

RRZE

BENUTZERINFORMATION

BI 5 - ERLANGEN - 15. DEZEMBER 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

Organisatorische Hinweise

Kurzbeschreibung der Rechenanlage CYBER 172.

Vorwort:

Mit dem Beginn des Jahres 1977 wird das Regionale Rechenzentrum Erlangen den regulären Betrieb mit dem Dreifachprozessor TR 440 aufnehmen. Gleichzeitig wird die CYBER 172 installiert.

Diese letzte Benutzerinformation im Jahre 1976 enthält deshalb, neben einigen organisatorischen Hinweisen, eine kurze Einführung in den Aufbau der CYBER 172.

Wir wünschen allen Benutzern ein frohes Weihnachtsfest und einen guten Rutsch ins Neue Jahr.

Vorstand, Leitung und Mitarbeiter
des Regionalen Rechenzentrums

Organisatorische Hinweise

1. Aus Gründen der Kosteneinsparung wird - entgegen der bisherigen Ankündigung - der Betrieb des Regionalen Rechenzentrums Erlangen vom 24.12.76 bis 9.1.77 eingestellt. Falls dadurch bei der Durchführung einzelner Arbeiten Engpässe auftreten, werden wir uns bemühen, diese entweder kurzfristig vor Weihnachten oder nach der Wiederaufnahme des Rechenbetriebes am 10. Januar auch außerhalb des normalen Schichtbetriebes zu befriedigen.



2. Ab Januar 1977 kann die Rechenzeit am Dreifachprozessor TR 440 nicht mehr kostenlos zur Verfügung gestellt werden. Die Rechenzeitabrechnung erfolgt im wesentlichen zunächst nach den bereits 1976 praktizierten Richtlinien. Aus programmtechnischen Gründen wird die Abrechnung vorerst weiterhin in CD 3300-Einheiten vorgenommen und zwar nach der Formel, die neben der CPU Zeit auch die Verweilzeit im Speicher und den Papierverbrauch berücksichtigt. Im Mittel ergeben sich dabei Rechenzeitkosten in Höhe von DM 40,-- pro CPU Stunde. Es ist jedoch zu berücksichtigen, daß eine CPU Stunde TR 440 mit 3,4 CPU Stunden 3300 angesetzt wird.
3. Die bisherige Informatikanlage TR 440 wurde termingemäß außer Dienst gestellt.
4. Der Umbau des bisherigen 3300-Rechnerraumes geht planmäßig vonstatten. Die Installation der CYBER 172 ist für den 10. Januar 1977 vorgesehen, so daß der Probetrieb spätestens am 1.2.1977 beginnen kann.
5. Die Entscheidung über die Stapelfernverarbeitungsstationen ist noch nicht gefallen. Die Bestellung soll jedoch noch im Jahr 1976 erfolgen.
6. Auf folgende Termine des RZ-Kolloquiums soll an dieser Stelle noch einmal hingewiesen werden:
 - 11.1.77 Einführung in die Rechenanlage CYBER 172,
Steuerkartenbeispiele
 - 18.1.77 Umstellung CD 3300 - CYBER 172, insbesondere
FORTRAN

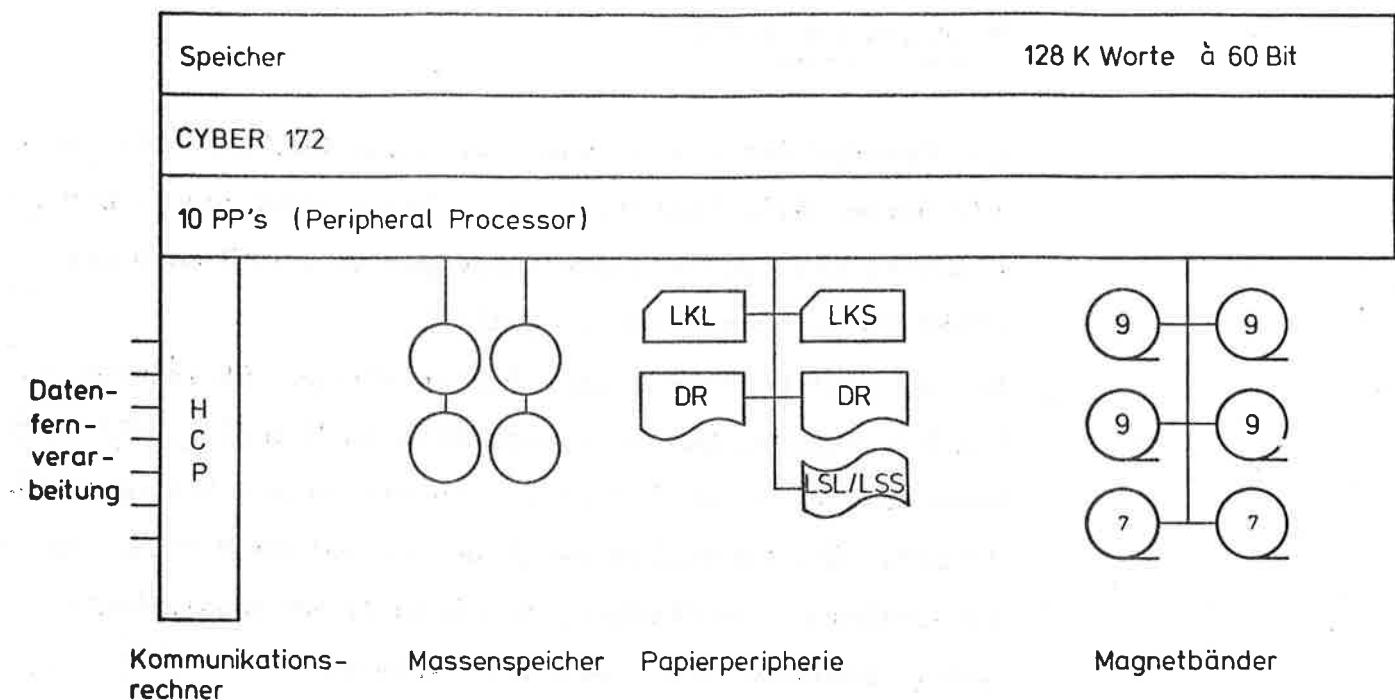
Zusätzlich ist Ende Januar ein 2-tägiger Einführungskurs geplant. Wir bitten um Ihr Verständnis, daß wir diese Termine erst kurzfristig vorher bekannt geben können.

Kurzbeschreibung der Rechenanlage CD CYBER 172

<u>Gliederung</u>	<u>Seite</u>
o. Geräteliste und Konfigurationsplan	1
1. Hardware	3
1.1 Zentralprozessor	3
1.2 Hauptspeicher	5
1.3 System der peripheren Prozessoren und Kanäle	6
1.4 Kommunikationsrechner	9
1.5 Bedienplatz	11
1.6 Plattenspeicher Typ CD 844-41	12
1.7 Die Magnetbandeinheiten der Serie 66X	13
2. Software	15
2.1 Systemsoftware	15
2.2 Auftragsverwaltung	19
2.3 Dateiverwaltung	29
3. Assembler und Compiler	35
3.1 Assembler	35
3.2 Übersetzer für Programmiersprachen	37
3.2.1 ALGOL 60	37
3.2.2 BASIC	38
3.2.3 COBOL und SORT/MERGE	39
3.2.4 FORTRAN EXTENDED IV	40
Anhang	
Benutzerkomfort unter NOS	41
1. NOS Kommandosprache	42
2. NOS Dienstprogramme	43
3. NOS LOADER (Laden + Binden)	45
4. NOS Record Manager (Ein-/Ausgabe)	47
5. NOS Text Editor (nur Dialogbetrieb)	48
6. FORTRAN	50
7. ALGOL 60	53
8. COBOL	58
9. PL/1	61
10. Austauschbarkeit von Unterprogrammen	62
11. Dialog mit Programmen	63

o. Geräteliste und Konfigurationsplan

Am 11.1.1977 wird die Rechenanlage CYBER 172 geliefert und im Rechnerraum der CD 3300 aufgestellt. Sie finden auf der nächsten Seite den Konfigurationsplan und eine Aufstellung der Geräte mit den wichtigsten technischen Daten. Die folgende einführende Beschreibung der neuen Rechenanlage (ab Seite 3) wurde uns von der Firma Control Data zur Verfügung gestellt.



CYBER 172 Zentraleinheit

128 K 60 Bit-Worte Hauptspeicher

10 periphere Prozessoren mit je 4096 Worte Speicher

4 Wechselplattenspeicher à 200 Mio. Zeichen über 2 Steuereinheiten

4 Magnetbandgeräte, 9-Spur, 120 bzw. 240 kHz, 800-1600 CPI

2 Magnetbandgeräte, 7-Spur, 40 kHz

1 Lochkartenleser, 1200 Karten/Min.

2 Drucker, 1200 Zeilen/Min.

1 Lochkartenstanzer, 250 Karten/Min.

1 Lochstreifenleser/Stanzer (FAZIT)

1 Kommunikationsrechner (HCP) mit 32 K Speicher incl.

Loop Multiplexer und Adapter

14 Sichtgeräte CD 751

16 Sichtgeräte CD 751 (Fernschreiber kompatibel)

} Geräte werden von
der CD 3300 übernommen

1. Hardware CYBER 172

Der Rechner ist das Anfangsmodell der Control Data CYBER 170 Serie. Alle Modelle dieser Serie sind sowohl untereinander als auch mit den Vorgängerserien CD 6000 und CYBER 70 softwaremäßig kompatibel.

Bis zu vier Rechner dieser Serien können miteinander gekoppelt werden. Das leistungsfähigste Modell (CYBER 175) kann maximal 40 Millionen Instruktionen pro Sekunde ausführen. Die Ausbaufähigkeit der einzelnen Modelle und der reibungslose Übergang zu einem leistungsstärkeren Modell zeichnen die CYBER 170 Serie aus.

Modernste Technologie (MOS-Speicher, ECL-Schaltkreise) und das Konzept des verteilten Rechnens sind die wesentlichen Leistungsmerkmale.

1.1 Zentralprozessor

Die Informationseinheiten der Zentralprozessoren sind 60-Bit Worte:

Zeichendarstellung:	10 Zeichen zu 6 Bit
Zahlendarstellung:	Festpunkt 48 Bit
	Gleitpunkt 12 Bit Exponent
	48 Bit Mantisse

Doppelte Genauigkeit mit
96 Bit Mantisse

Ausführungszeiten (Gleitpunkt)

Addition	0.85 us
Subtraktion	0.85 us
Multiplikation	3.20 us
Division	3.20 us
Laden implizit über Adress- Abspeichern register	1.05 us 0.25 us

Befehlsvorrat: 78 Befehle, zwei bis vier Befehle pro Wort. Überwiegend Dreiaadress-Befehle.

Programmwechsel: 2,45 us für Register-Austausch und Starten; kann auch von peripheren Prozessoren angestoßen werden.

24 Arbeitsregister: 8 Arithmetikregister zu 60 Bit
8 Adressregister zu 18 Bit
8 Indexregister zu 18 Bit

Zeichenmanipulation: 4 Compare/Move Befehlsarten

Taktzeit: 50 ns

Hilfsregister: 7 Register für zusätzliche Funktionen

Rechenwerke für: Befehls- und Adressaufbereitung
60 Bit Arithmetik
18 Bit Arithmetik

Besonderheiten: Maschineninterne Darstellung von infiniten und indefiniten Werten, keine E/A-Befehle

1.2 Hauptspeicher

Der Hauptspeicher ist von 32 K Worten bis zu 128 K Worten (262 K bei den Modellen 173, 174 und 175) ausbaubar. Er ist in acht (bis zu 16) Bänken organisiert. Die Hauptspeicher-Zugriffe können achtfach überlappt erfolgen, so daß sich bei einer Zykluszeit von 400 ns eine maximale Übertragungsrate von einem Wort in 50 ns (200 Mio. Zeichen/Sek.) ergibt. Er kann durch einen Erweiterungsspeicher (ECS, bis zu 2 Millionen 60-Bit Worte) mit einer Übertragungsrate von 100 Millionen Zeichen pro Sekunde ergänzt werden.

Die Hauptspeicher-Kontrolleinheit (CMC) koordiniert prioritengesteuert den Zugriff von maximal 22 Prozessoren (2 Zentralprozessoren und 20 peripheren Prozessoren) und die Verbindung zum Erweiterungsspeicher.

Sie führt außerdem Paritäts- und Breakpointprüfungen, Programmwechselkontrollen und die Aufgaben der SECDED-Logik aus. Die SECDED-Logik (Single Error Correction, Double Error Detection) basiert auf der Hauptspeicher-Struktur von 68 Bit (60 Bit Daten und 8 Bit für Einzelbit-Fehlerkorrektur/Mehrbit-Fehlererkennung).

Der Hauptspeicher kann ohne Eingriffe ins Betriebssystem in Stufen zu 32 K rekonfiguriert werden. Dabei bleibt die achtfache Zugriffsüberlappung erhalten, weil sich nur die

Größe der verschränkten Blöcke, nicht aber deren Anzahl verändert.

Die im Hauptspeicher angelegten Adressräume für Benutzerprogramme werden hardwaremäßig gegen Bereichsüberschreitung geschützt.

1.3

System der peripheren Prozessoren und Kanäle

Da nach dem Prinzip des verteilten Rechnens der Zentralprozessor keine langsamten Ein-/Ausgabe- und Verwaltungsaufgaben ausführen sollte, verfügt er bei den CYBER-Modellen konsequenterweise über keine derartigen Befehle.

Diese Aufgaben werden vielmehr von ein (CYBER 172) oder zwei (CYBER 173-175) Systemen peripherer Prozessoren übernommen. Ein peripheres Prozessor-System (PP-System) besteht aus jeweils bis zu 10 (Abstufungen 10, 14, 17, 20) funktionell unabhängigen Prozessoren mit eigenem Speicher.

Rechenwerk: Jedes PP-System besitzt ein schnelles Rechenwerk mit 64 Instruktionen, das in einem multiplex-Verfahren den 10 Registersätzen der peripheren Prozessoren (PP) zugewiesen wird. Alle 500 ns oder 1000 ns (softwaremäßig wählbar) kann der PP das Rechenwerk für 50 ns verwenden.

Speicher: Jeder PP hat einen eigenen MOS-Speicher mit 4 K Worten zu 12 Bit (+ 1 Paritätsbit). In ihm werden die Programme ausgeführt und Pufferbereiche verwaltet.

Kanäle: Zu jedem PP-System gehört ein Satz von 12 E/A-Kanälen. Ein Kanal kann 4 Millionen Zeichen pro Sekunde übertragen. Jeder PP kann über eine Matrix-Schaltung jeden Kanal selektieren und somit auch mit jedem anderen PP kommunizieren.

Hauptspeicherzugriff:

Jeder periphere Prozessor kann auf den Hauptspeicher (auch auf den Erweiterungsspeicher) zugreifen. Dabei wird ein 60-Bit Wort in 5 12-Bit Worte zerlegt und umgekehrt werden 5 12-Bit Worte zu einem 60-Bit Wort vereinigt. Pro PP-System sind acht gleichzeitige Zugriffe möglich (4 Leseoperationen und 4 Schreiboperationen).

Status- und

Kontroll-

Register : Bestandteil des PP-Systems ist ein 204 Bit umfassendes programmgesteuertes Register. Es erlaubt die Steuerung der Paritäts- und SECDED-Netzwerke, Überwachung von Breakpoint-Adressen, Umschaltung der PP-Geschwindigkeit und Durchführung von vielen Wartungsaufgaben

(wie beispielsweise die programmgesteuerte Veränderung der Spannungspegel).

Beim Systemstart wird das Urladeprogramm in den peripheren Prozessor Null (PP0) geladen. Um beim Ausfall dieses Prozessors trotzdem einen System- oder Wartungsdeadstart (Kaltstart/Warmstart) ausführen zu können, erlauben Schalter am PP-Chassis die Selektierung eines beliebigen PP als PP0.

Bleibt ein PP während der Ausführung einer Instruktion infolge fehlender Rückmeldung stehen (z.B. bei einem Kanalfehler), so kann er über das Status- und Kontrollregister zum Durchlaufen der vorgeschriebenen Hardwaresequenz gezwungen werden. Auf ähnliche Weise kann auch ein fehlerhaft arbeitender PP durch einen anderen PP neu geladen und gestartet werden (PP-Deadstart).

1.4

Kommunikationsrechner

Die Kommunikationsrechner vom Typ 2550 können in zwei Leistungsklassen mit 10000 bzw. 30000 Zeichen/Sek. Durchsatz reiner Daten angeboten werden.

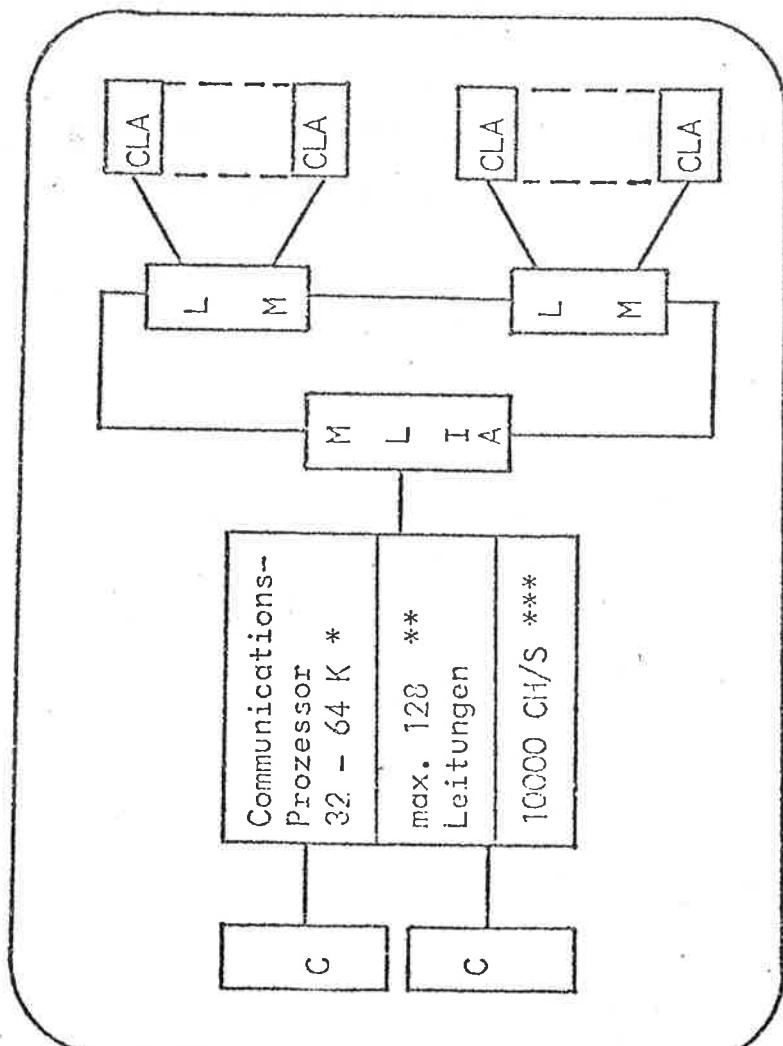
Es handelt sich um mikroprogrammierbare Rechner, die in hohem Maße Parallel-Verarbeitung durchführen. Der Befehlsvorrat ist speziell auf die Datenübertragung und Zeichenverarbeitung zugeschnitten. Die Befehlausführungszeit beträgt 168 ns. Der MOS-Speicher kann von 32 K auf 128 K 16-Bit Worte ausgebaut werden und hat eine Zykluszeit von 550 ns.

Die Übertragungsleitungen werden von einem bedarfsgesteuerten Multiplex-System bedient. Dieses wird nur belastet, wenn tatsächlich Daten zur Verarbeitung anstehen. Die Eingabegeschwindigkeit ist unabhängig von der Ausgabegeschwindigkeit.

Auf der Leitungsseite werden die Schnittstellen V 24, V 35 oder AT&T 301/303 versorgt. Auf der Hauptrechnerseite ist der Anschluß an einen E/A-Kanal der CYBER 170 Modelle standardmäßig vorhanden.

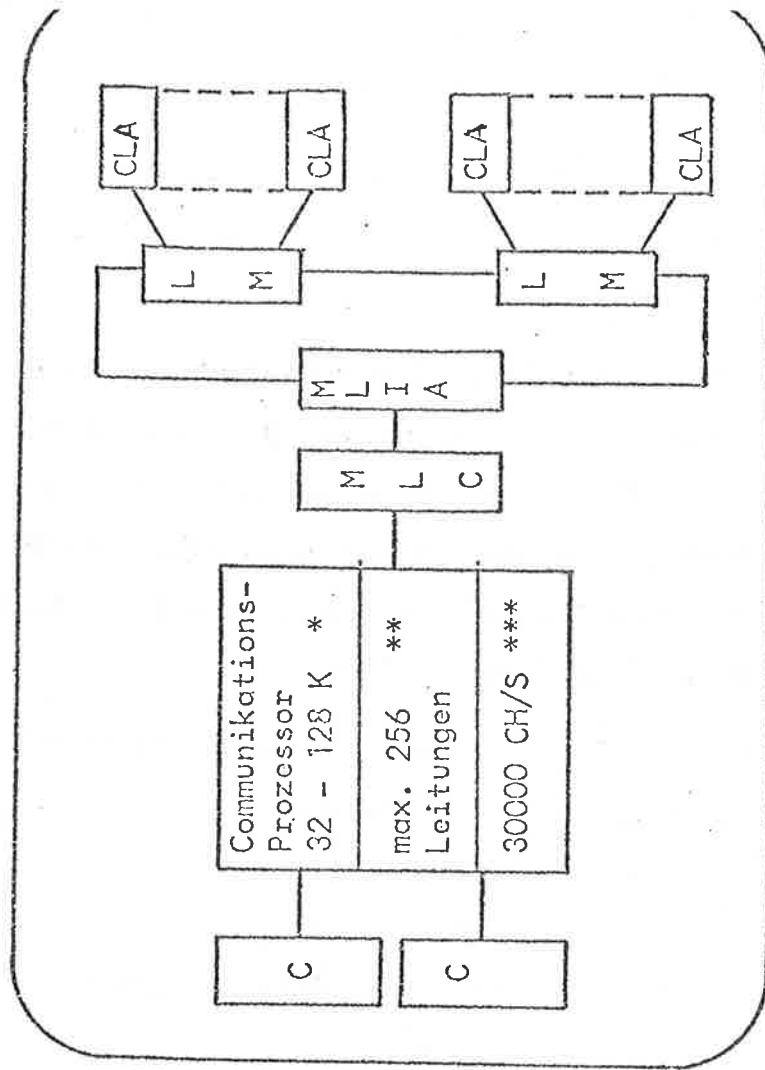
Kommunikations-Rechner

Modell 2550-2



* 16-bit Worte
 ** 2 Leitungen / CLA
 *** Durchsatz in Zeichen/Sek.
 C Channel Coupler

Modell 2552-1



MLIA Multiplex Loop Interface Adapter
 MC Multiplex Loop Controller
 LM Loop Multiplexer
 CLA Communication Line Adapter

1.5

Bedienplatz

Als Bedienplatz für den Operateur wird eine Bildschirmkonsole mit Blink-Einrichtung verwendet. Als Nebenbedienplätze können weitere Konsolen oder beliebige Sichtgeräte angeschlossen werden. Die Protokollierung der Operateurmeldungen ist über jeden beliebigen Drucker möglich.

Die Konsole kann auf mehrere Arten betrieben werden:

- a. Punktraster-Verfahren
(262 144 Bildpunkte)
- b. Buchstaben-Verfahren
(Es stehen drei verschiedene Buchstabengrößen zur Verfügung).
- c. Wahlweise Aufteilung in zwei unabhängige Bild-Hälften auf einem Bildschirm.

Die Bedienplätze werden über einen eigenen Kanal von einem eigens dazu bestimmten peripheren Rechner betrieben.

1.6 Plattenspeicher Typ CD 844-41

Der Plattenspeicher Typ 844-41 ist ein Massenspeicher mit austauschbaren Plattenstapeln. Die Steuereinheit gestattet die Steuerung von 8 Plattenspeichern. Außerdem besteht die Möglichkeit, 2 Steuereinheiten an je einen Plattenspeicher anzuschließen (multi access) und die 2 Steuereinheiten über separate Kanäle zu betreiben. Zudem hat jede Steuereinheit Eingänge für 2 Kanäle (dual channel). Bei Verwendung einer einzigen Steuereinheit besteht damit die Möglichkeit, während eines Lese- oder Schreibvorgangs mehrfach überlappende Suchoperationen durchzuführen. Zwei Steuereinheiten ergeben unter Beibehaltung der mehrfach überlappten Suchoperationen zwei gleichzeitige Datenübertragungen zu zwei beliebigen Laufwerken.

Diesen Plattenspeicher liefert CD auch mit halber Spurdichte und einer Kapazität von ca. 118 Mio. Zeichen (Typ 844-21), sowie einem Zusatz (Expander), der den Anschluß von maximal 64 Laufwerken pro Steuereinheit ermöglicht.

844-41 Technische Daten

Bit-Dichte (BPI)	4040
Spuren-Dichte (SPI)	384
Spuren pro Oberfläche	808
Oberflächen pro Stapel	19
Zahl der Spuren/Stapel	15352
Kapazität/Stapel	237 Mio. Zeichen
Transferrate, Zeichen/sec.	1075 Mio. Zeichen Durchschnittl.
Positionierungszeit	30 ms Durchschnittl.
Latenzzeit	8.3 ms

1.7 Die Magnetbandeinheiten der Serie 66X

Diese neue Serie von Magnetbandeinheiten hat die folgenden charakteristischen Eigenschaften:

- Bandantriebsmotor
- automatische Bandeinfädelung
- automatische Befestigung und Lösung der Bandrolle
- Bandkassettenbetrieb oder Betrieb mit normalen Bandspulen
- automatisches Verschließen des Ladefensters
- Schreibschutz (Schreibring)
- Vorwärts- und rückwärts lesen
- Schreibverfahren: NRZI/PE
- Vertikal-/Longitudinal-/Zyklische Redundanz- und bit-Verlust-Prüfung. 1-Spur-Fehler-Korrektur

Spezifikationen:

Typ	Spuren	Lese/Schreibdichte Zeichen/inch	max. Trans- ferrate Zeichen/sec	Rückspulzeit 2400 Fuß/Band
667-2	7	200*, 556,800	80000	60 sec.
667-3	7	200*, 556,800	120000	50 sec.
667-4	7	200*, 556,800	160000	45 sec.
669-2	9	800, 1600	160000	60 sec.
669-3	9	800, 1600	240000	50 sec.
669-4	9	800, 1600	320000	45 sec.

* 200 BPI: nur Lesen möglich

Die Steuereinheiten CDC 7021-21 und CDC 7021-22 können jeweils bis zu 8 Magnetbandeinheiten einer beliebigen Typkombination der 66X-Serie betreiben. Die Steuereinheiten enthalten 1 bis 2 kleine Prozessoren, deren austauschbare Programme als Controlware bezeichnet werden. Diese Controlware betreibt die angeschlossenen Bandeinheiten im Parallelbetrieb. Dies ermöglicht das Abschalten einzelner Bandgeräte während des laufenden Betriebs und das off-line-Testen dieser Einheiten.

Das Modell 7021-22 enthält 2 Prozessoren und 2 Kanalkoppler. Dies erlaubt gleichzeitiges Lesen und/oder Schreiben zweier Bandeinheiten.

2. SOFTWARE

2.1 Systemsoftware

NOS - Network Operating System

NOS realisiert ein als "Distributed Computing" bekanntes Konzept. Danach werden alle anfallenden Aufgaben den für die Durchführung jeweils am besten geeigneten Systemkomponenten zugeteilt.

Für die CYBER-Modelle bedeutet das:

- Ausführung der Benutzerprogramme auf schnellen Zentraleinheiten (mit 60-Bit-Arithmetik und Zeichenbehandlung)
- Ausführung der Verwaltungs- und Ein/Ausgabeprogramme in peripheren Prozessoren mit eigenem Speicher
- Behandlung von E/A- und Leitungsprozeduren in mikroprogrammierbaren Steuereinheiten (Controlware).

Entsprechend gliedert sich auch das streng modular aufgebaute Betriebssystem in einen kleinen Kern von Zentralrechner-Programmen für die Verwaltung der Zentraleinheit und eine Fülle von Programmen peripherer Prozessoren für alle anderen Verwaltungs- und Ein/Ausgabe-Aufgaben.

Als Multiprocessing-System verwaltet NOS ein oder zwei Zentralprozessoren sowie zehn bis zwanzig periphere Prozessoren. Als Network-System ist NOS außerdem in der Lage, mehrere gekoppelte CYBER-Systeme und große Time-

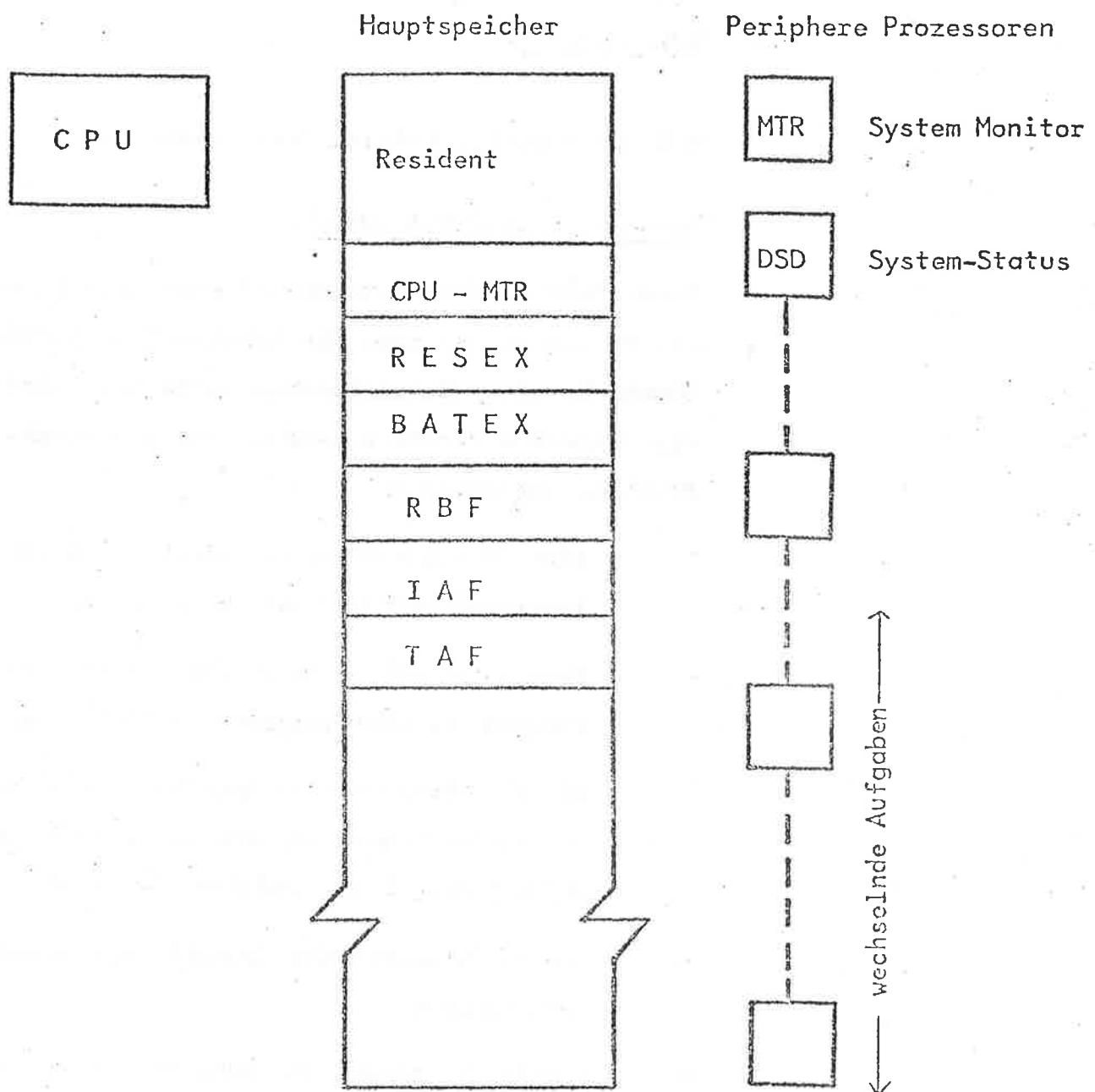
sharing-Netze zu betreiben. Der hardwaremäßige Bereichsschutz erlaubt simultane Bearbeitung von 23 Benutzerprogrammen mit allen Möglichkeiten des Multiprogramming und Multiprocessing.

Dank der modularen Struktur besitzt NOS einige hervorragende Eigenschaften:

- Der residente Teil des Betriebssystems ist unabhängig von den gewünschten Betriebsarten. Er kann beim Systemstart für die gerade benötigte oder vorhandene Hardware-Konfiguration generiert werden.
- Der residente Teil des Betriebssystems kann sehr klein gehalten werden, da sich alle anderen Module nach Bedarf aktivieren und deaktivieren lassen. Dies geschieht im laufenden Betrieb.
- Module für lokale Stapelverarbeitung, externe Betriebsmittelverwaltung, Fernstapelverarbeitung und Dialogbetrieb können einzeln oder in jeder beliebigen Kombination arbeiten. Im Rahmen des Dialogbetriebes steht ein zusätzlicher Modul für Realtime/Transaction-Betrieb zur Verfügung, der aber ebenfalls nur bei Bedarf in den Hauptspeicher geladen wird.

- Die einzelnen Module können im laufenden Betrieb modifiziert werden. Hautspeicher-Modifikationen erfolgen direkt über die Bildschirmkonsole. Änderungen auf dem System-Massenspeicher ermöglicht ein vielseitiges Dienstprogramm.
- Die möglichen Betriebsarten sind untereinander gleichberechtigt. Sie werden durch Parameter so aufeinander abgestimmt, daß die bestmögliche Betriebsmittel-Ausnutzung für die jeweilige Auftrags-Zusammensetzung erreicht wird.

SYSTEM - AUFBAU



RESEX: Betriebsmittelverwaltung
BATEX: Lokaler Stapelbetrieb
RBF: Fernstapelbetrieb
IAF: Dialogbetrieb
TAF: Transaktionsbetrieb

2.2

Auftragsverwaltung

a) Betriebsarten

NOS unterstützt folgende Betriebsarten:

Stapelbetrieb (LOCAL BATCH)

Diese "klassische" Betriebsart unterstützt den Zugang zum System über die lokalen Eingabemedien. Obwohl in allen Betriebsarten verfügbar, sind die folgenden Punkte besonders für den Batch-Benutzer interessant:

- eine flexible Kommandosprache (job control language) und Kommando-Prozeduren
- Programme und Daten werden bei Ein- und Ausgabe zwischengespeichert (Spooling System)
- ein in einem eigenen Speicherbereich arbeitendes Planungssystem für externe Betriebsmittel wie Bänder und Plattenstapel.
- die Möglichkeit des Zugriffs zum Transaktionsundersystem
- ein vielseitiges und außerordentlich störungssicheres Konzept für die langfristige Datenhaltung
- ein breites Spektrum an leistungsfähiger Datenbank-Software

Dialog-Stapelbetrieb (DEFERRED BATCH)

Dieser Control Data - Terminus bedeutet, daß Dialog-Teilnehmer Aufträge an das Batch-Untersystem abgeben können. Diese Aufträge werden parallel zum noch laufenden Dialogverkehr bearbeitet. Ergebnisse solcher "SUBMIT-Jobs" können an beliebiger Stelle, also auch beim sendenden Terminal, ausgegeben werden.

Fernstapelbetrieb (REMOTE BATCH)

REMOTE BATCH ermöglicht die Kommunikation mit der Rechenanlage von irgendeiner Außenstation aus über Endgeräte wie Kartenleser oder Drucker mit derselben Kommandosprache wie im Stapel-Betrieb bei hohen Übertragungsraten.

Dialogbetrieb (TIME-SHARING)

Die Dialog-Eigenschaften von NOS beruhen auf den Erfahrungen mit unserem Time-Sharing Betriebssystem KRONOS. Bis zu 512 simultan aktive Terminals können unterstützt werden, wobei die durchschnittlichen Antwortzeiten im Sekundenbereich liegen.

- Die gleiche Kommandosprache wie im Stapelbetrieb wird verwendet (dazu einige zusätzliche Kommandos)
- Drei sogenannte Compile-to-core Compiler stehen speziell dem Dialog-Benutzer zur Verfügung.

- Ein weiter Bereich von Übertragungsgeschwindigkeiten wird unterstützt.
- Ein Hardware- oder Softwarefehler oder ein kurzfristiger Stromausfall zerstören die Verbindung mit der zentralen Rechenanlage nur vorübergehend, da für eine bestimmte Zeit auch nach der Unterbrechung die vom Benutzer angelegten Dateien erhalten bleiben und nach Wiederherstellung der Verbindung weiterbearbeitet werden können.
- Das Permanent-File-System in NOS wird in besonderem Maße den Bedürfnissen von Dialog-Benutzern gerecht.

Transaktionsbetrieb (TRANSACTION)

Das NOS-Transaktionssystem wurde für Transaktionsanwendungen mittleren Ausmaßes konzipiert, wie es im Wirtschafts- oder Geschäftsleben heute vornehmlich Anwendung findet. Es können über 1000 Terminals angeschlossen und auf mehrere Datenbanken kann gleichzeitig zugegriffen werden.

Für jede dieser fünf Betriebsarten existiert eine Reihe von Parametersätzen. Sie ermöglichen die Einstellung des Systems im Hinblick auf die für jede Betriebsart typischen Eigenarten, Betriebsziele und externen Anforderungen.

Diese Struktur gewährleistet ein außerordentlich hohes Maß an Sicherheit gegen unerwünschte Wechselwirkungen zwischen Aufträgen verschiedener Art untereinander. Gleichzeitig wird dadurch erreicht, daß die Auswirkung von Parameteränderungen transparent bleibt und keine verfälschenden Nebenerscheinungen auftreten. Die Protokollierung aller Systemdaten ermöglicht eine detaillierte Analyse des Systemverhaltens. Daraus kann beispielsweise die Rechen- und E/A-Intensität in Abhängigkeit von der Echtzeit ermittelt werden.

b) Anpassung an die Auftragslast

Die Auftragslast kann nach verschiedenen Gesichtspunkten gegliedert werden, z.B.:

- Aufträge, die bestimmten Betriebsarten zugeordnet sind (Stapelauftrag, Dialogsitzung, etc.).
- Aufträge mit speziellen Betriebsmittelanforderungen (Bandgeräte, bestimmte Druckerformulare, Eingriffe des Operateurs, Wechselplatten, etc.).
- Aufträge, die eine besondere Behandlung aus Sicherheitsgründen erfordern (Änderung des laufenden Systems, Diagnoseprogramme etc.).
- Aufträge mit besonderen Ablauf-Anforderungen bzw. -Restriktionen (Antwortzeiten, externe Prioritäten, Erlaubnis nur zu bestimmten Tageszeiten, etc.).
- Aufträge mit bestimmten Ablaufverhalten bezüglich der Betriebsmittelauslastung (rechenintensiv, E/A-intensiv).

Mit Hilfe von Parametern lassen sich solche Betriebsziele und Betriebsanforderungen steuern. Die Parameter können bei der Systemgenerierung, vom Operateur im laufenden Betrieb oder beim Systemstart, vom Benutzer, vom Systempersonal im

laufenden Betrieb und vom System selbst vorgegeben oder dynamisch verändert werden.

Dies ermöglicht eine schnelle und fein abgestufte Anpassung an ständig wechselnde und nicht kalkulierbare Betriebssituationen.

Hochgeschwindigkeits-Bildschirm-Konsolen geben den Operateuren die Möglichkeit, sich laufend über die aktuelle Betriebssituation zu informieren. Alle verfügbaren Systeminformationen können selektiv in verschiedenen Gruppierungen und Feinheitsgraden abgerufen werden. Eingaben der Operateure werden von einem interpretierenden Programm geprüft, ergänzt und diagnostiziert oder ausgeführt. Da dieses Programm in einem eigens reservierten peripheren Prozessor arbeitet, ist es möglich, die Systeminformationen verzögerungsfrei (im Millisekundenbereich) ständig auf dem neuesten Stand zu halten.

c) Strategien der Ablaufsteuerung

Ein komplexes Verbundsystem muß seine Ablaufstrategien vor allem im Hinblick auf optimale Ausnutzung der Betriebsmittel ausrichten. Nur dann ist es möglich, gewisse Betriebsziele (wie z.B. Antwortzeitverhalten oder Durchlaufzeitverhalten) bei gleichzeitigem Einhalten einer geforderten Gesamtleistung zu erreichen.

Ein wichtiges Kriterium dafür ist die ständige Beschäftigung der zentralen Prozessoren mit Benutzeroberträgen.

Das Network Operating System wird bei standardmäßiger Vorbereitung der Einflußgrößen folgende Strategien verwenden:

- Einhaltung einer Obergrenze für Antwortzeiten (mit höchster Priorität)
- Erzielung eines guten Durchlaufverhaltens, wobei jedoch Kompromisse zugunsten optimaler Betriebsmittelnutzung erlaubt sind.
- Erreichung eines unter obigen Randbedingungen größtmöglichen Gesamtdurchsatzes.

d) Einflußgrößen

Zur Ablaufsteuerung unterscheidet das System fünf Auftragsklassen entsprechend der Herkunft

- System-Aufträge (von der Bedienkonsole)
- Stapelaufträge
- Fernstapelaufträge
- Dialog-Aufträge
- Multi-Terminal-Aufträge

Für jede Auftragsklasse existiert ein Satz von Parametern, von denen die wichtigsten die Prioritäten sind.

Drei Größen bestimmen die Abarbeitung des Auftrags während er sich im Hauptspeicher befindet:

- Zentralrechner-Priorität (CP)
- Zentralrechner-Zeitscheibe (TC)
- Hauptspeicher-Zeitscheibe (TM)

Weitere wichtige Parameter beziehen sich auf die Warteschlangen:

- Eingabe-Warteschlange (IQ)
- Warteschlange der ausgelagerten Aufträge (RQ)
- Ausgabe-Warteschlange (OQ)

Für jede Kombination von Auftragsklasse und Warteschlange gibt es folgende Parameter:

- Priorität bei Eintritt in die Warteschlange (EP)
- Untere Warteschlangen-Priorität (LP)
- Obere Warteschlangen-Priorität (UP)
- Alterungsrate (IN)

Die Behandlung der Prioritäten geschieht folgendermaßen:

- Ein Auftrag tritt mit der Priorität EP in die Eingabe-Warteschlange ein.
- Nach Ablauf von IN Zeiteinheiten wird die Priorität um 1 erhöht (Alterung der Priorität).
- Ein Algorithmus ermittelt den Auftrag mit der höchsten Priorität, dessen Betriebsmittel-Anforderungen mit minimalem Aufwand erfüllt werden können. Dieser Auftrag gelangt mit der Priorität CP in den Hauptspeicher.
- Es wird nach Ablauf einer Mindestzeit (TC und TM) ausgelagert (auf den Erweiterungsspeicher oder Massenspeicher), wenn Prioritäten und Betriebsmittel-Anforderungen anderer Aufträge dies erfordern.
Mit der Priorität LP befindet er sich dann

in der Warteschlange RQ, wo eine erneute Alterung beginnt (maximal bis zum Wert UP). Dialog-Aufträge erhalten bei Eintritt in diese Warteschlange die höhere Priorität EP und werden somit schneller abgefertigt.

- Nach einer gewissen Zeit (Alterung) wird der Auftrag erneut in den Hauptspeicher geladen.
- Die beiden vorangegangenen Schritte wiederholen sich bis zur Beendigung des Auftrags. Nun gelangt er mit der Priorität EP in die Ausgabe-Warteschlange und wird dort wiederum gealtert.

2.3

Dateiverwaltung

a) Arten von Dateien:

Es werden eine Reihe von Dateitypen gemäß Verwendungsart unterschieden:

- Local Files vom Benutzer generierte Arbeitsfiles.

Als Sonderfall von Local Files:

- Primary Terminal Files die Arbeitsdateien des Terminalbenutzers.
- System Files werden ausschließlich vom System benutzt.
- Common Files können von mehreren Programmen gleichzeitig benutzt werden.
- Library Files sind "read-only" common Files mit der Möglichkeit eines gleichzeitigen Zugriffs durch mehrere Benutzer, jedoch mit besonderer Erlaubnis.
- Permanent Files unter den permanenten Dateien gibt es 2 Haupttypen:

Permanente Files mit direktem Zugriff sind in großen Blöcken, abhängig vom Speichermedium, angeordnet und haben eine Schreibsperrre. Das bedeutet, daß jeweils nur ein Benutzer auf den File im Schreibmodus zugreifen kann.

Bei permanenten Files mit indirektem Zugriff erhält der Benutzer eine Kopie des Files. Beliebig viele Kopien eines Files können so parallel verarbeitet werden. Diese Dateiart eignet sich besonders für Steuerkartenprozeduren und Programmdateien im Time-sharing-Betrieb.

b) Logische Dateistruktur:

Für die logische Struktur von Dateien, also Zugriffsverfahren, Satzarten etc. ist der "Record Manager" zuständig. Dieses Paket enthält alle Routinen zur logischen Dateiverarbeitung und zur Abbildung logischer Einheiten auf gerätespezifische Einheiten. Nur die in einem Programm benötigten Routinen werden hinzugeladen. Man erhält also für jedes Programm ein "maßgeschneidertes" Ein/Ausgabesystem. Streng definierte Schnittstellen erlauben auch die Behandlung von Fremd-Dateien oder von Dateien mit unbekannter Struktur.

Der Benutzer kommuniziert mit dem Record-Manager durch eine Tabelle (File Information Table), mit deren Hilfe er die gewünschte logische Struktur seiner Datei aufbaut. Dieses Verfahren bietet große Flexibilität.

Der Benutzer braucht einerseits von der Existenz dieser Tabelle nichts zu wissen, da die Compiler deren Verwaltung übernehmen, hat aber andererseits die Möglichkeit (Systempersonal), die Datei-Behandlung bis ins kleinste Detail kontrollieren und steuern zu können (Anschluß von Own-Code-Prozeduren, spezielle Fehlerbehandlung usw.).

Der Record-Manager unterstützt eine Vielzahl von Zugriffsverfahren:

- Sequentiell "klassischer" sequentieller Zugriff
- Wortadressierbar wahlfreier Zugriff auf Einzelworte oder sequentieller Zugriff
- Satzadressierbar wahlfreier Zugriff auf Datensätze oder sequentieller Zugriff. Diese Methode ist eine Untergruppe des wortadressierten Zugriffs und stellt COBOL-Norm für RANDOM ACCESS dar.
- Direkt wahlfreier Zugriff auf Datensätze durch Adressrechnung (Hashing) oder sequentieller Zugriff.

- Indexsequentiell wahlfreier Zugriff auf Datensätze über Indextabellen oder sequentieller Zugriff. Dateien dieser Art sind "wartungsfrei"; es existieren keine Überlaufbereiche, so daß die Zugriffszeit konstant bleibt und Reorganisation überflüssig ist.
- Actual Key wahlfreier Zugriff auf Datensätze über systeminterne, unveränderliche Schlüssel oder sequentieller Zugriff.
- Standard wahlfreier Zugriff (COBOL) über einstufigen Index.

Die Sätze können dabei zu einem der folgenden Typen gehören:

- Typ F Feste Länge
- Typ D Länge als Zeichenzahl in einem Längenfeld innerhalb des Satzes
- Typ R Abschluß durch ein vom Benutzer definierte Endzeichen
- Typ T Ein Führungssatz fester Länge steht vor einer variablen Zahl von Folgesätzen.
Der Führungssatz enthält die Angabe der Folgesätze als Folgezähler.

- Typ U Der Benutzer definiert die Satz-Struktur.
- Typ W Die Länge des Satzes ist in einem Kontrollwort enthalten, das nicht im Satz liegt, sondern dem Satz vom Record-Manger zugeteilt wird.
- Typ Z Der Satz wird durch zwei "zero character" abgeschlossen.
- Typ S Ein logischer Satz besteht aus einem oder mehreren "Physical Record Units" (geräteabhängige physikalische Sätze) und wird mit einem "kurzen" (d.h. nicht gefüllten) physikalischen Satz abgeschlossen.

Für sequentielle Dateien ist außerdem noch die Blockungsart wählbar.

c) Schutz von Dateien

Dateien sind gegen illegalen Zugriff geschützt. Ein mehrstufiges Passwörtschema, nämlich Benutzernummer, Benutzerpasswort und Dateipasswort, wird zur Sicherung verwendet. Darüber hinaus wird überprüft, ob ein Benutzer für die Zugriffsart zugelassen ist.

Die Zugriffsarten sind:

- Verändern (Modify)
- Schreiben (Write)
- Verlängern (Append)
- Lesen (Read)
- Lesen und Verändern (RM)
- Lesen und Verlängern (RA)
- Ausführung (Execute)
- Löschen früher erzielter Zugriffserlaubnisse (Null)

Simultaner Zugriff durch mehrere Benutzer ist durch eine 'multi read' Kennung beim Lesen der Datei möglich. Nur ein Benutzer darf zu einer gegebenen Zeit die Datei beschreiben. Es können Zugriffsprotokolle geführt (z.B. letztes Zugriffsdatum, Benutzerkennung) und mit Dienstprogrammen abgezogen werden.

Bei Verwendung von 'indirekten Dateien' wird automatisch eine Arbeitskopie erstellt. Die Sicherheitskopie bleibt erhalten. Für den Dialogbenutzer sind Kurzbefehle zum Erstellen von Sicherheitskopien vorhanden. Sie übernehmen automatisch seine Identifikation in den Katalog.

3. Assembler und Compiler

3.1 Assembler

Die Maschinensprache der Cyber 170 Serie wird durch den COMPASS Assembler zu einem flexiblen und vielseitigen Arbeitsmittel für spezielle Anwendungen.

Einige Haupteigenschaften seien hier aufgeführt:

- Befehlseingabe nicht an feste Felder gebunden
- lokale und gemeinsame Blöcke
- Vorbesetzen von Datenfeldern
- Daten-Darstellung in allen gebräuchlichen Formen (Gleitkomma, ganzzahlig, alphanumerisch)
- vielseitige Adress-Arithmetik
- Gleichsetzen und Umdefinieren von Symbolen
- Spezifikation der Zuordnung von Symbolen zu Gruppen
- Wahl des erzeugten Maschinencodes (CPU, PPU, absolut, relativ)
- Selektive Assemblierung von Befehlsfolgen
- Spezifizierung der Standard-Datendarstellung
- Listen-Steuerung durch Pseudo-Befehle
- Zeichenfolgen können symbolisch angesprochen werden (MIKROS)
- Befehlsfolgen können symbolisch angesprochen werden (MAKROS)
- Spezifizierung oder Änderung der Syntax eines CPU- oder PPU-Befehls
- Neudefinition von Befehlen, Ersatz durch Befehlsfolgen
- Wiederholung von Befehlsfolgen
- Wahl der Assemblierungsreihenfolge von Codesequenzen
- Aufruf von Routinen aus Systembibliotheken
- Ausführliche Diagnostiken

Der Assembler-Programmierer kann auf eine Fülle von Systemdiensten (Macros aus Systembibliotheken) zurückgreifen, die als Untermenge auch den Funktionssatz der Kommandosprache enthalten. Außerdem können in separaten Benutzer-Bibliotheken oder in den Systembibliotheken selbstdefinierte Systemdienste generiert und vom Assembler aufgerufen werden. Die Struktur der Cyber-Serie beinhaltet, daß keine sogenannte Unterbrechung (Interrupt), sondern nur die vom Monitor gesteuerte Aktivierung eines peripheren Prozessors oder der Zentraleinheit zur Ausführung von Systemdiensten stattfindet.

3.2 Übersetzer für Programmiersprachen

3.2.1 ALGOL 60

Algol 60 entspricht dem ACM (1963) Standard, dem "Revised Report on the Algorithmic Language, ALGOL 60" und den E/A-Vorschlägen der ISO-Empfehlung Nr. 1538 und ist ebenfalls im Dialog- und Stapelbetrieb einsetzbar.

In einigen Punkten geht es über den Standard hinaus:

- Speicherauszug angegebener Variablen
- Ein/Ausgabebefehle für variable Formate
- Verwendung externer Dateien über "CHANNEL"-Karten

Rund 30 zusätzliche Prozeduren (GET/PUT, GETARRAY/PUTARRAY, PARITY, EOF, SKIPF, BACKSPACE, DUMP, CLOCK u.a.m) tragen erheblich zum Komfort dieses Compilers bei. Block- und Schleifenschachtelungen sind bis zu 64 Ebenen möglich.

Es können drei Optimierungsstufen gewählt werden, um optimalen Maschinencode zu generieren.

Des weiteren sind Befehle zur Generierung von Overlaystrukturen vorhanden.

Auch hier stehen umfangreiche Hilfsmittel zur Fehlerbehandlung auf allen Stufen zur Verfügung, u.a. auch ein ähnliches "Debug" Paket, wie es der Fortran Compiler besitzt.

3.2.2 BASIC

Der BASIC Compiler ist für den interaktiven Betrieb ausgelegt, kann aber auch von Stapelprogrammen verwendet werden. Er stellt eine Erweiterung der am Dartmouth College Computation Center entwickelten Basic-Sprache dar.

BASIC 2 ist in COMPASS geschrieben und umfaßt etwa 5500 Maschinenworte. Im Run-Time-Mode benötigt BASIC lediglich zwischen 1000 (Minimum) und 2000 (Maximum) Maschinenworte. Der restliche Code ist nur für Stapel-Kompilierung erforderlich.

Er wurde bezüglich E/A-Behandlung und Matrix-Verarbeitung um einige Sprachelemente erweitert.

3.2.3 COBOL und SORT/MERGE

Der COBOL Compiler und das auch getrennt anwendbare System SORT/MERGE gewährleisten volle ANSI-COBOL (Level 3) Implementierung und Sort/Merge Fähigkeit.

Die über ANSI-COBOL hinausgehenden Spracherweiterungen werden wahlweise als NON-ANSI gekennzeichnet.

Solche Erweiterungen sind z.B.:

- Zusätzliche "Edit" Möglichkeiten im Report Writer
- Kürzel für viele Schlüsselworte
- Programmteile können aus der Cobol Quell-Bibliothek kopiert werden.

Der Compiler erzeugt optimierten Code für den jeweiligen Maschinentyp.

Das SORT/MERGE Paket kann von COBOL und FORTRAN ange-sprochen werden.

SORT/MERGE kann aber auch mit Kontrollkarten als selb-ständiges System kontrolliert werden und besitzt oben-drein die Möglichkeit, vom Benutzer selbst erzeugte Routinen (OWNCODE) aufzurufen.

Beide Systeme sind interaktiv verwendbar und verfügen über Fehlerbehandlungshilfen auf allen Stufen. COBOL enthält zusätzlich ein "Tracing" mit Protokollierung von Paragraphen-Namen und Zeitangabe.

3.2.4 FORTRAN EXTENDED IV

Mit dem FORTRAN EXTENDED Compiler bieten wir den zur Zeit ausgedehntesten und komfortabelsten Fortran Übersetzer an. Er entspricht mindestens dem USASI Standard 2.1.3.9 und ist im Dialog- und Stapelbetrieb einsetzbar. Außerdem verfügt er über ein IMPLICIT Statement zur Definition des Typs einer Variablen durch den Anfangsbuchstaben ihres Namens, ein RETURNS Statement zur Modifizierung der Rücksprungadressen während eines Unterprogrammdurchlaufes.

Die Standard-Bibliothek ist um über 20 Routinen erweitert:

Drei mögliche Optimierungsstufen erlauben die Generierung optimalen Maschinencodes, z.B. durch Eliminierung redundanter Operationen, Ermittlung möglicher Parallelarbeit nach der Methode des kritischen Weges, Do-Loop-Optimierung unter dem Gesichtspunkt hardwaremäßig schnellster Ausführung u.a.m.

Ein spezielles "Debug" Paket ergänzt die standardmäßig auf Quell- und Maschinencode-Ebene vorhandene ausführliche Fehlerbehandlung. Damit ist die Verfolgung von Sprüngen, Aufrufen, Feldindices usw. sowohl durch Steuerkarten als auch auf Programmebene möglich.

Anhang

Benutzerkomfort unter NOS

NOS und seine Compiler bieten dem Benutzer ein breites Spektrum erprobter Hilfen für alle Ebenen der Programmierung an. Da die Verschiedenartigkeit der Compiler (d.h. ihre sprachenbedingten Eigenarten) eine rein tabellarische Darstellung der Hilfen nicht erlaubt, seien sie hier kurz der Reihe nach im Rahmen der von NOS selbst zur Verfügung gestellten Hilfsmittel beschrieben:

1. NOS Kommandosprache
2. NOS Dienstprogramme
3. NOS Loader (Laden + Binden)
4. NOS Record Manager (Ein-/Ausgabe)
5. NOS Text Editor
6. FORTRAN
7. Algol 60
8. Cobol
9. PL/I
10. Austauschbarkeit von Unterprogrammen
11. Dialog mit Programmen

Alle Eigenschaften sind im Batch- und Dialogbetrieb vorhanden wenn sie nicht ausdrücklich für Dialogbetrieb gekennzeichnet sind.

Auftretende Fehler werden je nach Art und Schwere im Auftragsprotokoll (Job-Dayfile), in der Übersetzerliste, in der Ausführungsliste oder in einer speziellen Fehlerdatei protokolliert.

Die Protokolle können vom Benutzer weitgehend gesteuert werden.

1. NOS Kommandosprache

- a. Die Befehle der Kommandosprache werden nicht in den Datenstrom des Auftrags eingeschoben, sondern in zusammenhängendem Block als erster Datensatz eines Auftrags geführt. Sie benötigen daher auch keine Fluchtsymbole.
- b. Die komfortable Kommandosprache erlaubt die Haltung vordefinierter Kommandoprozeduren, die bei Aufruf mit aktuellen Parametern versorgt werden. Damit ist es z.B. möglich, den Zustand oder das Medium einer Datei zu ermitteln und abhängig davon eine bestimmte Steuersprachen-Prozedur anzuspringen oder aufzurufen. Zusätzlich steht über den SUBMIT-Befehl eine Fülle von Reformatierungsmöglichkeiten zur Verfügung. Diese Möglichkeiten können insbesondere bei der Programmentwicklung in der Testphase hervorragende Dienste leisten.
- c. Die Kommandosprache enthält Befehle für das Setzen und Löschen programmierten Schalter, für die Ablaufsteuerung nach dem Auftreten von Fehlern im Auftragsablauf und für die Aktivierung bzw. Deaktivierung der Fehlerbehandlung für bestimmte Fehlerarten. Außerdem können Speicherabzüge in verschiedenen Formaten angefordert werden.

2. NOS Dienstprogramme

a. Quellbibliotheken

Die Dienstprogramme UPDATE, MODIFY und DEVOUR bieten folgende Möglichkeiten an:

- Generierung von Quellbibliotheken in komprimierter Form (sequentiell oder random).
- Modifikation dieser Bibliotheken durch zeilenorientiertes oder zeichenkettenorientiertes Löschen, Ersetzen, Einfügen oder Anhängen von Zeichenketten, Zeilen, Gruppen von Zeilen, Programmen oder ganzen Programm-/Änderungspaketen.
- Chronologische Buchführung der Modifikationen.
Jede Modifikation kann unter ihrem Namen gefunden, aktiviert oder deaktiviert werden (permanent oder temporär).
- Generierung von Dateien für die verschiedensten Aufgaben der Weiterverarbeitung (z.B. als Eingabe für Übersetzer oder als neugeordnete Bibliothek oder Eingabewarteschlange für NOS etc.)
- Auswahl verschiedener Modifikations-Stufen (komplette Bibliothek oder nur spezifizierte Teile).
- Flexible Ablauf- und Ausgabesteuerung
- Gemeinsame Datensätze (Common Decks) können durch Aufruf an bestimmten Stellen eingefügt werden.
- Nicht nur Quellprogramme sondern Sätze kompletter Aufträge (einschließlich der Steuerbefehle) können gewartet werden

- Automatische Änderung bestimmter Jobkartenparameter
- Umstellung von Programm paketen (z.B. Benchmarks) durch eine Reihe automatischer Funktionen.

b. Modulbibliotheken (Objekt Module)

Generierung und Wartung (LIBGEN/LIBEDIT) von Modulbibliotheken die vom NOS-Loader wie Systembibliotheken behandelt werden, jedoch der Kontrolle des Benutzers unterliegen.

Zur Wartung von einfachen Moduldateien kann das GTR Kommando verwendet werden.

Diese Programme dienen im allgemeinen dazu ,Module einer Bibliothek/Datei durch neue Module einer anderen Datei zu ersetzen, alte Module zu löschen oder neue Module einzufügen. Eine ausführliche Inhaltsangabe über Name, Art und Größe der Module solcher Dateien liefert das Kommando CATALOG. Auch ein Kommentarfeld der Objekt-Module (z.B. Kurzbeschreibung) und das Erstellungsdatum werden ausgegeben.

3. NOS Loader (Laden + Binden)

Zusätzlich zu den flexiblen bibliotheks- oder dateibezogenen Steuerbefehlen zum Laden und Binden von Objekt-Modulen, Vorbesetzen von Speicherbereichen und Spezifizieren bestimmter Systembedingungen bietet der Loader durch seine als Steuerbefehle verfügbaren TRACK- und FRAME-Direktiven folgende Testhilfen:

- Speicherabzüge bestimmter Bereiche bei Durchlaufen bestimmter Programme oder Adressen.

Beispiel: AT Name / Adr.FROM Name 1/Adr. 1 FOR 1000

START 10 EVERY 2 UNTIL 26 REG

Diese Anweisung bewirkt, daß beim Durchlaufen der Adresse Adr. im Programm Name ein Dump (einschließlich Registerabzug, REG) von Adresse Adr.1 in Programm Name 1 ausgegeben wird. Dies geschieht zwischen dem zehnten und sechsundzwanzigsten bei jedem zweiten Durchlauf. Der abgezogene Bereich umfaßt 1000 Worte.

- Speicherabzüge bestimmter Bereiche für den Fall, daß sich der Inhalt einer bestimmten Zelle oder eines Registers ändert

Beispiel: FROM Name 1/Adr.1 TO Name2/Adr.2 WHEN Name3/Adr.3
oder Register START 1 EVERY 10 UNTIL 100

Diese Anweisung liefert bei jeder zehnten Änderung (zwischen 1 und 100) der Zelle Name3/Adr.3 oder eines angegebenen Registers einen Speicherabzug des Bereiches Name1/Adr.1 bis Name2/Adr.2.

Die TRACK und FRAME Direktiven können als Laufzeit-Testhilfe für alle ladefähigen Programme eingesetzt werden. Sie sind völlig unabhängig von den sonstigen Testhilfen des Systems und der Compiler.

Eine Neu-Übersetzung von Programmen ist nicht notwendig.

Selbstverständlich verfügt der Loader auch über ein System von Fehlermeldungen, um den Benutzer rechtzeitig auf sinnlose, gefährliche oder verbotene Lade-/Bindeoperationen aufmerksam zu machen. Eine weitere Hilfe bietet sich in den verschiedenen ausführlich abrufbaren Lade-/Binde-Kontroll-Listen.

4. NOS Record Manager (Ein-/Ausgabe)

- Einheitliches E/A-Paket für Assembler und Compiler
- Dateien, die unter verschiedenen Compilern erstellt wurden, können von anderen Sprachen her angesprochen werden.
- Einheitliche Fehlermeldungen für E/A-Operationen verschiedener Sprachen (eine für die jeweilige Sprache spezifische Angabe kann vorangestellt werden).
- Fehlergruppen:
 - Tabellenführung
 - Datei-Organisation
 - Block-/Satztyp
 - Eröffnen/Schließen
 - Bearbeitungsrichtung
 - Positionierung
 - Operationsfolge
 - Schlüssel
 - Daten
 - Geräte
 - Kennsätze
- Fehlerklassen:
 - trivial
 - fatal für die Datei
 - fatal für den Auftrag
 - trivial beim Schreiben/fatal beim Lesen

5. NOS Text Editor (nur Dialogbetrieb)

a. Text-Modus

Eingabe von Textzeilen ohne Editierungsfunktionen, schnelle und effektive Methode (die den Zentralrechner nicht belastet, da sie in peripheren Prozessoren abgewickelt wird).

b. Edit-Modus

Eingabe, Korrektur, Löschen, Einfügen, Ersetzen und Mischen von Zeichenketten, Zeilen oder Gruppen von Zeilen anhand von Such-Zeichenketten, Zeilenzeigern oder sequentiell.

- Ausgabe (Display) von Teilen einer Datei oder ganzer Dateien
- Selektives Löschen von Information
- Selektives Hinzufügen oder Einschieben von Information
- Zwischenspeichern und späteres Einfügen von Informationen an anderer Stelle
- Mischen von Information bzw. Dateien
- Zählung des Vorkommens spezifizierter Zeichenketten
- Ausgabe und Verwaltung der lfd. Zeilennummer

- Formatsteuerung der Editierungs-Datei
- Angabe von einfachen und geklammerten Suchschlüsseln (z.B. für Zeichenketten, die sich über mehrere Zeilen erstrecken).
- Linemode/Stringmode; die Editierungsbefehle können sich wahlweise auf eine ganze Zeile oder nur auf eine spezifizierte Zeichenkette beziehen.
Dies ist unabhängig von evtl. spezifizierten Such-Zeichenketten.
- Tabulierung
- Ausgabe von Fehlermeldungen und Handhabungsinformationen (z.B. HELP-Kommando)
- Schnelle (oft implizite) Datei-Handhabung
- Weiterverarbeitung der Editierungsdateien nach Abschluß der Arbeiten durch Programme und Compiler.

6. Fortran

a. Testhilfen zur Übersetzungszeit

- Ausgabe einer mehrstufig abrufbaren 'Cross Reference Map'. Sie enthält Angaben über Art und Lage von Variablen, Feldern und Blöcken (sowohl quellbezogen als auch auf die AnfangsAdresse des Programmblocks bezogen). Weiterhin können externe Referenzen, DO-Schleifen, Datei-Referenzen und Blockstrukturen untersucht werden. Variable, die nur einmalig vorkommen werden gekennzeichnet. Es werden auch alle Quellzeilen aufgeführt, von denen aus ein Name angeprochen wird. Zusätzlich werden Optimierungskriterien ausgewiesen.
- Fehlermeldungen:

ANSI Der Befehl entspricht nicht dem ANSI-Standard

FATAL Ausführung wird unterbunden

INFORMATIVE Hinweis auf mögliche Fehlerquelle

Für die Dialog-Version des Compilers:

ANSI und FATAL wie oben

WARNING Ein vom Compiler automatisch korrigierter Fehler

NOTE

Hinweis auf unbedeutenden Fehler der keine Folgen erwarten läßt.

Die Fehlerklassen können über Aufrufparameter selektiert werden.

- Compiler-Aufrufparameter für:
 - Syntaxprüfung ohne Codegenerierung
 - Timesharing-Modus
 - Debug-Modus (Übersetzung von Laufzeit-Testhilfen)
 - Error-Traceback (call by name)
 - Error-Recovery (Protokollierung)
 - Error-Level
 - Exit-Behandlung

b. Testhilfen zur Laufzeit

Zusätzlich zu den klassifizierten Laufzeit-Fehlermeldungen des Compilers existiert ein DEBUG-Paket.

Diese Testhilfen können in Form von Befehlen in einer separaten Datei, vor den einzelnen Programm-Einheiten oder zwischen die Quellbefehle eingestreut mitgeführt werden.

Sie werden als Kommentar gedeutet wenn im Fortran-Aufruf nicht der D-Parameter gesetzt ist. Die entsprechenden Meldungen werden wahlweise inmitten der Ausführungsliste oder auf einer separaten Datei ausgegeben.

Die Testhilfen umfassen:

- Feldgrenzen-Überwachung

- Aufruf-Verfolgung (auch Funktionen)
- Änderung von Werten wird anhand arithmetischer/ logischer Bedingungen verfolgt
- Sprungverfolgung für ASSIGNED GOTO
- Tracing aller Verzweigungen
- Aktivierung / Deaktivierung von Bereichen

Weiter steht eine Reihe von Funktionen zur Verfügung, die durch Aufruf aktiviert werden können:

- ERRSET/RECOVR eigene Fehlerbehandlung durch den Benutzer
- SYSTEM/SYSTEMC Steuerung und Änderung der Fehlerbehandlung und der Fehlermeldungen
- STRACE Traceback Information
- LEGVAR Legalitätsprüfung von Variablen
- DUMP/PDUMP Speicherabzüge in diversen Formaten
- Interaktive Laufzeit-Tests können durch das (von Control Data nicht standardmäßig unterstützte) Paket FIDA (Fortran Interactive Debug Aid) durchgeführt werden. Dies geschieht durch "Breakpoints" an denen dem Benutzer die Kontrolle über sein Programm gegeben wird, um beispielsweise Werte von Variablen zu prüfen.

- c. Die Fehlerrückverfolgung nach dem Abbruch des Programms übernimmt ein Zusatzpaket (Post Mortem Dump).

7. Algol 60

a. Testhilfen zur Übersetzungszeit

- Fehlermeldungen werden am Ende der Quellprogrammliste gesammelt ausgegeben:

lexikographische/syntaktische Fehler
pre-semantische/semantische Fehler
Prozedurprüfung (Optimierung)
GOTO und LABEL
Datei-Prüfung (symbolisch)

Alle Meldungen beziehen sich auf die entsprechende Zeilennummer des Programms.

Sie sind noch nach der Schwere des Fehlers untergliedert in:

ALARM	keine Code-Generierung
ADVISORY	mögliche Fehlerquellen
RECOVERY	vom Compiler selbst-korrigierte Fehler werden angezeigt.

- Der Algol Interactive Syntax Checker ALGEDIT erlaubt die Eingabe von Quellprogrammen beliebiger Sprachen wie ein Text-Editor, kann aber gleichzeitig für die interaktive (zeilen-/gruppenweise) Syntaxprüfung von Algol-Programmen aktiviert und deaktiviert werden.

Einfügen, Löschen, Ersetzen von Information nach Suchkriterien (Zeichenketten oder Zeilennummern) sowie Ausführung von Programmen ist möglich.

- Einige Parameter des Algol-Aufrufes bieten weitere Testhilfen an:

C = 0	Kommentare werden nicht interpretiert
C = 1	Debug - Direktiven innerhalb der Kommentare werden übersetzt
C = 2	Overlay-Handhabung
C = 3	CHECKON/CHECKOFF wird für die Feldgrenzen-Überwachung aktiviert
E	Exit-Behandlung
D	Dumpfile-Zuweisung
F	Syntax-Fehler Behandlung
Q	ALGOL Interactive Debug Aids (AIDA)

b. Testhilfen zur Laufzeit

- Fehlermeldungen des Compilers und Aufrufe der Routinen ERROR und CHANERROR; der Benutzer kann bestimmte Fehlerklassen selektiv ein-/ausschalten.
- Aufrufe der DUMP-Routine geben in lesbärer Form alle lokalen und formalen Variablen und Felder des gerade aktiven Blockes aus (z.B. *T* oder *F* für boole'sche Werte).

- Eine 'Cross Reference List' gibt alle Zeilennummern und Referenzen für Standard Prozeduren, Blöcke, Prozeduren mit Angabe der Verschachtelungstiefe aus. Alle Namen eines Blockes werden mit folgender Information versehen gelistet:

Name

Typ (real, integer, boolean)

Art (array, procedure, switch, label)

Zeile der Definition sowie alle Zeilen von
denen aus 'Name' angesprochen wird

Dimension (falls array)

Anzahl der formalen Parameter (falls procedure)

Außerdem wird angegeben, ob die Variable

als aktueller Parameter oder in einer Wert-
zuweisung verwendet wird (in der entsprechenden
Zeile)

- Ausführlichste Information liefert eine STACK-Liste für fortgeschrittene Anwender
- Während der Ausführung können u.a. folgende Parameter der Ausführungsbefehls wirksam werden:

S Ausgabe der STACK-Statistik

D Dump und Traceback-Information in 4 Stufen

P Feld-Vorbesetzung

T Begrenzung der Debug- und Dump-Ausgabe

- Eine Reihe von Debug-Direktiven kann in das Algol-Programm kompiliert werden:

TRACE

Überwachung von Verzweigungen,
Prozeduren, Ablauf-Kontrolle,
Label-Überwachung

SNAP

Änderung spezifizierter
Variablen oder Felder wird
angezeigt

SNAPOFF

Ausschalten von SNAP

- Das ALGOL Interactive Debugging Aids Paket (AIDA) verwendet 'Breakpoints' um dem Benutzer die Kontrolle über sein Programm zu geben (auch im Batchbetrieb). Der Benutzer kann dann Variable anzeigen lassen und modifizieren (quellbezogen unter Namensangabe). Es können auch Felder komplett oder teilweise durch Weglassen der Indizes oder Angabe von * statt eines von mehreren Indizes ausgegeben werden.

Die Angabe der 'Breakpoints' geschieht folgendermaßen:

BREAK, $l_1, l_2, \dots, l_n i, f, s (v_1, \dots, v_5)$

$l_1, l_2 \dots l_n$ sind Zeilennummern an denen das Programm angehalten werden soll.

Es können durch $l_1/l_2 \dots$ auch Bereiche definiert werden.

i, f, s Dies sind Kontrollvariable für Anfangswert
 und Endwert sowie Schnittweite der Anzahl
 der Durchläufe für die Aktivierung des
 "Breakpoint".

V₁--V₅ Variablen-Namen die ausgegeben oder modifiziert
 werden sollen.

Durch RELEASE können "Breakpoints" selektiv wieder ge-
löscht werden.

Eine ganze Reihe weiterer Befehle ergänzen zusammen mit
der Möglichkeit, die Geschichte der "Breakpoint"-Läufe
in einer Datei aufzuzeichnen, diese komfortable Testhilfe.

8. Cobol

a. Testhilfen zur Übersetzungszeit

- Zusätzlich zur Quell-Liste und einem umfangreichen Satz von Fehlermeldungen erlaubt Cobol die Ausgabe einer Liste aller Data-Items und Procedure-Namen mit Adressen- und Zeilenangabe, eine Liste aus der hervorgeht, von welchen Zeilen aus ein Name ange- sprochen wurde und eine Diagnostic-Liste in der die aufgetretenen Fehler mit Zeilen- und Spaltenangabe sowie Angabe der Schwere des Fehlers geführt wird.
- Fehlermeldungen

TRIVIAL fragwürdiger aber syntaktisch richtiger Einsatz eines Befehls (Hier können auch NON-ANSI-Fehler geführt werden).

WARNING Vom Compiler durchgeföhrte Korrektur eines Syntaxfehlers wird gemeldet

FATAL Der Befehl kann nicht übersetzt werden

CATASTROPHIC Die Übersetzung des Programms muß abgebrochen werden.

Die Meldungen können durch Compiler Aufrufparameter selektiert werden.

- Compiler Aufrufparameter:

NON-ANSI Befehle als trivial / fatal
DEBUG-Mode (Paragraph Tracing / Debug Lines/
Feldgrenzenüberwachung)
Fehler-Datei
Fehler-Stufen (Auswahl der Meldungen)
Fehler-Behandlung (Abbruch-Kriterien)
Syntax-Prüfung

b) Testhilfen zur Laufzeit

- Paragraph Tracing

Bei Antreffen einer Trace Ein-/Ausschaltanweisung im Programm wird eine Statusmeldung ausgegeben. Alle dazwischen ausgeführten Paragraphen werden mit Namen und aufgelaufener Rechenzeit gemeldet. (Nur nach Selektierung des Aufrufparameters TR).

- Debugging Lines

Im Programm enthaltene Überwachungsanweisungen (z.B. USE FOR DEBUGGING ON ALL REFERENCES OF ID-1 u.s.w) dienen zur Kontrolle von Variablen, Feldern, Aufrufen und Verzweigungen. Sie werden als Kommentar behandelt, wenn der Aufrufparameter DL nicht angegeben wurde.

- Debugging Sections

Übersetzung der Hilfsanweisungen nach WITH DEBUGGING MODE-Befehl, sonst als Kommentar gewertet.

Die Überwachung von Namen, Feldern, Sprüngen, Aufrufen und Data-Referenzen geschieht durch Steuerung von innen oder außen über einen programmierbaren Schalter (SWITCH 6).

Die Meldungen sind quellbezogen.

In einem DEBUG ITEM REGISTER werden die aktuellen Bedingungen abgelegt, so daß sie auch vom Programm dynamisch verwertet werden können (Name, Inhalt, Zeilennummer u.a.m).

9. PL/I

a. Testhilfen zur Übersetzungszeit

- Ausgabe einer 'Cross Reference Map'. Sie enthält Angaben über Art und Lage von Variablen, Marken, Feldern und Blöcken. Außerdem können Datei-Referenzen untersucht werden.
- Fehlermeldungen werden am Ende der Quellprogrammliste ausgegeben, die ein genaues Diagnostizieren der Fehlerursache ermöglichen.

b. Testhilfen zur Laufzeit

- ON-Einheit und Bedingungspräfixe ermöglichen es dem Benutzer, sein Programm zu überwachen, bzw. auf verschiedene erwartete oder unerwartete Programmzustände zu reagieren.
- durch die Bindemöglichkeit zu FORTRAN-Unterprogrammen, können Dumproutinen aufgerufen werden, die Speicherabzüge in diversen Formaten liefern. (Weitere FehleraufrufROUTINEN siehe FORTRAN-Beschreibung).

10. Austauschbarkeit von Unterprogrammen

Alle Compiler generieren einheitliche Objekt-Module, die vom Loader gebunden und geladen werden können.

Die Behandlung der sprachenspezifischen Schnittstellenprobleme (Parameterübergabe, Common Blöcke, Switch etc.) ist von Fall zu Fall unterschiedlich.

Fortran Es können Assembler-Routinen eingebunden werden.

Die meisten Compiler verfügen über Aufruf-Parameter für eine Ausgabe des übersetzten Programms in Assembler-Form (Fortran, Algol, Cobol).

Algol Es existiert ein Satz von Interface-Macros. Sie berücksichtigen Algol-Struktur. Damit können Unterprogramme der anderen Compiler eingebunden werden.

Cobol In Cobol erlaubt die ENTER-Anweisung (mit USING für die Parameterübergabe und COMMON Storage für gemeinsame Bereiche) das Einbinden von Unterprogrammen der anderen Compiler.

PL/I Über Fortran-Interface können die oben beschriebenen Einbindungen ebenfalls realisiert werden.

11. Dialog mit Programmen

Hier sei ganz allgemein noch erwähnt, daß der Dialog mit Programmen sowohl intern (CONNEC, DISCON in Fortran/ACCEPT, DISPLAY in Cobol, INPUT/OUTPUT in Algol (Algedit) und PL/I oder CONIN/CONOUT (AIDA)) als auch extern über Kommandos gesteuert werden kann. Der Benutzer hat die Möglichkeit, ein ausführendes Programm zu unterbrechen und fortzusetzen.

