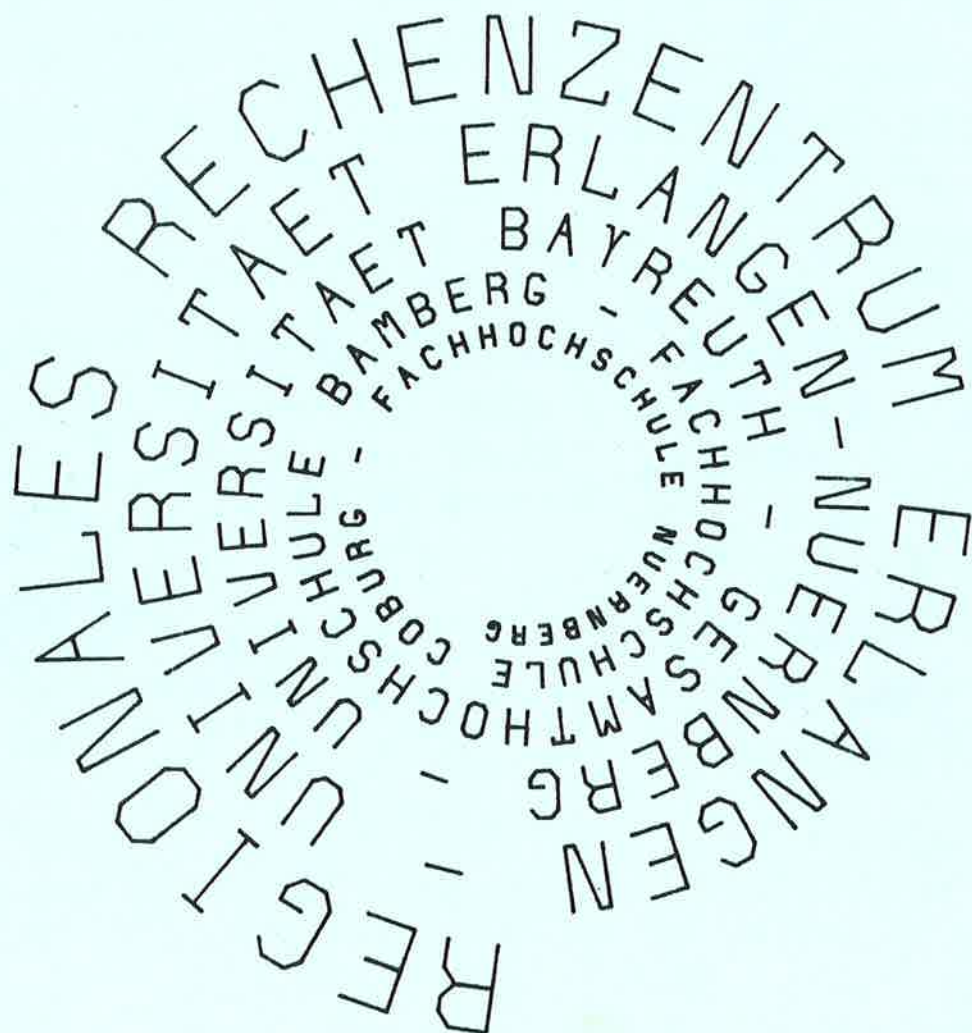


RRZE

BENUTZERINFORMATION



BI 7 -ERLANGEN- 15.FEBRUAR 1977

R R Z E

REGIONALES RECHENZENTRUM

MARTENSSTRASSE 1

8520 ERLANGEN

TEL: 09131 / 85 70 31 - 85 70 32

BETEILIGTE EINRICHTUNGEN :

UNIVERSITÄT ERLANGEN - NÜRNBERG MIT

RECHENZENTRUM

AUSSENSTELLE ERLANGEN INNENSTADT

AUSSENSTELLE ERLANGEN SÜDGELÄNDE

AUSSENSTELLE NÜRNBERG TUCHERGELÄNDE

AUSSENSTELLE NÜRNBERG FINDELGASSE

UNIVERSITÄT BAYREUTH

GESAMTHOCHSCHULE BAMBERG

FACHHOCHSCHULE COBURG

FACHHOCHSCHULE NÜRNBERG

HERAUSGEGEBEN VOM REGIONALEN RECHENZENTRUM ERLANGEN

Inhalt

1.	Aktuelle Information	1
1.1	Rundschreiben	1
1.2	Protokoll des Benutzerkolloquiums vom 1.2.77	1
1.3	Wählanschlüsse	2
1.4	Handbücher für TR 440 und CYBER	3
1.5	Vergabe von Magnetbändern	7
1.6	"Kummerkasten"	9
1.7	Aufsicht während der CYBER-Einführung	9
1.8	Rechenzeitkosten	9
1.9	Fragebogen zum Umstellungsaufwand	10
2.	Neues zum Stand der Software	11
2.1	CYBER-Programmbibliothek	11
2.2	TR 440-Programmbibliothek	11
3.	Probleme aus der Beratung	25
3.1	Bemerkungen zu den Kommandos TKAPEINFUEGE und TNUMERIERE	25
3.1.1	Einfügen von Kapiteln mit TKAPEINFUEGE	25
3.1.2	Numerieren von RAN-Dateien mit TNUMERIERE	25
3.2	Korrektur zum Stanzen von Lochstreifen an der TR 440	26
3.3	Hinweise für die Benutzung der CYBER 172	26
3.4	Sie fragen - wir antworten	27
4.	Ausgewählte Themen	29
4.1	Steuerkartenbeispiele für die CYBER	29
4.1.1	Handhabung von Magnetbändern	29
4.1.2	Umgang mit dem Lader	32
4.2	FORTTRAN-Unterschiede CD 3300 - CYBER	36
4.3	Conversion Aids	37
4.4	Übertragung von Daten von CD 3300 auf CYBER mit CDPICK	46
4.5	Steuerkarten für UPDATE	47
	Anhang: Programmbeschreibungen TR 440	54
	Kommandos TR 440	63
	Fragebogen Umstellungsaufwand	69

1. Aktuelle Informationen

1.1 Rundschreiben

Seit Erscheinen der BI 6 sind allen Benutzern folgende Rundschreiben zugegangen.

Rundschreiben 3/77

enthält einen Hinweis auf den Probetrieb der CYBER 172 und den Einschränkungen des Betriebs der CD 3300.

Rundschreiben 4/77

gibt einige Termine bekannt. Zur Wiederholung:

- 24.2. - 26.2.77 : Kompaktkurs (9 hct, 2.159). Einführung in die Benutzung der CYBER
- | | | |
|---------|---|--|
| 1.3.77 | : | Benutzerkolloquium (16 hct, H4) |
| 8.3.77 | : | } RZ-Kolloquium, Einführung in die Dialogbenutzung (CYBER) |
| 15.3.77 | : | |

1.2 Protokoll des Benutzer-Kolloquiums vom 1.2.1977

T.O.P. 1

Bericht des Rechenzentrumsleiters Dr. Wolf

Zustand der Rechenanlagen:

- Die TR 440 läuft im Normalbetrieb, nachts ohne Operateur.
- Die CYBER 172 läuft im Probetrieb.
- Die CDC 3300 Doppelprozessor wurde wieder zum Einfachprozessor konfiguriert.
- An der CDC 3300 wird künftig keine Nachtschicht mehr gefahren. Außerdienststellung der Maschine voraussichtlich am 1.4.1977.
- Die Remote-Job-Entry-Stationen sind bestellt.

Rechnungsmodus für die verschiedenen Rechenanlagen:

- Die Rechnung für das 4. Quartal 1976 steht wegen formaler Schwierigkeiten bei der Rechnungsstellung noch aus.
- Die Berechnung der Rechenkosten an der CDC 3300 bleibt unverändert.
- An der TR 440 wird in der Formel für die Rechenkosten in Zukunft die Kanalzeit nicht mit dem CPU-Zeit-Faktor hochgerechnet.
- Die Schwankungen der CPU-Zeit an der TR 440 ergaben sich durch Ein- und Ausschalten von Assoziativspeichern (wird inzwischen nicht mehr gemacht).
- Es wird versucht, an der TR 440 die Führung des aktuellen Kontostandes zu ermöglichen. Bis zu einer Lösung dieses schwierigen Problems soll eine monatliche Rechnungsübersicht den Benutzern zugänglich gemacht werden.

T.O.P. 2

Beantwortung von Fragen durch Mitarbeiter des RRZE.

Ergebnisse: Es wird eine Übersicht über den Stand der Arbeiten an der Uhrensynchronisation des Tripel-Prozessors TR 440 gegeben.

Die Anlage von Bibliotheken an der TR 440 ist wegen der E/A-Prozeduren teuer, aber komfortabel.

Der Datenschutz bei Bändern wird verbessert.

T.O.P. 3

Der Termin für das nächste Benutzerkolloquium wird auf den 1. März 1977 festgelegt.

T.O.P. 4

Sonstiges:

Der Parkplatz vor dem RZZE soll in Zukunft für die Benutzer besser freigehalten werden.

Für die nächsten RZZE-Kolloquien werden folgende Programmpunkte vorgeschlagen:

Update, Editor und Conversion Aids an der Cyber 172.

Genaue Termine werden durch einen Aushang bekanntgegeben.

1.3 Wählanschlüsse an CYBER 172, CD 3300 und TR 440

Ab 1.3.1977 stehen für den Dialogverkehr mit der Cyber 172 8 Wählanschlüsse 300 Bd zur Verfügung.

Es sind die Telefonnummern der Hausapparate

7691 ... 7698

Es wird empfohlen, die Nummer 7691 zu wählen, da von dieser Nummer ausgehend innerhalb der Kaskade automatisch die nächste freie Leitung vermittelt wird.

Externe Benutzer können sich, sofern sie mit einem Postmodem ausgerüstet sind, für den Dialogbetrieb mit 300 Bd an die Cyber 172 über die Telefonnummer (09131) 85 7444 anwählen.

Zum Anschluß an die CD 3300 steht nur noch die Nummer 7690 zur Verfügung.

Wählanschlüsse am TR 440:

Es stehen für den Dialogbetrieb mit 300 Bd die Hausapparate-Nummern 7996 ... 7999 zur Verfügung. Die Einrichtung eines Postmodems ist in Vorbereitung.

1.4 CYBER und TR 440 - Unterlagen

Die Benutzer von CYBER und TR 440 haben auf folgende Weise einen allgemeinen Zugang zu den Schriften der Hersteller:

- a) Im Benutzerraum (1.07) der CYBER ist ein Satz von Manuals zur Einsicht aufgestellt. Das sind:
FORTRAN Extended
NOS1 Ref. Man. Band 1
NOS1 Ref. Man. Band 2
Loader Ref. Man.
SORT/MERGE Ref. Man.
Time-sharing Users Ref. Man.
Text Editor Ref. Man.
UPDATE Ref. Man.
COBOL 4 Ref. Man.
- b) Am Terminal in Nürnberg steht der obige Satz an CYBER-Manuals (COBOL Man. wird nachgeliefert), dazu ein Satz TR 440 Literatur wie im Verteilungsplan für Außenstellen angegeben.
- c) Ein TR 440-Unterlagensatz steht im Benutzerraum der TR 440 (s. Verteilungsplan auf der nächsten Seite).
- d) Weiter ist die Aufstellung eines Satzes von Schriften beider Rechenanlagen im Raum 1.17 geplant.
- e) Ein vollständiger Satz von Unterlagen beider Rechner befindet sich im Zimmer der Beratung.
- f) Einige Handbücher bzw. Kopien davon können in der Aufsicht käuflich erworben werden, und zwar derzeit
zur CYBER : NOS 1 VOL I Reference Manual DM 5.--
 NOS 1 VOL II Reference Manual DM 7.--
 FORTRAN Extended 4 Reference Manual DM 7.--
 SORT/MERGE 4 Reference Manual DM 5.--
zur TR440 : Kommando-Taschenbuch DM 1.50
 FORTRAN-Sprachbeschreibung DM 5.--
 Vorlesungsskript FORTRAN (vom RRZE) DM 2.--
- g) Soweit Unterlagen vom Rechenzentrum erstellt werden, werden diese in der Regel in die BI aufgenommen.
- h) Alle weiteren Unterlagen können zur Zeit nur direkt durch Bestellung beim jeweiligen Hersteller bezogen werden. Eine Liste der CYBER-Literatur ist unten angegeben. Ein Literaturverzeichnis für den TR 440 (~100 Seiten) kann in der Aufsicht eingesehen werden.

Verteilungsplan

Benutzer-
räume

Außen-
stellen

TR 440-Literatur

Daten
Große Befehlsliste
Befehlslexikon
Systemdienste
Programmiersystem Einführung
Teilnehmer-Betriebssystem Einführung
Teilnehmer-Rechensystem Kurzbeschreibung
Grafik-Software Benutzerbeschreibung
Kommandosprache TR440 Sprachbeschreibung
Kommandosprache TR440 Taschenbuch
COBOL Sprachbeschreibung
FORTRAN Sprachbeschreibung
ALGOL Sprachbeschreibung
Abschnittsbetrieb Teil 1 des Benutzerhandbuches
Dialogbetrieb Teil 2 des Benutzerhandbuches
BASIC Sprachbeschreibung
BCPL Sprachbeschreibung
Datenhaltung im TR440
PL/I Einführung in den Sprachumfang
PL/I RECORD-EA
PL/I Attribute BASED, POINTER
PL/I Blockstruktur
PL/I Attribute DEFINED, POSITION
PL/I Unterprogrammtechnik
PL/I Datentypen
PL/I STREAM-EA
TAS Sprachbeschreibung
SORT Benutzerbeschreibung
FORTRAN Hilfsprogramme
ALGOL Arithmetik und Hilfsprogramme
ALGOL Elementare Funktionen
FORTRAN Elementare und spezielle Funktionen
ALGOL Spezielle Funktionen
ALGOL-FORTRAN Polynome Reihen etc
ALGOL Lineare Algebra
ALGOL Lineare Algebra 1. Nachtrag
FORTRAN Lineare Algebra
FORMAT-440
FORTRAN Statistik
TELDOK Benutzerbeschreibung
DBS 440
Informationsdarstellung auf Magnetbändern
Druckschriftenverzeichnis
Begriffe
Namen zur Informationsdarstellung

NOS 1 PUBLICATIONS

Preise: Stand Oktober 1976

Bemerkung: Alle aufgeführten Preise sind als Richtpreise zu verstehen und nicht verbindlich.

Anschrift: CONTROL DATA GmbH
zu Hd. Herrn P. Noack
Stresemannallee 30-32

6000 Frankfurt/Main

SAB DATE---8/76
OS PSR LEVEL---430
PS PSR LEVEL---428 } Installationslevel Erlangen

<u>TITLE</u>	<u>TYPE</u>	<u>PUB.NO.</u>	<u>PREIS (\$)</u>
Operating System			
NOS 1 Vol I	RM	60435400 B	8,65
Vol II	RM	60445300 B	9,15
	OG	60435600 B	8,30
	IH	60435700 D	15,35
Applications Prog	IN	60436000 B	1,35
System Prog	IN	60449200 B	2,60
NOS Text Editor	RM	60436100 C	2,35
EXPORT/IMPORT	RM	60436200 B	1,--
NOS Time Sharing	RM	60435500 B	4,15
	UG	60436400 A	1,65
	IN	60435800 B	1,70
CYBER LOADER	RM	60429800 C(1.2)	6,35
	IN	60449800 A	1,50
UPDATE	RM	60449900 A	2,75
	IN	60450000 A	1,35
CYBER Common Utilities	RM	60495600 B	1,--
MODIFY	RM	60450100 A	2,40
	IN	60450200 A	1,55
Common Products			
ALGOL 4	RM	60496600 A	8,55
APL 1	RM	19980400 D	4,--
BASIC 3	RM	19983900 C	6,75
COBOL 4	RM	60496800 A	11,90
	IN	60497000 A	1,35
COMPASS 3	RM	60492600 B	9,35
	IN	60492800 A	1,50

<u>TITLE</u>	<u>TYPE</u>	<u>PUB.NO.</u>	<u>PREIS (\$)</u>
CYBER Record Manager 1	RM	60495700 B	5,95
COBOL 4	UG	60496000 A	nicht bekannt
FORTRAN Extended 4	RM	60497800 B	10,--
	IN	60497900 A	1,35
DEBUG	UG	60498000 A	2,60
Common Math Lib	RM	60498200 B	4,95
SIFT	PSB	60496500 A	1,20
SORT/MERGE 4	RM	60497500 B	4,40
	IN	60497600 A	1,30
SYMPL 1	RM	60496400 A	4,20
8-Bit Subroutines	RM	60495500 A	3,90
Convr Aids FTN (3000L)		19980600 C	3,--
Convr Aids COBOL (3000L)		19980700 B	5,50
Convr Aids System (3000L)	UG	19980900 B	5,50
FORM 1	RM	60496200 A	2,85
Graphics	GIM	76077000 A	nicht bekannt
Beginning Graphics	UG	76077300 A	nicht bekannt
LCGT/IGS 2	RM	76079100 A	nicht bekannt
	UG	76077400 A	nicht bekannt
SIMULA 1	RM	60234800 F	8,30
TOTAL 1	RM	76070300 C(1.4)	5,40
ATHENA	UIM	76071400 D	4,--

Abkürzungen

RM	=	Reference Manual
OG	=	Operators Guide
IH	=	Installation Handbook
IN	=	Instant
UG	=	Users Guide
PSB	=	Program Subsystem Bulletin
GIM	=	General Information Manual
UIM	=	Users Information Manual

1.5 Bandorganisation

1. Standort der Bänder

Mit Inbetriebnahme der Rechananlage TR 440 Dreifachprozessor und CYBER wurde am RRZE damit begonnen, eine neue Bandorganisation aufzubauen. Das erste Zeichen der Bandkennzeichnung gibt den Standort der Bänder an (T = TR 440, C = CYBER). Standort der Bänder ist der jeweilige Rechenraum, gleichgültig ob es sich um benutzereigene oder RZ-Bänder handelt. Der Benutzer kann über die Aufsicht Bänder für die jeweilige RA leihen oder kaufen und sie wie folgt benutzen:

2. Schreibschutz

Normalerweise werden die Bänder mit Schreibring aufbewahrt, so daß Lesen und Schreiben auf dem Band möglich ist. Um wichtige Bänder vor Überschreiben zu sichern, kann der Benutzer auf einer speziellen Lochkarte (liegen im jeweiligen Ein-/Ausgaberaum aus) angeben, daß dieses Band ohne Schreibring aufbewahrt werden soll. Diese Karte ist dem Operateur abzugeben. Falls dieses Band dann irgendwann doch beschrieben werden soll, muß er schriftlich dazu über eine weitere Karte eine Anweisung an den Operateur geben.

REGIONALES RECHENZENTRUM ERLANGEN

Das Band
mit dem Kennzeichen

--	--	--	--	--	--	--	--

- ☐ darf ab sofort nicht mehr beschrieben werden;
- ☐ kann wieder beschrieben werden;
- ☐ darf nur von folgenden Benutzernummern angefordert werden:
- | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
- ☐ wurde heute ausgeliehen;

Datum

Benutzernummer

Unterschrift

3. Zugriffsschutz

Um zu verhindern, daß nicht autorisierte Benutzer auf geschützte Bänder zugreifen können, kann auf dem Band zusätzlich ein Aufkleber mit bis zu 5 berechtigten Benutzernummern angebracht werden. Diese zusätzliche Sicherung muß über die gleiche Bandkarte beim Operateur angefordert werden.

Zur Zeit ist diese Prüfung durch den Operateur jedoch äußerst schwierig durchzuführen.

Sowohl an der CYBER als auch am TR 440 müssen zur Realisierung dieser Operateurprüfung Systemänderungen vorgenommen werden. Daher wird sie vorerst nur bei Verdacht auf Mißbrauch durchgeführt.

Das RRZE bittet darum um sparsame Anforderung dieser Dienstleistung, da zur Durchführung zusätzlicher personeller Aufwand notwendig ist. Eine kleine zusätzliche Sicherheit kann der Bandbenutzer jedoch an der CYBER selbst durchführen: nach der Bandbenutzung wird die UNLOAD-Funktion aufgerufen, die das Band zum Bandanfang zurückspult und das Bandgerät ausschaltet.

4. Ausleihen von Bändern

Benutzer, die aus irgendeinem Grund T und C Bänder, d.h. Bänder mit Standort TR 440-Rechnerraum oder CYBER-Rechnerraum, ausleihen wollen, müssen wiederum eine Bandkarte, die dann an Stelle des Bandes aufbewahrt wird, ausfüllen. Wie in einer Präsenzbibliothek sollen Bänder nur in Ausnahmefällen und auch nur kurzfristig ausgeliehen werden.

5. Sicherheitskopien

Die Benutzer sind für die Sicherheit der im Rechenzentrum hinterlegten Daten selbst verantwortlich. Bei besonders wichtigen Daten wird empfohlen

- a) der Benutzer kauft selbst Sicherungsbänder und nimmt entsprechende Kopien mit nach Hause. Diese Bänder sind genauso zu kennzeichnen, wie die Bänder mit Standort Rechenzentrum, jedoch wird der erste Buchstabe (T oder C) in ein Z umgewandelt.
- b) Benutzer, die am TR 440 arbeiten, können über die Aufsicht auch Bänder an der CYBER beantragen und diese Bänder als Kopien der am TR 440 benutzten Bänder mit Standort CYBER (C) aufbewahren lassen.
Umgekehrt ist das gleiche natürlich mit an der CYBER benutzten Bändern auch möglich.

In jedem Fall handelt es sich bei der Erstellung und weiteren Benutzung der Kopien am jeweils anderen REchner immer um Spezialläufe.

6. Größerer Bandbedarf

Falls ein Benutzer viele Bänder benötigt (mehr als 10), so wird er gebeten, diese selbst zu kaufen. Der Standort kann aber weiterhin einer der beiden Rechnerräume sein, so daß die Kennzeichnung mit T oder C erfolgt.

7. Fremdbänder

Fremdbänder, die von außerhalb kommen und mehrmals im Rechenzentrum benutzt werden sollen, müssen in T oder C Bändern umgewandelt werden. Bei Schwierigkeiten kann die Beratung in Anspruch genommen werden.

Bänder von anderen Anlagen, (z.B. PDP 15 der Informatik)

werden ebenfalls als Fremdbänder behandelt. Es ist sinnvoll, diese dann mit PDP und einer laufenden Nummer zu kennzeichnen.

8. Bandnummer auf Auftragskarte

Die Benutzung von Bändern innerhalb eines Jobs muß über die Auftragskarte mit den entsprechenden Bandnummern dem Operateur mitgeteilt werden. Insbesondere müssen alle Folgebänder in der richtigen Reihenfolge angegeben werden. Fehlt diese Angabe und wird über die Konsole vom Operateur ein Folgeband verlangt, hat der Operateur die Anweisung, den Job abnormal zu beenden.

1.6 Kummerkasten

Das Beschwerdebuch an der CD 3300 ist seit einem Jahr nicht benutzt worden. Wir möchten deshalb - entgegen der früheren Ankündigung - keine Beschwerdebücher an den Rechnern auslegen, sondern Sie bitten, nur den Briefkasten an der Aufsicht für Anregungen, Kritik und Beschwerden zu benutzen. Ihre Mitteilungen werden, wenn Sie den Absender angegeben haben (möglichst mit Telefonnummer), direkt beantwortet und bei allgemeiner Bedeutung auch in der BI oder auf dem nächsten Benutzerkolloquium behandelt.

1.7 Aufsicht während der CYBER-Einführung

Die Mitarbeiter in der Aufsicht hatten bisher keine Gelegenheit, sich für die CYBER auszubilden, sie werden deshalb an der Einführung in die Benutzung der CYBER am 24. und 25. Februar teilnehmen. Die Aufsicht wird an diesen Tagen in der üblichen Zeit mit einem Notdienst besetzt.

Auch der beratende Mitarbeiter nimmt an diesem Kurs teil, er ist deshalb nur in den Pausen zu sprechen. Wir bitten um Ihr Verständnis für diese Regelung.

1.8 Rechenzeitkosten

Mitte Februar wurden folgende Rechenzeitübersichten mit Rechnungen an die Benutzer verschickt.

1. Rechenzeitübersicht 4. Quartal 1976 (CD 3300 + TR 440 der Informatik)
2. Rechnungen über Rechenzeitverbrauch im 4. Quartal 76
3. Rechenzeitübersicht CD 3300 für Januar 1977
4. Rechenzeitübersicht TR 440 für Januar 1977

Die Rechenzeitkosten am TR 440 wurden dabei nach einer Formel ermittelt, die sich von der im Rundschreiben Nr. 1 angegebenen Formel im wesentlichen dadurch unterscheidet, daß die Kanalzeit nicht mehr mit dem CPU-Zeit-Faktor hochgerechnet wird:

$$KJ = F_1 \cdot (T \cdot P + S) + Z \cdot F_2 + K \cdot F_3$$

KJ = Kosten eines Jobs

P = Programmlaufzeit (verbrauchte CPU-Zeit)

T = Kostenfaktor der verschiedenen Typen

CD 3300 T = 1

TR 440 T = 3.4

CYBER T = noch nicht ermittelt

S = Speicherbelegungszeit, d.h. Zeit, in der der Hauptspeicher anteilig durch das Programm belegt ist.

= $(T \cdot P + KA) \cdot SP/MSP$

KA = Kanalzeit

SP = in Anspruch genommener Hauptspeicher
bei CD 3300 in Quarterpages
bei TR 440 in K Speicher

MSP = 128 Quarterpages bei CD 3300

= 80 K bei TR 440

Bei der CYBER wird der Wert für S direkt vom System ermittelt, Angaben über die Kanalzeit existieren dagegen nicht.

F₁ = Kosten einer CD 3300 Abrechnungseinheit
(CPU und Speicherkosten)

= DM 20,-- / Std.

F₂ = Kosten einer Druckzeile (0.5 Pfg.)

F₃ = Kosten einer gestanzten Karte (0.5 Pfg.)

Z = Zahl der gedruckten Zeilen

K = Zahl der gestanzten Karten

1.9 Fragebogen zum Umstellungsaufwand

Von der SEKORA wird derzeit eine Umfrage zur Ermittlung des Umstellungsaufwandes durchgeführt. Bei der Auswahl eines Rechners für das Regionale Rechenzentrum Erlangen war der Umstellungsaufwand für die Benutzer eine viel zitierte, aber nicht abschätzbare Größe. Es erreichen uns laufend Anfragen von offiziellen Stellen und anderen Interessenten über unsere Erfahrungen bei der Umstellung von Programmen auf die neuen Anlagen des Regionalen Rechenzentrums. Um die Fragen sorgfältig beantworten zu können und um Anhaltspunkte für zukünftige Ausbaupläne zu gewinnen, werden alle Benutzer um Mithilfe gebeten. Die Fragebögen werden mit gesonderter Post geschickt. Im Anhang ist ein Muster beigelegt.

2 Neues zum Stand der Software

2.1 CYBER-Programmbibliothek

Die Programme der CD 3300-Programmbibliothek sind noch nicht auf die CYBER umgestellt. Es wird versucht, zusammenhängende Programmpakete wie SSP und EISPACK als Binärbibliotheken zur Verfügung zu stellen. Die im CD 3300-Programmbibliotheks-katalog mit ERL und UNI gekennzeichneten Programme werden nur auf Benutzerwunsch umgestellt. Als Ersatz für diese Programme wird die CERN-LIBRARY adaptiert, die umfangreich und ausgetestet ist.

Die Kataloge der Programmbibliothek stehen wie bisher in der Aufsicht. Sollten Sie dringend Programme auf der CYBER benötigen, so teilen Sie bitte deren Identifikation der Aufsicht mit. Es wird in Kürze ein Verzeichnis der CYBER-Programmbibliotheken erscheinen.

2.2 Programmbibliothek TR 440 - &STARG

Seit der letzten BI wurden folgende Objekte neu implementiert.

1. Pascalcompiler

Compiler für die Programmiersprache PASCAL, Aufruf:

▣ BIBANMELDE,&STARG,LFD
▣ UEBERSETZE,SPRACHE=PASCAL,...

2. Zeichenverarbeitung in FORTRAN

Zur Umwandlung vom Format A4 nach dem Format A1 bzw. umgekehrt wurden je zwei Unterprogramme implementiert. Die genaue Programmbeschreibung steht im Anhang.

3. Generierung von Auftragsketten

Mit dem Kommando AKETTE (Beschreibung im Anhang) ist es möglich Auftragsketten (job sequencing) zu realisieren. Da einige der vorgesehenen Leistungen des Kommandos noch nicht implementiert sind empfehlen wir ein Vorgehen nach folgendem Beispiel:

a) Zusammenstellung aller Aufträge der Auftragskette.
Entgegen den sonstigen Ablochkonventionen ist dabei das XEN-Kommando am Ende jedes Auftrages wegzulassen und das XBA-Kommando mit der Angabe aller benötigten Betriebsmittel in der Form (Beispiel!)

▣ XBA,BEN=UNRZXY FACH99,KSB=40,TSB=80,PSB=200,DRS=20,
RZS=1,BGB=1▣.

statt in der üblichen Form

▣ 2XBA,...

zu lochen.

Das Kommando FEHLERHALT verhindert die Ausführung von Aufträgen der Auftragskette, wenn ein früherer Auftrag mit Fehler beendet wurde.

(Fortsetzung am Ende des Beispiels)

0491*UNRZVF

ER1

RRZE

ERLANGEN

15.02.77 180120 SEITE

```

      EE      EE      EE      EEEEEEEEEEE      EEEEEEEEEEE      EEEEEEEEEEE      EEEEEEEEEEE
    EEEE      EE      EE      EEEEEEEEEEE      EEEEEEEEEEE      EEEEEEEEEEE      EEEEEEEEEEE
    EE  EE      EE      EE      EE      EE      EE      EE      EE      EE      EE      EE
    EE  EE      EE      EE      EE      EE      EE      EE      EE      EE      EE      EE
    EE      EE      EE      EE      EEEEEEEEE      EE      EE      EE      EEEEEEEEE
    EE      EE      EE      EE      EEEEEEEEE      EE      EE      EE      EEEEEEEEE
EEEEEEEEEEEE      EEEE  EE      EE      EE      EE      EE      EE      EE      EE
EEEEEEEEEEEE      EEE   EE      EE      EE      EE      EE      EE      EE      EE
EE      EE      EE      EE      EEEEEEEEEEE      EE      EE      EE      EEEEEEEEEEE
EE      EE      EE      EE      EEEEEEEEEEE      EE      EE      EE      EEEEEEEEEEE

```

AUFTRAG 0491 15.02.77 17:06 180120 (0001.01)

BEISPIEL FUER DAS KOMMANDO A K E T T E
1. SCHRITT

ZUSAMMENSTELLUNG DER AUFTRAEGE
UND EINTRAGEN IN EINE LF-DATEI

DRPROTOKOLL,GERAET=DR-DC2

LFDATEI,LFTEMP,AUFTRAEGE,,U100

KREIERT; AUFTRAEGE(0001.00) KAT: LFTEMP

TEINTRAGE,AUFTRAEGE,PROTOKOLL=-STD-,INFORMATION=/FREMDSTRING

```

000010  XBA,BEN=UNRZVF NR.1,KSB=40,TSB=120,PSB=200,DRS=20,RZS=1.
000020  DRPROTOKOLL,GERAET=DR-DC2
000030  FEHLERHALT
000040  MIT DEM KOMMANDO FEHLERHAFT WIRD IM FEHLERFALL
000050  DIE AUFTRAGSKETTE GESTOPPT.
000060  LFANMELDE,UNRZPB,GED
000070  GEDAECHTNIS,GED
000080  LFDATEI,LFTEMP,QPROG,,U100
000090  TEINTRAGE,QPROG,PROT.=STD-,INF.=/
000100  C
000110  C
000120  C      *** DO STATEMENT ***
000130  C      *** LOGISCHES IF STATEMENT ***
000140  C      *** GOTO STATEMENT ***
000150  C
000160  C
000170      INTEGER IPLUS/1H+/, IMINUS/1H=/, ITIMES/1H*/, IDIV/1H//
000180      1      IEXP/1HE/
000190      READ (5,1000) IANZ
000200  1000  FORMAT (I4)
000210      DO 100 I=1,IANZ
000220      READ (5,1001) IL,IOP,IR
000230  1001  FORMAT (I4,A1,I4)
000240      IF ( IOP .EQ. IPLUS ) GOTO 101
000250      IF ( IOP .EQ. IMINUS ) GOTO 109
000260      IF ( IOP .EQ. ITIMES ) GOTO 103
000270      IF ( IOP .EQ. IDIV ) GOTO 104
000280      IF ( IOP .EQ. IEXP ) GOTO 105
000290      WRITE (6,1002) IOP
000300  1002  FORMAT (1H,'***** FALSCHER OPERATOR : ',1A1)
000310      GOTO 100

```



```

000320 101 IERG = IL+ IR
000330      GOTO 90
000340 102 IERG = IL - IR
000350      GOTO 90
000360 103 IERG = IL * IR
000370      GOTO 90
000380 104 REALL =IL
000390      REALR =IR
000400      ERG = REALL/REALR
000410      WRITE (6,1003) IL,IDP,IR,ERG
000420 1003 FORMAT (1H ,I4,X,A1,I4,3H = ,F10.4)
000430      GOTO 100
000440 105 IERG = IL ** IR
000450 90   WRITE (6,1004) IL,IDP,IR,IERG
000460 1004 FORMAT (1H ,I4,X,A1,X,I4,3H = ,I8)
000470 100 CONTINUE
000480      STOP
000490      END
000500  □LFDATEI,LFTEMP,EDAT,,U100
000510  □TEINTRAGE,EDAT,PROT,==STD-,INF,=/
000520      10
000530      11+ 123
000540      15* 15
000550      100/ 11
000560      2E 20
000570 9999-1234
000580      22) 11
000590      19*1257
000600 9658/ 475
000610      12E 5
000620      9E 9
000630
000640
000650  □LFRESERVIERE,LFTEMP,QPROG
000660  □LFRESERVIERE,LFTEMP,EDAT
000670  □XBA,BEN=UNRZVF NR.2,KSB=32,TSB=80□.
000680  □DRPROTOKOLL,GERAET=DR-DC2
000690  □FEHLERHALT
000700  □LFANMELDE,UNRZPB,GED
000710  □GEDAECHTNIS,GED
000720  □LFANMELDE,LESEN=LFTEMP,QPROG
000730  □UEBERSETZE,QPROG
000740  □LFDATEI,LFTEMP,BPROG,SEQ,U1000,G18W
000750  □MONTIERE
000760  □BINAERAUS,BPROG,PROGRAMM=STDHP
000770  □XBA,BEN=UNRZVF NR.3,SBG=9□.
000780  □DRPROTOKOLL,GERAET=DR-DC2
000790  □LFANMELDE,LFTEMP,BPROG'LFTEMP,EDAT
000800  □LFANMELDE,UNRZPB,GED
000810  □GEDAECHTNIS,GED
000820  □BINAEREIN,BPROG
000830  □STARTE,DATEI=5-EDAT

```

ENDE TEINTRAGE (7.02) 0.62

□= 2. SCHRITT

ABMELDEN DER DATEI AUFTRAEGE
UND START DER AUFTRAGSKETTE

□LFABMELDE,AUFTRAEGE

ABGEMELDET: AUFTRAEGE(0001.00) KAT: LFTEMP
□LFANMELDE,UNRZPB,GED

0491*UNRZVF ER1 RRZE ERLANGEN 15.02.77 180120 3

ANGEMELDET: GED(0004,00) KAT: UNRZPB
#GEDAECHTNIS,GED,EIN

GEDAECHTNIS : GED(0004,00) V.11.02.77
#AKETTE,LFTEMP,AUFTRAEGE,1-999999,LF0,124,NORM

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0.20

AUFTRAG 492 WURDE KREIERT;BEREICH 10 671

ENDE AKETTE (2.06) 0.24
#LISTINPUT

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0.18
ANGEMELDET: &BIBL1.&M3(9999,99) KAT: &BIBL1

START LISTINPUT

#= BEISPIEL FUER DAS KOMMANDO A K E T T E

1. SCHRITT

 ZUSAMMENSTELLUNG DER AUFTRAEGE
 UND EINTRAGEN IN EINE LF-DATEI

#DRPROTOKOLL,GERAET=DR-DC2

#LFDATEI,LFTEMP,AUFTRAEGE,,U100

#TEINTRAGE,AUFTRAEGE,PROTOKOLL=-STD-,INFORMATION=

#XBA,BEN=UNRZVF NR.1,KSB=40,TSB=120,PSB=200,DRS=20,RZS=10.

#DRPROTOKOLL,GERAET=DR-DC2

#FEHLERHALT

#= MIT DEM KOMMANDO FEHLERHAFT WIRD IM FEHLERFALL
 DIE AUFTRAGSKETTE GESTOPPT.

#LFANMELDE,UNRZPB,GED

#GEDAECHTNIS,GED

#LFDATEI,LFTEMP,QPROG,,U100

#TEINTRAGE,QPROG,PROT.=-STD-,INF.=/

C
C
C
C
C
C
C
C

*** DO STATEMENT ***

*** LOGISCHES IF STATEMENT ***

*** GOTO STATEMENT ***

INTEGER IPLUS/1H+/, IMINUS/1H-/, ITIMES/1H*/, IDIV/1H//,

1 IEXP/1HE/

READ (5,1000) IANZ

1000 FORMAT (I4)

DO 100 I=1,IANZ

READ (5,1001) IL,IOP,IR

1001 FORMAT (I4,A1,I4)

IF (IOP ,EQ. IPLUS) GOTO 101

IF (IOP ,EQ. IMINUS) GOTO 109

IF (IOP ,EQ. ITIMES) GOTO 103

IF (IOP ,EQ. IDIV) GOTO 104

IF (IOP ,EQ. IEXP) GOTO 105

WRITE (6,1002) IOP

1002 FORMAT (1H , '***** FALSCHER OPERATOR I ',1A1)

```

      GOTO 100
101   IERG = IL + IR
      GOTO 90
102   IERG = IL - IR
      GOTO 90
103   IERG = IL * IR
      GOTO 90
104   REALL = IL
      REALR = IR
      ERG = REALL/REALR
      WRITE (6,1003) IL,IOP,IR,ERG
1003  FORMAT (1H,I4,X,A1,I4,3H = ,F10.4)
      GOTO 100
105   IERG = IL ** IR
90    WRITE (6,1004) IL,IOP,IR,IERG
1004  FORMAT (1H,I4,X,A1,X,I4,3H = ,I8)
100   CONTINUE
      STOP
      END

```

```

□LFDATEI,LFTEMP,EDAT,,U100
□TEINTRAGE,EDAT,PROT.=-STD-,INF,=/
10
11+ 123
15* 15
100/ 11
2E 20
9999-1234
22) 11
19*1257
9658/ 475
12E 5
9E 9

```

```

□LFRESERVIERE,LFTEMP,QPROG
□LFRESERVIERE,LFTEMP,EDAT
□XBA,BEN=UNRZVF NR.2,KSB=32,TSB=80□,
□DRPROTOKOLL,GERAET=DR-DC2
□FEHLERHALT
□LFANMELDE,UNRZPB.GED
□GEDAECHTNIS,GED
□LFANMELDE,LESEN=LFTEMP,QPROG
□UEBERSETZE,QPROG
□LFDATEI,LFTEMP,BPROG,SEQ,U1000,G18W
□MONTIERE
□BINAERAUS,BPROG,PROGRAMM=STDHP
□XBA,BEN=UNRZVF NR.3,SBG=9□,
□DRPROTOKOLL,GERAET=DR-DC2
□LFANMELDE,LFTEMP,BPROG,LFTEMP,EDAT
□LFANMELDE,UNRZPB.GED
□GEDAECHTNIS,GED
□BINAEREIN,BPROG
□STARTE,DATEI=5-EDAT
□ 2. SCHRITT
      ABMELDEN DER DATEI AUFTRAEGE
      UND START DER AUFTRAGSKETTE
□LFANMELDE,AUFTRAEGE
□LFANMELDE,UNRZPB.GED
□GEDAECHTNIS,GED,EIN
□AKETTE,LFTEMP,AUFTRAEGE,1-999999,LPD,124,NORM
□LISTINPUT

```

0492*UNRZVP ERI RRZE ERLANGEN 15.02.77 180120 SEITE

```

EE      EE      EEEEEEE      EE
EE      EE      EEEEEEEEEE
EEEE    EE      EE      EEE
EE EE    EE      EE      EE
EE      EE      EE      EEE
EE      EE      EEEEEEEEEE
EE      EE      EEEEEEEE
EE      EEEE    EE      EE
EE      EEE     EE      EEE
EE      EE      EE      EEE
EE      EE      EE      EEEEE

```

AUFTRAG 0492 15.02.77 17:06 180120 (0001.01)

DRPROTOKOLL,GERAET=DR-DC2

FEHLERHALT

MIT DEM KOMMANDO FEHLERHAFT WIRD IM FEHLERFALL
DIE AUFTRAGSKETTE GESTOPPT.

LFANMELDE,UNRZPB,GED

ANGEMELDET: GED(0004.00) KAT: UNRZPB
GEDAECHTNIS,GED

GEDAECHTNIS : GED(0004.00) V.11.02.77
LFDATEI,LFTEMP,QPROG,,U100

KREIERT: QPROG(0001.00) KAT: LFTEMP
INTRAGE,QPROG,PROT,=-STD-,INF,=/FREMDSTRING

```

000010 C
000020 C
000030 C      *** DO STATEMENT ***
000040 C      *** LOGISCHES IF STATEMENT ***
000050 C      *** GOTO STATEMENT ***
000060 C
000070 C
000080      INTEGER IPLUS/1H+/, IMINUS/1H-/, ITIMES/1H*/, IDIV/1H//,
000090      1      IEXP/1HE/
000100      READ (5,1000) IANZ
000110      1000      FORMAT (I4)
000120      DO 100 I=1,IANZ
000130      READ (5,1001) IL,IOP,IR
000140      1001      FORMAT (I4,A1,I4)
000150      IF ( IOP .EQ. IPLUS ) GOTO 101
000160      IF ( IOP .EQ. IMINUS ) GOTO 109
000170      IF ( IOP .EQ. ITIMES ) GOTO 103
000180      IF ( IOP .EQ. IDIV ) GOTO 104
000190      IF ( IOP .EQ. IEXP ) GOTO 105
000200      WRITE (6,1002) IOP
000210      1002      FORMAT (1H,'***** FALSCHER OPERATOR : ',1A1)
000220      GOTO 100
000230      101      IERG = IL+ IR
000240      GOTO 90

```

0492*UNRZVF

ERI RRZE

ERLANGEN

15.02.77 180120

2

```

000250 102 IERG = IL - IR
000260      GOTO 90
000270 103 IERG = IL * IR
000280      GOTO 90
000290 104 REALL = IL
000300      REALR = IR
000310      ERG = REALL/REALR
000320      WRITE (6,1003) IL,IOP,IR,ERG
000330 1003 FORMAT (1H ,I4,X,A1,I4,3H = ,F10.4)
000340      GOTO 100
000350 105 IERG = IL ** IR
000360 90    WRITE (6,1004) IL,IOP,IR,IERG
000370 1004 FORMAT (1H ,I4,X,A1,X,I4,3H = ,I8)
000380 100 CONTINUE
000390      STOP
000400      END

```

ENDE TEINTRAGE (7.02) 0.34
 #LFDATEI,LFTEMP.EDAT,,U100

KREIERT: EDAT(0001.00) KAT: LFTEMP
 #TEINTRAGE,EDAT,PROT.=-STD-,INF.=/FREMDSTRING

```

000010 10
000020 11+ 123
000030 15* 13
000040 100/ 11
000050 2E 20
000060 9999-1234
000070 22) 11
000080 19*1257
000090 9658/ 475
000100 12E 5
000110 9E 9
000120
000130

```

ENDE TEINTRAGE (7.02) 0.14
 #LFRRESERVIERE,LFTEMP.QPROG

RESERVIERT: QPROG(0001.00) KAT: LFTEMP BELEGT: 2K
 #LFRRESERVIERE,LFTEMP.EDAT

RESERVIERT: EDAT(0001.00) KAT: LFTEMP BELEGT: 2K
 #AKETTE,LFTEMP.AUFTRAEGE(1. 0) , 669-999999,LPD
 , 124,

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0.20
 AUFTRAG 492 NORMAL BEENDET

STAND DER AUFTRAGSKETTE
 AUFTRAG 492 WURDE VON AUFTRAG 491 KREIERT, BEREICH 10 - 671

AUFTRAG 493 WURDE KREIERT, BEREICH 670 - 771

ENDE AKETTE (2.06) 0.13

0493*UNRZVF

ERI RRZE

ERLANGEN

15.02.77

180120 SEITE

```

EE      EE      EEEEEEE      EEEE
EE      EE      EEEEEEEEEE    EEEEEEEE
EEEE    EE      EE      EEE    BEE      EEE
EE EE    EE      EE      EE    EE      EE
EE      EE      EE      EE      EEE    EEE
EE      EE      EEEEEEEEEE    EEE
EE      EE      EEEEEEEE      EEE
EE      EEEE    EE      EE      EE      EEE
EE      EEE    EE      EE      EEE    EEEEEEEEEE
EE      EE      EE      EE      EEE    EEEEEEEEEE

```

AUFTRAG 0493 15.02.77 17:06 180120 (0001.01)

HDRPROTOKOLL,GERAET=DR-DC2

FEHLERHALT

LFANMELDE,UNRZPB,GED

ANGEMELDET: GED(0004.00) KAT: UNRZPB

GEDAECHTNIS,GED

GEDAECHTNIS : GED(0004.00) V.11.02.77

LFANMELDE,LESEN=LFTEMP.QPROG

ANGEMELDET: QPROG(0001.00) KAT: LFTEMP

UEBERSETZE,QPROG

START PS&FTNCOMP (0074.04)

ANFANG PROTOKOLL

```

000010 C
000020 C
000030 C      *** DD STATEMENT ***
000040 C      *** LOGISCHES IF STATEMENT ***
000050 C      *** GOTO STATEMENT ***
000060 C
000070 C
000080      INTEGER IPLUS/1H+/, IMINUS/1H-/, ITIMES/1H*/, IDIV/1H//,
000090      1      IEXP/1HE/
000100      READ (5,1000) IANZ
000110 1000      FORMAT (I4)
000120      DO 100 I=1,IANZ
000130      READ (5,1001) IL,IOP,IR
000140 1001      FORMAT (I4,A1,I4)
000150      IF ( IOP .EQ. IPLUS ) GOTO 101
000160      IF ( IOP .EQ. IMINUS ) GOTO 109
000170      IF ( IOP .EQ. ITIMES ) GOTO 103
000180      IF ( IOP .EQ. IDIV ) GOTO 104
000190      IF ( IOP .EQ. IEXP ) GOTO 105
000200      WRITE (6,1002) IOP
000210 1002      FORMAT (1H,'***** FALSCHER OPERATOR : ',1A1)
000220      GOTO 100
000230 101      IERG = IL+ IR

```

0493*UNRZVF

ER1

RRZE

ERLANGEN

15.02.77 180120

2

```
000240      GOTO 90
000250 102   IERG = IL - IR
000260      GOTO 90
000270 103   IERG = IL * IR
000280      GOTO 90
000290 104   REALL = IL
000300      REALR = IR
000310      ERG = REALL/REALR
000320      WRITE (6,1003) IL,IOP,IR,ERG
000330 1003   FORMAT (1H ,I4,X,A1,I4,3H = ,F10.4)
000340      GOTO 100
000350 105   IERG = IL ** IR
000360 90     WRITE (6,1004) IL,IOP,IR,IERG
000370 1004   FORMAT (1H ,I4,X,A1,X,I4,3H = ,I8)
000380 100   CONTINUE
000390      STOP
000400      END
```

**FEHLER: LABEL 109 NICHT VORGEKOMMEN

**WARNG.: UEBERSETZUNG HIER ABGEBROCHEN

ENDE PROTOKOLL

UEBERSETZERABBRUCH WEGEN FEHLER KEIN MO ERZEUGT

ENDE PS&FTNCOMP (0074,04) 0.45

+++++OPERATORLAUF MIT FEHLER BEENDET: PS&FTNCOMP

ABBRUCH DURCH KOMMANDO FEHLERHALT
AUFTRAGSBEARBEITUNG ABGEBROCHEN

0510*UNRZVF ER1 RRZE ERLANGEN 15.02.77 180120 SEITE

```
EE      EE      EEEE      EEEEEEE      EEEEEEE
EE      EE      EEEEEEEE      EEEEEEEEE      EEEEEEEEE
EE      EE      EEE      EEE      EE      EEE      EE      EEE
EE      EE      EE      EE      EE      EE      EE      EE
EE      EE      EE      EE      EE      EEE      EE      EEE
EE      EEE      EE      EE      EEEEEEEEE      EEEEEEEEE
EEEE      EE      EE      EE      EEEEEEE      EEEEEEE
EEE      EE      EEE      EEE      EE      EE      EE      EE
EE      EE      EEEEEEEE      EE      EE      EE      EE
EE      EE      EEEE      EE      EE      EE      EE      EE
```

AUFTRAG 0510 15.02.77 17:10 180120 (0001.01)

□= SCHRITT 3:

KORREKTUR DES FEHLERHAFTEN PROGRAMMS
UND WIEDERAUFSETZEN DER UNTERBROCHENEN AUFTRAGSKETTE

□DRP.,GER.=DR-DC2

□LFANMELDE,UNRZPB.GED

ANGEMELDET: GED(0004.00) KAT: UNRZPB

□GEDAECHTNIS,GED

GEDAECHTNIS : GED(0004.00) V.11.02.77

□LFANMELDE,,LFTEMP,QPROG

ANGEMELDET: QPROG(0001.00) KAT: LFTEMP

□LFRESERVIERE,LFTEMP,QPROG,100

RESERVIRT: QPROG(0001.00) KAT: LFTEMP BELEGT: 5K

□TEINTRAGE

,NAME=QPROG,

NUMERIERUNG=160,

MODUS=EM,

INFORMATION=/FREHDSTRING

GLEICHE NUMMER: 000160 DIE ZEILE WIRD ERSETZT

ENDE TEINTRAGE (7.02) 0.18

□AKETTE,LFTEMP.AUFTRAEGE,669-999999,LFID,124,NORM

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0.25

AUFTRAG 511 WURDE KREIERT;BEREICH 670 - 771

ENDE AKETTE (2.06) 0.16

0511*UNRZVF ER1 RRZE ERLANGEN 15.02.77 180120 SEITE

```

EE      EE  EEEEEEE
EEE     EE  EEEEEEEEE
EEEE    EE  EE      GEE
EE EE   EE  EE      EE
EE  EE  EE  EE      GEE
EE  EE  EE  EEEEEEEE
EE  EE  EE  EEEEEEE
EE      EEEE  EE      EE
EE      EEE  EE      EE      EEE
EE      EE  EE      EE      EEE
      EEE  EE      EEEEEEEEE
      EE  EE      EEEEEEEEE

```

AUFTRAG 0511 15.02.77 17:10 180120 (0001.01)

DRPROTOKOLL,GERAET=DR-DC2

FEHLERHALT

LFANMELDE,UNRZPB,GED

ANGEMELDET: GED(0004.00) KAT: UNRZPB

GEDAECHTNIS,GED

GEDAECHTNIS : GED(0004.00) V.11.02.77

LFANMELDE,LESEN=LFTEMP,QPROG

ANGEMELDET: QPROG(0001.00) KAT: LFTEMP

UEBERSETZE,QPROG

START PS&FTNCOMP (0074.04)

MD STDHP WURDE ERZEUGT

ANFANG PROTOKOLL

```

000010 C
000020 C
000030 C      *** DO STATEMENT ***
000040 C      *** LOGISCHES IF STATEMENT ***
000050 C      *** GOTO STATEMENT ***
000060 C
000070 C
000080      INTEGER IPLUS/1H+/, IMINUS/1H-/, ITIMES/1H*/, IDIV/1H//,
000090      1      IEXP/1HE/
000100      READ (5,1000) IANZ
000110 1000      FORMAT (I4)
000120      DO 100 I=1,IANZ
000130      READ (5,1001) IL,IOP,IR
000140 1001      FORMAT (I4,A1,I4)
000150      IF ( IOP .EQ. IPLUS ) GOTO 101
000160      IF ( IOP .EQ. IMINUS ) GOTO 102
000170      IF ( IOP .EQ. ITIMES ) GOTO 103
000180      IF ( IOP .EQ. IDIV ) GOTO 104
000190      IF ( IOP .EQ. IEXP ) GOTO 105
000200      WRITE (6,1002) IOP
000210 1002      FORMAT (1H,'***** FALSCHER OPERATOR : ',1A1)
000220      GOTO 100

```

0511*UNRZVF	ER1	RRZE	ERLANGEN	15.02.77	180120	2
-------------	-----	------	----------	----------	--------	---

```

000230 101      IERG = IL + IR
000240      GOTO 90
000250 102      IERG = IL - IR
000260      GOTO 90
000270 103      IERG = IL * IR
000280      GOTO 90
000290 104      REALL = IL
000300      REALR = IR
000310      ERG = REALL/REALR
000320      WRITE (6,1003) IL,IOP,IR,ERG
000330 1003     FORMAT (1H ,I4,X,A1,I4,3H = ,F10.4)
000340      GOTO 100
000350 105      IERG = IL ** IR
000360 90        WRITE (6,1004) IL,IOP,IR,IERG
000370 1004     FORMAT (1H ,I4,X,A1,X,I4,3H = ,I8)
000380 100      CONTINUE
000390      STOP
000400      END

```

ENDE PROTOKOLL
KEINE SYNTAXFEHLER

ENDE PS&FTNCOMP (0074.04) 0.70
 □LFDATEI,LFTEMP,BPRDG,SEQ,U1000,G18W

KREIERT: BPROG(0001,00) KAT: LFTEMP
 MONTIERE

ENDE MONTIERE (22,05) 2.12
 ▢BINAERAUS,BPROG,PROGRAMM=STOHP

FOLGENDE OBJEKTE WURDEN IN BINAERFORM AUSGEGEBEN

OPERATOR : STDHP

ENDE BINAERAUS (19.00) 2.37
 MAKETTE, LFTEMP. AUFTRAEGE (1, 0) , 769-999999, LFD
 , 124,

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0,26
AUFTRAG 511 NORMAL BEENDET

STAND DER AUFTRAGSKETTE
AUFTRAG 511 WURDE VON AUFTRAG 510 KREIERT, BEREICH 670 - 771

AUFTRAG 514 WURDE KREIERT;BEREICH 770 - 999999

ENDE AKETTE (2.06) 0.13

0514*UNRZVF ER1 RRZE ERLANGEN 15.02.77 180120 SEITE

```
EE      EE  EEEEEEE      EEEE
EEE     EE  EEEEEEEEE    EEEEEEE
EEEE    EE  EE      FEE    EE      EE
EE EE    EE  FE      FE      FF
EE  EE    EE  EE      FEE    EEEE
EE  EE    EE  EEEEEEEEE    EEEE
EE  EE    EE  EEEEEEE      EEE
EE      EEEE EE      EE    EEE      EE
EE      EEE  EE      EE    EEEEEEE
EE      EE  EE      EE    EEE      EEE
```

AUFTRAG 0514 15.02.77 17:12 180120 (0001.01)

▣DRPROTOKOLL,GERAET=DR-DC2

▣LFANMELDE,LFTEMP,BPROG,LFTEMP,EDAT

ANGEMELDET: BPROG(0001.00) KAT: LFTEMP

ANGEMELDET: EDAT(0001.00) KAT: LFTEMP

▣LFANMELDE,UNRZPB,GED

ANGEMELDET: GED(0004.00) KAT: UNRZPB

▣GEDAECHTNIS,GED

GEDAECHTNIS : GED(0004.00) V.11.02.77

▣BINAEREIN,BPROG

FOLGENDE OBJEKTE WURDEN ERSTELLT

OPERATOR : STDHP V.15.02.77

ENDE BINAEREIN (15.01) 1.44

▣STARTE,DATEI=5-EDAT

START STDHP

11 + 123 = 134

15 * 15 = 225

100 / 11 = 9.0909

2 E 20 = 1048576

9999 - 1234 = 8765

***** FALSCHER OPERATOR :)

19 * 1257 = 23883

9658 / 475 = 20.3326

12 E 5 = 248832

9 E 9 = *****

STOP

ENDE STDHP 0.15

▣AKETTE,LFTEMP,AUFTRAEGE(0001.00)

, 830-999999,LFD

, 124,

START PS&BIBTRANS (17.01)

ENDE PS&BIBTRANS (17.01) 0.24

0514*UNRZVF ER1 RRZE ERLANGEN 15.02.77 180120 2
AUFTRAG 514 NORMAL BEENDET
ENDE DER AUFTRAGSKETTE

ENDE AKETTE (2.06) 0.06

- b) Kreation einer LF-Datei oder WSP-Datei und Eintragen der in
a) zusammengestellten Aufträge
- c) Abmelden der Datei und Start der Auftragskette mit dem
Kommando AKETTE
- d) Im Ablaufprotokoll der Aufträge der Auftragskette wird der
jeweilige Stand der Auftragskette dokumentiert. Ist ein Fehler
aufgetreten, so kann man die Auftragskette durch eine ent-
sprechende Angabe zur Spezifikation Bereich an dieser Stelle
wieder aufsetzen.

3. Probleme aus der Beratung

3.1 Bemerkungen zu den Kommandos TKAPEINFUEGE und TNUMERIERE der TR440

Aus gegebenem Anlaß wird auf zwei Kommandos hingewiesen, deren unsachgemäßer Gebrauch zur Zerstörung von LFD-Dateien führen kann.

3.1.1 Einfügen von Kapiteln mit TKAPEINFUEGE

Aufruf:

□TKAP., <ZIELKAPITEL>, <QUELLKAPITEL>, <EINFUEGUNGEN> ...

Wird bei "Zielkapitel" und "Quellkapitel" der gleiche Name angegeben, dann kann die Datei zerstört werden, wenn die anzufügenden Kapitel nicht mehr in den für die Datei in der LFD reservierten Platz hineinpassen.

Abhilfe:

1. Als "Zielkapitel" eine Hilfsdatei kreieren und erst nach erfolgreichem TKAP. in die Quelldatei zurückschreiben.

z.B.

```
□DATEI,HILF,RAM,U1,M8000
□TKAP.,HILF, <QUELLKAPITEL> ,...
□TEINT.,< QUELLKAPITEL> ,I.=HILF
```

2. Wem das unter 1. beschriebene Verfahren zu umständlich ist:

- . Quelldatei bereinigen
□TN., <QUELLKAPITEL> ,NU.--
- . Quelldatei erweitern
□LFRES., <QUELLKAPITEL> ,n
- . Kapitel einfügen
□TKAP., <QUELLKAPITEL> , <QUELLKAPITEL> ,...

Dabei ist die Anzahl n so zu wählen, daß sie mindestens gleich der Satzanzahl der Einfügungen ist.

3.1.2 Numerieren von RAN-Dateien mit TNUMERIERE

Bei Dateien mit Direktzugriff (RAN oder RAM) benötigt das System eine Schlüsselliste, um auf die Sätze der Datei gezielt zugreifen zu können. Die Länge dieser Schlüsselliste ist bei RAM-Dateien proportional zur Anzahl der Sätze, bei RAN-Dateien proportional zur höchsten Satznummer (Zeilennummer).

Wird nun eine LFD-Datei vom Typ RAN neu numeriert, dann verlängert sich gegebenenfalls die Schlüsselliste. Wenn dann der Platz, der für die Datei in der LFD reserviert wurde, nicht ausreicht, kann die Datei zerstört werden. (Eine Datei ist zerstört, wenn die Schlüsselliste zerstört ist).

Abhilfe:

1. Vorzugsweise RAM-Dateien verwenden
2. Bei RAN-Dateien die Dateien vor dem Numerieren bereinigen und erweitern (vgl. 3.1)

3.2 Korrektur zum Stanzen von Lochstreifen an der TR440

In den unter Punkt 3.1 der BI6 (S. 22/23) angegebenen Beispielen zum Stanzen von Lochstreifen ist die Spezifikation GERAET des STANZE-Kommandos mit SS8-Sc4 zu besetzen.

Also: STANZE, GERAET=SS8-SC4 richtig!
anstatt STANZE, GERAET=SS8(1,0)-SC4 falsch!



3.3 Hinweise für die Benutzung der Rechananlage CYBER 172

- a) Bei Fehlern auf der Jobkarte oder auf der USER-Karte wird auf der Titelseite der Jobliste nicht wie sonst die Benutzernummer in Großbuchstaben ausgegeben, sondern ein verstümmelter interner Name, z.B. ERROACX. (ERR steht für "error".) Solche Listen werden normal zurückgegeben. Wenn Sie die Liste zu einem Job vermissen, denken Sie daher bitte an diese Möglichkeit und durchsuchen Sie die ausgelegten Listen mit außergewöhnlichen Jobnamen.
- b) Beim Ausstanzen von Lochkarten wird jeweils eine Titeltkarte erzeugt. Diese trägt jedoch nicht den Jobnamen in Großbuchstaben, sondern nur die ersten 4 Zeichen der Benutzernummer und danach eine Kombination von 3 Buchstaben.
- c) Beim Arbeiten mit Algol wird eine Felddlänge (CM-Parameter der Jobkarte) von mindestens 46000 benötigt, meistens mehr. Die automatische Reduktion der Felddlänge muß mit der Steuerkarte REDUCE,-. abgeschaltet werden, da sonst unsinnige Fehlermeldungen auftreten, z.B. "implementation restriction" bei OUTPUT-Anwendungen. Dasselbe gilt für SIMULA.
- d) Die Übersetzer an der CYBER und am TR440 (bzw. IBM) unterscheiden sich u.a. in folgendem:
Als Literalbegrenzungszeichen erwartet der TR440 einen Apostroph (z.B. 'TEXT')

Darstellung am Drucker	:	'
Am Bildschirm	:	'
Lochung KC2	:	5/8
KC3	:	4/8

Die CYBER erwartet statt dessen ein Anführungszeichen ("quote", z.B. "TEXT").

Darstellung am Drucker	:	≠
am Bildschirm	:	"
Lochung 29-Code	:	7/8
26-Code	:	4/8

Um Programme, die am TR440 dieses Zeichen verwenden und in KC2 gelocht sind, auch an der CYBER unverändert rechnen zu können, wurde als Zwischenlösung bis zur endgültigen Behebung dieses Problems der Lochkartencode "27" eingeführt, bei dem gegenüber dem Code 29 die beiden Zeichen "und" vertauscht sind. Diese Codeangabe erfolgt wie die anderen in Spalte 79 und 80 der Jobkarte.

Hinweis: Beim Duplizieren von Karten bitte stets Code 26 oder 29 verwenden.

- e) Zum Testen von FORTRAN-Programmen wird für die FORTRAN-Steuerkarte die Verwendung der Parameter ER und T empfohlen.
Beispiel: FTN,ER,T.

3.4 Sie fragen - wir antworten

Fragen zum TR 440:

- a) Benutzer haben festgestellt, daß am TR 440 gegenüber der 3300 eine zu hohe Kanalzeit abgerechnet wird.

Antwort

Die Abrechnungsformel wird derzeit anhand des Erlanger Benchmarcks überprüft. Die Abrechnung soll so erfolgen, daß der Benchmarck auf beiden Rechnern gleich teuer ist.

- b) Kontostand und Rechenzeitkosten fehlen auf den Listen oder die Rechenzeit wird auf volle DM aufgerundet.

Antwort

An dem Problem der aktuellen Kontoführung wird noch gearbeitet. Der Wert wird in Zukunft auf den Pfennig genau angegeben. Es wird jedoch wahrscheinlich nicht möglich sein, den aktuellen Kontostand anzugeben sondern einen Kontostand der 8 bis 14 Tage zurückliegt und zusätzlich die Anzahl der bisher durchgeführten Läufe.

- c) Der Aufbau einer Bibliothek ist sehr teuer am TR 440.

Antwort

Dies ist bekannt. Die Bibliotheksorganisation am TR 440 ist aufwendiger und insbesondere die Systemdienste nehmen wesentlich mehr Rechenzeit als auf der 3300 in Anspruch. Komfort kostet Rechenzeit.

- d) Die interne Uhr des TR 440 funktioniert nicht ordnungsgemäß.

Antwort

Dies ist richtig. Jeder der Rechnerkerne hat eine eigene Uhr. Es ist der Firma CGK bisher nicht gelungen, diese 3 Uhren softwaremäßig zu synchronisieren. CGK arbeitet an dem Problem; gleichzeitig wird sich die Systemgruppe des Rechenzentrums darum kümmern, um in absehbarer Zeit eine Korrektur durchführen zu können.

- e) Die Programme binärein und binäraus sind am TR 440 zu teuer.

Antwort

siehe unter c)

- f) Am TR 440 ist kein Schutz gegen einen mißbräuchlichen Zugriff auf Dateien gegeben. Über das Kommando SINFORMIERE kann man sich alles über fremde Files bekanntgeben lassen.

Antwort

Mit SINFORMIERE kann man sich nicht über Dateien informieren. LFINFORMIERE bzw. INFORMIERE liefern keine Passwörter und keine Information über fremde Privat-Dateien aus.

- g) Nicht genutzte Wartungszeiten werden nicht an die Benutzer weitergegeben.

Antwort

Es besteht die klare Anweisung an die Operateure, die bei der Hardware- oder Software-Wartung nicht genutzte Zeit voll an den Betrieb d.h. die Benutzer weiterzugeben. Bei kurzen Unterbrechungen ist dies jedoch nicht immer möglich.

- h) Falls ein Gespräch hängt, merkt dies der Operateur nicht. Der Benutzer muß dann zum Operateur gehen und veranlassen, daß dieser Bildschirm wieder freigemacht wird. Kann sich der Operateur nicht an der Konsole darüber informieren?

Antwort

Die Informationsmöglichkeiten des Operateurs sind so schlecht und umständlich, daß dies nicht möglich ist. In solchen Fällen ist über den Anrufbeantworter (7639) zu klären, ob die Maschine überhaupt noch läuft; falls ja, ist der Konsoloperator (7629) anzurufen.

- i) In den BI wird zum Arbeiten mit Binärdateien innerhalb von Bibliotheken nur die Kommandos BINAERAUS und BINAEREIN besprochen. Warum wird nicht BIBVERLAGERE empfohlen?

Antwort

Wir haben BINAERAUS und BINAEREIN empfohlen, weil dabei die Dumpfähigkeit der Montageobjekte erhalten bleibt. Nach unserer Meinung rechtfertigt dies - insbesondere im Teststadium - den erheblichen Mehraufwand an Rechenzeit. Für ausgetestete Bibliotheken, in denen keine Fehler mehr auftreten, ist BIBVERLAGERE eine angemessene Möglichkeit.

4. Ausgewählte Themen

4.1 Steuerkartenbeispiele für die CYBER

4.1.1 Steuerkarten für Magnetbänder

Normalerweise sind an der CYBER 172 Bänder mit ANSI-Labels zu verwenden. Diese Bänder werden durch die Aufsicht mit einem Band-Label versehen, der die Bandnummer und die Benutzernummer des Eigentümers enthält und der von einem Benutzer nicht geändert werden kann.

Solche Bänder werden mit der LABEL-Steuerkarte angefordert, die die Form

LABEL(Filename, Parameter₁ = Wert₁,, Parameter_n = Wert_n)
hat und diesen wichtigsten Parameter im folgenden erklärt werden sollen.

Filename: Name, unter dem die Banddatei zugegriffen werden soll.
7 Zeichen langer Name im Sinne von FORTRAN.

Parameter	Bedeutung	Mögliche Werte
PO	<u>P</u> rocessing <u>O</u> ption	R <u>R</u> ead Band ohne Schreibring auflegen W <u>W</u> rite Band mit Schreibring auflegen
VSN	<u>V</u> olume <u>S</u> erial <u>N</u> umber	Bandnummer bei Benutzerband (z.B. CX00099) SCRATCH Scratchband
SI	<u>S</u> et <u>I</u> dentifier	Name der Datenmenge bei mehreren Dateien auf einem oder mehreren Bändern
FI	<u>F</u> ile <u>I</u> dentifier	Datenbezeichnung im File Label (HDR1) 1 - 17 Zeichen. Enthält FI Sonderzeichen, ist er in \$ einzuschließen. (z.B. \$BAND-DATEI\$)
QN	<u>S</u> equ <u>e</u> nce <u>N</u> umber	Dateinummer auf dem Band (1 für die 1. Datei, 2 für die 2.) 9999 bei der Kration einer hinter der letzten Datei des Bandes anzufügenden neuen Datei.
T	<u>R</u> etention <u>T</u> ime	Ø-999: Anzahl der Tage, bis die Datei wieder überschrieben werden darf.

FA	<u>File Access</u>	<p>leer: Die Datei kann von jedem angesprochen werden.</p> <p>A : Nur der Band Eigentümer darf die Datei ansprechen</p> <p>sonst: Das angegebene Zeichen muß bei allen Zugriffen angegeben werden.</p>
R	<u>Read Label</u>	Existierenden File-Label lesen, keine Wert-Angabe
W	<u>Write Label</u>	Neuen File-Label schreiben, keine Wert-angabe

Beispiel:

```

LABEL(TAPE70,PO=W,VSN=CX0099,SI=CX0099,FI=$TAPE70-FILE$,QN=1,
      T=365,FA=A,W)

```

schreibt auf das Band CX0099 an 1. Stelle einen File mit dem Identifier TAPE70-FILE, der 1 Jahr lang nicht überschrieben und nur vom Eigentümer des Bandes verwendet werden kann. Aus einem FORTRAN-Programm würde dieser File als Kanal 70 beschrieben werden können.

Dieser Bandfile könnte in einem späteren Job etwa mit

```

LABEL(TAPE30,PO=R,VSN=CX0099,SI=CX0099,FI=$TAPE70-FILE$)

```

als Kanal 30 in einem Programm verwendet werden.

An der CD3300 in FORTRAN mit WRITE mit Formatangabe ungeblockt (d.h. ohne CALL PACKD) beschriebene Bänder können mit

```

REQUEST(TAPEnn,D=556800,C=136,MT,PO=R,F=E,LB=KU,VSN=Bandnummer)

```

angefordert und in FORTRAN als Kanal nn direkt gelesen werden.

An der TR440 mit

```

DATEI,NAME=BAND,TRAEGER=B60H(BANDNUMMER),TYP=SEQ-E,SATZBAU=G800,
SATZZAHL=U1,BLOCKUNG=1F

```

erzeugte Bänder können mit

```

REQUEST(TAPEnn,D=PE,FC=136,NT,PO=R,F=E,LB=KU,VSN=Bandnummer)
SKIPF(TAPEnn,1,C)

```

gelesen werden. Dabei wird der Label des Bandes, den die TR440 erzeugt, mit SKIPF übersprungen.

In allen Fällen können Bandnummern den Filenamen auch durch eine VSN-Karte zugeordnet werden. Dabei können auch die Folgebänder einer Bandreihe angegeben werden. Die Angaben in einer VSN-Karte haben Vorrang vor den Angaben in LABEL oder REQUEST.

Beispiel:

VSN(TAPE7Ø=CYØ1ØØ,TAPE8Ø=CZØ1Ø1/CAØ1Ø2/CBØ1Ø3)

Sind die Angaben der LABEL,REQUEST und VSN-Karten nicht in 8Ø Spalten unterzubringen, so können Fortsetzungskarten verwendet werden. In diesen muß die 1. Spalte frei bleiben. In §-Zeichen eingeschlossene Zeichenketten dürfen nicht von einer Karte auf die nächste fortgesetzt werden.

4.1.2 Einfache Ladesteuerkarten

a) Steuerung des Ladevorgangs mit LDSET

Mit den Parametern der LDSET-Karte, die im einfachsten Fall vor die LGO-Karte zu legen ist, kann der Ladevorgang in verschiedener Weise beeinflusst werden. Insbesondere können zu verwendende Benutzerbibliotheken, Vorbesetzung von Feldern und Variablen, Umfang der Montageprotokollierung und Fehlerverhalten des Laders festgelegt werden:

Format:

LDSET(Parameter₁=Wert₁,...,Parameter_n=Wert_n)

Reicht der Platz auf einer Karte nicht aus, so können mehrere LDSET-Karten verwendet werden.

Parameter	Mögliche Werte
LIB	Namen von Binärdeck-Bibliotheken, d.h. Filenamen der Hilfsbibliotheken oder Namen von Systembibliotheken. Mehrere Namen durch/zu trennen. Wird nur LIB angegeben, werden die vorhergehenden Bibliotheksdefinitionen gelöscht.
MAP	N Keine Loadermap S Statistik (kurze Liste) B (Block map (Unterprogramme, Commonblöcke) E Entry-Point-Liste X Entry Point Cross Reference Liste Mehrere Angaben aneinanderfügen (z.B. MAP=SBEX) Hinter einem / kann ein File angegeben werden, auf den die Map geschrieben werden soll.
PRESET	NONE Keine Vorbesetzung ZERO Vorbesetzung mit \emptyset und weitere
PRESETA	INF Vorbesetzung mit " ∞ " und Adresse und weitere
ERR	ALL Abbruch bei allen Fehlern FATAL Abbruch bei schweren Fehlern Fehlende oder doppelte Entrypoints sind insbesondere keine schweren Fehler

NONE

Abbruch nur bei Fehlern, die Laden unmöglich machen (z.B. angegebene Dateien existieren nicht).

LDSET-Karten können innerhalb einer Folge von Ladersteuerkarten mehrfach auftreten; dabei können Parameterwerte unterschiedlich sein.

Solche Loadersteuerkartenfolgen werden durch EXECUTE, NOGO oder eine File-Name-Karte (z.B. LGO) abgeschlossen.

Beispiel:

```
LDSET(LIB=AUX1,MAP=SBE,PRESET=ZERO,ERR=ALL)
LGO.
```

b) Absolutierung von Programmen

Ein geladenes Programm kann absolutiert auf einen File geschrieben werden. Dazu muß die Ladersteuerkartenfolge durch eine NOGO-Karte mit Angabe des Absolut-Files abgeschlossen werden.

Eine File-Name-Karte (z.B. LGO.) muß durch eine LOAD-Steuerkarte (z.B. LOAD(LGO)) ersetzt werden, da sie die Steuerkartenfolge beenden und die Ausführung des Programms anstoßen würde.

Beispiel:

```
LDSET(LIB=AUX2)
LOAD(LGO)
NOGO(ABS)
```

Dieses so erzeugte absolutierte Programm kann nun etwa mit
ABS.
aufgerufen werden.

c) Overlay-Generierung in FORTRAN

Die Anordnung der Quellenprogramme ist ähnlich wie bei der CD3300: das Hauptprogramm, jedes Overlay und jedes Segment müssen ein Programm mit PROGRAM-Statement enthalten, das als Einsprungelement für das jeweilige Overlay dient. Vor das erste Unterprogramm eines jeden solchen Overlays ist ein Overlay-Statement zu legen. Dieses enthält den Namen des Files, auf den das Overlay zu schreiben ist, die Overlay-Nummer und die Segmentnummer. Das Hauptprogramm

hat Overlay- und Segmentnummern \emptyset . Segmente müssen unmittelbar hinter dem zugehörigen Overlay liegen. Wird der Name des Overlay-files nicht angegeben, so wird der zuletzt angegebene verwendet, das Overlay-Statement des Hauptprogramms muß den Filenamen enthalten. Overlay- und Segmentnummern werden oktal angegeben und liegen im Bereich von \emptyset - 77.

Die Overlays werden durch

CALL OVERLAY(Filename,Overlay-Nr.,Segment-Nr.)
aufgerufen, geladen und gestartet.

Der Lader erzeugt bei einer der Absolutierung entsprechenden Steuerkartenfolge die in den Overlay-Statements angegebenen Overlay-Files. NOGO enthält dabei keinen Filenamen als Parameter. Das Programm wird durch eine File-Name-Karte mit dem Namen des Overlay-Files des Hauptprogramms gestartet.

Beispiel:

```
:  
FTN.  
LOAD(LGO)  
NOGO.  
OVERFIL.  
789  
    OVERLAY(OVERFIL, $\emptyset$ , $\emptyset$ )  
    PROGRAM MAIN(INPUT,OUTPUT)  
    :  
    CALL OVERLAY(7LOVERFIL,1, $\emptyset$ )  
    CALL OVERLAY(7LOVERFIL,76B, $\emptyset$ )  
    :  
    STOP  
    END  
    :  
    :  
    OVERLAY(1, $\emptyset$ )  
    PROGRAM OVER1  
    :  
    CALL OVERLAY(7LOVERFIL,1,1)
```

```
RETURN
END
:
OVERLAY(1,1)
PROGRAM SEGM11
:
RETURN
END
:
OVERLAY (76,Ø)
PROGRAM OVER76
:
RETURN
END
:
```

7₈₉

Datenkarten

6₇₈₉

d) Hilfsbibliotheks-Erzeugung

Aus einem File, der Binärdecks enthält, kann mit Hilfe von LIBGEN eine Bibliothek erzeugt und auf einen File geschrieben werden. Der Name des Files ist dabei der Bibliotheksname und muß bei jeder Verwendung wieder als Filename angegeben werden.

Format des LIBGEN-Aufrufes:

LIBGEN(F=Binärdeck-Filename,P=Bibliotheksfilename)

Beispiel:

```
BIBL(T1ØØ,CM46ØØØ)
USER(INSTØ1,)
CHARGE(INSTØ1,INSTØ1)
FTN.
LIBGEN(F=LGO,P=INSTBIB)
SAVE(INSTBIB/CT=PU,M=R)
```

7₈₉

FORTTRAN-Quellenprogramme

6₇₈₉

```

USEBIBL(T100,CM46000)
USER(INST10,)
CHARGE(INST10,INST10)
FTN.
GET(INSTLIB/UN=INST01)
LDSET(LIB=INSTLIB)
LGO.

```

7₈₉

FORTTRAN-Hauptprogramm

7₈₉

Datenkarten

6₇₈₉

4.2 FORTRAN-Unterschiede CD 3300 - CYBER 172

4.2.1 Hardware-bedingte Unterschiede

Wortlänge: 24 bit - 60 bit

Zeichensatz: unterschiedliche Internverschlüsselungen von
Zeichen, bei beiden Maschinen 6-bit-Zeichen

Abfragbare Maschinenzustände: bei der CYBER können einige
Fehlerbedingungen (ARFLT,DVFLT,
EXPOFLT) nicht getestet werden

4.2.2 Sprachbezogene Unterschiede

- a) Symbolische Namen: 8 Zeichen lang (CD 3300)
7 Zeichen lang (CYBER)
6 Zeichen lang (Norm)

b) Datentypen:

EQUIVALENCE und COMMON

Vorsicht bei S-Option CD 3300!

Unterschiede bei Verwendung von verschiedenen Datentypen
(z.B. INTEGER und REAL), die den gleichen Speicher teilen
durch Benutzung von EQUIVALENCE und COMMON.

LOGICAL: CYBER interner Wert = 0 : .FALSE.
≠ 0 : .TRUE.

Bei EQUIVALENCE von INTEGER und LOGICAL
Daten unterschiedliche Integer-Werte

CHARACTER: Typ CHARACTER auf CYBER nicht erlaubt

CD 3300: 4 bzw. 8 characters pro 'Wort'

CYBER: 10 characters pro Wort

TABLE auf CYBER nicht erlaubt.

c) COMPUTED GOTO

CYBER: Index 0 und Index Anzahl der Labels ergibt Programm-
abbruch.

d) DO-LOOPS

CYBER: Schleife wird immer mindestens einmal durchlaufen
(auch bei Endwert Anfangswert)

e) Program

CYBER: keine Task-Parameter möglich, I/O-File-Liste erforderlich!

f) Bibliotheksroutinen

Einige Routinen haben andere Namen und andere Funktion
(z.B. IFEOF - EOF)
einige Routinen sind auf der CYBER nicht vorhanden
(z.B. SELARFLT, CANARFLT,...)

g) OVERLAY

Unterschiede bei Overlay-Strukturen, zusätzliche Möglichkeiten bei der CYBER durch Lader-Steuerkarten ohne Eingriffe ins Programm.

4.3 COSY-UPDATE-Conversion

Cosy-File auf Band (Tape 10), COUP erzeugt Input für UPDATE auf TAPE11

```
COUP(T100,CM46000)
USER(XXXX)
CHARGE(XXXX,XXXX)
VSN(TAPE10=rce1 identifier)
LABEL(TAPE10,D=density,MT,I'O=R,F=S,LB=KU)
ATTACH(COUP/UN=LIBRARY)
COUP.
UPDATE(I=TAPE11,N,F,W)
FTN(I)
SAVE(NEWPL=filename)
```

6₇
8₉

Fortran-Conversion-Aids

Job-Deck auf Input wird nach TAPE11 kopiert.

Convertiertes Card-Deck auf TAPE25.

LCS-Directives siehe Conversion Aids System - Fortran
Conversion 2.1.-3 (Translation Option)

```
LCS(T100,CM46000)
USER(XXXX)
CHARGE(XXX,XXXX)
ATTACH(LCS=LCAS/UN=LIBRARY)
ATTACH(TAPE10=LSYNTAX/UN=LIBRARY)
COPYCR(INPUT, TAPE11)
LCS.
```

7₈
9

CD 3300-Deck
inclusive Job- und EOF-Karte

7₈
9

JOB

FOPT,program-name,LOOP=YES,GOTO=YES,PROGRAM=YES,LIBSUB=YES

6₇
8₉

TRANSLATION OPTIONS

User options control the actual translation process. Their format and placement is described in the CONVERSION AIDS SYSTEM, Language Conversion, Version 1 User's Guide. Directives are coded in the following form:

FOPT, name[, option (s)]

where

- name can be:

- program name
- subroutine name
- function name

options, with default values for each option underlined, can be:

MSGTXT = $\left\{ \begin{array}{c} \text{YES} \\ \underline{\text{NO}} \end{array} \right\}$

YES - The complete text of each LCP message, in addition to the message code, will appear immediately after the pertinent statement on the output listing. A message summary will not be produced.

NO - Only the message code will appear beside the pertinent statement. A message summary will be produced.

LBLORG = $\left\{ \begin{array}{c} n \\ \underline{10000} \end{array} \right\}$

n - The first statement label generated by LCP will be n, where n is an integer.

OCTAL = $\left\{ \begin{array}{c} \underline{\text{NO}} \\ \text{R/YES} \\ \text{H} \end{array} \right\}$

NO - no action will be taken by the LCP in the case of octal constants.

R/YES - each octal constant will be translated to an equivalent hollerith character string, justified right.

H - each octal constant will be translated to an equivalent hollerith character string, justified left.

LBLINC = $\left\{ \begin{array}{c} n \\ 10 \end{array} \right\}$

n - The value of the increment used to generate statement labels. For example, if:

LBLORG=100 and LBLINC=10,

then the sequence of generated statement labels is 100, 110, 120, etc.

LIBSUB = $\left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}$

YES - References to CDC 3000L library subprograms will be changed to references to the equivalent CDC CYBER 70 library subprograms. References to library subprograms with no CDC CYBER 70 equivalents will be flagged.

NO - References to both types of library subprograms above will be flagged.

TRUNCATE = $\left\{ \begin{array}{c} 0 \\ n \end{array} \right\}$

0 - 8-character identifiers will not be truncated, but all occurrences will be flagged on the output listing.

n - a digit in the range (1-7) is to be used to truncate all 8-character variables; truncation will be to size n characters. (LCP-generated variable names are not affected).

DECK = $\left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}$

YES- LCP will produce an output file for the translated program in punched card format.

LOOP = $\left\{ \begin{array}{l} \text{YES} \\ \underline{\text{NO}} \end{array} \right\}$

YES - Code will be generated to test Do loop parameters and conditionally bypass the execution of Do loops. For example, the following sequence of statements:

```
DO 10 I = I1, I2
10 X (I) = 0
will be translated to:
IF (I1.GT.I2) GO TO 10000
DO 10 I = I1, I2
10 X (I) = 0
10000 CONTINUE
```

The generated code simulates the effect of CDC 3000L FORTRAN by bypassing execution of the loop if the lower limit of the index is greater than the upper limit.

NO - This code will not be generated.

GOTO = $\left\{ \begin{array}{l} \text{YES} \\ \underline{\text{NO}} \end{array} \right\}$

YES - Code will be generated to test if the index value is out of range in a computed GO TO statement. For example, the following statement:

```
GO TO (10, 20, 30), I
will be translated to:
IF (I. LE. 1) GO TO 10
IF (I. GE. 3) GO TO 30
GO TO (10, 20), I
```

NO - This code will not be generated.

CHAR

$$= \left\{ \begin{array}{c} 0 \\ 1 \\ 4 \text{ or } 8 \\ R4 \text{ or } H4 \\ R8 \text{ or } H8 \\ 10 \end{array} \right\}$$

- 0 - This turns off the CHAR option if previously selected for a program or subprogram.
- 1 - Changes CHARACTER declarations to INTEGER.
- 4 or 8 - In the translated program translate character constants which are greater in size than the value selected by the CHAR option. The type of constant is preserved.
- R4 or H4 - Character constants will be stored four/eight
(R8 or H8) characters per word, right justified (R4/R8) or left (H4 or H8). The types of constants are also translated to comply with the setting of the option.
- 10 - Character constants in DATA statements or used as actual parameters are examined. Those greater than 10 in size are translated to two or more H-type constants, of size less than or equal to ten characters each.

CARITH

$$= \left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\}$$

- YES - This option is used in conjunction with the CHAR option, and pertains to the translation of character type variables in assignment statements. Each character element is translated into a simple expression consisting of the original character element decremented by the fixed constant 1R0. The translation is performed if CHAR = R4, H4, R8, or H8 is in effect, and is restricted to character variables appearing in right hand sides of mixed mode assignment statements where no character type left hand side exists.
- NO - This translation is not performed, but if CHAR = 4, R4, H4, 8, R8 or H8 is in effect, the character variable in the right hand side of an assignment statement as described above is flagged.

DOUBLE = $\left\{ \begin{array}{c} \text{REAL/YES} \\ \underline{\text{NO}} \end{array} \right\}$

REAL/YES - Variables declared DOUBLE PRECISION, TYPE DFP or TYPE DFPRIME are translated to type REAL. If the REQI option is also selected, only the originally declared variables are translated.

NO - No action is taken.

REQI = $\left\{ \begin{array}{c} \text{DOUBLE/YES} \\ \text{FLAG} \\ \underline{\text{NO}} \end{array} \right\}$

DOUBLE/YES - REAL type variables which are equivalenced to type INTEGER and/or type LOGICAL are translated to type DOUBLE PRECISION. Multi-equivalenced type REAL variables are flagged. If the DOUBLE = REAL option is also selected, only original type REAL variables are affected.

FLAG - REAL type variables which are equivalenced to type INTEGER and/or type LOGICAL are flagged. Multi-equivalenced type REAL variables are flagged. If the DOUBLE option is also selected, only original type REAL variables are affected.

NO - No action is taken

Note that this option pertains to M.S. FORTRAN and MASTER FORTRAN with the 24 bit integer option.

PROGRAM = $\left\{ \begin{array}{c} \underline{\text{NO}} \\ \text{YES} \end{array} \right\}$

NO - for each PROGRAM statement encountered while this option is in effect, any task parameters present will be deleted and a file list will not be inserted.

YES- for each PROGRAM statement encountered while this option is in effect, any task parameters will be deleted and a file list will be inserted consisting of an entry for each logical I/O unit explicitly referred to in a transfer statement as well as any logical units which may be implied (INPUT, OUTPUT, PUNCH).

An option is in effect until either: (1) changed by a subsequent directive, or (2) the end of the translator input is encountered.

FUNCTION IGETC AND SUBROUTINE PUTC

D

The source code for the function IGETC and the subroutine PUTC must be appended to a source program, which has been converted using the CHAR option, before compilation and execution can be attempted on the CDC CYBER 70 system.

(a) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = H4:

```
FUNCTION IGETC (CHARAY, INDEX)
INTEGER CHARAY (1)
DATA ICPW/4/
IWD=(INDEX-1)/ICPW+1
ISH=-(10-(INDEX-(IWD-1)*ICPW))*6
IGETC=AND(SHIFT(CHARAY(IWD), ISH), 77B)
RETURN
END
```

```
SUBROUTINE PUTC(CHARAY, INDEX, IVALUE)
INTEGER CHARAY(1)
DATA ICPW/4/
IWD=(INDEX-1)/ICPW+1
ISH=(10-(INDEX-(IWD-1)*ICPW))*6
CHARAY (IWD)=OR(AND(COMPL(SHIFT(77B, ISH)+77777777777B),
C CHARAY(IWD)), (SHIFT(AND(IVALUE, 77B), ISH)+55555555555B))
RETURN
END
```

(b) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = H8:

IGETC same as in (a) except initialize ICPW to 8.

```
SUBROUTINE PUTC(CHARAY, INDEX, IVALUE)
INTEGER CHARAY(1)
DATA ICPW/8/
IWD = (INDEX-1)/ICPW + 1
ISH = (10-(INDEX-(IWD-1)*ICPW))*6
CHARAY(IWD) = OR(AND(COMPL(SHIFT(77B, ISH)+7777B),
C CHARAY(IWD)), (SHIFT(AND(IVALUE, 77B), ISH)+5555B))
RETURN
END
```

- (c) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = 10;

IGETC same as in (a) except initialize ICPW to 10.

```

SUBROUTINE PUTC (CHARAY,INDEX,IVALUE)
INTEGER CHARAY(1)
DATA ICPW/10/
IWD=(INDEX-1)/ICPW+1
ISH=(10-(INDEX-(IWD-1)*ICPW))*6
CHARAY(IWD)=OR(AND(COMPL(SHIFT(77B,ISH)),
C CHARAY(IWD)), (SHIFT(AND(IVALUE, 77B),ISH))
RETURN
END

```

- (d) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = R4:

```

FUNCTION IGETC(CHARAY, INDEX)
INTEGER CHARAY(1)
DATA ICPW/4/
IWD = (INDEX-1)/ICPW+1
ISH = -(4-(INDEX-(IWD-1)*ICPW))*6
IGETC = AND(SHIFT(CHARAY(IWD), ISH), 77B)
RETURN
END

```

```

SUBROUTINE PUTC(CHARAY, INDEX, IVALUE)
INTEGER CHARAY(1)
DATA ICPW/4/
IWD = (INDEX-1)/ICPW+1
ISH = (4-(INDEX-(IWD-1)*ICPW))*6
CHARAY(IWD)=OR(AND(COMPL(SHIFT(77B, ISH))+
C 777777777777700000000B), CHARAY(IWD)),
C (SHIFT(AND(IVALUE, 77B), ISH)))
RETURN
END

```


- (e) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = R8:

```
FUNCTION IGETC(CHARAY, INDEX)
INTEGER CHARAY(1)
DATA ICPW/8/
IWD = (INDEX-1)/ICPW+1
ISH = -(8-(INDEX-(IWD-1)*ICPW))*6
IGETC = AND(SHIFT(CHARAY(IWD), ISH), 77B)
RETURN
END

SUBROUTINE PUTC(CHARAY, INDEX, IVALUE)
INTEGER CHARAY(1)
DATA ICPW/8/
IWD = (INDEX-1)/ICPW+1
ISH = (8 - INDEX - (IWD - 1)*ICPW))*6
CHARAY(IWD) = OR(AND(COMPL(SHIFT(77B, ISH)+
C 77770000000000000000000000000000B), CHARAY(IWD)),
C (SHIFT(AND(IVALUE, 77B), ISH)))
RETURN
END
```

- (f) Function IGETC and Subroutine PUTC corresponding to option selected CHAR = 4 or 8:

Under this setting of the CHAR option, only those character constants are converted which are bigger in size than the option selected (4 or 8). This, character storage and justification in character arrays must be consistent in order to use the routines provided. If this is so, the IGETC, PUTC of (a) or (d) or the IGETC, PUTC of (b) or (e) should be used corresponding to CHAR = 4 or CHAR = 8 respectively.

4.4 Übertragung von Daten von der CD 3300 auf die CYBER mit CDPICK

Das Programm CDPICK liest einen MASTER-Standard-geblockten File von einem 7-Spur-Band (TAPE2o) und überträgt die BCD-Records auf einen File (TAPE3o).

Die Benutzung erfolgt gemäß nachfolgendem Beispiel:

```
CDPICK (T2oo,CM35ooo)
USER (usernumber,password)
CHARGE (usernumber,usernumber)
GET (CDPICK/UN=UNRZPB)
VSN (TAPE2o=reelid)
LABEL (TAPE2o,D=8oo,MT,PO=R,F=S,LB=KU,NS=18)
CDPICK.
SAVE (TAPE3o)
6789   EOI           oder andere Verarbeitung von TAPE3o
                        nach vorherigem Rewind
```

Danach stehen die BCD-Records des ersten Files des Bandes reelid auf der permanenten Datei TAPE3o.

Alle Binary-Records werden überlesen.

Das Band sollte auf der CD 3300 ohne System-Noise-Records (FORMAT-Steuerkarte s.u.) geschrieben sein.

\$FORMAT(dsi,,NSNR)

4.5 Steuerkarten für UPDATE

1. Grundsätzliches

UPDATE ist ein sehr leistungsfähiger Editor, der (ähnlich wie COSY und CUP an der CD 3300) vor allem für die Haltung und Wartung von Quellenprogrammen gedacht ist. Im Gegensatz zu COSY/CUP werden dabei nicht nur die Originalkarten, sondern auch die Korrekturen in komprimierter Form gehalten. Die Korrekturen sind in einem UPDATE-File an den Stellen eingefügt, an die sie durch entsprechende Direktiven plaziert wurden; mit DELETE gelöschte Karten bleiben im UPDATE-File erhalten und werden nur als inaktiv gekennzeichnet (Reaktivierung erfolgt z.B. nicht wie bei CUP durch (CUP-)DELETE der (COSY-)DELETE-Karten, sondern durch RESTORE der inaktivierten Karten); solche inaktiven Karten werden nicht auf den dem SHO-File entsprechenden COMPILE-File geschrieben. Alle Korrekturen werden mit einem sog. Correction Set Identifier benannt, die eingefügten Korrekturtextkarten werden innerhalb des jeweiligen Correction Sets fortlaufend durchnummeriert und so (ohne Bezug auf das Quellenprogrammdeck, zu dem sie gehören) identifiziert. Korrekturtextkarten können nicht nur einzeln wieder inaktiviert werden, dies kann auch für eine Korrektur als ganze vorgenommen werden.

Bei jeder Korrektur wird ein vollständiger neuer UPDATE-File (entsprechend einem neuen COSY-File) erzeugt, wobei sämtliche Numerierungen erhalten bleiben.

2. UPDATE-Steuerkarte

Format : UPDATE (Option₁ = Wert₁, ..., Option_n = Wert_n)

Hier sollen nur die wichtigsten Options erwähnt werden.

Option	Bedeutung	Mögliche Werte
C	Filename für editierte Quellenprogramme als Eingabe für <u>C</u> ompiler	Keine Angabe oder C: Quellenprogramme auf File COMPILE
		C = 0 : Keine Ausgabe von Quellenprogrammen
		C = File : Ausgabe auf dem spezifizierten File
F	<u>F</u> ull Update	keine Angabe : COMPILE-File enthält nur die geänderten Programme.
		F : COMPILE-File enthält alle Programme des UPDATE-Files

I	Input für Direktiven	Keine Angabe oder I: I = File	Steuerkarten auf INPUT Steuerkarten auf dem spezifizierten File
N	New Program Library	Keine Angabe N N = File	: Kein neuer, geän- derter UPDATE-File : Neuer UPDATE-File NEWPL : Neuer UPDATE-File auf dem angegebe- nen File
P	Old Program Library	Keine Angabe oder P: P = File	: Alter UPDATE-File auf OLDPL : Alter UPDATE-File auf dem spezifi- zierten File

3. Format der UPDATE-Direktiven

UPDATE-Direktiven beginnen mit einem * in Spalte 1 gefolgt von einem Schlüsselwort; die Parameter folgen hinter einem Leerzeichen und werden untereinander durch Kommata getrennt.

Beispiel:

*COMPILE PROG1,SUB1,FUN1

bewirkt, daß die Quellendecks PROG1, SUB1 und FUN1 auch dann auf den COMPILE-File geschrieben werden, wenn keine Korrekturen an ihnen gemacht wurden.

Namen

Namen von Decks oder Korrekturen (Correction Set Identifiers) bestehen aus 1-9 Zeichen langen Folgen der Zeichen

A-Z 0-9 + - * / () \$ =

Beispiel:

DECK 723A

ERROR1

Kartenidentifikationen

In UPDATE-Direktiven werden Quellenprogrammarten des UPDATE-Files durch Angabe von Deck- oder Korrekturname und der laufenden Nummer innerhalb des Decks bzw. der Korrektur identifiziert; diese beiden Angaben werden durch einen Punkt getrennt. Die laufende Nummer bezieht sich bei Korrekturen nur auf die Quellenprogrammarten, die durch die Korrektur eingefügt wurden.

Die Angabe des Deck- bzw. Korrekturnamens und des Punktes kann entfallen, wenn sich die zu identifizierende Karte zum gleichen Deck bzw. zur gleichen Korrektur gehört wie die letzte Karte, bei der die Kartenidentifikation vollständig angegeben wurde.

Beispiel:

*DELETE PROG1.318,379

Die Karten 318 bis 379 in Deck PROG1 werden als inaktiv gekennzeichnet, sie bleiben im UPDATE-File, werden jedoch nicht auf den COMPILE-File geschrieben.

4. Erstellung eines UPDATE-Files

Die Quellen-Programme, aus denen der UPDATE-File erzeugt werden soll, müssen auf INPUT stehen, vor jedem Quellendeck muß eine *DECK-Karte liegen, die den Namen des Quellenprogramms enthält

Beispiel:

```

:
UPDATE(N,F)
SAVE (NEWPL=QUELLE1)
FTN(I)
LIBGEN(F=LGO,P=USERLIB,NX)
SAVE(USERLIB)
```

7₈₉

```
*DECK  PROG1
      PROGRAM  PROG1
      :
      :
      END
*DECK  SUB1
      SUBROUTINE SUB1
      :
      :
      END
*DECK  FUN1
      FUNCTION FUN1(I)
      :
      :
      END
```

67₈₉

Stehen die Quellenprogramme auf Files, so können etwa folgende Direktiven verwendet werden:

```
*DECK PROG1
*READ FILEP1
*DECK SUB1
*READ FILES1
*DECK FUN1
*READ FILEF1
```

5. Hinzufügen von Decks auf einem UPDATE-FILE

Auf dem INPUT-File müssen hinter einer *ADDFILE-Direktive die neuen Quellendecks mit ihren zugehörigen *DECK-Karten liegen.
Beispiel:

```
:
:
GET(OLDPL=QUELLE1)
UPDATE(N)
SAVE(NEWPL=QUELLE2)
FTN(1)
GET(ULIBMOD/UN=LIBRARY)
GET(USERLIB)
CALL(ULIBMOD(LIBNAME=USERLIB)
SAVE(OLDLIB)
REPLACE(USERLIB)
:
:
7 8 9
*ADDFILE
*DECK SUB2
      SUBROUTINE SUB2
6 7 8 9
```

Steht das Quellenprogramm auf einem File, lauten die Direktiven etwa:

```
*ADDFILE
*DECK SUB2
*READ FILES2
```

6. Korrektur von einzelnen Quellenprogrammkarten

Alle Korrekturen müssen mit einem Korrekturnamen versehen werden, mit dem die einzelnen durch die Korrektur eingefügten Quellenprogrammkarten (fortlaufend durchnummeriert) identifiziert werden. Dieser Korrekturname wird auch bei den durch die Korrektur (z.B. mit *DELETE) inaktivierten oder (z.B. mit *RESTORE) reaktivierten Quellenprogrammkarten vermerkt, die zum ursprünglichen Deck- oder zu anderen Korrekturen gehören.

Diese Benennung geschieht durch die Direktive
*IDENT Korrekturname.

Ein so definierter Korrekturname bezieht sich auf alle folgenden Korrektur-Direktiven (die andere Quellprogrammkarten deaktivieren oder reaktivieren können) und durch die Korrektur eingefügten Quellprogrammkarten bis zur nächsten *IDENT-Direktive.

Auf eine solche *IDENT-Direktive können insbesondere folgende Korrekturdirektiven folgen:

*INSERT Kartenidentifikation

Einfügung von neuen Quellprogrammkarten hinter die angegebene Karte

*BEFORE Kartenidentifikation

Einfügung von neuen Quellprogrammkarten vor die angegebene Karte.

*DELETE Kartenidentifikation bzw.

*DELETE Kartenidentifikation₁, Kartenidentifikation₂.

Die angegebene Karte bzw. die Karten zwischen den beiden angegebenen Karten werden deaktiviert. Bereits deaktivierte Karten im angegebenen Bereich bleiben inaktiv.

*RESTORE Kartenidentifikation bzw.

*RESTORE Kartenidentifikation₁, Kartenidentifikation₂.

Die angegebene Karte bzw. die Karten zwischen den beiden angegebenen Karten werden reaktiviert. Aktive Karten im angegebenen Bereich bleiben aktiv.

Auf *DELETE und *RESTORE können ebenfalls neue Quellenprogrammkarten folgen. Diese werden wie bei *INSERT am Ende des angegebenen Bereichs eingefügt.

Beispiel:

```

:
GET(OLDPL=QUELLE2,USERLIB)
UPDATE(N)
SAVE(NEWPL=QUELLE3)
FTN(I)
GET(ULIBMOD/UN=LIBRARY)
CALL(ULIBMOD(LIBNAME=USERLIB)
REPLACE(USERLIB,OLDLIB)
789
*IDENT KORR1
*DELETE PROG1.30,70
    CALL SUB2(A,B)
*INSERT 85
    IF(A .NE. 0.) CALLSUB2(A,C)
*INSERT SUB2.7
    IF(P1 .EQ. 0.) GOTO 3
*IDENT KORR2
*INSERT SUB1.12
    A=0.
6789
```

Bei folgenden Korrekturen könnte nun z.B. die Quellprogrammkarte
IF(P1 .EQ. 0.) GOTO 3
mit KORR1.3 angesprochen werden.

7. Rücknahme ganzer Korrekturen

Die Wirkung einer ganzen Korrektur (d.h. Deaktivieren, Reaktivieren und Einfügen von Karten u.s.w.) kann durch die *YANK-Direktive aufgehoben werden. Dadurch werden Karten die durch die ursprüngliche Korrektur deaktiviert wurden reaktiviert und umgekehrt, eingefügte Karten werden deaktiviert.

Eine *YANK-Direktive ist selbst Teil einer Korrektur.

Beispiel:

```
*IDENT  ANTIKORR1
*YANK   KORR1
ist äquivalent zu
*IDENT  ANTIKORR1
*RESTORE PROG1.30,70
*DELETE KORR1.1
*DELETE 2
*DELETE 3
```

8. Verwendung von Common Decks

Enthalten mehrere Quellendecks innerhalb eines UPDATE-Files gleiche Codestücke (z.B. gemeinsame Tabellendefinitionen), so ist es günstig, diese zu einem sog. COMMON-Deck zusammenzufassen und dessen Inhalt von UPDATE mit Hilfe der *CALL-Direktive an den gewünschten Stellen in den COMPILE-File einfügen zu lassen. Die Definition eines solchen Common-Decks geschieht mit

```
*COMDECK Deckname
```

anstelle der *DECK-Direktive. Es wird mit der Direktive

```
*CALL Deckname
```

an der Stelle der *CALL-Direktive beim Erstellen des COMPILE-Files eingesetzt; die *CALL-Direktive wird innerhalb des UPDATE-Files als Quellenprogrammkarte verwaltet (kann also z.B. mit *DELETE entfernt werden) und erscheint nicht im COMPILE-File. Wird ein Common-Deck korrigiert, werden alle Decks, die ein *CALL für das veränderte Common-Deck enthalten, auf den COMPILE-File geschrieben (also neu übersetzt).

Beispiel:

```
*COMDECK TABDEF
      COMMON/TABLE/IV1(20),IY(13),Z(8),PQ(3)
      INTEGER Z
      COMPLEX PQ
*DECK MAIN
      PROGRAM MAIN(INPUT,OUTPUT)
      :
*CALL TABDEF
      :
      END

      SUBROUTINE A
```

*CALL TABDEF

:

END

Die Vereinbarung des COMMON/TABLE/ wird dadurch in MAIN und A eingefügt.

9. Abkürzungen für Direktiven

Die Direktiven können auch abgekürzt werden:

Langform	Kurzform	Langform	Kurzform
*ADDFILE	*AF	*DELETE	*D
*BEFORE	*B	*IDENT	*ID
*CALL	*CA	*INSERT	*I
*COMDECK	*CD	*READ	*RD
*COMPILE	*C	*RESTORE	*R
*DECK	*DK	*YANK	*Y

A n h a n g :

TR44o-
Programmbeschreibungen: S5UNPK
S5PACK
FORMA1
FORMA4

TR44o-
Kommandos

AKETTE

Regionales Rechenzentrum Erlangen

Programmbibliothek TR440-&STARG

I FORTRAN	I S5UNPK	I	I
I Prozedur	I Umspeichern eines A4-Feldes	I	I
I	I in ein A1-Feld	I	I
I	I	I Datum	I
I TAS	I	I 02.77	I

Das Unterprogramm S5UNPK ist in der Quellsprache TAS geschrieben.

Unterprogrammanschluss:

```
SUBROUTINE S5UNPK(A,B,N)
INTEGER A(*), B(*), N
```

Programmbeschreibung:

A(1), A(2), ..., A(n) enthalten Zeichen im A4-Format. Die ersten N Zeichen werden in B(1), B(2), ..., B(N) im Format A1 abgespeichert. Die weiteren Elemente von B werden nicht verändert.

Regionales Rechenzentrum Erlangen
 Programmbibliothek TR440-6STARG

I	FORTTRAN	I	S5PACK	I		I
I	Prozedur	I	Umspeichern eines A1-Feldes	I		I
I		I	in ein A4-Feld	I		I
I		I		I	Datum	I
I	TAS	I		I	02.77	I

Das Unterprogramm S5PACK ist in der Quellsprache TAS geschrieben.

Unterprogrammabschluss:

SUBROUTINE S5PACK(A,B,N)
 INTEGER A(*), B(*), N

Programmbeschreibung:

A(1), A(2), ..., A(n) enthalten Zeichen im A1-Format. S5PACK packt diese Zeichen im A4-Format in das Feld B. Dabei werden nur so viele Elemente von B verändert wie nötig. Das letzte benötigte Element von B wird rechtsbündig mit Blank aufgefüllt.

STAENDIGE ARBEITSGRUPPE DER TR440 - RECHENZENTREN
ANWENDERPROGRAMM-BIBLIOTHEK

I FORTRAN-	I FORM1	I Bestell-Nr.	I
I Prozedur	I UMSPEICHERN VON ZWEI A4-VARIABLEN	I E7.HH.01.04	I
I	I AUF ACHT A1-VARIABLEN	I	I
I	I	I Datum	I
I TAS	I	I 09.73	I

Das Unterprogramm FORM1 ist in der Quellsprache TAS
geschrieben.

Unterprogrammanschluß:
SUBROUTINE FORM1 (IEIN,IAUS)
DIMENSION IEIN(2), IAUS(8)

PROGRAMMBESCHREIBUNG

=====

FORM1 SPEICHERT 2 IM FORTRAN-FORMAT-CODE A4 AUF DAS
FELD IEIN EINGELESENE ZEICHENKETTEN UM AUF DAS FELD IAUS.
IAUS ENTHAELT DIE ZEICHEN IM FORTRAN-FORMAT-CODE 8A1

PROGRAMMBESCHREIBUNG VOM 05.03.75
ZOELLER,G., RECHENZENTRUM DER UNIVERSITAET
Hamburg

STAENDIGE ARBEITSGRUPPE DER TR440 - RECHENZENTREN

ANWENDERPROGRAMM-BIBLIOTHEK

I FORTRAN-	I FORMA4	I Bestell-Nr.	I
I Prozedur	I UMSPEICHERN VON ACHT A1-VARIABLEN	I E7.HH.01.03	I
I	I AUF ZWEI A4-VARIABLEN	I	I
I	I	I Datum	I
I TAS	I	I 09.73	I

Das Unterprogramm FORMA4 ist in der Quellsprache TAS
geschrieben.

Unterprogrammanschluß:
SUBROUTINE FORMA4 (IEIN, IAUS)
DIMENSION IEIN(8), IAUS(2)

PROGRAMMBESCHREIBUNG

=====

FORMA4 SPEICHERT 8 IM FORTRAN-FORMAT-CODE A1 AUF DAS FELD IEIN
EINGELESENE ZEICHEN UM AUF DAS FELD IAUS.
IAUS ENTHAELT DIE 8 ZEICHEN IM FORTRAN-FORMAT-CODE 2A4.

PROGRAMMBESCHREIBUNG VOM 05.03.75
ZOELLER, G., RECHENZENTRUM DER UNIVERSITAET
Hamburg

AKETTE

Abarbeiten von Auftragsketten

Spezifikation :

- | | | |
|-------|---------|---|
| 1 | DATEI | Angabe der zu bearbeitenden Datei |
| 2 | BEREICH | Angabe des Dateibereiches, der bearbeitet werden soll |
| 3 | TRAEGER | Angabe zu Dateiträger |
| ----- | | |
| 4 | FLS | Angabe zur Codierung des Fluchtsymbols in der Datei |
| 5 | MODUS | Angabe zum Modus |

Kommando der RRZE-Programmbibliothek

anlagenspezifische
Voreinstellung :

Einschränkung : Das Kommando ist nur zulässig für Texthaltungsdateien auf LFD oder Wechselplatte.
Weitere Einschränkungen siehe Beispiel.

Wirkung : Die Datei wird in dem angegebenen Bereich auf das erste Auftreten eines XBA-Kommandos untersucht. Die im XBA-Kommando angegebenen Spezifikationswerte werden für die Kreation eines neuen Abschnittes übernommen. Nicht besetzte Spezifikationen werden mit Ausnahme der Spezifikationen für Bandgerätebedarf (BGB, B60, usw.) vom aufrufenden Auftrag übernommen. Der Bereich der Datei vom XBA-Kommando bis zu einem weiteren XBA-Kommando bzw. bis zum Datei- oder Bereichsende wird in ein Hintergrundgebiet transportiert. Hinter den letzten Satz wird ein neues AKETTE-Kommando eingefügt. Anschließend wird ein Abschnitt kreiert. Bei einem Folgeaufruf wird geprüft, ob der kreierte Abschnitt mit Erfolg abgelaufen ist und anschließend ein weiterer Auftrag kreiert bis der unter Bereich angegebene Dateibereich abgearbeitet ist. Gegen die Fortsetzung der Auftragskette kann man sich mit einem Kommando FEHLERHALT schützen

1 DATEI

DATEI

Angabe der zu bearbeitenden Datei

Spezifikation :

DATEI : Name der zu bearbeitenden Datei

KAT. DATEI : Datei im Katalog KAT

Der Katalogname kann ein Dateimengen-
oder Benutzerkennzeichen sein.

Bei passwortgeschützten Dateien muß das
Schreibpasswort abgegeben werden.

obligate Spezifikation zum Kommando AKETTE

anlagenspezifische
Voreinstellung : -

Einschränkung :

Wirkung : Die angegebene Datei wird bearbeitet. Sie darf nicht
eingeschleust sein.

2 BEREICH

BEREICH

Angabe des Dateibereiches,
der bearbeitet werden soll.

Spezifikation :

a - b : es wird der Zeilenbereich a - b der
Datei bearbeitet.

obligate Spezifikation des Kommandos AKETTE

anlagenspezifische
Voreinstellung : 1-999 999

Einschränkung :

Wirkung : Der angegebene Zeilenbereich der Datei wird bearbeitet.

3 TRAEGER

TRAEGER

Angabe zu Dateiträger

Spezifikation :

LFD : es soll eine LF-Datei bearbeitet werden
W32 (exdkz) : es soll eine Datei auf einer Wechselplatte
WSP 432 bearbeitet werden
W14 (exdkz) : es soll eine Datei auf einer Wechselplatte
WSP 414 bearbeitet werden

obligate Spezifikation zum Kommando AKETTE

anlagenspezifische
Voreinstellung : -

Einschränkung :

Wirkung :

FLS

Angabe zur Codierung des Fluchtsymbols

Spezifikation :

- : Es wird die Oktade '35' als Fluchtsymbol interpretiert
- Z : Die angegebene Dezimalzahl wird umgeschlüsselt und als Fluchtsymbol interpretiert.
- Z : 53 (Oktade '35' = FL)
- 114 (Oktade '72' = #)
- 115 (Oktade '73' = §)
- 124 (Oktade '7o' = □)

optimale Spezifikation zum Kommando AKETTE

anlagenspezifische Voreinstellung :	124
--	-----

Einschränkung :

Wirkung : Beim Transport aus der Datei in das Hintergrundgebiet wird die angegebene Oktade in die Zentralcodeoktade '35' (FL) umgeschlüsselt. Die Adressen der Fluchtsymbole werden in der Fluchtsymbolverweisliste vermerkt, die den Entschlüssler als Grundlage zur Abarbeitung des Hintergrundgebietes dient.

5 MODUS

MODUS

Angabe zum Modus

Spezifikation :

- : Der nächste Auftrag wird bearbeitet
- NORM : Die Datei wird normiert

optimale Spezifikation zum Kommando AKETTE

anlagenspezifische Voreinstellung : -

Einschränkung : Z.Z. ist nur der MODUS NORM möglich

Wirkung : Beim Normieren wird die Datei normiert und entsprechend den Spezifikationen 1-3 eine neue Auftragskette erzeugt und bearbeitet.
Der Stand der Auftragskette wird in der Datei dokumentiert.
Bei einem Fehler innerhalb der Auftragskette wird bei einem weiteren AKETTE-Kommando an der Fehlerstelle aufgesetzt und die unterbrochene Kette hier fortgesetzt.

Fragebogen zur Ermittlung des Umstellungsaufwandes

Bitte für jedes Programm(paket) einen gesonderten Fragebogen ausfüllen.

1. Name des Benutzers: Aktuelle Accounting-Nummer:

--	--	--	--	--	--

Institut:
.....

2. Programm(paket)name:
Kurzbeschreibung:
.....

Falls das Programm neu geschrieben wird und keinen Vorgänger auf einer anderen Anlage hatte, bitte im nebenstehenden Kreis an der Seite ankreuzen. Der Rest des Fragebogens erübrigt sich in diesem Falle.



Programmiersprache: Sprach-Standard ☐ eingehalten
☐ weitgehend eingehalten
☐ nicht eingehalten

Umfang: Umfang (Daten):
(Zahl der Programmanweisungen)

3. Die Umstellung ist zu% abgeschlossen und erfolgte
von der Anlage auf die Anlage.
Auf der Ausgangsanlage wurde dabei folgende Acc.Nr. benutzt:
Pro Jahr betrug der Rechenzeitverbrauch CPU-Std. und die Zahl der
Produktionsläufe betrug
Auf der neuen Anlage wird für dieses Programmpaket ein Rechenzeit-Verbrauch von
.....CPU-Std. bei insgesamt Produktionsläufen erwartet.

Die Umstellung erstreckte sich über einen Zeitraum von Monaten und erforderte CPU Stunden auf der Anlage mit Mann-Monate Personaleinsatz.

4. Als Datenträger für die Übernahme wurden benutzt:
bei Programmen: ☐ Karten ☐ Band ☐ Lochstreifen
bei Daten: ☐ Karten ☐ Band ☐ Lochstreifen

5. Bei der Umstellung wurden folgende Kommandos/Programme benutzt:

	mit Schwierigkeiten	mit Erfolg	ohne Erfolg	weil
FTNPRE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SAM	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CDPICK	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WANDLE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CYBER/CONV.AIDS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SONSTIGE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Umstellungsprobleme ☐ keine
☐ geringfügig, nicht erwähnenswert
☐ normal, wie erwartet
☐ groß
☐ so groß, daß Umstellung noch nicht erfolgreich
beendet werden konnte
☐ Umstellung abgebrochen, Neuprogrammierung erforderlich

7. Worin liegen die Umstellungsschwierigkeiten?

- ☐ bei Übernahme der Programme
- ☐ bei Übernahme der Daten
- ☐ umzustellende Programme nicht gemäß Standard
- ☐ im wesentlichen nur Steuerkartenprobleme
- ☐ geringfügige Programmänderungen
- ☐ völlige Neukonzeption
- ☐ mangelnde Dokumentation
- ☐ kein Ersatz für Unterprogramme der CD 3300 vorhanden
es fehlen z.B.
- ☐ mangelnde Unterstützung durch das Rechenzentrum in
folgenden Punkten:
.....
.....
- ☐ sonstige Schwierigkeiten
.....
.....

8. Besondere Schwierigkeiten bei der Umstellung entstanden durch:

.....
.....
.....

9. Wodurch hätten Ihre Umstellungsprobleme beseitigt oder zumindest stark reduziert werden können?

.....
.....
.....

10. Hier ist noch Platz für Antworten auf Fragen, die wir versäumt haben zu stellen:

