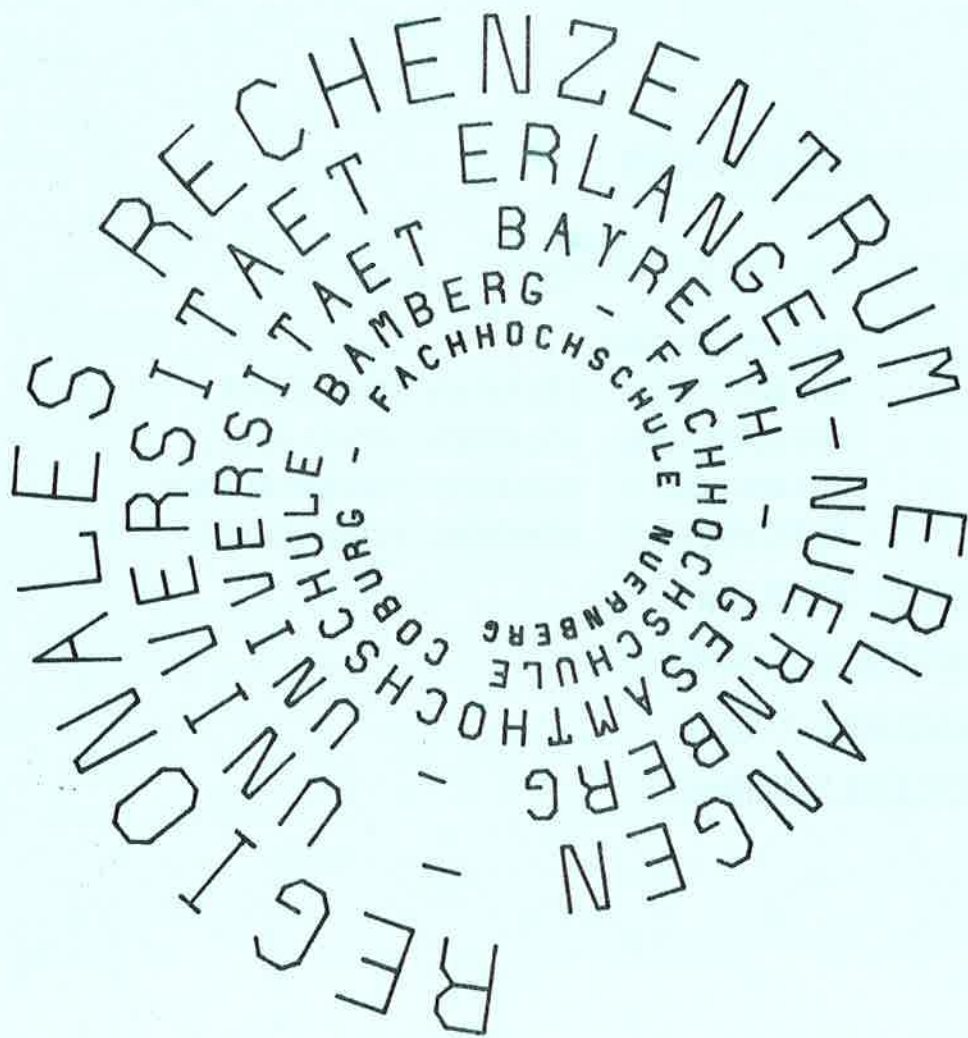


RRZE

BENUTZERINFORMATION



BI 3 - ERLANGEN - 18. OKTOBER 1976

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: Vorwort: Stand der Realisierung des RRZE

EINFÜHRUNG IN DAS SYSTEM TR 440

1. Konfigurationsübersicht
2. Übersicht Hardware (Rechner)
3. Übersicht Betriebssystem
4. Übersicht Programmiersystem

Stand der Realisierung des RRZE

Das RRZE befindet sich bereits in der Realisierungsphase, ist also über die Planungsphase weitgehend hinaus. Die Verträge für den Dreifachprozessor TR 440 und die CYBER 172 sind rechtskräftig unterschrieben. Die Liefertermine stehen fest:

Anlieferung TR 440	19.10.76
Anlieferung CYBER 172	10. 1.76

d.h. der in der BI 2 angegebene Terminplan für die zentralen Systeme wird eingehalten.

Bei den Stapelfernverarbeitungsstationen wurde die Herstellerfrage nochmals diskutiert und endgültig festgelegt. Jetzt ist zu hoffen, daß auch hier die Verträge zügig abgeschlossen werden können. Das RRZE wird sich unmittelbar nach Vertragsabschluß mit den Außenstellen in Verbindung setzen um die Installation, mit der frühestens 4-5 Monate später zu rechnen ist, vorzubereiten (Klima und Elektroversorgung in den vorgesehenen Räumen, Telefonanschlüsse, Einrichtung, gemeinsame Beschaffung der Locher).

Der Probetrieb des Dreifachprozessors TR 440 kann voraussichtlich in der ersten Novemberwoche beginnen. Hierzu vorab einige organisatorische Hinweise (beachten Sie bitte jeweils die aktuellen Aushänge).

1. Der Probetrieb dient zur Überprüfung des gelieferten Hard- und Softwaresystems. Benutzerprogramme können während dieser Zeit kostenlos bearbeitet werden. Dem Charakter des Probetriebes entsprechend können jedoch keine festen Rückgabzeiten garantiert werden.

II

2. Mit der Aufnahme des Probebetriebes wird die Beratung am RRZE wesentlich verstärkt:
 - a) Die "Aufsicht" wird wie bisher weitergeführt, wird sich jedoch im wesentlichen auf organisatorische Probleme beschränken.
 - b) Über die Aufsicht steht tagsüber jeweils ein wissenschaftlicher Mitarbeiter zur programmtechnischen Beratung zur Verfügung.
3. Sobald der Probebetrieb zufriedenstellend läuft, wird der Stapelbetrieb am TR 440 der Informatik stark eingeschränkt und spätestens mit Abschluß des Probebetriebes ganz eingestellt.
4. Benutzer aus der Region können selbstverständlich ebenfalls Rechenkapazität auf dem Dreifachprozessor in Anspruch nehmen, sind jedoch vorerst auf den Postweg angewiesen. Interessenten setzen sich bitte mit der Aufsicht (Tel. 85-7039) in Verbindung.
5. Das nächste Benutzerkolloquium findet am 2.11.1976 um 16.15 Uhr im Hörsaal H4 statt (als erste Veranstaltung des RZ-Kolloquiums im WS). Hierbei sollen insbesondere organisatorische Probleme besprochen werden.

Um allen Benutzern die keine Zeit hatten, einen Einführungskurs zu besuchen, die Gelegenheit zu geben, sich mit den Eigenschaften des TR 440 bekannt zu machen, werden im folgenden einige Informationen über Hard- und Software zusammengestellt.

1. Konfigurationsübersicht

Das Regionale Rechenzentrum Erlangen wird mit einem Dreifachprozessor TR 440 ausgestattet. In bezug auf Rechenleistung und Speicherkapazität wird es sich dabei um die derzeitig größte Anlage dieses Typs handeln. Einzelheiten ergeben sich aus der folgenden Geräte-
liste (ABB. 1 zeigt die entsprechende Konfigurationsskizze):

Dreikernanlage	siehe Datenblatt
348 K 48 Bit-Worte Halbleiterspeicher	Rechnerkern
8 Standardkanalwerke, Übertragungsrate max. 700.000 Zeichen/s	
5 Schnellkanalwerke, Übertragungsrate max. 3 Mio. Zeichen/s	
Operateurplatz	
5 Plattenstrecken mit 10 (je 2) Wechsellattenspeichern (à 140 Millionen Zeichen) 38.95 ms mittlere Zugriffszeit, Übertragungsrate max. 800.000 Zeichen/s	
2 Magnetbandgeräte (7-Spur, 556 bzw. 800 bpi)	
6 Magnetbandgeräte (9-Spur, 800 bzw. 1.600 bpi)	
Lochstreifenleser, 2000 Z/sek.	
Lochstreifenstanzer, 150 Z/sek.	
2 Lochkartenleser, 1200 Karten/min.	
1 Lochkartenstanzer, 250 Karten/min.	
2 Schnelldrucker, 1200 Zeilen/min., kleiner Zeichenvorrat (63)	
1 Schnelldrucker, 600 Zeilen/min., großer Zeichenvorrat (114)	
Datenfernübertragungseinheit DUET mit Anschlüssen, siehe Abb. 2	
30 Sichtgeräte SIG 51-8, 20 Zeilen à 80 Zeichen, Groß-/ Kleinschreibung (94 Zeichen)	
10 Fernschreiber FSR 208-4, max. 118 Zeichen/Zeile, Groß-/ Kleinschreibung (95 Zeichen), 300 baud	

Ein am Universitätsrechenzentrum bereits vorhandener OFF-LINE
Plotter vom Typ BENSON wird auch in Zukunft weiterverwendet werden.

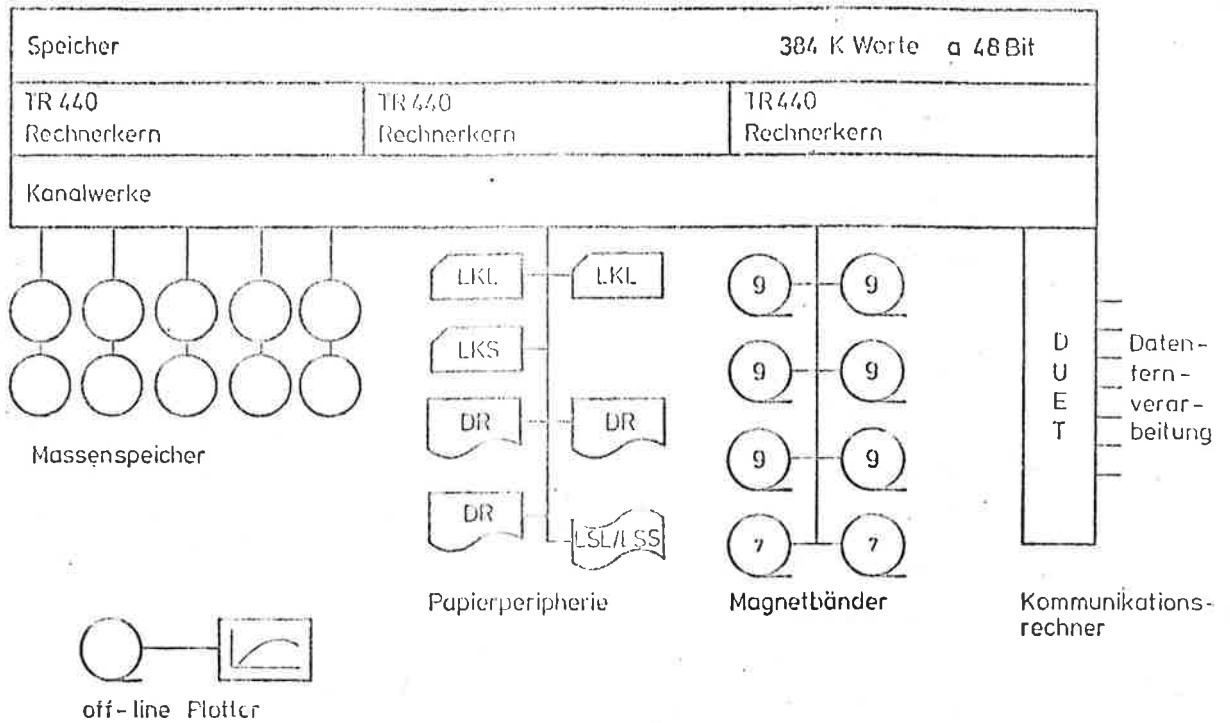


Abb.:1 Geräteausstattung des Regionalen Rechenzentrums Erlangen

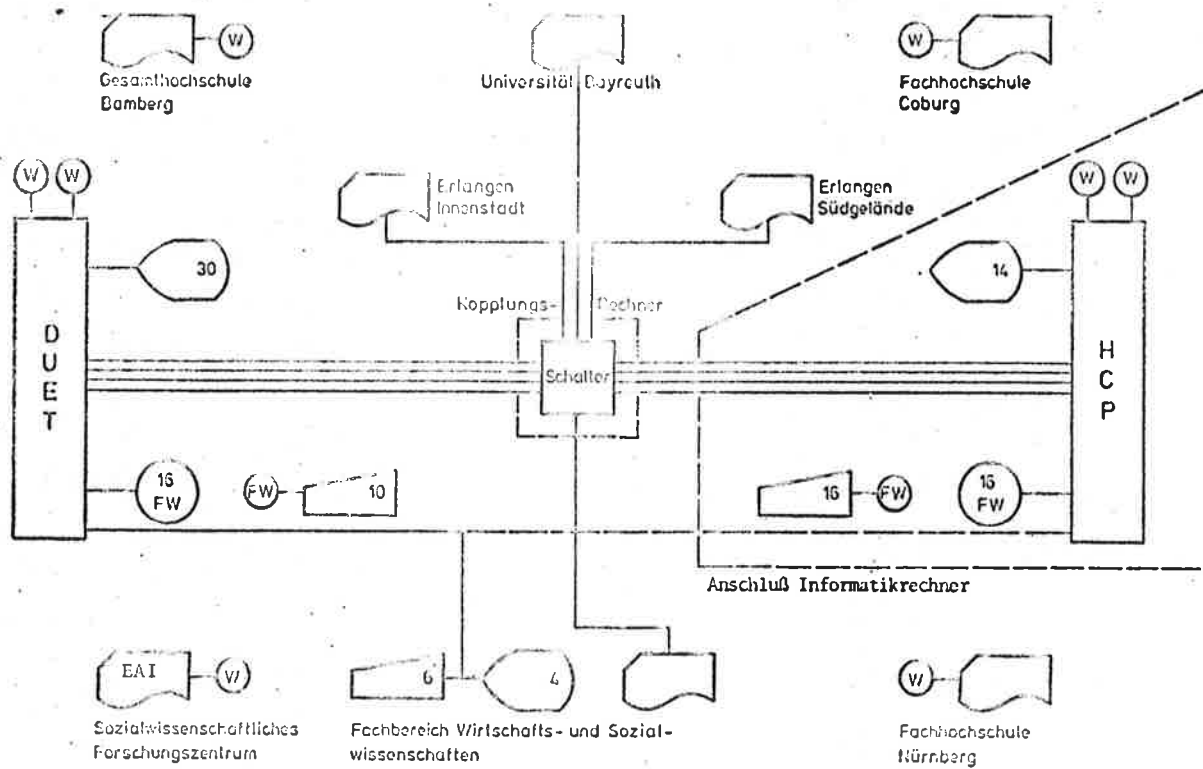


Abb.:2 Datenfernverarbeitung am Regionalen Rechenzentrum Erlangen mit dem Informatikrechner der Universität Erlangen-Nürnberg

W = Wählanschluß für Stapelstation
FW = Wählanschluß für Fernschreiber

Datenblatt Rechnerkern

Rechnerbasis RBA 442

Wortlänge	52 Bits, davon 48 Bits Information 2 Bits Typenkennung 2 Bits Dreierprobe	
Verarbeitung	Vorwiegend wortweise parallel	
Informationsdarstellung		
Text	Bevorzugt in Bytes zu 8 Bits (mit 6 Bytes pro Wort)	
Zahlen	Festpunkt dual	Gleitpunkt sedezimal
Halbwort	6 Dezimalstellen	—
Ganzwort	13 Dezimalstellen	10 Dezimalstellen
Doppelwort	—	24 Dezimalstellen
Rechenzeit	Festpunkt	Gleitpunkt
Addition	ca. 0,5 μ s	ca. 1,75 μ s
Subtraktion	ca. 0,5 μ s	ca. 1,75 μ s
Multiplikation	ca. 3,44 μ s	ca. 3,38 μ s
Division	ca. 13,75 μ s	ca. 13,31 μ s
Mittlere Operationsgeschwindigkeit	ca. 830.000 Operationen/s, entsprechend ca. 2 Mio Additionen/s	
Gibson-Mix	ca. 1,23 μ s	
GAMM-Mix	TAS:	4,4 μ s
	FORTTRAN:	5,3 μ s
	ALGOL:	6,4 μ s
Befehlsvorrat	240 Befehle (zu 24 Bits), 2 Befehle pro Wort; vorwiegend Einadreßbefehle, z. T. Spezifikationen im Adreßteil.	
	7 Befehle (zu 48 Bits) für das EA-Werk.	

Schneller Halbleiterspeicher HLS 270

Aufbau	MOS-Technik, 1 K-Chip gegliedert in Moduln mit je 32 K Worte (zu 59 Bits)
Ausbau	Minimal 4 HLS-Moduln mit 786.432 Bytes (128 K Worte, 1024 Blöcke mit je 128 Worten) Maximal 16 HLS-Moduln mit 3.145.728 Bytes (512 K Worte, 4096 Blöcke mit je 128 Worten)
Zugriffszeit	ca. 400 ns
Zykluszeit	Schreiben 350 ns Lesen 435 ns Reduzierte Schreib-/Lesezyklen durch Zyklusüberlappung
Fehlerkorrektur	Hohe Verfügbarkeit durch automatische Ein-Bit-Fehler-Korrektur und Erkennung aller Zwei-Bit- und der meisten Mehrbit-Fehler

2. Übersicht Hardware

TR 440/400
Triprozessor

Die Zentraleinheit des Systems TR 440/400 Triprozessor setzt sich aus einer Reihe von autonomen Werken zusammen, die simultan zueinander arbeiten können. Es sind dies

- Drei Rechnerkerne mit je Befehlswerk
Rechenwerk
Assoziativspeicher
- Vorrangwerk
- EA-Werk mit Eingriffswerk
1 bis 6 EA-Befehlswerken (BW) für max.
6 Schnellkanäle

1 EA-BW für max. 4 Standard-
kanäle, die im Multiplexbetrieb
arbeiten

1 EA-BW für max. 8 Standardkanäle,
die im Multiplexbetrieb arbeiten
- Speicherwerk mit max. 16 Moduln zu 32 K Worten

Der TR 440/400 Triprozessor ist mit schnellen MECL-2 Schaltkreisen aufgebaut. Um eine hohe Verarbeitungsleistung zu erreichen, wurden oft durchlaufene Routineschritte zu Mikroprogrammbausteinen zusammengefaßt (z. B. Befehlssentschlüsselung, Speicheranwahl, EA-Verkehr Eingriffe usw.).

Die Informationsverarbeitung des Rechnerkerns ist vorwiegend wortweise parallel.

Es existieren weiterhin leistungsfähige Befehle zur Zeichenverarbeitung. Doppelwort- und Blockverarbeitung.

Das Wort besteht aus

- 48 bit Information
- 2 bit Typenkennung
- 2 bit Dreierprobe.

Die Typenkennung (TK) dient zur Unterscheidung von

- Gleitkommazahlen TK = 0
- Festkommazahlen TK = 1
- Befehlen TK = 2
- Alphatext TK = 3,

während die Dreierprobe (DP) zur Fehlererkennung beim Verkehr mit dem Speicher, dem Rechnerkern und den EA-Kanälen benutzt wird.

Im Halbleiterspeicher des TR 440/400 Triprozessor werden zusätzlich über das TR 440 Wort (von 52 Bit) 6 ECC-Bit (Error-correcting code) gebildet, die zur Fehlerkorrektur bei Einbitfehlern dienen. Ein weiteres, über die 58 bit erzeugtes Paritybit dient zur Erkennung von Zweibitfehlern. Ein Wort ist demnach 59 bit lang.

Rechnerkern

Der Rechnerkern mit Befehlswerk und Rechenwerk dient zur Bearbeitung von Programmen. Mit den 228 für den Benutzer implementierten Rechner-Befehlen lassen sich folgende Operationen ausführen

- Festpunktoperation
- Gleitpunktoperationen einfacher und doppelter Genauigkeit
- Transportoperationen
- Boolesche Operationen
- Registeroperationen
- Zeichentransporte
- Modifizierungen und Ersetzungen
- Indexoperationen
- Tabellensuchoperationen
- Setz- und Löschoptionen
- Verzweigungsoperationen

Im Ablauf eines Befehls wird zwischen Abruf- und Ausführungsphase unterschieden. Die Ausführungsphase von Befehlen kann überlappt ablaufen, wenn die Ausführung in verschiedenen Werken liegt. Benutzt ein Mikroprogramm zur Ausführung eines Befehls z. B. nur das Rechenwerk (bei arithmetischen Operationen), so können gleichzeitig vom Befehlswerk weitere Befehle verarbeitet werden, die zu ihrer Ausführung nicht das Rechenwerk benötigen.

Befehlswerk

Das Befehlswerk setzt sich aus einer Reihe Registern und festverdrahteten Mikroprogrammbausteinen zusammen, die den Befehlsablauf und die Befehlsausführung steuern und kontrollieren. Von den 31 Registern des Befehlswerks sind 9 Register adressierbar. Die Befehle werden in Halbworten (24 bit) abgelegt und zerfallen ihrerseits in einen Befehlscode (8 bit) und einen Adreßteil (16 bit).

Operationscode	8	Adreßteil	16
----------------	---	-----------	----

Die wichtigsten Aufgaben des Befehlswerks sind

- Auslesen des Befehls
- Befehle vorentschlüsseln
- Umschaltung der Adressierungsmodi
- Befehlsablauf steuern
- Befehlsfolgezähler auf die aktuelle Fortsetzadresse einstellen
- Umrechnung der programmrelativen 16-bit-Adresse in die absolute 22-bit-Adresse (paging).

Rechneroperationen bei der Indexarithmetik wie

- Zählen und
- Adreßmodifikation

werden ebenfalls im Register des Befehlswerks ausgeführt.

Jeder im TR 440 ablaufende Befehl kann in der Ausführungsphase unterbrochen werden, vorausgesetzt, gewisse Sperren sind nicht gesetzt. In TR 440 unterscheidet man 4 Arten, auf die ein Prozessor unterbrochen werden kann

Eingriff	(kanalspezifische Rückmeldung des EAW, auch Fehlermeldung)
Alarm	(Fehlermeldung, z. B. Registerüberlauf bei arithmetischer Operation oder Signal, z. B. Weckzeitpunkt erreicht)
Makro-Befehle	(nicht belegter Befehlscode)
SSR-Befehle	(Systemsprung)

Unterbrechungen sind immer mit einer hardwareseitigen Ablage der wichtigsten Register verbunden.

Die vier Unterbrechungsarten enden in der Regel in den Unterbrechungsbehandlungen des Systemkerns (Supervisor des Betriebssystems).

Rechenwerk

Das Rechenwerk führt die durch einen Befehl geforderten Rechenoperationen aus. Es arbeitet bei reinen Rechenoperationen völlig unabhängig von allen anderen autonomen Werken. Mit dem Befehlswerk besteht eine gemeinsame Verbindung zum Speicher über das RS-Sammelregister. Dieses Register puffert und verteilt die in beiden Richtungen fließende Information zum Rechenwerk oder Befehlswerk.

Das Rechenwerk umfaßt insgesamt 16 Register, von denen 6 Register dem Programmierer direkt zugänglich sind.

Mit der sogenannten Dreierprobe wird das Rechenwerk bei allen Operationen auf richtige Funktion überprüft.

Schneller Assoziativspeicher

Vor dem Rechnerkern des TR 440/400 Triprozessor liegt ein schneller Assoziativspeicher, der zum Vorladen von Programmteilen (Befehlen und Operanden) dient. Durch diesen Zwischenpuffer wird die Zugriffszeit des Rechnerkerns auf den Speicher stark reduziert, da das Vorladen parallel zur Arbeit im Rechnerkern abläuft. Dadurch wird die Leistungsfähigkeit des Rechnerkerns voll ausgenutzt.

Der schnelle Assoziativspeicher besteht aus

- 16 assoziativ adressierbaren Registern für die Aufnahme von Befehlspaaren (also für 32 Befehle). Die angesprochenen Befehlsworte werden nacheinander eingeschrieben, wobei Befehlsfolgen durch einen Look-ahead-Mechanismus vorgeladen werden.
- 8 assoziativ adressierbaren Registern für Operanden. Die angesprochenen Operanden werden hier der Reihe nach eingeschrieben. Dabei wird laufend geprüft, ob es sich um eine Sequenz handelt. Ist dies der Fall, wird diese durch "Look-ahead" ebenfalls vorgeladen.

Der Look-ahead-Mechanismus des schnellen Assoziativspeichers besitzt eine eigene Schnittstelle zum Vorrangwerk für direkte Zugriffe zum Speicher, die unabhängig von der des Rechnerkerns ist. Damit hat der Rechnerkern die Möglichkeit, Speicherzugriffe, die der Assoziativspeicher nicht ausführen kann, selbst abzuwickeln (z. B. bei einem Sprung auf eine Adresse, die nicht im Assoziativspeicher vorhanden ist).

Vorrangwerk

Die Aufgabe des Vorrangwerkes ist es, alle auf den gemeinsamen Speicher gerichteten Zugriffswünsche der verschiedenen Zugreifer zu koordinieren.

Diese Koordination wird mit einer Vorrangpyramide erreicht, indem jedem Zugreifer eine feste Priorität zugeordnet ist. Dabei wird einem Zugreifer nur dann Vorrang zum Speicher erteilt, wenn das angesprochene Speichermodul auch frei ist.

Sämtliche Zugreifer sind im Vorrangwerk als Mikroprogramm-bausteine realisiert und arbeiten parallel zueinander. Damit sind die allen Zugreifern gemeinsamen Adreß-, Speicher- und Steuerleitungen optimal ausgenutzt und evtl. Wartezeiten zu einem belegten Speichermodul auf ein Minimum reduziert. Durch diesen Aufbau ergibt sich für das TR 440-System eine maximale Lese- und Schreibrate von

- 3.500.000 Leseoperationen/sec
- 8.000.000 Schreiboperationen/sec

In diesen Zeiten ist die Dreierprobenfehlererkennung und Benachrichtigung des Zugreifers enthalten.

Speicherwerk

Der Hauptspeicher ILS 270 des TR 440/400 Triprozessor ist ein Halbleiter-Speicher in MOS-FET-Technik. Der Speicher ist aus einzelnen Modulen von 32 K Ganzworten zu je 59 bit aufgebaut. Jeder Speichermodul besitzt eine eigene Ablaufsteuerung mit Adreß- und Informationsregister sowie ECC-(ERROR CORRECTING CODE)- und Dreierprobenprüfung, so daß Parallelarbeit der einzelnen Module möglich ist.

Ist die Ablaufsteuerung eines Moduls durch einen Speicherbefehl belegt, so ist innerhalb von 400 ns kein weiterer Schreib- oder Lesezugriff auf diesen Modul möglich. Durch Adressenverschränkung werden aufeinanderfolgende Adressen in verschiedenen Modulen abgelegt. Dadurch wird die Parallelarbeit der Speichermodul optimal ausgenutzt.

Ein Lesezyklus dauert pro Modul 400 ns
Ein Schreibzyklus dauert pro Modul 350 ns

Mit dem Ausbau des Speichers und dem Verschränkungsgrad verringern sich die effektiven Zyklus- und Zugriffszeiten für den Hauptspeicher erheblich.

Im ILLS 270 werden über das TR 440-Wort (von 52 Bits) 6 ECC-Bits gebildet, die eine Fehlerkorrektur von Einbitfehlern erlauben. Ein weiteres über die 53 Bits erzeugtes Paritybit dient zur Erkennung von Zweibitfehlern. Dreierprobe und ECC-Prüfung erlauben darüberhinaus das Erkennen von Mehrbitfehlern mit großer Sicherheit.

Der Speicher ist in den Ausbaustufen 128 KW, 256 KW, 384 KW und 512 KW lieferbar. Bei einer Wortlänge von 6 Bytes entspricht dies ca. 3 M Bytes. Eine Verschränkung - und damit eine reduzierte Zykluszeit - ist bei den Ausbaustufen 128 KW (vierfach verschränkt), 256 KW (achtfach verschränkt), 384 KW (vierfach verschränkt) und 512 KW (achtfach verschränkt) möglich.

An dem Anzeigefeld des Speichers werden Ein- und Zweibitfehler im Speicher angezeigt, die Einbitfehler bis auf das Chip genau. Dadurch wurde die Wartung des Speichers wesentlich vereinfacht.

EA-Werk

Das EA-Werk hat die Aufgabe, den unabhängig vom Rechnerkern betriebenen EA-Verkehr über die angeschlossenen Kanäle und Geräte abzuwickeln.

Im einzelnen besteht das EA-Werk aus

- bis zu 6 Schnellkanalwerken mit je 1 Unerkanal
- bis zu 12 Standardkanälen mit je 6 Unterkanälen
- bis zu 8 parallel zueinander arbeitenden EA-Befehlswerken
- dem Eingriffswerk

Um die Übertragungsgeschwindigkeit moderner EA-Geräte verarbeiten zu können, sind die 6 Schnellkanäle jeweils mit einem eigenen EA-Befehlswerk ausgerüstet, so daß die Kanalwerke parallel zueinander arbeiten können. Sie dienen zum Anschluß von schnellen Trommel- und Wechselplattenspeichern.

Den 4 Standardkanälen 6 - 9 und den restlichen 8 Standardkanälen 10 - 17 stehen je ein EA-Befehlswerk zur Verfügung, so daß die Standardkanäle parallel arbeiten können. Die Kanäle unter sich werden vom EA-Befehlswerk entsprechend ihrer Priorität bedient. (Der Kanal mit der niedrigeren Nummer hat die höhere Priorität). Sie dienen zum Anschluß von Wechselplattenspeichern mittlerer Transferrate, schnellen Magnetbandgeräten und papierverarbeitenden Geräten.

Die Übertragungsgeschwindigkeiten der Kanäle betragen

- max. 700 000 Byte/s bei Standardkanälen
- max. 3.000 000 Byte/s bei Schnellkanälen

Der Zugriff der Kanäle zum Speicher wird durch die Vorrangpyramide des Vorrangwerks gesteuert. Die Unterkanäle 0 und 1 aller Standardkanäle sind mit einem Anrufsender für die Rechnerkopplung ausgestattet.

Eingriffswerk

Meldungen über den Zustand, Fehler oder Beendigung eines EA-Vorganges werden dem Betriebssystem (Systemkern) über das Eingriffswerk mitgeteilt. Eingriffe sind grundsätzlich ein Verlangen an den Rechnerkern, sein Programm zu unterbrechen und eine Eingriffsbehandlung durchzuführen. Eingriffe können nur Meldungen von externen Geräten sein. In dem mitgelieferten Eingriffswort steht die Ursache der Unterbrechung.

Das Eingriffswerk besteht aus Mikroprogrammbausteinen, die parallel zu sich selbst und insgesamt parallel zu anderen autonomen Werken arbeiten. Die Aufgaben des Eingriffswerkes sind

- Speichern der Eingriffswünsche von den Kanälen mit Angabe der Unterkanäle
- Ermitteln des Rechnerkerns, der die Eingriffe bearbeiten soll
- Unterbrechung des Rechnerkerns
- Übergabe des Eingriffswortes zur Auswertung und Behandlung durch den Systemkern (Supervisor).

Die im Eingriffswerk anstehenden Eingriffe werden nach der Priorität der Kanäle abgearbeitet. Damit ist sichergestellt, daß die schnellen Geräte bevorzugt bedient werden.

Virtuelle Adressierung (paging)

Operatoren - das sind die im TR 440 ablaufenden Benutzerprogramme - werden der Seitenadressierung (paging) unterworfen. Der Zentralspeicher ist dazu in max. 512 Kacheln (physikalische Seiten) unterteilt. Die Kachel ist eine Einheit von 1024 konsekutiv adressierten Ganzworten. Sie nimmt genau eine Programmseite (logische Seite) an Information auf. Die Zuordnung von Seiten zu Kacheln geschieht durch Seiten-Kachel-Zuordnungstabellen. Da die Seiten relativ adressiert werden, können zusammenhängend adressierte Seiten im Zentralspeicher auch in nicht unmittelbar aufeinanderfolgenden Kacheln liegen. Dies hat den Vorteil, daß der Systemkern (Supervisor) im Kernspeicher entstandene Lücken ohne Rücksicht auf Zusammenhangbedingungen belegen kann.

					Zuordnung	
Seite	Bit 1.....12/13.....24/25.....36/37.....48				Seiten-Nr.	Kachel-Nr.
0-3	Kachel 4	7	32	5	0	4
4-7	16	29	34		1	7
8-11					2	32
12-15					3	5
16-19					4	16
20-23					5	29
24-27					6	34
28-31						
32-35						
36-39						
40-43						
44-47					48	2
48-51	2	3			49	3

Seiten-Kachel-Zuordnungstabelle

Die virtuelle Adressierung stellt sicher, daß der Adressenraum eines Programms durch ein anderes Programm nicht verletzt werden kann.

Überschreitet ein Programm seinen ihm zugewiesenen Adressenraum, wird ein Speicherschutzalarm erzeugt. Durch Auswertung dieses Alarms wird im Betriebssystem BS 3 Demand segmenting bzw. Demand paging realisiert. Da das System (der Job-Abwickler) einen größeren Adressenraum pro Job zur Verfügung hat, können Systemteile, die von mehreren Jobs benötigt werden, in einen gemeinsamen Bereich gelegt werden. Codesharing von Systemprogrammen ergibt sich so ganz automatisch.

Für jede Programmseite kann ein Schreibschutzbit gesetzt werden. Sämtliche Compiler legen den Befehlsteil von Programmen in schreibgeschützten Seiten ab, während die Variablen in Seiten ohne Schreibschutz stehen. Demnach sind sämtliche erzeugten Programme ablaufinvariant (reentrant). Das paging hat weiterhin noch den Vorteil, daß eine Reorganisation des Speichers unnötig wird und demnach Verschnitt im Speicher vermieden werden kann.

Zentralspeicher- ansteuerung

Erfolgt in einem Operator durch einen Befehl ein Zugriff auf den Zentralspeicher, so wird die im Adreßteil enthaltene virtuelle Adresse durch die Hardware transformiert. Die als Dualzahl aufgefaßte Adresse wird in zwei Teile unterteilt. Der linke Teil wird als Seitennummer verwendet. Mit der Seitennummer als Index wird aus der Seiten-Kachel-Tabelle das entsprechende Element, das die Kachelnummer enthält, ausgelesen. Der rechte Teil der virtuellen Adresse stellt die Seiten- bzw. Kachelrelativ-Adresse dar.

Seitenassoziativregister

Nach dem beschriebenen Verfahren muß bei jedem beabsichtigten Speicherzugriff vorher ein Zugriff auf die Seiten-Kachel-Tabelle erfolgen, die ebenfalls im Kernspeicher liegt. Um die Zeit zum Auslesen der Kachelnummer in möglichst vielen Fällen einsparen zu können, verfügt der zentrale Rechner des TR 440 über vier Assoziativregister, die die vier zuletzt aufgetretenen Kachelnummern enthalten. Tatsächlich wird durch diese vier Register in 96 % der Fälle ein Zugriff auf die Seiten-Kachel-Tabelle vermieden.

3 Übersicht Betriebssystem

3.1 Funktioneller Aufbau

Das Betriebssystem ist durch einen modularen Aufbau gekennzeichnet und liegt in seinem residenten Teil sowohl im Arbeitsspeicher des RD 441 als auch im Kernspeicher des Satellitenrechners. Der zuladbare Teil liegt (wie das gesamte Programmiersystem) auf einem Hintergrundspeicher (Trommel, Wechselplatte).

Der Teil des Betriebssystems, der im RD 441 liegt, gliedert sich in Systemkern und eine feste Anzahl von selbständigen Programmeinheiten, Akteure genannt.

Die Aufgaben der Akteure lassen sich wie folgt klassifizieren:

- Aufgaben, die zeitlich absolut vorrangig sind:
Erledigung der Aufgaben durch die Systemteile
- Aufgaben, die wenig rechenintensiv und stark ein-/ausgabintensiv sind; sie verursachen lange Wartezeiten auf die - im Vergleich zum Rechnerkern - langsamen peripheren Einheiten:
Erledigung der Aufgaben durch die Vermittlerprozesse.
- Aufgaben, die rechenintensiv sein können und deren Erledigung zeitlichen Bedingungen (Reaktionszeit an den Konsolen) unterliegt.
Erledigung der Aufgaben durch die (Gesprächs-)Abwickler.
- Aufgaben, die rechenintensiv sein können und deren Erledigung keinen zeitlichen Bedingungen unterliegt.
Erledigung der Aufgaben durch die (Abschnitts-)Abwickler.

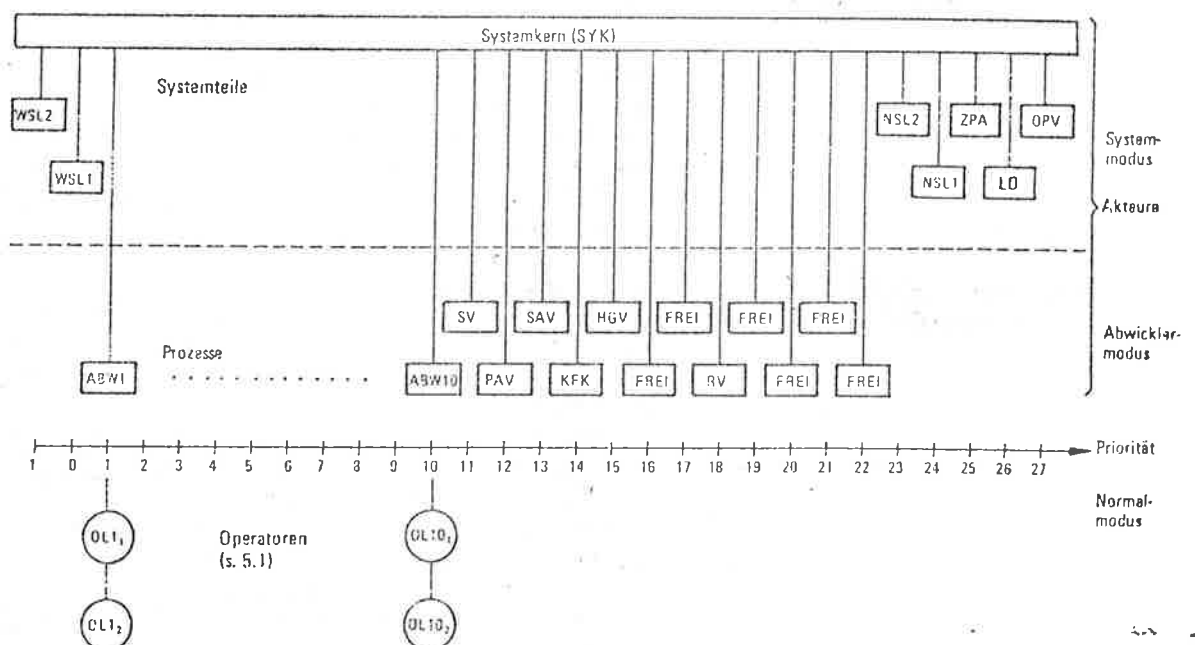


Bild 5 Funktioneller Aufbau

Aufgrund der aufgeführten Aufgabenklassen ist eine feste Zuordnung zwischen Akteur und Priorität getroffen worden, siehe Bild 5. (Die Prioritäten der Abwicklerprozesse werden dynamisch vergeben.)

Die angegebene Reihenfolge der Aufgabenklassen, die sich in den Prioritäten (s. Bild 5) wiederholt, ist notwendig, um der Forderung nach vernünftiger Antwortzeit und gutem Durchsatz, d.h. optimaler Anlagenauslastung zu genügen.

Die Menge der Akteure zerfällt in Systemteile und Prozesse, die sich durch unterschiedliche Aufgabenstellung und durch ihren Adressierungsmodus unterscheiden. Bild 5 ist nach Priorität und nach Adressierungsmodus gegliedert. Die Erläuterungen der verwendeten Abkürzungen folgen im Text.

3.2 Systemkern

Der Systemkern verwaltet im Betriebssystem die Betriebsmittel Rechnerkern, Arbeitsspeicher (im RD 441), Ein-/Ausgabegeräte und Kanäle.

Mit Verwaltung ist hierbei die "physikalische" Verwaltung (Vergabe und Rückgabe) der Betriebsmittel gemeint; die "logische" Verwaltung (Verplanen und Entplanen) führt die Kontrollfunktion durch (s. 3.5).

Im einzelnen erfüllt der Systemkern (Nukleus) folgende Aufgaben:

Rechnerkernvergabe

Dem Systemkern stehen selbständige Programmeneinheiten - Akteure gegenüber, die zueinander eine feste Priorität haben. Bedingt durch die Abhängigkeit der einzelnen Akteure voneinander, gibt es Situationen, in denen einzelnen Akteure an der Zuteilung des Rechnerkerns uninteressiert sind, d.h. sie haben den Zustand "nicht rechenwillig".

Andere Akteure bewerben sich um den Rechnerkern, sie besitzen den Zustand "rechenwillig". Der Systemkern teilt dem rechenwilligen Akteur mit der höchsten Priorität den Rechnerkern zu.

Speicherverwaltung

Die Speichermedien (Arbeitsspeicher und Hintergrundspeicher) werden in Einheiten von 1 K (1024) Ganzworten verwaltet, der Arbeitsspeicher vom Systemkern, die Hintergrundspeicher vom Hintergrundvermittler (s. 3.4). An diese Verwaltung werden von den Prozessen Forderungen nach freiem Speicherraum und nach Beendigung ihrer Aufgaben deren Freigabe gemeldet. Der Arbeitsspeicher wird mit Hilfe von Assoziativregistern virtuell adressiert.

Koordination der Peripheriegeräte

Der Systemkern übernimmt die Ein- und Ausgabeaufträge von den Akteuren, verteilt diese auf Warteschlangen vor den EA-Kanälen und arbeitet sie über die Kanalwerke ab. Die Rückmeldungen von den EA-Kanälen werden den betroffenen Akteuren anschließend zugestellt.

Die Vermittlerprozesse melden bei Ein-Ausgabeverkehr das betreffende periphere Gerät beim Systemkern an und nach Beendigung des Transports wieder ab.

In der Behandlung durch den Systemkern nimmt der Satellitenrechner und dessen periphere Geräte (Fernschreiber und Sichtgeräte) keine Sonderstellung ein. Ausgabeaufträge an die Benutzerstationen werden an das Kanalwerk geleitet, an dem der Satellitenrechner angehängt ist.

3.3 Systemteile

Die Systemteile erfüllen systemnahe und hardwarenahe Aufgaben, die nachstehend stichwortartig beschrieben sind. Die Warteschleife WSL (WSL1 für Rechnerkern 1, WSL2 für Rechnerkern 2) erhält immer dann den Rechnerkern, wenn kein Akteur den Zustand "rechenwillig" besitzt. Die Warteschleife hat zu diesem Zweck immer den Zustand "rechenwillig". Die Notschleife NSL wird bei Systemfehlersituationen rechenwillig gesetzt. Die freien Prioritäten sind für Systeme mit mehr als zwei Rechnerkernen reserviert.

Zentralprotokoll

Das Zentralprotokoll ZP gestattet, auf einem Hintergrundspeicher Speicherabzüge (Dumps) und Zusatzinformation über einen Systemlauf (Engpässe etc.) zu lagern.

Lader

Der Lader LD hat die Aufgabe, während des Systemaufbaus die Prozesse vom Systemband in den Rechner zu transportieren.

Operateurvermittler

Der Operateurvermittler OPV belegt die höchste Priorität, um dem Operateur die Möglichkeit zu geben, jederzeit mit Hilfe von Operateurkommandos (z. B. zum Abfragen der Betriebssituation) in den Bearbeitungsablauf einzugreifen. Außerdem verwaltet der Operateurvermittler die Kontrollschreibmaschine und ermöglicht das Ein- und Ausgeben von Information zwischen den Akteuren und dem Operateur.

3.4 Prozesse

Unter Prozessen sind Programme zu verstehen, die für den Benutzer des Rechensystems genau definierte Leistungen erbringen. Während die Systemteile ihre Aufträge von Prozessen erhalten und systemnahe Aufgaben erfüllen, also vom Benutzer nicht direkt erreichbar sind, stehen die Prozesse stellvertretend für den Benutzer dem Systemkern gegenüber. Dabei werden Prozesse unterschieden, die die EA-Geräte oder Hintergrundspeicher bedienen - Vermittlerprozesse - und die Rechenaufträge des Benutzers verwalten - Abwicklerprozesse -.

Vermittlerprozesse

Der Hintergrundvermittler HGV verwaltet die Trommel- und Wechsell Plattenspeicher und organisiert den Informationstransport zwischen diesen Speichermedien und dem Arbeitsspeicher. Der Papiervermittler PAV organisiert den Informationstransport zwischen Arbeitsspeicher und den peripheren Geräten Streifenleser, Kartenleser, Plotter, Streifenleser, Kartenstanzer und Schnelldrucker. Dabei hat er die Aufgabe, die Information in den Zentralcode oder aus diesem in den entsprechenden Gerätecode umzuschlüsseln und die entsprechenden Geräte zu betreiben. Magnetbandgeräte werden durch einen Vermittler betrieben, der Bestandteil der Abwicklerprozesse ist. Aufgrund dieser Tatsache kann der Magnetbandverkehr - wie durch die Eigenheit des Magnetbands erforderlich - direkt betrieben werden.

Satellitensystem

Der Satellitenvermittler SAV ist gemeinsamer Kommunikationspartner aller mit Benutzerstationen in Verbindung stehender Programmläufe. Er nimmt im Arbeitsspeicher oder auf einem Hintergrundspeicher gepufferte Ausgaben entgegen und übergibt sie in Teilstücken nach Aufforderung an den Satellitenrechner. Eingabeinformation von den Benutzerstationen nimmt der Satellitenvermittler vom Satellitenrechner entgegen.

Unvollständige Eingaben puffert er auf dem Hintergrundspeicher, vollständige Eingaben werden an die betreffenden Programmläufe weitergeleitet (siehe Bild 6).

Die Software im Satellitenrechner wird ihrer Aufgabe gemäß Satellitenprogramm (SAP) genannt. Dieses Programm organisiert die Ein- und Ausgabe von den Benutzerstationen und die Übertragung der Information vom und zum Satellitenvermittler über die Rechnerkopplung. Dabei besorgt es die Registrierung der verschiedenen Zustände, puffert geringe Ein-/Ausgabemengen und sorgt für den geregelten Fluß der Information.

Die Programmkomplexe Satellitenvermittler SAV (im RD441) und Satellitenprogramm SAP (im Satellitenrechner) bilden zusammen das Satellitensystem SAS.

Abwicklerprozesse (job management)

Ein vom Benutzer gestellter Auftrag (Gespräch oder Abschnitt) hat den Ablauf einer Operatorlauf-Kette (Entschlüsselerlauf, Übersetzerlauf usw.) zur Folge. Das vom Benutzer eingegebene Programm wird zu einem Glied - dem Objektlauf - in dieser Kette (job step).

Ein Abwicklerprozess hat die Aufgabe, diese Operatorlaufkette zu verwalten und den einzelnen Operatorläufen Systemdienste anzubieten. Aus dieser Sicht eines Operatorlaufs stellt der Abwicklerprozeß ABW ein benutzerspezifisches Betriebssystem dar. Die Operatoren selbst sind Bestandteil des Programmiersystems (s. Abschnitt 4).

3.5 Kontrollfunktion (task management)

Die Bearbeitung eines Benutzerauftrags beginnt normalerweise mit der Entgegennahme des Quellprogramms und endet mit dem Ausgeben der Ergebnisse auf den gewünschten Ausgabemedien. Dazwischen liegt die Bearbeitung des Auftrags durch die Prozesse, zu der "Betriebsmittel" benötigt werden. Unter diesem Begriff werden alle Hardware-Einrichtungen, Programme und Informationen zusammengefaßt, deren Verfügbarkeit nicht zu jedem Zeitpunkt gewährleistet ist, von denen es also zu wenige gibt, um in jedem Fall alle Interessenten gleichzeitig bedienen zu können.

Betriebsmittel

Die wesentlichste Aufgabe des Prozesses Kontrollfunktion KFK ist es, über die Vergabe der Betriebsmittel zu entscheiden. Betriebsmittel sind in diesem Zusammenhang:

- Rechnerkern
- Speichermedien (Arbeits- und Hintergrundspeicher)
- Abwicklerprozesse
- Ein- und Ausgabebefehle
- Auftragsselemente (Speicherbereiche zur internen Beschreibung eines Auftrags).

Beim Einsatz der zur Verfügung stehenden Betriebsmittel beachtet die Kontrollfunktion folgende Forderungen:

1. Die Betriebsmittel, insbesondere Rechnerkern und Arbeitsspeicher, sind möglichst gut auszunutzen.
2. Die Bearbeitung erfolgt gemäß einer definierten Wichtigkeit (Gewicht, Reihenfolge des Eintreffens o.ä.).
3. Aufträge sollen so schnell wie möglich bearbeitet werden. Insbesondere sollen sich bei Gesprächen (Dialog) die Reaktionszeiten innerhalb solcher Grenzen bewegen, daß der Benutzer den Eindruck hat, in einer - seinem eigenen Arbeitstempo und dem Arbeitsumfang der von ihm gestellten Aufgaben - angemessenen Geschwindigkeit bedient zu werden.

Diese Forderungen widersprechen sich zum Teil, so daß die Kontrollfunktion eine Parameterbewertung durchführt und aufgrund folgender Festlegungen entscheidet:

- Eine gute Rechnerauslastung soll durch das parallele Ablaufenlassen von bis zu 10 Aufträgen erreicht werden (multiprogramming). Wartezeiten eines Auftrags (z.B. wegen EA-Verkehr) werden durch andere Aufträge ausgenutzt.
- Gespräche sind wichtiger als Abschnitte (Reaktionszeit).
- Für Abschnitte wird nach einer durch die Rechenzentrumsleitung vorgebbaren Formel aus Kenndaten des Benutzers ein Gewicht berechnet, das die Wichtigkeit des Abschnitts ausdrückt. Das Eintreffen eines wichtigen Abschnitts bewirkt unter Umständen den vorübergehenden Entzug gewisser Betriebsmittel bei unwichtigeren Abschnitten.
- Die insgesamt zur Verfügung stehende Anlagenzeit (nicht nur die Rechnerkernzeit) soll unter die Aufträge so aufgeteilt werden, daß die Forderung 3 erfüllt wird (time sharing). Dabei soll eine angemessene Reaktionszeit durch zyklische Zuteilung des benötigten Kernspeichers und des Rechnerkerns an alle laufenden Gespräche erreicht werden (time slicing). Das in diesem Zyklus "nächste" Gespräch ist damit immer das wichtigste.

Um die skizzierte Betriebsmittelvergabe durchführen zu können, wird die Kontrollfunktion nach festen Zeitintervallen oder aufgrund von Rückmeldungen der Prozesse "rechenwillig" gesetzt.

Der Sendungsvermittler SV hat die Aufgabe einer Kommunikationszentrale und stellt die von KOMSYS benötigten Dienstleitungen zur Verfügung

Der Rechnervermittler RV übernimmt die Koordination der Zugriffe zweier Rechner auf eine gemeinsame LFD (GEMLFD) und die Organisation der Datenübertragung zwischen mehreren direkt gekoppelten TR 440.

3.6

Bearbeitung eines Auftrags

Ein Auftrag beginnt mit dem XBA-(Beginne-Abschnitt-) oder XBG-(Beginne-Gespräch-)Kommando, beinhaltet eine Folge von Programmiersystem-Kommandos (s. Abschnitt 5) und wird durch ein XEN-(Beende-Auftrag-)Kommando abgeschlossen. Wird der Auftrag im Gesprächsmodus eingegeben, so kann der Benutzer auf den Ablauf einwirken. Andernfalls liegt mit den Kommandos die Reihe der Anforderungen fest, die an die unter der Regie eines Abwicklers laufenden Operatoren gestellt werden. In Bild 6 wird stellvertretend für andere Auftragsformen der Ablauf eines Gesprächs über ein Sichtgerät wiedergegeben, das als Ausgabe nur einen Druckauftrag absetzt.

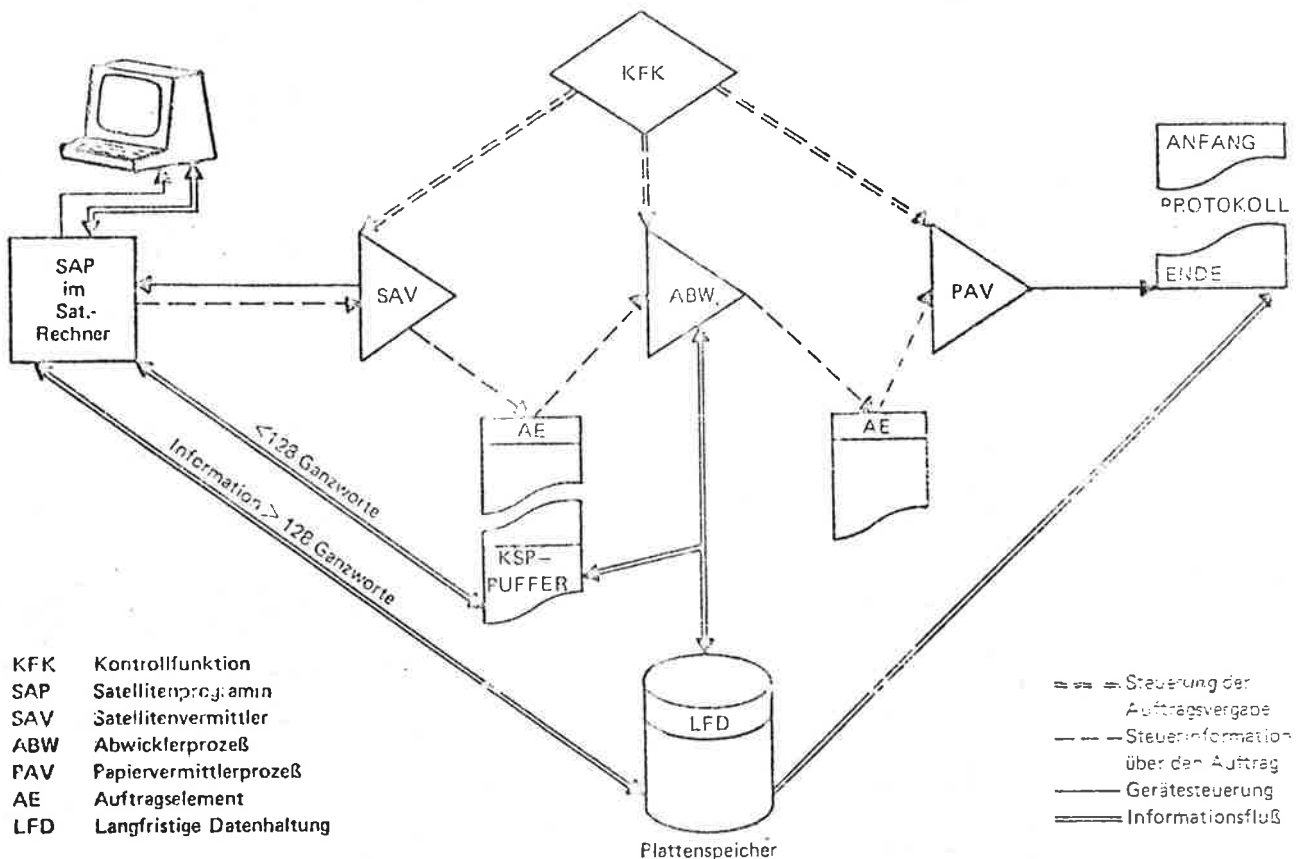


Bild 6 Gesprächsablauf

Der Satellitenvermittler nimmt das XBG-Kommando entgegen und teilt der Kontrollfunktion mit, daß ein Abwicklerprozeß den Auftrag übernehmen muß. Es entsteht ein Wechselgespräch zwischen dem Benutzer und den von ihm angesprochenen Operatoren, die von dem Abwickler verwaltet werden. Die Information wird dabei je nach Umfang im Arbeitsspeicher oder über einen Plattenspeicher ausgetauscht. Zusätzlich kann der Benutzer auf die Datenbestände in der langfristigen Datenhaltung LFD zugreifen (vergl. 3.7). Nach Gesprächsende wird der Papiervermittler beauftragt, die auf dem Plattenspeicher angefallene Druckinformation auszugeben.

Verdrängen von Gesprächen

Der Grundgedanke eines Teilnehmer-Rechensystems ist die zeitliche Aufteilung (time sharing) der begrenzt zur Verfügung stehenden Betriebsmittel an die Programme der einzelnen Benutzer.

Da der Mensch eine gegenüber dem Rechner große Reaktionszeit besitzt, merkt er in den meisten Fällen von der Zuteilung nichts. Es hat den Eindruck, die Leistung der Rechenanlage stehe im allein zur Verfügung.

Jedes Gespräch darf seit der letzten Zuteilung eines Abwicklerprozesses ein bestimmtes Zeitintervall - Bedienungsintervall - im Besitz des Rechnerkerns sein.

Gibt ein Gespräch innerhalb des Zeitintervalls nicht von sich aus einen Anlass zur Verdrängung - Ausgabe an eine Benutzerstation mit anschließender Eingabe - so wird der Programmbestand des Gesprächs zwangsweise auf den Trommelspeicher verdrängt (swapping). Die frei gewordenen Betriebsmittel stehen anderen Gesprächen zur Verfügung.

3.7 Datenorganisation (data management)

Die Datenorganisation bildet als Teil der Abwicklerprozesse einen abgeschlossenen Programmkomplex. Zur Speicherung von Datenbeständen stehen dem Benutzer des Teilnehmer-Rechensystems TR 440 die Speichermedien Trommel, Wechselplatte und Magnetband zur Verfügung. Zum Erstellen, zur Verwaltung und zur Verarbeitung dieser Datenbestände werden ihm die Dienstleistungen der Datenorganisation angeboten.

Dateien (files)

Der Benutzer (und mit Ausnahmen das Betriebssystem) hält seine Information in Form von Dateien. Der Begriff Datei beinhaltet, daß die Information strukturiert ist und daß über die Datei Verweisinformation gehalten wird. Die Dateien werden aus Sätzen - den kleinsten adressierbaren Informationseinheiten - aufgebaut. Nach der Art des Zugriffs auf diese Sätze werden vier Dateitypen unterschieden:

- SEQ Sequentieller Zugriff auf die Sätze
Die physikalische Reihenfolge der Sätze auf dem Speichermedium bestimmt die logische Folge.
- RAN Randomzugriff mit Satznummern
Die Information liegt dicht gepackt in der Reihenfolge ihres Eintreffens. Die logische Reihenfolge wird durch Stellvertreter repräsentiert, die nach ihrem Binärwert lückenlos aufsteigend abgelegt sind.
- RAM Randomzugriff mit Satzmarken
Der Unterschied zu RAN liegt darin, daß die Variationsmöglichkeit der Stellvertreter bei gleichem Speicherbedarf ungleich höher ist, da für undefinierte Sätze auch kein Stellvertreter angelegt wird.
- RAS Randomzugriff mit Satzschlüsseln
Während bei einer RAM-Datei die Kennzeichnung durch eine Satzmarke mit max. 6 Bytes erfolgt, kann zur Kennzeichnung einer RAS-Datei ein Satzschlüssel mit einer Länge von max. 255 Bytes verwendet werden.

Ein fünfter Dateityp - PIYS - bezieht sich nicht auf Sätze, sondern auf Blöcke (zu 896 Bytes). Ein Zugriff ist nur auf physikalischer Ebene direkt auf die Blöcke möglich.

Innerhalb eines Auftrages können simultan bis zu 255 Dateien verwaltet werden. Der Benutzer hat die Möglichkeit, die Dateien logisch zusammenzufassen unter dem Begriff einer Datenbasis. Der Dateiname braucht dann nur innerhalb einer Datenbasis eindeutig zu sein.

Die Information einer Datei kann der Benutzer nach Belieben handhaben, insbesondere kann die Datei für Zugriffe gesperrt werden.

Bibliotheken

Die Zahl der Datenbasen, die ein Abwicklerprozeß verwalten kann, ist auf 8 begrenzt.

Zwei ausgezeichnete Datenbasen werden durch das Betriebssystem aufgebaut und bearbeitet:

- Eine öffentliche Datenbasis (Systembibliothek)
Die öffentliche Datenbasis enthält den Programmbestand des Programmiersystems. Sie steht allen Benutzern für lesenden Zugriff zur Verfügung. Die öffentliche Datenbasis wird bei jedem Systemaufbau neu aufgebaut, ihre Lebensdauer stimmt also mit der des Systems überein.
- Eine Standard-Datenbasis (Auftragsbibliothek)
Jedem Benutzerauftrag steht eine Standard-Datenbasis zur Verfügung, die auftragsspezifische Daten enthält (z.B. Montageobjekte, Rückverfolgungslisten, Operatoränder-Beschreibungen).
Die Standard-Datenbasis wird von einem Operator - dem Entschlüssler (s. unter 5) - aufgebaut und mit Standarddateien besetzt. Auch der Benutzer legt implizit seine Dateien in die Standard-Datenbasis, wenn er keine eigene Datenbasis (s. o.) definiert.
Die Standard-Datenbasis wird am Ende eines Benutzerauftrages gelöscht.

Langfristige Datenhaltung

Die langfristige Datenhaltung bietet auch dem Benutzer die Möglichkeit, Dateien permanent zu speichern. Dabei werden die Dateien des Benutzers unter einem gemeinsamen Benutzerkennzeichen verwaltet, das an die Stelle eines Datenbasennamens tritt (Benutzerbibliothek). Die Entscheidung über den Inhalt und die Lebensdauer der Dateien in der LFD liegt beim Benutzer, ähnlich wie die Rechenzentrumsleitung über die Maintenance den Inhalt der öffentlichen Datenbasis festlegen kann.

Die Dateien in der LFD - alle oben genannten Typen sind zugelassen - unterscheiden sich in ihrer Zugriffseigenschaft:

- G-Datei (Gemeinschaftsdatei), die für den Zugriff mehrerer Benutzer bestimmt ist, und für die zwangsweise eine Koordinierung simultaner Bearbeitungen stattfindet.
- P-Datei (Privatdatei), die nur vom Benutzer veränderbar und löschar ist, der diese Datei kreiert hat.

Manipulationen an Dateien der langfristigen Datenhaltung sind durch Kommandos der Texthaltung (s. 4.7) und von höheren Programmiersprachen her möglich.

Gebietskonzept

Die Notwendigkeit der Verlagerbarkeit von größeren Informationseinheiten zwischen verschiedenen Speichermedien - z. B. Verdrängung eines Programmbestandes - führt zur indirekten Datenadressierung.

Zusammenhängende Informationseinheiten von einer oder mehreren Seiten werden als Gebiet bezeichnet.

Auf in Gebieten organisierte Informationsbestände kann der Benutzer implizit (über die Datenorganisation) oder explizit (Dienstleistungen des Abwicklers) zugreifen.

4

ÜBERBLICK ÜBER DAS PROGRAMMIERSYSTEM

Die Gesamtheit aller Operatoren, die jedem Benutzer des Teilnehmer-Rechensystems TR 440 zur Verfügung stehen - Standardoperatoren - und aller Prozeduren, die als Montageobjekte gehalten werden, wird als Programmiersystem bezeichnet.

4.1 Operatoren

Ein Operator im Sinne der Software-Organisation des TR 440 ist jedes Programm, das folgenden Bedingungen genügt:

- alle Befehle laufen im Normalmodus ab
- alle Beziehungen zum Betriebssystem werden über einen Abwicklerprozeß hergestellt, der den zugehörigen Benutzerauftrag bearbeitet.
- im Abwickler existiert eine Beschreibung dieses Programms (Startadresse, Größe usw.).

Aufgrund der zuletzt genannten Beschreibung ist der Abwickler in der Lage, einen Operator zu starten, d.h. einen Operatorlauf aus ihm zu machen.

Montierer (linkage editor)

Auch das Programm des Benutzers wird zu einem Operatorlauf in der Kette der Bearbeitungen - dem Objektlauf -.

Es ist natürlich nicht Aufgabe des Benutzers, die dazu notwendige Beschreibung zu erstellen. Diese Aufgabe übernimmt ein spezieller Operator - der Montierer -.

Er baut aus einzelnen Montageobjekten (vergl. 4.3) ein lauffähiges Programm auf und hinterlegt die Beschreibung in der Standard-Datenbasis des Benutzers (vergl. 3.7).

Zu dieser Beschreibung gehören auf Wunsch des Benutzers auch Referenzlisten, die es den zahlreichen Analyse-Operatoren (s. 5.6) erlauben, im Fehlerfall des Programmablaufs Informationen mit den Bezeichnungen in der Quelle des Benutzers auszugeben (quellenbezogener Dump) und die Fehlerstelle zu lokalisieren.

Entschlüssler

Ein Benutzerauftrag besteht aus der eigentlichen Information, die von der Rechanlage zu bearbeiten ist (Quelle, Daten) und Angaben darüber, in welcher Form das geschehen soll. Die Steuerinformation in Form von Kommandos (s.u.) wird von einem speziellen Operator, dem (Programmier-) Entschlüssler, ausgewertet. Der Entschlüssler veranlaßt daraufhin den Start der geforderten Operatoren.

4.2

Kommandosprache (command language)

Um dem Benutzer ein einheitliches und bequemes Steuermittel für den Bearbeitungsablauf in die Hand zu geben, wurde die TR 440-Kommandosprache geschaffen. Ihre Gültigkeit beschränkt sich nicht nur auf das Programmiersystem, vielmehr läßt sich mit ihr die gesamte Externsteuerung des Rechensystems formulieren.

Die Elemente der Sprache sind Kommandos, die mit einem speziellen Zeichen, dem Fluchtsymbol (im weiteren durch \square dargestellt), beginnen müssen, um sie von den übrigen Eingabedaten trennen zu können.

Tätigkeitskommandos

Das Grundelement der Kommandosprache ist das Tätigkeitskommando, mit dem eine bestimmte Leistung vom Programmiersystem verlangt wird. Es beginnt mit einem Tätigkeitsnamen, auf den Spezifikationen folgen, z. B.:

\square UEBERSETZE, SPRACHE = FTN, QUELLE = TEXT, ...

Für jede Tätigkeit ist eine Menge von Spezifikationen definiert (im Beispiel SPRACHE, QUELLE usw.). Hält man eine vorgegebene Reihenfolge ein, können die Spezifikationsnamen entfallen. Auch das Abkürzen der Tätigkeits- und Spezifikationsnamen ist möglich, solange Eindeutigkeit gewährleistet ist, z. B.:

\square UEB., FTN, QU. = TEXT, ...

Voreinstellung

Zu den einzelnen Spezifikationen, die in obligate und optionale unterteilt werden, existiert eine globale Voreinstellung. Diese Voreinstellung tritt in Kraft, wenn zu einer Spezifikation keine Aussage gemacht wird. Zu den Tätigkeitskommandos zählen:

- Grunddienste
Übersetzen, Montieren, Starten und Löschen von Objekten
- Allgemeine Dateidienste
Einrichten, Einschleusen, Sichern und Sperren von Dateien
- Texthaltungsdienste
Eintragen, Korrigieren, Mischen und Kopieren von Texten
- Datenmanipulationen
Sortieren und Mischen von Sätzen, Komprimieren von Texten
- Datentransportdienste
Ein- und Ausgabe von Binärobjekten, Ausgabe von Daten
- Assembler-Makrodienste
Eintragen, Löschen und Informieren
- Langfristige Datenhaltung
Einrichten, Aufgeben und Verwalten von langlebigen Dateien

Erweiterung des Kommandovorrats

Der vorhandene Kommandovorrat läßt sich über das DEFINIERE-Kommando oder über eine Prozedurvereinbarung erweitern, ohne daß dabei bestehende Kommandos versehentlich gelöscht werden können.

Durch das DEFINIERE-Kommando neu geschaffene Kommandos starten und versorgen bei Aufruf den bei der Definition angegebenen Operator, den der Benutzer selbst geschrieben hat.

Durch eine Prozedurvereinbarung wird eine Folge von vorhandenen Kommandos zu einem neuen Kommando zusammengefaßt. Dabei können formale Parameter eingeführt werden, die dann als Spezifikationen des neuen Kommandos auftreten.

Beispiel:

Prozedurvereinbarung:

```
□ *RECHNE (TEXT, SPRACHE, PROG)
□ UEBERS., QUELLE=*TEXT, SPR.=*SPRACHE
□ MONTIERE, PROGRAMM=*PROG
□ STARTE, PROGR.=*PROG, DUMP=F-NEST' A-NEST
□ * *
```

Prozeduraufruf:

```
□ RECHNE, TEXT=QUELLE, SPR.=FTN, PROG=TEST
```

Kommandoablauf

Wird der Auftrag des Benutzers im Abschnittsmodus bearbeitet, so kann die Übersetzung einer Quelle fehlerhaft verlaufen, ohne daß der Benutzer wie bei einem Gespräch dies feststellen kann. Ein nachfolgender Montage- und Startversuch wären sinnlos. Für solche Fälle gibt es zwei spezielle Kommandos, die den Auftrag nach einem Fehler abbrechen (FEHLERHALT) oder in Abhängigkeit von auftragsspezifischen Variablen die vorgegebene Kommandoabfolge verlassen (SPRINGE).

4.3 Übersetzungsvorgang (source translation)

Eine der wichtigsten Leistungen des Programmiersystems ist das Umsetzen einer Quelle in die darin gewünschte Rechnerleistung. Die hierzu erforderliche Übersetzung erfolgt grundsätzlich zweistufig (s. Bild 7).

Die Quelle, die entweder in einer Datei oder in der Form von Eingabedaten auf einem Hintergrundspeicher liegt, wird zunächst von einem Sprachübersetzer verarbeitet. Dieser erzeugt daraus ein Montageobjekt, das aus dem entsprechenden Montagecode und aus Adreßbüchern für die Rückverfolgung im Fehlerfall besteht, falls Dumpfähigkeit verlangt wird (Spezifikation im Kommando). Der Montagecode selbst stellt keinen lauffähigen Maschinencode dar. Dieser wird erst bei der sich anschließenden Montage erzeugt.

Der Montierer sorgt für die Verknüpfung und günstige Anordnung von einem oder mehreren Montageobjekten in dem von ihm daraus erstellten lauffähigen Operator (vergl. 4.1).

Im Bild 7 ist der Übersetzungsvorgang dargestellt, wie er beispielsweise verlaufen kann, solange der Abwicklerprozeß den Auftrag bearbeitet (vergl. Bild 6).

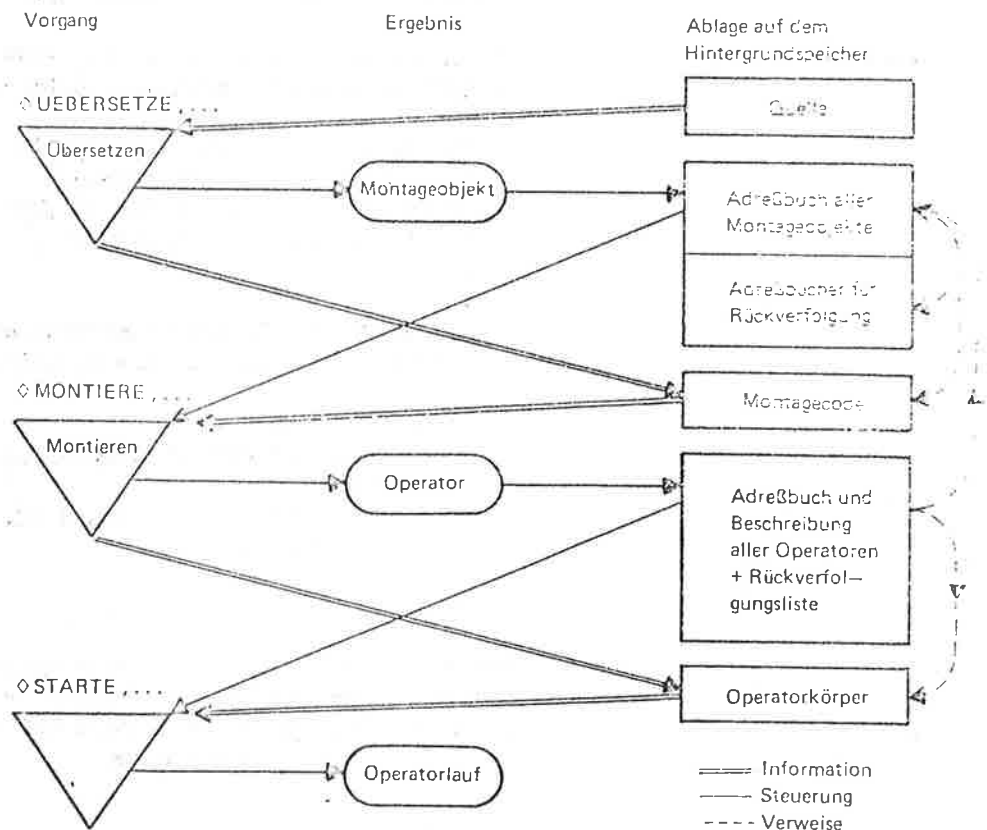


Bild 7 Übersetzungsvorgang

4.4

Mischung von Sprachen

Die Zweistufigkeit der Übersetzung dient u. a. dazu, Prozeduren aus verschiedenen Sprachen gemeinsam verwenden zu können. Dabei ist der gegenseitige Aufruf in den Sprachen ALGOL 60, FORTRAN und TAS (Telefunken-Assembler) realisiert. Aus TAS heraus können auch COBOL-Routinen, aus COBOL Routinen der restlichen genannten Sprachen aufgerufen werden.

Der Zusammenschluß erfolgt während des Montagevorgangs. Der Übergang zwischen zwei Sprachen innerhalb einer Prozedur (z. B. Einfügen von Assembler-Befehlen in ein ALGOL-Programm) ist nicht möglich. Für den Anschluß von Assembler-Prozeduren an höhere Sprachen sind nur die Konventionen einzuhalten, nach denen ein bestimmter Compiler seine Montageobjekte erzeugt. Ebenso sind beim Anruf von Prozeduren höherer Sprachen aus einem TAS-Programm die Konventionen der jeweiligen Sprache zu berücksichtigen. Der Zusammenschluß von Prozeduren höherer Sprachen soll an einem Beispiel zwischen ALGOL und FORTRAN erläutert werden. Wegen der verschiedenartigen Programmstruktur sind einige Anpassungsarbeiten erforderlich, die von einer für den Benutzer unsichtbaren Zwischenprozedur durchgeführt werden, die auch rekursiv aufgerufen werden kann.

Beispiel

Es soll erreicht werden, auch in FORTRAN mit Hilfe einer ALGOL-Prozedur dynamische Felder zu verwalten.

- Das FORTRAN-Programm für die Matrixmanipulationen wird als Subroutine organisiert:
SUBROUTINE MATRIX (N, A1, A2)
DIMENSION A1 (N, N), A2 (N)
:
:
:
- Das folgende ALGOL-Programm enthält die Deklaration der beiden Felder a1, a2 und den Aufruf der FORTRAN-Prozedur:

```
'begin' 'integer' n;  
      'procedure' matrix (x, y, z); 'fortran';  
      read (n);  
      'begin' 'array' a1 [1: n, 1:n], a2 [1:n] ;  
      matrix (n, a1, a2)  
      'end'  
'end'
```

Der Übergang zwischen beiden Sprachen ermöglicht auch die fast uneingeschränkte Verwendung von Prozeduren der Programmbibliothek ohne Rücksicht auf die Sprache, in der diese Prozeduren geschrieben werden.

4.5 Sprachübersetzer

Im folgenden soll ein kurzer Überblick über die im TR 440-Teilnehmer-Rechensystem implementierten Programmiersprachen gegeben werden.

TAS Telefunken- Assemblersprache

TAS ist eine formatfreie Sprache mit symbolischer Adressierung, die neben den eigentlichen 240 Befehlen (Maschinenanweisungen) noch sogenannte Pseudobefehle (Anweisungen an den Assembler) kennt. Eine Makrobibliothek (mit Makros für Ein- und Ausgabe) steht dem Benutzer zur Verfügung und kann von ihm erweitert werden.

FORTTRAN

Der FORTRAN-Compiler bearbeitet zwei Versionen:

- FORTRAN IV nach ASA-Standard
- FORTRAN-440, das vollkompatibel zu FORTRAN IV ist, mit Spracherweiterungen für die Ein- und Ausgabe an Terminals.

Neben den FORTRAN-Standardprozeduren existieren FORTRAN-Routinen, die Stringmanipulationen, d.h. Textverarbeitung ermöglichen.

ALGOL

Es wurde bis auf own-Variable der volle Sprachumfang von ALGOL 60 implementiert. Als Erweiterung ist ein fortran-ähnliches COMMON-Konzept realisiert worden. Die zur Verfügung stehenden Ein- und Ausgaberroutinen umfassen die ALCOR- und ISO-Prozeduren, alle Prozeduren des Knuth-Proposals und einige spezielle Prozeduren für Dateimanipulationen.

COBOL

Der realisierte Sprachumfang entspricht dem ANSI-Standard COBOL 68 ohne Reportwriter, aber mit der COBOL-Bibliothek der sprachspezifischen Segmentierung, der Verarbeitung von dreidimensionalen Feldern der Sortierung.

Für die COBOL-Projekte gibt es neben dynamischen Kontrollen und dem Prozedurtracing (vergl. 4.6) eine Überwachung von Variablen. Es existiert ein Dump, der auf Wunsch Variable gemäß ihrer Beschreibung in der Datenhierarchie in lesbarer Form ausdrückt.

BASIC

BASIC ist eine fortranähnliche Sprache, die speziell für Teilnehmersysteme gedacht und leicht zu erlernen ist. Der Interpreter ist voll gesprächsfähig und ermöglicht nahezu unbeschränkte Manipulationen der Quellen. Der realisierte Sprachumfang umfaßt die Implementierungen von SIEMENS und GE.

PL/I

PL/I ist eine problemorientierte formatfreie Sprache für die mathematische und kaufmännische Programmierung. Der im TR 440 implementierte Sprachumfang entspricht dem ECMA/ANSI-Standard mit einigen Erweiterungen. Die Erweiterungen beziehen sich im wesentlichen auf zusätzliche Built in-Funktionen und Macro-Facilities.

GPSS
General Purpose
Simulation System

GPSS ist eine Sprache zur direkten Systemsimulation. Sie dient der Formulierung und Behandlung von Problemen auf den Gebieten der Unternehmensforschung und der Organisationsanalyse ebenso wie dem technisch-wissenschaftlichen Bereich. GPSS wird als interpretatives System mit einem Vorübersetzer und dem Simulationsprozessor implementiert, wobei auf weitgehende Kompatibilität mit GPSS/360 geachtet wird.

RPG
Report Program Generator

RPG ist eine problemorientierte, formatgebundene Sprache für kommerzielle und organisationsbezogene Programmierung. Wesentliche Spracherweiterungen beziehen sich auf den Aufbau von Tabellen. Der Zugriff wurde auf direkten Zugriff (indiziert) und sequentiellen Zugriff (vorwärts und rückwärts) erweitert.

4.6
Testhilfen

In diesem Abschnitt sollen Hilfen beschrieben werden, die das Programmiersystem zum Austesten von Programmen zur Verfügung stellt.

Quellbezogene Dumps

Hierzu gehören in erster Linie quellbezogene Dumps, die bei irregulären Abbrüchen der Bearbeitung oder auch auf explizites Verlangen des Benutzers die Variablen ausdrucken. Dabei werden die in der Quelle angegebenen Namen von Variablen und deren Inhalt in lesbarer Form ausgedruckt, so daß Referenzlisten nicht benutzt werden müssen. Die Druckausgabe des Dumps kann weitgehend gesteuert werden. So kann u. a. der Ausdruck aller Variablen mit einzelnen Ausnahmen (z. B. große Felder) oder nur der Variablen, die an der aktuellen Aufrufverschachtelung beteiligt sind, oder der Druck explizit angegebener Variablen verlangt werden.

Rückverfolger

Von einem ausgezeichneten Operator - dem Rückverfolger - wird außerdem die Fehlerstelle, ebenfalls unter Bezugnahme auf die Originalquelle, lokalisiert. Zusätzlich wird die aktuelle Aufrufverschachtelung ausgegeben. Welche Dumps in welcher Form im Alarmfall ausgeführt werden sollen, wird im STARTE-Kommando spezifiziert.

Tracing

Eine weitere Testhilfe besteht in der Möglichkeit, ein Programm überwacht ausführen zu lassen. Der Ausdruck von Registerständen oder Speicherbereichen kann dabei vom Ansprechen eines bestimmten Speicherbereichs oder vom Auftreten eines bestimmten Befehlscodes abhängig gemacht werden (z. B. Protokollierung aller auftretenden Sprungbefehle). Ähnliche Ablaufüberwachungen sind auch bei Programmen möglich, die aus Quellen in höheren Programmiersprachen entstanden sind. Die Routinen werden im UEBERSETZE-Kommando angefordert und im STARTE-Kommando aktiviert.

Dynamische Kontrollen

Auf Wunsch des Benutzers werden beim Übersetzungsvorgang dynamische Kontrollen eincompiliert, die sich beziehen auf:

- Einhaltung von Indexgrenzen
- Verträglichkeit von aktuellen und formalen Parametern
- Zulässigkeit von Schleifenparametern.

Damit lassen sich statistisch und syntaktisch nicht erkennbare Fehler lokalisieren.

4.7

Dialogbetrieb

Im Gespräch von einer Benutzerstation aus stehen bis auf unwesentliche, durch den Gesprächsmodus bedingte Einschränkungen alle Dienstleistungen des Programmiersystems zur Verfügung.

Nach der Eröffnung eines Gesprächs besteht die Möglichkeit, ein einzelnes oder eine Folge von Kommandos einzugeben.

Im Gesprächsmodus können dabei in eine bereits eingegebene Kommandofolge Kommandos eingefügt werden, die dann vorrangig ausgeführt werden.

Beispielsweise läßt sich bei einer fehlerhaften Übersetzung, der Fehler mit einem Kommando korrigieren, um anschließend die Übersetzung zu wiederholen. Nach den vorrangig ausgeführten Kommandos wird nur auf expliziten Wunsch des Benutzers an der Unterbrechungsstelle fortgefahren.

Anweisungen

Im Gesprächsmodus können außer Kommandos auch Anweisungen an den Abwicklerprozeß, der den Auftrag bearbeitet, oder den Entschlüssler und den Objektlauf des Benutzers gegeben werden. Anweisungen werden immer nur einzeln und auf Anforderung verarbeitet. Der Entschlüssler versteht z.B. Anweisungen zum Fortsetzen an der Unterbrechungsstelle und zum Löschen aller eingegebenen und noch nicht ausgeführten Kommandos.

Texthaltung

Arbeitet der Benutzer im Gespräch mit umfangreichen Quellen und Datenmengen, so ist es unbequem, diese von einer Benutzerstation aus einzugeben. Vielmehr ist in diesem Fall der Gebrauch der Langfristigen Datenhaltung zu empfehlen (vergl. 3.7).

Unabhängig von oder auch im Zusammenhang mit LFD-Dateien arbeitet die Texthaltung von Quellen. Mit ihrer Hilfe lassen sich Quellen in Dateien eintragen, einzelne Zeilen korrigieren, löschen oder auch vertauschen. Auch einzelne Zeichen einer Textzeile lassen sich eintragen, auswechseln oder löschen. Ferner kann der Benutzer mehrere Quellen zu einem Quellentext zusammenfügen, mischen oder Teile korrigieren.

Erweiterte Testmöglichkeiten

Das Austesten eines Programms wird gerade im Gesprächsmodus durch die Möglichkeit, auf Ergebnisse sofort zu reagieren, vereinfacht und beschleunigt. Ein gesprächsfähig montierter Operator kann jederzeit während eines Laufs angehalten werden. Im UEBERSETZE-Kommando werden sogenannte Kontrollereignisse definiert, indem den Nummern von Quellzeilen Namen zugeordnet werden.

Ist ein Kontrollereignis aktiv (Angabe im STARTE-Kommando), so hält der Operatorlauf an der entsprechenden Stelle an und meldet den Namen des Ereignisses auf die Benutzerstation. Darauf sind mehrere Reaktionen möglich:

- Fortsetzen oder Beenden des Operatorlaufs
- Aktivieren oder Passivieren von definierten Kontrollereignissen
- Bringen oder Umsetzen der Werte einzelner Variablen, die frei aus dem Programm wählbar sind
- Dump aller Variablen nach Anweisung auf die Benutzerstation oder dem Schnelldrucker (gepuffert)
- Einfügen von Kommandos aller Art
- Anhalten bei Operatorlaufende, d.h. vor Ausführung des nächsten anstehenden Kommandos.

