor  $\square$  einen Zeilenwechsel eintippen. Ist eine Zeile länger als 67 Zeichen, so wird sie nach und nach in Teilzeilen von 67 Zeichen ausgegeben, und die Korrekturinformation wird für jede Teilzeile angefordert. Das Kdo TZKØRRIGIERE kann auch zum reinen Eintragen von Zeilen verwendet werden, wenn die Zeilennummern

unter der Spz NUMERIERUNG angegeben werden.

Soll eine Zeichenfolge, die im Kapitel mehrfach vorkommt, auf einmal korrigiert werden, so verwende man die Kdoprozedur ~~TOK~~KORRIGIERE (BO).

## 42. Reaktionen auf Fehlermeldungen

### 421. Normaler Abbruch eines Kommandos

Wird bei der Ausführung eines Kdos ein Fehler gefunden (z.B. ein unzulässiger Spezifikationswert, ein syntaktischer Fehler in einem zu übersetzenden Programm oder eine fehlerhafte Dateneingabe), so wird die Ausführung abgebrochen; es folgt eine Fehlermeldung und die Anfrage nach weiteren Kdos. War das fehlerhafte Kdo einzel oder am Ende einer Kdofolge eingegeben worden, so kommt die Aufforderung "GIB KOMMANDOS  $\square$ :", stand das Kdo innerhalb einer Kdo-folge, so heißt es "GIB KOMMANDOS MIT VORRANG  $\square$ :".

In beiden Fällen können nun Kdos eingegeben werden, im zweiten Fall auch Anweisungen (s. 432.) oder einfach " $\square$ .", d.h. es soll das nächste Kdo der Kdofolge verarbeitet werden. Kdos, die im zweiten Fall eingegeben werden, werden vorrangig behandelt (s. 44.).

Falls ein Kdo wegen Speichermangel nicht ausgeführt werden konnte, kann man nur noch solche Kdos ausführen lassen, die

im betreffenden Speichermedium weniger Platz benötigen, oder man muß (nicht mehr benötigte) Dateien, Montageobjekte oder Operatoren, die in diesem Speicher liegen, löschen.

#### 422. Alarm

Läuft ein Operator auf einen Alarm (z.B. arithmetischen Alarm), so erfolgt (vgl. 241.) "normalerweise" ein Dump. Ist der Operator jedoch gesprächsfähig (Kdo UEBERSETZE (3), VARIANTE=GS oder KV) so kommt statt dessen die Meldung "`<Operatorlaufname> *KE=*ALARM □:`". Gibt man jetzt Kdos ein, so werden diese vorrangig behandelt (s. 44.); gibt man "`□.`" ein, wird der Operatorlauf fortgesetzt (evtl. mit erneuter Alarmmeldung). Im übrigen können Anweisungen eingegeben werden (s. 434.).

#### 423. Weitere Meldungen

In selteneren Fällen (z.B. nach Eingabe von XAN) meldet sich der Abwickler, und zwar mit "`□ □ ABW*□:`"; Weiteres hierzu s. 414.b).

In manchen Fällen, bei denen eine Fortsetzung des Gesprächs sinnlos ist, kommt die Meldung "SCHWERER FEHLER. GIB XEN □:". Erwartete Reaktion: "`□ XEN □.`" (vgl. 414.c)). In einem Gespräch können außerdem die unter 3. genannten Meldungen des SAS und der Operateure erscheinen.

43. Anweisungen431. Allgemeines

Eine "Anweisung" im Sinn der Kommandosprache ist eine bestimmte Zeichenfolge, die an gewissen Stellen eines Gesprächs eingegeben werden kann. Anweisungen werden im Unterschied zu Kdos nicht mit einem Fluchtsymbol eingeleitet und können immer nur einzeln und auf Anforderung eingegeben werden. Eine Anweisung beginnt mit einem Anweisungsnamen, evtl. gefolgt von Parametern, und wird wie jede Eingabe mit "▣." abgeschlossen.

432. Anweisungen nach Fehlern in einer Kommandofolge

Bei einem Fehler innerhalb einer Kdofolge (s. 421.) erscheint die Aufforderung "GIB KOMMANDOS MIT VORRANG ▣:". Statt "▣." kann man auch die Anweisungen "NEIN ▣." oder "WEITER ▣." eingeben, wenn das nächste anstehende Kdo verarbeitet werden soll. Sollen dagegen die restlichen Kdos der Folge übergangen werden, so gebe man die Anweisungen "KLOESCHE ▣." oder "LOESCHE ▣." ein. Die Maschine meldet sich dann mit "GIB KOMMANDOS ▣:" und erwartet nur noch Kdos.

433. Anweisungen an den Abwickler

Nach einer Meldung des Abwicklers mit "▣▣ ABW\*▣:" (s. 414.b) und 423.) kann eine der folgenden Anweisungen



eingegeben werden:

- a) BEENDE  $\square$ . Eine noch laufende Ausführung eines Kdos wird abgebrochen, noch nicht ausgeführte Kdos werden übersprungen, es folgt "GIB KOMMANDOS  $\square$ :".
- b) HALT  $\square$ . Ein bereits angefangenes Kdo wird abgearbeitet; vor Abarbeitung des nächsten Kdos kommt die Anforderung von vorrangigen Kdos oder Anweisungen.
- c) HALT, <Operatorlaufname>  $\square$ .  
Der betreffende Operatorlauf wird angehalten, Weiteres s. 434.

#### 434. Anweisungen nach Anhalten eines Operatorlaufs

Man kann Anweisungen eingeben, wenn ein Operatorlauf durch eines der folgenden Ereignisse angehalten wird:

- a) Alarm (s. 422.),  
b) Kontrollereignis (s. 451.),  
c) Anweisung an den Abwickler (s. 433. c)).

(Wird ein Operatorlauf angehalten, weil Daten von der Konsole erwartet werden (s. 413.), so kann keine Anweisung

eingegeben werden.)

Soll der Operatorlauf fortgesetzt werden, so gebe man "N." ein. Der Operatorlauf kann aber auch beendet werden mit einer der beiden folgenden Anweisungen:

ØPSTØFN. beendet den Operatorlauf ohne Ausgabe von Dumps.

ØPABRRUCH N. beendet den Operatorlauf mit Ausgabe der im Kdo STARTE (3), Spz DUMP, vorgesehenen Dumps; der Operatorlauf muß aber dumpfähig sein (Kdo UEBERSETZE (3), VARIANTE = D oder GS).

Bevor ein Operatorlauf fortgesetzt oder beendet wird, können noch vorrangige Kdos (s. 44.) oder Anweisungen für Testzwecke eingegeben werden (s. 45.).

#### 44. Vorrangige Kommandos

##### 441. Übergang von der Grundstufe zur Vorrangstufe

Kdos, die zu Beginn eines Gesprächs eingegeben werden, werden in der "Grundstufe" bearbeitet. Es gibt nun verschiedene Gelegenheiten, wo Kdos vorrangig, d.h. in der "Vorrangstufe" bearbeitet werden:

- a) bei Anfrage nach Daten von der Konsole (s. 413.),
- b) nach einem Fehler innerhalb einer Kdofolge (s. 421.),
- c) bei einer Anfrage des Abwicklers (s. 414. b)),

- d) beim Anhalten eines Operatorlaufs (s. 434.),
- e) durch Eingabe mit CALL KØMMDØ (S. 445.).

#### 442. Normale Gesprächsform in der Vorrangstufe

Ist ein in der Vorrangstufe eingegebenes Kdo (oder eine Kdofolge) abgearbeitet worden, so kommt die Anfrage: "WEITERE KOMMANDOS MIT VORRANG ⚡:". Gibt man jetzt weitere Kdos ein, so werden diese ebenfalls in der Vorrangstufe behandelt. Möchte man jedoch keine weiteren Kdos mehr einfügen, sondern in die Grundstufe zurückkehren und an der Unterbrechungsstelle fortfahren, gibt man die leere Antwort "⚡." oder die Anweisung "NEIN ⚡." oder "WEITER ⚡." (vgl. 432.). Je nachdem, wie die Vorrangstufe begonnen wurde (vgl. 441.), erfolgt dann

- a) die Anfrage nach Daten,
- b) die Bearbeitung des nächsten Kdos in der Grundstufe,
- c) Fortsetzung an der unterbrochenen Stelle,
- d) Anfrage nach Anweisungen (s. 434.).

#### 443. Fehlermeldungen in der Vorrangstufe

Tritt in einem einzelnen eingegebenen vorrangigen Kdo ein Fehler auf, so kommt die Anfrage "WEITERE KOMMANDOS MIT VORRANG ⚡:". Man kann jetzt so verfahren wie bei derselben

Anfrage, wenn das letzte Kdo abgearbeitet wurde (s. 442.). Tritt jedoch innerhalb einer vorrangigen Kdofolge ein Fehler auf, so erscheint die Meldung "KEIN KOMMANDO EINFUEGBAR, GIB ANWEISUNG  $\mu$ :". Jetzt ist nur die Eingabe einer Anweisung möglich (deshalb empfiehlt es sich, in der Vorrangstufe von vornherein nur einzelne Kdos einzugeben). Es können die unter 432. genannten Anweisungen (oder die Antwort " $\mu$ .") eingegeben werden; die Maschine meldet sich (ggf. nach Abarbeiten der begonnenen Kdofolge) mit der Anfrage nach weiteren Kdos mit Vorrang.

#### 444. Sonstige Anfragen in der Vorrangstufe

Allgemein kann man bei einer Anfrage in der Vorrangstufe keine Kdos eingeben, wenn bei derselben Anfrage in der Grundstufe ein Übergang von der Grundstufe in die Vorrangstufe erfolgen würde. Dies ist der Fall bei

- a) der Anfrage nach Daten: jetzt sind wirklich Daten einzugeben,
- b) Fehler in einer Kdofolge (s. 443.),
- c) einer Anfrage des Abwicklers; außer " $\mu$ ." können die in 433. besprochenen Anweisungen eingegeben werden; es kommt die Anfrage nach weiteren vorrangigen Kdos oder nach Anweisungen,
- d) Anhalten eines Operatorlaufs (s. 434.).

445. Eingabe vorrangiger Kommandos  
in einem FORTRAN - Programm

Die Anweisung CALL KØMMDØ (' <Kommando> ' [ ,&n] ) bewirkt, daß das (ohne Fluchtsymbol) eingetragene Kdo vorrangig ausgeführt wird; nach Ausführung wird normal fortgefahren. Treten so grobe Fehler auf, daß das Kdo nicht ausgeführt werden kann, so erfolgt ggf. Rücksprung auf die Anweisung mit der Nr. n.

45. Testhilfen

451. Kontrollereignisse

Mit Hilfe eines Kontrollereignisses kann man einen Operatorlauf an einer bestimmten Stelle anhalten, um Anweisungen oder vorrangige Kdos einzugeben; danach kann der Operatorlauf abgebrochen oder fortgesetzt werden. Ein Kontrollereignis ist die erste Anweisung einer bestimmten Zeile des Quellprogramms. In einem Programm dürfen bis zu 50 Kontrollereignisse festgelegt werden. Im einzelnen sind folgende Schritte nötig:

- a) Im Kdo UEBERSETZE (3) ist VARIANTE = KV oder GS zu setzen, und unter der Spz KE müssen die Kontrollereignisse definiert werden, indem den gewünschten Zeilen geeignete Bezeichnungen zugeordnet werden (max. 6stellige Zahlen oder FORTRAN-Namen).

- b) Die Kontrollereignisse müssen aktiviert werden. Dies geschieht entweder im Kdo STARTE (3) in der Spz AKTIV oder mit der Anweisung KEAKTIV. Diese Anweisung kann nach Anhalten des Operatorlaufs (s. 434.) eingegeben werden, z.B. wenn vor Erreichen des zu aktivierenden Kontrollereignisses ein Alarm aufgetreten ist. Die Anweisung hat die Form:

KEAKTIV (KE1, KE2, ...) □.

Dabei sind für KE1, KE2, ... die Namen der zu aktivierenden Kontrollereignisse einzusetzen, evtl. gefolgt von " -n" ( $1 \leq n \leq 65536$ ), z.B.:

KEAKTIV (ALPHA - 10, BETA, GAMMA - 2) □.

Bevor ein aktiviertes Kontrollereignis zum n-ten Mal (fehlendes n bedeutet 1) ausgeführt wird, so wird der Operatorlauf angehalten, und die Maschine meldet sich mit

"⟨Operatorlaufname⟩ \* KE = ⟨KE-Bezeichnung⟩ □:"

Jetzt können Anweisungen und evtl. vorrangige Kdos eingegeben werden (s. 434.).

Mit der Anweisung "KEPASSIV (KE1, KE2, ...) ." können die Kontrollereignisse KE1, KE2, ... passiviert werden, d.h. vom n-ten Durchlauf ab wird bei ihnen der Operatorlauf nicht mehr angehalten.

Nach Eingabe von KEAKTIV oder KEPASSIV kommt die Anfrage nach weiteren Anweisungen oder vorrangigen Kdos; mit "Y." wird der Operatorlauf fortgesetzt.

Weiß man bereits bei der Eingabe des Kdos STARTE (3), welche Anweisungen nach Erreichen eines bestimmten Kontrollereignisses einzugeben sind, so kann man diese Anweisungen gleich unter der Spz KONTROLLE angeben (ohne abschließendes "Y."). Bei Erreichen dieses Kontrollereignisses werden dann diese Anweisungen sofort ausgeführt, sofern es aktiviert ist. (Dies gilt auch bei einem Operatorlauf im Abschnittsmodus; jedoch nur für Anweisungen, die keine Ein-/Ausgabe auf der Konsole verlangen; fehlt die Angabe unter Spz KONTROLLE, so wird das Kontrollereignis wie ein passives behandelt.)

#### 452. Dumps

Bei einem nicht gesprächsfähigen Operator kann ein Dump durch einen Fehler im Operatorlauf bewirkt werden (s. 241. und 422.). Man kann aber auch nach Anhalten eines Operatorlaufs (s. 434.), also z.B. nach Alarm beim Lauf eines gesprächsfähigen Operators, durch Anweisungen einen Dump verlangen, sofern der Operator dumpfähig ist (Kdo UEBERSETZE (3), VARIANTE=D oder GS). Außer ØPABBRUCH (s. 434.) gibt es noch die Anweisungen:

a) KDUMPE  $\square$ . Es werden die Dumps ausgeführt, die im Kdo STARTE (3) in der Spz DUMP angegeben wurden.

b) KDUMPE (< Dump-Angabe > )  $\square$ .

Es werden Dumps entsprechend der Dump-Angabe ausgeführt; diese Angabe hat dieselbe Form wie sonst im Kdo STARTE (3), Spz DUMP.

Nach Eingabe einer dieser Anweisungen kommt die Anfrage nach weiteren Anweisungen oder vorrangigen Kdos; mit " $\square$ ." wird der Operatorlauf fortgesetzt.

#### 453. Überwacher

Der Überwacher wird gemäß 243. im Kdo UEBERSETZE (3), Spz TRACE vereinbart und wird dann mit dem Kdo STARTE (3) automatisch eingeschaltet. Man kann nun nach Anhalten des Operatorlaufs (s. 434.) durch Eingabe von Anweisungen den Überwacher ganz oder teilweise aus- und einschalten. Einschalten des gesamten vereinbarten Überwachers erfolgt mit TRACEEIN  $\square$ . bzw. KTRACEEIN  $\square$ ., wenn die Protokollierung nur im Ablaufprotokoll bzw. im Ablauf- und Konsolprotokoll vorgenommen werden soll; Ausschalten erfolgt entsprechend mit TRACEAUS  $\square$ . bzw. KTRACEAUS  $\square$ .

Will man nur die Überwachung einer bestimmten Art ein- oder ausschalten, so ist dies hinter dem Anweisungsnamen in ( )



anzugeben; z.B.

TRACEEIN (GØTØ)M .    oder    KTRACEAUS (ASSIGN,IF)M .

Im 1. Beispiel wird der Überwacher nur für Sprünge eingeschaltet, im 2. Beispiel wird die Protokollierung von Wertzuweisungen und von logischen IF-Anweisungen auf der Konsole ausgeschaltet.

Das Backtracing (s. 243.) wird durch BTRACEEIN M . bzw. BTRACEAUS M . ein- bzw. ausgeschaltet. Die Anweisung BTRACEM . bewirkt eine einmalige Ausgabe der letzten 20 überwachten Schritte; durch den Zusatz (KØ) erfolgt die Ausgabe auch auf der Konsole. Anschließend kommt die Anfrage nach weiteren Anweisungen oder vorrangigen Kdos; mit "M ." wird der Operatorlauf fortgesetzt.

#### 454. Bringen und Setzen von Variablen

Die Ausgabe spezieller Werte von Variablen ist nicht nur durch einen Dump möglich (s. 452.), sondern auch (ohne Rückverfolgungsliste) mit der Anweisung

BRINGE (<Montageobjekt> (<Variable> , <Variable>,...)) M .

Diese Anweisung kann nach Anhalten eines dumpfähigen Operators, der aus dem angegebenen Montageobjekt (unter andern) montiert wurde, eingegeben werden. Es werden dann

die Werte der angegebenen Variablen des Montageobjekts wie bei einem Dump ausgedruckt. Gibt man keine Variablen an, so werden die Werte aller Variablen des Montageobjekts ausgedruckt.

Mit der Anweisung

```
SETZE (<Montageobjekt> (<Variable> = <Konstante> ,...)) ▯.
```

können Variablen neue Werte erhalten.

(Näheres über Bringen und Setzen s. HKS (6), 2.3.2.9.)

Nach Ausführung von BRINGE oder SETZE kommt die Anfrage nach weiteren Anweisungen oder vorrangigen Kdos; mit "▯." wird der Operatorlauf fortgesetzt.

## 5. Dateien

### 51. Grundbegriffe

#### 511. Interne und externe Dateien

Eine Datei ist eine organisierte Sammlung von Daten auf einem "Träger": Trommel, Platte, LFD oder Magnetband. Man unterscheidet:

- a) Interne Dateien (Scratch-Dateien): Sie können auf Trommel oder Platte angelegt werden und bleiben dort nur bis zum Ende eines Abschnitts oder Gesprächs erhalten. (Mit Hilfe der FORTRAN-Subroutine SPEDA läßt sich auch ein Feld wie eine Datei im Kernspeicher bearbeiten.)
  
- b) Externe Dateien: Diese bleiben entweder in der "Langfristigen Datenhaltung" (LFD) (in einem Plattenspeicher) oder auf Magnetband. Im folgenden sei unter einer LFD-Datei stets eine Datei in der LFD verstanden. Es gibt kurz-, mittel- und langfristige LFD-Dateien. Eine kurz- bzw. mittelfristige Datei wird am Ende eines Tages bzw. einer Woche gelöscht; eine langfristige Datei bleibt erhalten, bis sie vom Benutzer gelöscht wird. LFD-Dateien können im Gegensatz zu Magnetbändern in Gesprächen verwendet werden. Eine LFD-Datei läßt sich als Privatdatei, die nur für den einzelnen Benutzer bestimmt ist, oder als Gemeinschaftsdatei für mehrere Benutzer anlegen.

### 512. Struktur einer Datei

Die "Struktur" einer Datei gibt an, in welcher Form die Datei mit Daten belegt werden soll, und nicht, wie die Datei tatsächlich belegt ist. Die Daten werden "satzweise" abgelegt; ein Satz ist also eine gewisse Menge von Daten. Der Typ der Datei bezeichnet die Art des Zugriffs zu diesen Sätzen:

SEQ = sequentieller Zugriff, d.h. die Sätze werden hintereinander eingetragen und können nur unmittelbar hintereinander verarbeitet werden. Diese Art des Zugriffs ist typisch für die normale Ein- und Ausgabe in FORTRAN und ALGOL und für Magnetbandverkehr.

PHYS = sequentieller Zugriff, wobei dem Begriff (logischer) Satz genau ein (physikalischer) Block entspricht.

RAN = Random-Zugriff über Satznummer

RAM = Random-Zugriff über Satzmarke

Bei Random-Zugriff können Sätze an beliebiger Stelle einer Datei eingetragen oder aufgerufen werden. Der Unterschied zwischen RAN und RAM besteht im wesentlichen darin, daß eine RAN-Datei durchgehend numeriert ist (also auch Speicherplatz für leere Sätze belegt), während die RAM-Datei nur den wirklich benötigten Speicherplatz belegt; dafür ist bei RAM der Zugriff komplizierter organisiert als bei RAN. Quellprogramme für das Kdo UEBERSETZE (3) müssen (falls sie nicht als Fremdstring eingegeben werden) in einer RAN- oder RAM-Datei stehen.

Die Satzzahl gibt an, wieviel Sätze genau, maximal oder ungefähr einzugeben sind.

Der Satzbau gibt an, welche Satzelemente (Oktaden, Ausgabezeichen, Ganz- oder Viertelwörter) und wieviel ein einzugebender Satz genau, maximal oder ungefähr haben soll ("Oktade" bedeutet beliebiges Zeichen, das durch eine Oktade im Zentralcode verschlüsselt ist; bei "Ausgabezeichen" muß dagegen das 1. Zeichen jedes Satzes ein Vorschubzeichen sein, s. 232.).

### 513. Datenbasen und Kataloge

Jede Datei, mit der gearbeitet werden soll, muß in einer Datenbasis angemeldet sein; d.h. man kann eine Datenbasis als eine Zusammenfassung von Dateien betrachten, wie sie von der Maschine oder vom Benutzer vorgegeben wird. Zu Beginn jedes Auftrags wird automatisch eine auftrags-spezifische Standarddatenbasis (STDDB) angelegt; der Benutzer kann außerdem mit dem Kdo DATENBASIS (4) bis zu 6 weitere Datenbasen kreieren. Alle diese Datenbasen werden bei Auftragsende gelöscht, können aber mit dem Kdo LÖESCHE (3) auch schon vorher gelöscht werden. Im Gegensatz dazu ist die öffentliche Datenbasis (ØEFDB) nicht auftrags-spezifisch; sie enthält die Montageobjekte und Programme, die allen Benutzern zur Verfügung stehen.

LFD-Dateien sind durch Benutzerkennzeichen (BKZ) zusammengefaßt. Für kurz- bzw. mittelfristige LFD-Dateien lautet das BKZ stets KFD bzw. MFD. BKZ und Datenbasis werden als Kataloge bezeichnet. Ein "Arbeitskatalog" ist ein Katalog, in dem eine Datei bearbeitet wird, d.h. er ist stets eine Datenbasis; das BKZ einer LFD-Datei ist dagegen ein "Verwaltungskatalog". Bei internen Dateien sind Arbeits- und Verwaltungskatalog identisch.

#### 514. Dateibezeichnung

Die volle Bezeichnung einer Datei hat die Form:

⟨Katalogname⟩ .N(G.V) - ⟨Paßwort⟩

Der "eigentliche" Name der Datei, der bei jedem Aufruf der Datei benötigt wird, ist N (Name von Standardlänge, s. 124.). Die Generations-Versions-Nummer (G.V) wird nur dann explizit gebraucht, wenn mehrere gleichnamige Dateien im selben Katalog stehen. Der Katalogname ist ein FORTRAN-Name (s. 124.) und bezeichnet den Katalog, in dem die Datei angemeldet ist. Das Paßwort (maximal 6 Zeichen) ist nur bei externen Dateien sinnvoll. Näheres s. beim Kdo DATEI (4), Spz NAME.

## 52. Dateiverkehr allgemein

### 521. Kreation und Anmelden einer Datei

Eine Datei wird kreierte (d.h. angelegt oder eröffnet) mit dem Kdo DATEI (4). Dieses Kdo bestimmt Bezeichnung, Struktur und Träger der Datei. (Dateikenndaten, die noch nicht besetzt werden, lassen sich in einem FORTRAN-Programm mit der Subroutine FIDE nachbesetzen.) Für Dateien auf Magnetband können noch Angaben zu Blockung, Verfallsdaten und Folgeträger gemacht werden; für Gemeinschaftsdateien ist eine Spz ZEIT vorhanden.

Soll mit einer externen Datei, die in einem früheren Auftrag kreierte wurde, gearbeitet werden, so ist diese mit dem Kdo EINSCHLEUSE (4) einzuschleusen; sie wird dann in einer Datenbasis angemeldet. Statt dessen kann man mit VERLAGERE (4) die Datei auf Platte oder Trommel verlagern (wobei die externe Datei noch erhalten bleibt, aber nicht mehr zugänglich ist) und dort wie eine andere interne Datei behandeln.

### 522. Eintragen von Information

Bei der Eingabe von Daten in eine Datei wird die Länge jedes Satzes festgelegt; falls Satzlänge und Satzzahl nicht der Struktur der Datei entsprechen, wird die Eingabe mit einer Fehlermeldung abgebrochen. Es gibt folgende Möglichkeiten der Eingabe:

- a) Mit dem Kdo EINTRAGE (4): als Fremdstring oder aus einer anderen Datei.

Daten, die schon vor Beginn des eintragenden Kdos in der Datei stehen, werden gelöscht.

- b) In einem FORTRAN-Programm mit einer WRITE-Anweisung; im Kdo STARTE (3) muß in der Spz DATEI die Geräte-Nr. der Dateibezeichnung zugeordnet werden. Falls die Datei nicht vom Typ SEQ ist, muß sie mit einer DEFINE-FILE-Anweisung im selben Programm definiert worden sein, und in der WRITE-Anweisung muß ein Parameter für direkten Zugriff stehen, der einen definierten momentanen Wert hat. In diesem Fall wird bereits eingetragene Information in der Datei nur gelöscht, wenn sie in dem neu zu beschreibenden Satz steht; bei einer SEQ-Datei wird beim Beschreiben alle vor dem Programmablauf vorhandene Information gelöscht.

Die Daten können formatfrei (Satzelemente: Ganzwörter) oder formatgebunden (Satzelemente: Oktaden oder Ausgabebzeichen) eingetragen werden. Mit jeder WRITE-Anweisung wird ein neuer Satz begonnen. Bei formatgebundener Eingabe können mit einem WRITE mehrere Sätze eingegeben werden. Für den Verkehr mit einer SEQ-Datei sind noch die Anweisungen REWIND, BACKSPACE und ENDFILE zu verwenden.



c) In einem ALGOL-Programm:

- I) für SEQ-Dateien die IFIP-Prozeduren ØUTSYMBOL, ØUTREAL, ØUTARRAY, die Knuth-Prozeduren ØUTPUT, ØUTLIST, sowie PUTS,
- II) für RAM/RAN-Dateien die Prozedur PUT,
- III) für Dateien beliebigen Typs die Prozedur ØUT.

Im Kdo STARTE (3) muß in der Spz DATEI eine Zuordnung Geräte-Nr.-Dateibezeichnung stehen. Weitere Prozeduren für den Dateiverkehr: REWIND, BACKSPACE, CLOSØ.

- d) Mit dem Kdo BINAERAUS (4) lassen sich Informationen in verschlüsseltem Binärcode in eine Datei eintragen (Satzbau: mindestens 18 Ganzworte).

### 523. Ausgabe von Information

Die Ausgabe von Daten aus einer Datei kann auf folgende Weisen geschehen:

- a) Mit den Kdos DRUCKE (4) oder STANZE (4) als Teilauftrag (s. 116.). Die Satzelemente müssen Oktaden oder Ausgabezeichen sein; bei binärer Ausgabe dagegen Ganz-

wörter. Sind es Ausgabezeichen, so wird das erste Zeichen eines Satzes nicht ausgegeben, sondern als Vorschubzeichen verwendet (s. 232.). Mit DRUCKE kann die Ausgabe auch auf Konsole erfolgen, jedoch ist das Druckbild dem Schnelldrucker angepaßt.

- b) In einem FORTRAN-Programm können die Daten mit READ gelesen (und dann weiterverarbeitet, z.B. ausgedruckt) werden. Es gelten die Ausführungen von 522.b) mit READ statt WRITE.
- c) In ALGOL stehen zur Verfügung:
  - I) für SEQ-Dateien die IFIP-Prozeduren INSYMBOL, INREAL, INARRAY, die Knuth-Prozeduren INPUT, INLIST,
  - II) für RAN/RAM-Dateien die Prozedur GET,
  - III) für Dateien beliebigen Typs die Prozeduren GETS und IN.

Im übrigen vgl. 522.c).

- d) Mit dem Kdo BINAERAUS (4) können beliebige Dateien samt Kenndaten gestanzt werden. (Bei binärer Ausgabe gemäß a) wird nur der Inhalt einer Datei ausgegeben.) Mit BINAEREIN (4) kann der binär verschlüsselte Inhalt von Dateien einer weiteren Bearbeitung zugänglich gemacht werden.

e) Mit dem Kdo KØPIERE (BO) läßt sich der Inhalt einer Datei ins Ablauf- oder Konsolprotokoll ausgeben; das Druckbild kann der Konsole angepaßt werden.

#### 524. Weitere Kommandos für beliebige Dateien

Mit den Kdos MISCHE (4), SØRTIERE (4) können die Sätze einer Datei nach gewissen Kriterien gemischt bzw. sortiert werden. Man kann (aus Sicherheitsgründen) eine Datei mit einer Schreibsperre versehen (Kdo SPERRE (4)); mit Kdo LØESE (4) wird die Sperre aufgehoben.

Mit dem Kdo LØESCHE (3) kann sowohl eine Datei, die nicht auf Magnetband steht, als auch eine Datenbasis gelöscht werden. Mit den Kdos DRUCKE (4) und STANZE (4), soweit keine Kopie verlangt wird, kann eine interne Datei ebenfalls gelöscht werden. Interne Dateien, die nicht gelöscht werden, bleiben bis zum Ende des Abschnitts bzw. Gesprächs erhalten.

#### 525. Spezielle Kommandos für LFD-Dateien

Eine LFD-Datei wird mit dem Kdo DATEI (4) kreiert (TRAEGER=LFD) und mit EINSCHLEUSE (4) angemeldet (vgl. 521.); stattdessen können z.Z. noch die Kdos LFDATTEI (5) und LFANMELDE (5) verwendet werden. Das Einschleusen zum Lesen bzw. Schreiben wird am einfachsten durch die Kdo-

prozeduren EL bzw. ES (s. Anhang III) angefordert. Zum Lesen werden von einem etwaigen Paßwort nur die auf das 3. Zeichen folgenden Zeichen benötigt, das Lesepaßwort. Der Schreibzugriff für eine Gemeinschaftsdatei kann auf eine bestimmte Tageszeit beschränkt werden (Kdo DATEI (4), Spz ZEIT). Ist eine solche Datei zum Schreiben angemeldet, so kann sie in keinem anderen Auftrag angemeldet werden. In eine zum Schreiben angemeldete Datei können Informationen nicht nur eingetragen werden (s. 522.), sondern auch verlagert werden (Kdo VERLAGERE (4)).

Eine LFD-Datei kann man mit LFABMELDE (5) (in Zukunft mit ABMELDE) abmelden, z.B. um eine Gemeinschaftsdatei zum Schreibzugriff freizugeben. Die Datei wird dadurch nicht gelöscht; zum Löschen verwende man das Kdo ~~L~~ÖSCHE (3) (z.Z. kann auch noch L~~F~~ÖSCHE (5) verwendet werden).

Nachträgliche Veränderungen der Satzzahl sind mit dem Kdo LFRESERVIERE (5) (in Zukunft mit RESERVIERE) möglich.

Mit dem Kdo LFINF~~O~~RMIERE (5) kann man sich informieren über

- a) den Dateienbestand unter einem bestimmten BKZ (Paßwörter werden nicht ausgedruckt; Privatdateien können nur unter dem auftragsspezifischen BKZ erfragt werden),
- b) den Zustand einzelner Dateien (ein etwaiges Paßwort muß angegeben werden).

53. Texthaltung531. Eigenschaften eines Kapitels

Ein Kapitel ist eine kurzfristige Datei vom Typ RAM mit dem Satzbau U 80 Ø, Träger ist Platte. Ein Kapitel wird mit dem Kdo TDEKLARIERE (4) kreiert; es wird wie eine Datei bezeichnet, aber ohne Paßwort. (Man kann ein Kapitel auch mit dem Kdo DATEI (4) oder mit LFDATEI (5) deklarieren, wenn man Typ = RAM, SATZBAU = U 80 Ø vorgibt.) Mit einem Kapitel lassen sich dieselben Manipulationen (durch Kdos oder Programme) vornehmen wie mit jeder anderen RAM-Datei, deren Satzelemente Oktaden sind; darüberhinaus kann man jedoch die Kdos für die Texthaltung verwenden (diese Kdos beginnen mit T, s. HKS4); diese Kdos haben u.a. den Vorteil, daß sie eine ausführliche Protokollierung auf der Konsole erlauben (PRØT.=KØ).

Allgemeiner läßt sich jede Datei vom Typ RAM oder RAN wie ein Kapitel behandeln, wenn ein Satz nicht mehr als 160 Oktaden enthält.

532. Eintragen von Information

Es gibt zunächst die in 522. genannten Möglichkeiten:

- a) mit Kdo EINTRAGE (4);
- b) in einem FORTRAN-Programm mit WRITE formatgebunden; das Kapitel muß mit DEFINEFILE im Programm zusätzlich deklariert sein;

- c) in einem ALGOL-Programm mit PUT;
- (d) mit BINAERAUS geht nicht, da die Satzelemente keine Ganzworte sind).

Bessere Möglichkeiten, Information in vorgegebene "Zeilen" des Kapitels einzutragen, bieten die Kdos TEINTRAGE (falls die Information als Fremdstring vorliegt), TKAPEINFUEGE, TMISCHE, TKØPIERE (falls die Information in einem Kapitel steht). Durch diese Kdos werden Daten, die schon im Kapitel standen, nur auf Wunsch gelöscht. In einem Gespräch sind mit TZKØRRIGIERE auch zeichenweise Korrekturen möglich, ebenso zeilenweise Eintragungen (s. 415.).

### 523. Ausgabe von Information

Es gibt gemäß 523. folgende Möglichkeiten:

- a) DRUCKE (4) oder STANZE (4);
- b) in FORTRAN: mit READ formatgebunden; beachte DEFINEFILE;
- c) in ALGOL mit GET;
- d) mit BINAERAUS (4) kann ein Kapitel binär verschlüsselt ausgegeben werden, um später mit BINAEREIN (4) wieder zugänglich gemacht zu werden;
- e) KØPIERE (BO).

Von den Kdos der Texthaltung kommt für die Ausgabe vor allem TKØPIERE in Frage (für sämtliche Ausgabegeräte außer dem

3-Kanal-Streifenstanzer); außerdem können bei den übrigen Texthaltungs-Kdos ausführliche Protokolle auf Schnelldrucker oder Konsole verlangt werden; ein solches Protokoll umfaßt evtl. das ganze Kapitel, d.h. der Inhalt des Kapitels wird dann ins Ablauf- oder Konsolprotokoll ausgegeben.

#### 534. Weitere Manipulationen

Auf Kapitel lassen sich (vgl. 524.) die Kdos MISCHE (4), SØRTIERE (4), SPERRE (4), LØESE (4), LØESCHE (3) anwenden; mit DRUCKE (4) und STANZE (4) kann ein Kapitel auch gelöscht werden.

Außer den bereits aufgeführten Texthaltungs-Kdos stehen noch zur Verfügung: TZLØESCHE, TVERTAUSCHE (zum Löschen bzw. Vertauschen einzelner Zeilen), TMISCHE (Mischen anhand von Zeilennummern), TNUMERIERE (Numerieren der Zeilen).

### 54. Bibliotheken

#### 541. Allgemeines

Eine Bibliothek ist eine organisierte Sammlung von Montageobjekten und Operatorkörpern. In der ØEFD steht die Bibliothek, über die jeder Benutzer ohne besondere Anmeldung verfügen kann. Eine Benutzerbibliothek steht

in der LFD unter einem BKZ; dieses BKZ dient als Name der Bibliothek. Die Bibliothek belegt 5 Dateien (&L, &L1, &MO, &M1, &M2) und beansprucht mindestens 17K; diese Werte werden auf die Speicherberechtigung des betreffenden BKZ angerechnet. Eine Benutzerbibliothek (im folgenden kurz als Bibliothek bezeichnet) muß zur Verwendung mit dem BKZ angemeldet werden; sie kann paßwortgeschützt sein.

#### 542. Kreation

Für jeden Auftrag werden in der STDDDB automatisch 5 Dateien eingerichtet, in denen - als Ergebnis von Übersetzungen und Montagen - Montageobjekte und Operatorkörper eingetragen werden. Diese auftragspezifische Bibliothek kann mit dem Kdo BIBVERLAGERE (3) oder mit der Kdoprozedur BA (Anhang III) in die LFD kopiert werden. Wenn bereits eine Bibliothek unter demselben BKZ existiert, und nicht paßwortgeschützt ist, werden die in ihr stehenden Montageobjekte bzw. Operatorkörper überschrieben, falls die neue Bibliothek solche enthält. Wenn unter dem BKZ noch keine Bibliothek existiert, wird eine angelegt (falls die Speicherberechtigung ausreicht).

#### 543. Verkehr mit einer Bibliothek

Will man eine Bibliothek verwenden, so muß man sie mit BIBANMELDE (3) anmelden. Hierbei wird die Hierarchie fest-



gelegt, nach der beim Montieren oder Starten verfahren wird (vgl. 222., 223.). Um Änderungen an der Bibliothek vorzunehmen, verlagere man sie mit BE (Anhang III) in die STDDDB, erzeuge neue Montageobjekte oder Operatorkörper und verlagere dann mit BA zurück in die LFD. Gelöscht wird eine Bibliothek mit

◇ BIBVERL., <BKZ> , LFD, LFD

Abgemeldet werden Bibliotheken automatisch mit Auftragsende oder durch Auslassen beim Neudefinieren der Hierarchie.

## 55. Magnetbänder

### 551. Eigenschaften von Magnetbanddateien

Auf Magnetband können Dateien beliebiger Struktur gespeichert werden. Falls jedoch eine Datei direkt auf dem Band verarbeitet (z.B. satzweise gelesen) werden soll, muß sie vom Typ SEQ sein.

Allgemein empfiehlt sich für Dateien auf Magnetband zur Sicherung vor Fremdzugriff die Angabe eines Paßwortes und eines Verfallsdatums (s. DATEI (4), Spz NAME, VERFALL).

Jedes Magnetband, das dem Benutzer auf Antrag zur Verfügung gestellt wird, hat ein Kennzeichen (6stellige Zahl).

Eine Datei kann sich über mehrere Bänder (mit verschiedenen Kennzeichen) erstrecken; umgekehrt kann ein Band mehrere Dateien enthalten. Auf einem Folgeband (d.h. auf einem Band, das eine Fortsetzung einer Datei enthält) darf aber keine neue Datei beginnen. Zusammengehörige Bänder werden auf demselben Bandgerät nacheinander vom Operateur aufgespannt (nach Aufforderung durch die Maschine). Die Anzahl der benötigten Bandgeräte wird im Kdo XBA oder XBG (Spz BGB, s. 142.) angefordert; bei Gesprächen ist ihre Verwendung nur nach Eingabe von XEG möglich.

#### 552. Anlegen einer Datei auf Band

Es gibt zwei Möglichkeiten, eine Datei auf Band zu eröffnen:

- a) Man lege eine Datei sofort auf Band an und schreibe die Daten unmittelbar auf Band. Die Datei wird kreiert mit dem Kdo DATEI (4), TYP = SEQ, TRAEGER = MB (mit Kennzeichen); beachte auch BLÖCKUNG (vgl. 351.). Danach kann die Datei wie jede andere SEQ-Datei beschrieben und gelesen werden (natürlich mit größerem Zeitaufwand); das Band kann so insbesondere als Zwischenspeicher dienen.

- b) Man kreierte zunächst eine interne Datei und trage dort die Daten ein. Anschließend wird die Datei mit dem Kdo SICHERE (4) auf Band gebracht. Mit diesem Kdo kann auch eine LFD-Datei auf Band gesichert werden. (Vgl. das vorletzte Beispiel in HKS 10.)

### 553. Verkehr mit einer Magnetbanddatei

Es gibt analog zu 552. zwei Möglichkeiten, die unter 522. - 524. besprochenen Kdos, Anweisungen und Prozeduren auf eine Magnetbanddatei anzuwenden:

- a) Man verkehrt mit der Datei unmittelbar auf Band. Zu diesem Zweck muß sie mit dem Kdo EINSCHLEUSE (4) in einer Datenbasis angemeldet werden. Geeignet sind hierfür nur SEQ-Dateien; ein noch nicht vergangenes Verfallsdatum verhindert schreibenden Zugriff.
- b) Man verlagere die Datei auf Trommel oder Platte mit dem Kdo VERLAGERE (4). Sie bleibt gleichzeitig auf Band erhalten, ist aber dort nicht zugänglich. Mit dem Kdo SICHERE (4) kann die Datei später wieder auf Band kopiert werden, sofern dies nicht durch ein Verfallsdatum der ursprünglichen Datei verhindert wird. Mit VERLAGERE kann eine Datei von Band noch in die LFD verlagert werden, wenn dort bereits eine LFD-Datei glei-

chen Namens und gleicher Struktur existiert und zum Schreiben angemeldet ist. (Vgl. das letzte Beispiel in HKS 10.)

Eine weitere Möglichkeit, den Dateibestand eines Bandes zu ändern, ist folgende:

Man kopiere das Band mit MBKØPIERE (4) auf ein zweites Band und ändere ggf. den Dateibestand mit den Spz EINFUEGUNGEN und LØSCHUNGEN. —

Bänder von anderen Rechenanlagen müssen erst mit dem Kdo BAND (BO) einer Bearbeitung zugänglich gemacht werden.

Bemerkungen:

- 1) Mit BVERLAGERE (BO) können Informationen, die durch Bandmarken getrennt sind, von solchen Bändern in interne oder externe Dateien verlagert werden, um dort mit normalen Kdos bearbeitet zu werden.
- 2) Bänder, deren Parameter durch BAND definiert werden, lassen sich mit den ALGOL-Prozeduren TGET, TPUT, TFSKIP, TBSKIP, TPØS, TCLØSE bearbeiten (BO.C2.03).

#### 554. Magnetbandfamilien

Mit dem Kdo MBFERØEFFNE (BO) wird in einem Gespräch eine Magnetbandfamilie eröffnet, d.h. eine geordnete Menge von mindestens 2 Magnetbändern, die aufeinander kopiert werden. Zum Kopieren und Ändern dient das Kdo MBFAENDERE (BO), bei dem das jüngste Band der Familie auf das älteste kopiert

wird. Hat die Familie die Nummer n, so ist unter dem internen Namen \*n das Kennzeichen des jüngsten Bandes gespeichert; d.h. man benötigt zum Aufruf des jüngsten Bandes nicht sein Kennzeichen, sondern nur die Nummer der Familie.

6. Sonstiges61. Informationsdienste611. Informieren über Kdos

Das Kdo INFØRMIERE (3) liefert Kurzbeschreibungen und Anwendungsbeispiele für Kdos und deren Spzn. Insbesondere kann man sich über anlagen- und benutzerspezifische Kdos und interne Namen sowie über Voreinstellungen von Spzn informieren. In einem Gespräch erhält man mit diesem Kdo ohne Angabe von Spzn Anfragen der Maschine, welche Informationen sie liefern soll.

612. Informieren über auftragsspezifische Größen

Die folgenden Kdos sind vor allem in Gesprächen von Interesse, wenn der weitere Verlauf des Gesprächs von momentanen Größen abhängt (die der Benutzer evtl. vergessen hat); dabei bedeutet z.B. BINF.: BINFØRMIERE (BO).

BINF.: Spz-Werte des Kdos XBA bzw. XBG  
 DINF.: Dateibestand  
 GINF.: Kennzeichnung des Eingabegeräts  
 HIINF.: aktuelle Hierarchie der Bibliotheken  
 HINF.: freier Hintergrundspeicher  
 MINF.: Montageobjekte einer Bibliothek  
 ØINF.: Operatoren einer Bibliothek  
 RINF.: restliche Rechnerkernzeit  
 ZINF.: Uhrzeit, Auftragsnummer

613. Informieren über Gesamtsituation des Rechnersystems

Das Kdo XAS (8) liefert durch Anfragen an die Kontrollfunktion Informationen über die momentane Situation des Rechnersystems, z.B. über Engpässe oder Anzahl von laufenden Gesprächen (s. 343.). (Während mit diesem Kdo Signale nur abgefragt werden können, lassen sich in FORTRAN- und ALGOL - Programmen Signale für Verzweigungen verwenden , s. 624.).

614. Weitergabe von Nachrichten der Benutzer

Ein Benutzer kann eigene Nachrichten mit XMØ (8) an den TR 440-Operator (s. 342.), mit SEND (s. 343.) an andere Benutzer senden. Mit dem Kdo BRIEF (B0) kann er von der Konsole aus Nachrichten für andere Benutzer in der LFD hinterlegen.

62. Steuergrößen und Sprünge621. Kommandomarken

Jeder Abschnitt besteht aus einer Folge von Kdos. Diese Kdos werden in der angegebenen Reihenfolge nacheinander alle ausgeführt (sofern der Abschnitt nicht vorzeitig, z.B. wegen Speichermangel, abgebrochen wird). Es ist nun manchmal vorteilhaft, Kdos zu überspringen und an einer markierten Stelle des Abschnitts weiterzumachen. Dazu dient die Kdomarke; sie wird formal wie ein Kdo in den Abschnitt eingefügt:     ◇ \* z \*

Dabei ist z eine Folge von 1 bis 6 alphanumerischen Zeichen; falls das 1. Zeichen eine Ziffer ist, dürfen nur noch Ziffern folgen.

Eine Kdomarke dient nur als Sprungziel; wird sie angesprungen (s. 623.), so wird das nächste Kdo im Abschnitt ausgeführt. (Mehrere aufeinanderfolgende Kdomarken sind gestattet.) Die implizit definierte Marke ENDE bedeutet das Ende des Abschnitts.

Verbot: Eine Kdomarke darf nicht innerhalb eines Kdos stehen und nicht vor einem sekundären Tätigkeitskdo (s. 133.). In einem Gespräch dürfen Kdomarken nur innerhalb von Prozedurvereinbarungen auftreten (s. 134.), und die Marke ENDE darf nicht benutzt werden.

### 622. Steuergrößen

Um bedingte Sprünge nach Kdomarken formulieren zu können, benötigt man Steuergrößen, die die Werte TRUE und FALSE annehmen können. Es gibt folgende Arten:

- a) Die Booleschen Variablen BV1, BV2, ..., BV8 lassen sich mit dem Kdo WAHLSCHALTER (3) setzen (auf TRUE) oder löschen (auf FALSE).



- b) Die Wahlschalter WS1, WS2, ..., WS8 lassen sich ebenfalls mit WAHLSCHALTER setzen und löschen, können aber außerdem in ALGOL und FORTRAN mit SETSW (N,B) verändert werden (N = Nr. des Wahlschalters, B = TRUE oder FALSE).
- c) Die Fehlervariablen FE1, FE2 werden automatisch gesetzt, wenn ein Kdo aufgrund eines Fehlers nicht vollständig ausgeführt werden kann. FE1 wird nach jedem fehlerfrei ausgeführten Kdo gelöscht, FE2 nur dann, wenn es im Kdo SPRINGE (3) abgefragt wurde. Mit FE1 bzw. FE2 läßt sich also bequem feststellen, ob ein bestimmtes Kdo bzw. eine Kdofolge fehlerfrei abgearbeitet wurde.
- d) Ein Boolescher Ausdruck entsteht durch Verknüpfung von Booleschen Variablen, Wahlschaltern und Fehlervariablen durch die Zeichen + ("oder") und \*("und").
- e) Die Signale (numeriert von 1 bis 24) sind Steuergrößen, die vom TR 440-Operator gesetzt und gelöscht werden können. Das Signal 10 soll in Zukunft einige Minuten vor Systemlauf-Ende gesetzt werden. Dadurch besteht die Möglichkeit, Zwischenergebnisse in der LFD oder auf Band rechtzeitig abzulegen und den Programmlauf irgendwann wieder fortzusetzen.

623. Sprünge in Kdofolgen

Das Kdo SPRINGE (3) erlaubt Sprünge nach Kdomarken, und zwar sowohl unbedingte Sprünge als auch Sprünge in Abhängigkeit von Booleschen Ausdrücken. Wird in einem Abschnitt als Sprungziel ENDE angegeben, so wird der Abschnitt beendet. Es sind nur Vorwärtssprünge erlaubt. In Gesprächen sind Sprünge nur innerhalb von Kdoprozeduren erlaubt.

Soll nach einem wegen Fehlers abgebrochenen Kdo kein weiteres Kdo ausgeführt werden, so kann mit dem Kdo FEHLERHALT (3) der Abschnitt abgebrochen werden. In der Kdoprozedur RECHNE (3) wird automatisch das Ende der Prozedur angesprungen, wenn beim Übersetzen oder Montieren ein Fehler auftritt (s. 21.).

Durch Verändern der Wahlschalter in einem ALGOL- oder FORTRAN-Programm (s. 622.b) können auch Sprünge in Kdofolgen von Rechenergebnissen abhängig gemacht werden.

624. Sprünge in Programmen

Innerhalb eines ALGOL- oder FORTRAN-Programms können Wahlschalter und Signale abgefragt werden und somit den Programmlauf beeinflussen. Wahlschalter werden abgefragt mit ASKSW (N), Signale mit SIGNAL (N) (N=Nr. des Wahlschalters bzw. Signals). Beides sind Boolesche Prozeduren (in ALGOL)

bzw. logische Funktionen (in FORTRAN); SIGNAL muß im aufrufenden Programm entsprechend vereinbart werden.

Sprünge in Abhängigkeit von einem Signal sind einfacher mit der Subroutine SSIGNL zu programmieren; Beispiel:

```
CALL SSIGNL (13, & 4)
GØTØ 1
4 STØP 'DURCH SIGNAL 13'
```

Mit der ALGOL-Prozedur SYSUNIT läßt sich außerdem abfragen, ob Gesprächszustand herrscht, ebenso mit der logischen Funktion GSP, die in FORTRAN aufgerufen werden kann.

Für Sprünge in Abhängigkeit von der Rechenzeit läßt sich die Prozedur TIME verwenden (s. 244.).

### 63. Speicherung von Kommandos

#### 631. Speicherung im Gedächtnis

Alle eingegebenen Kdos (mit Ausnahme der Vermittlerkdos) werden vom "Entschlüßler" entschlüsselt; die Voreinstellungen der Spzn und die Kdoprozeduren sind im "Gedächtnis" des Entschlüßlers gespeichert. Voreinstellungen und Prozeduren, die ein Benutzer durch Deklarationskdos eingibt (s. 112. und 134.), bleiben nur während eines Abschnitts oder Gesprächs

erhalten. Man kann aber das so vom Benutzer veränderte Gedächtnis mit dem Kdo GEDAECHTNIS (3) in eine Datei auf Band oder in der LFD eintragen und kann so die Voreinstellungen und Prozeduren über längere Zeit speichern; mit GEDAECHTNIS können sie jederzeit wieder verwendet werden. Auf diese Weise ist auch die langfristige Speicherung von Kdofolgen möglich, nämlich in der Gestalt von Kdoprozeduren.

Zu Beginn jedes Auftrages enthält das Gedächtnis alle anlagenspezifischen Voreinstellungen und Prozeduren; es belegt 6K. Will man (um Speicherplatz zu sparen) das vom Hersteller gelieferte Gedächtnis (3K) ohne die vom Rechenzentrum eingetragenen Prozeduren verwenden, so hole man es aus der LFD-Datei RZ.GEDAE:

```
*****  ◆ EL, RZ.GEDAE ◆ GEDAECHTNIS, GEDAE
```

Will man dagegen einige Kdos nur unkenntlich machen, um andere Kdos unter demselben Namen verwenden zu können, so verwende man das Kdo LØESCHE (3), Spz KØMMANDØ. Der Umfang des Gedächtnisses wird hierdurch nicht verkleinert.

### 632. Speicherung in beliebigen Dateien

Man kann eine Kdofolge in eine beliebige Datei eintragen, wenn man vorher mit XUM die Fluchtsymbol-Erkennung abschaltet (und nachher wieder einschaltet!). Ruft man eine

solche Datei, die etwa den Namen KD1 habe, mit dem Kdo  
 ◇TUE, KD1 auf, so wird die gespeicherte Kdofolge aus-  
 geführt. (Im Gesprächsmodus kommt nach Ausführung die  
 Anfrage nach "weiteren Kdos mit Vorrang"; diese kann  
 mit □. beantwortet werden.)

#### 64. Graphische Ausgabe

Die graphische Ausgabe kann auf dem Plotter oder auf dem  
 Sichtgerät SIG 100 erfolgen.

##### a) Fahrbefehle

Mit dem FORTRAN-Unterprogramm PLOT läßt sich der Nullpunkt  
 einer Zeichnung festlegen, und es lassen sich Fahrbefehle  
 zu vorgegebenen Koordinatenpunkten ausführen. Die Unter-  
 programme SCALE und AXIS sorgen für eine geeignete Ska-  
 lierung und Zeichnung der Achsen sowie für ihre Beschrif-  
 tung. Durch das Unterprogramm LINE werden die Koordinaten-  
 punkte verbunden (nähere Angaben in der Beschreibung zu  
 FOPLOT). Die FUPLLOT-Unterprogramme (siehe Beschreibung von  
 FUPLLOT) eignen sich zum Zeichnen von Funktionen und Funk-  
 tionenscharen, die als vorübersetzte FORTRAN-Unterprogram-  
 me vorliegen. Hiermit kann man durch einen einzigen Aufruf  
 eine Zeichnung der Funktion in einem wählbaren Achsenkreuz  
 bei geeigneter Skalierung und gewünschtem Ausgabeformat  
 (etwa DIN A3, DIN A4 usw.) erreichen, z.B. wird nach  
 anschließendem Schlußaufruf (vgl. b) und c)) durch

```
EXTERNAL SINH
CALL PFUNC (SINH, -2., 2., -61, 40HY = SINH (X),
           40H, 40H)
```

die Funktion  $Y = \text{SINH}(X)$  im Intervall  $[-2,2]$  im DIN A4-Format mit einer linearen Achseneinteilung gezeichnet.

b) Ausgabe auf dem Plotter

Alle Fahrbefehle werden bei der Ausgabe auf dem Plotter in der Standardplotterdatei STDPLD aufbewahrt und durch den Schlußaufruf `CALL PLØT (0.,0.,4)` ausgegeben. Erfolgt im Programm kein Schlußaufruf, so kann der Inhalt der Datei STDPLD durch das Kdo `ZEICHNE (4)` mit der Spz. `DATEI=STDPLD` ausgegeben werden.

c) Ausgabe über das Sichtgerät im Gespräch

Bei der Ausgabe über das Sichtgerät SIG 100 ist vor den `PLØT`-Befehlen (vgl. a)) das Sichtgerät zu initialisieren mit dem Aufruf `CALL PLOSIG(1)`. Danach wird der Benutzer aufgefordert, den Bildschirm für Graphikausgaben frei zu machen und dazu die Funktionstaste F2 zu drücken. Die folgenden Fahrbefehle werden dann in einer Zwischendatei gespeichert. Durch den Aufruf der Prozedur `BILD` wird der Inhalt der Zwischendatei auf dem Bildschirm ausgegeben. Durch das Unterprogramm `LØESIG` wird der Inhalt der Zwischendatei gelöscht und steht dementsprechend nicht mehr zur Verfügung. Mit dem Aufruf `CALL PLØSIG (0)` kann man

vom Sichtgerät wieder auf den Plotter umschalten. Es erfolgt hierauf die Aufforderung, den Bildschirm für Texte freizumachen und die Funktionstaste F3 zu drücken.

#### 65. Codewandlung (vgl. BO.E4.02)

Da bei Codefragen keine einheitliche Normierung existiert, gibt es erhebliche Schwierigkeiten beim Übergang von einer Anlage zur anderen. Am TR 440 ist die Codewandlung vom Fremdcode in den Zentralcode ZC1 (Eingabe) und die Codewandlung von ZC1 in Fremdcode (Ausgabe) möglich. Das Ziel der Codewandlung ist es, der Anlage einen Fremdcode verständlich zu machen. Dazu ist es etwa bei der Eingabe in Fremdcode notwendig, jedem Zeichen die entsprechende Bitkombination in ZC1 zuzuordnen. Dies geschieht mittels einer Codetabelle, die mit dem Kdo CEINTRAGE (BO) erstellt und in der unter der Spez. ZIEL angegebenen Datei abgelegt wird. Zu dem müssen im Kdo CEINTRAGE noch Angaben zu Steuergrößen (z.B. Umschaltzeichen, Anfangs- und Endzeichen und Codewortlänge) gemacht werden.

Mit dem Kdo CODEWANDLE (BO) läßt sich die vorgegebene Information mit Hilfe einer Codetabelle in einen anderen Code umschlüsseln. In der unter der Spez. ZIEL angegebenen Datei wird die gewandelte Information eingetragen. Sätze falscher Länge werden außerdem in die Fehlerdatei übernommen.

### 66. Analog - Digital - Wandlung

Hat ein Benutzer ein den IRIG-Normen (vgl. Beschreibung zu ADWAND) entsprechendes 1/2 Zoll breites Magnetband mit analoger Information bespielt, so kann er diese Analoginformation in digitale umwandeln lassen. Dazu muß er einen Antrag beim RZ einreichen, der mit den nötigen Kenndaten, wie Analogband-Nr., Digitalband-Nr., Rechnertakt (Bandtakt), Blocklänge usw., versehen ist.

Die Wandlung der Analoginformation auf dem Magnetband-Analogspeicher MAS 55 wird mit Hilfe des Programmes ADWAND am TR 86 A vorgenommen, das die gewonnenen Digitalwerte auf einem TR 440 - Band sammelt.

Um die Daten auf dem Digitalband der weiteren Verwendung im TR 440 zugänglich zu machen, müssen die Daten mit dem dafür spezifischen Kdo ADWANDLE (BO) in integer\*2-Zahlen gewandelt werden, die dann in einer unter der Spz ZIEL angegebenen Datei (mit dem Satzbau W!) abgelegt werden.



Anhang IVoreinstellungen und Maximalwerte

(werden ab und zu geändert!)

		Abschnitt	Gespräche
SBG	normal	9	9
	maximal	12	12
KSB	normal	32	32
	maximal	60	60
	Sondermax.*	150	70
TSB	normal	50	50
	maximal	400	100
	Sondermax.*	600	200
PSB	normal	100	100
	maximal	2000	300
	Sondermax.*	3000	2000
RZS	normal	1	1
	maximal	60	10
	Sondermax.*	1000	10
DRS	normal	40	40
	maximal	300	150
	Sondermax.*	150	150
UMF	normal	1200 1)	300
	maximal	22500	3750
BGB	normal	0	0
	maximal	4	4

\* Nur bei Sonderläufen erlaubt, die nicht zu den üblichen Betriebszeiten gefahren werden können.

1) für Konsolabschnitte nur 300

Anhang IISpeicherbedarfsgruppen

SBG	KSB	TSB	PSB	DRS
1	12	20	50	20
2	12	20	200	100
3	18	30	70	20
4	18	30	200	100
5	23	50	70	20
6	23	50	200	100
7	27	50	70	20
8	27	50	200	100
9	32	50	100	40
10	32	50	200	100
11	48	100	140	60
12	48	100	300	150

Anhang III

Anlagenspezifische Voreinstellungen der TR 440 - Anlage Bochum (können auch mit INFØRMIERE (3) erfragt werden)

KOMMANDO	SPEZIFIKATION	ANLAGENSPEZIFISCHE VOREINSTELLUNG
ABMELDE	DATEI	-STD-
BIBANMELDE	TRAEGER	LFD
BINAERAUS	GERAET	-
DATEI	TRAEGER	P
DATEI	TYP	RAM-G
DRPROTOKOLL	ANZAHL	-
DRPROTOKOLL	GERAET	-
DRUCKE	GERAET	DR-DC 1
EINSCHLEUSE	TRAEGER	LFD
LFDATEI	TYP	RAM-G
MONTIERE	MV	-STD-
RECHNE	DUMP	-
RECHNE	SPRACHE	ALG 60
STANZE	GERAET	KS-KC 1
SICHERE	ZIEL	MB
TDEKLARIERE	ZEILZAHL	U1
TNUMERIERE	NUMERIERUNG	(1-999999) (10,10)
UEBERSETZE	MV	-STD-
UEBERSETZE	PROTOKOLL	-
UEBERSETZE	SPRACHE	ALG 60

Kommandoprozeduren:

- \*EL (DATEI) □ EINSCHL., \*DATEI, LFD, , LESEN □ \*\*
- \*ES (DATEI) □ EINSCHL., \*DATEI, LFD, , SCHREIBEN □ \*\*
- \*BA (BIB) □ BIBV., \*BIB, LFD, LFD □ BIBV., \*BIB, -STD-,  
LFD, -STD-, -STD-, A'KO □ \*\*
- \*BE (BIB) □ BIBV., \*BIB, LFD, -STD-, -STD-, -STD-, A'KO □ \*\*

Beispiele für Lochkartenabschnitte

(Für Konsolabschnitte sind nur die Vermittlerkommandos zu ändern).

1. ALGØL-Programme ohne Daten

Ø1XBA, BEN=650012 MEYER, FKZ=AØD, SBG=7 Ø.

ØRECHNE, /

ALGØL-Programm
----------------

Ø1XEN Ø.

2. FØRTRAN-Programm mit Unterprogrammen und Daten

Ø1XBA, BEN=650012 MEYER, FKZ=FTUPDA Ø.

ØRE., S.=FTN, Q.= /

<pre> C FØRTRAN-Hauptprogramm        :::::       END       SUBROUTINE UP1(X,M)        :::::       END       FUNCTION PHI(Y)        :::::       END </pre>
---

Ø /, DATEN= /

Daten
-------

Ø1XEN Ø.

3. Großes FØRTRAN-Programm mit mehrmaligem Start

Ø1XBA, BEN=650012 MEYER, FKZ=GFMS, SBG=12, RZS=5 Ø.  
ØUEB., S.=FTN, VA.=GR, Q.=/

FØRTRAN-Programm

ØMØNT.

ØSTARTE,, GANG1, DATEN=/  
ØL.=GANG2, DATEN=/  
Ø1XEN Ø.

Daten

Daten

Daten

Ø1XEN Ø.

4. ALGØL-Programm mit getrennter Prozedur in Testversion

Ø1XBA, BEN=650012 MEYER, FKZ=AGP Ø.  
ØUEB., P.=R, -STD-, ASSIGN, Q.=/

```
real procedure ALPHA (x,y,z);  
begin  
:::  
end
```

ØUEB.,/

```
comment algol-hauptprogramm  
begin  
:::  
real procedure ALPHA (x,y,z);  
code;  
:::  
end
```

ØMØNT.

ØSTARTE, DUMP=A-ALLES, DATEN=/  
Ø1XEN Ø.

Daten

Ø1XEN Ø.

5. FØRTRAN-Programm mit BLØCK DATA-Programm

Ø1XBA, BEN=650012 MEYER, FKZ=FØBLDA ◇.  
◇\*SPRACHE=FTN, \*PRØT.(UEB.)= -STD-  
◇UEB.,/

FØRTRAN-Hauptprogramm FUNCTION- und SUBRØUTINE-Unterprogramme
--

◇UEB., MØ=BETA, Q.=/

BLØCKDATA  ::: END
-----------------------------

◇MØNT., STDHP'BETA  
◇STARTE, DATEN=/  
Daten

Ø1XEN ◇.

6. Dateiverkehr ohne Programm

Ø1XBA, BEN=650012 MEYER, FKZ=DØP, PSB=200 ◇.  
◇DATEI, STE, SEQ, M1000, M80 Ø, P  
◇NAME=STA  
◇EINTR., STE,/  
Daten

◇SØRT., (STE), (STA),,(L,1,4,F)  
◇DRUCKE,,STA  
Ø1XEN ◇.

7. FØRTRAN-Programm mit Dateiverkehr

Ø1XBA, BEN=650012 MEYER, FKZ=FPD, TSB=100 ◇.

◇UEB., S.=FTN, MØ=HINEIN, Q.=/

.....	
WRITE (12) X	FØRTRAN- Programm
.....	

◇ MØ=HERAUS, Q.=/

.....	
READ (11) Y	FØRTRAN- Programm
.....	

◇ MØNT., HINEIN

◇ MØ=HERAUS

◇ DATEI, ZWISP, SEQ, M5000, G1W, T

◇ STARTE, HINEIN, DATEI=12-ZWISP

◇ DATEI=11-ZWISP, P.=HERAUS

Ø1XEN ◇.

8. ALGØL-Programm in einem Kapitel

Ø1XBA, BEN=650012 MEYER, FKZ=APKAP ◇.

◇TDEKL., ALK, M200

◇TEINTR., ALK, P.= -STD- , I.=/

Ø1XUM, KC3 ◇.

ALGØL-Programm, im Code KC3 gelocht
-------------------------------------

Ø1XUM, KC1 ◇.

◇ RECHNE, ALK

◇ TEINTR., ALK, 90, /

Korrekturkarte im Code KC1
----------------------------

◇ RECHNE, ALK

Ø1XEN ◇.

9. Anlegen einer LF-Datei

- Ø 1XBA, BEN=650012 MEYER, FKZ=LF1 ◇.
- ◇ DATEI, SP-AUFBAU, RAM, M500, M80 Ø, LFD
- ◇ EINTR., SP, /

Daten
-------

- ◇ DRUCKE., SP, 1KØP
- Ø 1XEN ◇.

10. Verwendung der LF-Datei von 9., die ein FØRTRAN-Programm enthalte

- Ø 1XBA, BEN=650012 MEYER, FKZ=LF2 ◇.
- ◇ EINSCHL., SP-BAU, LFD, M.=LESEN
- ◇ RECHNE, SP,, FTN
- ◇ TD., KASP, M400
- ◇ TE., KASP, I.=/

Programm-Erweiterung
----------------------

- ◇ EINSCHL., SP-AUFBAU, LFD,, SCHREIBEN
- ◇ TKA., SP, SP, 1-KASP,, -STD-
- ◇ RECHNE, SP,, FTN
- Ø 1XEN ◇.