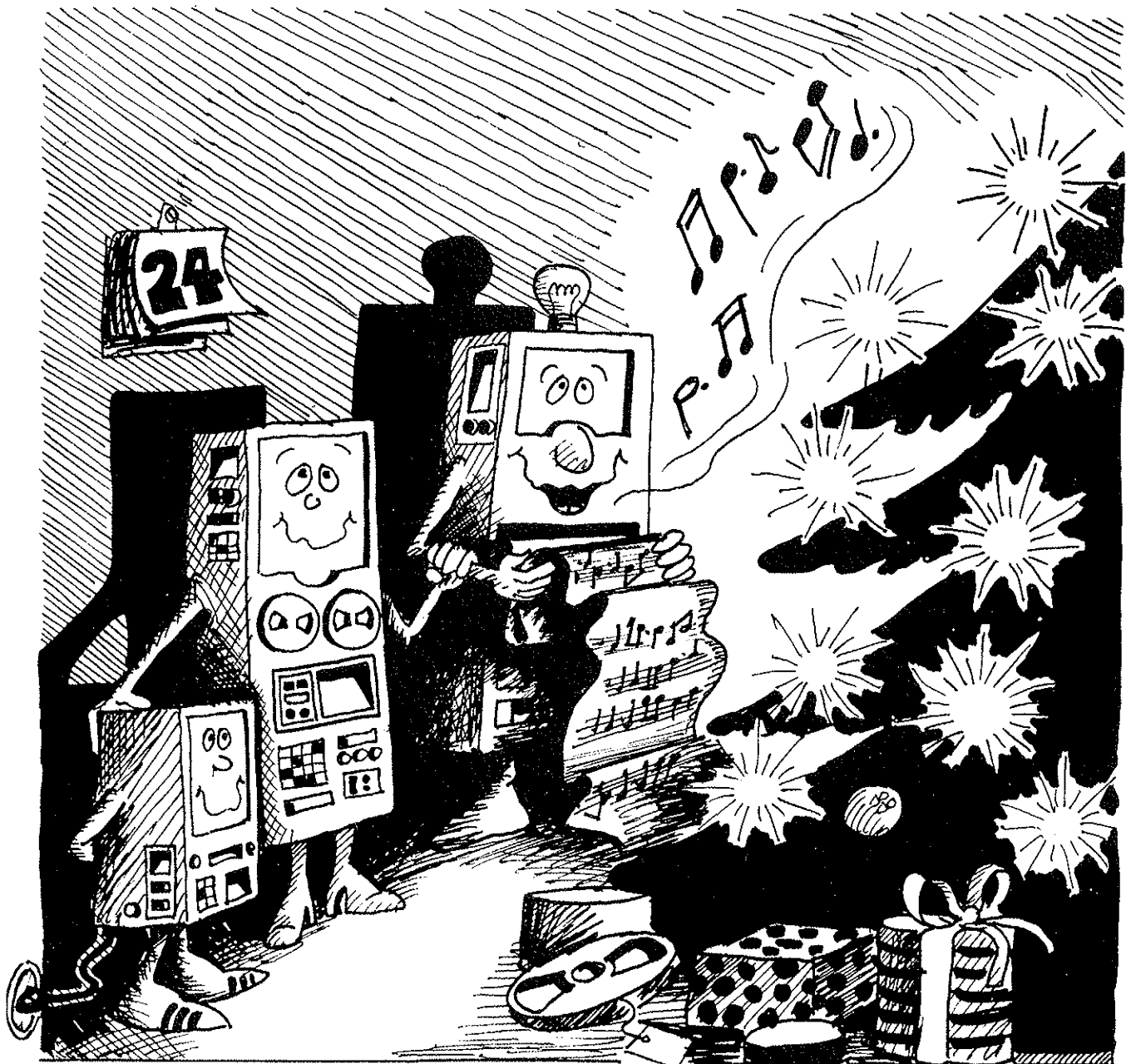


HEISSER DRAHT

RECHENZENTREN
UNI-TU WIEN
CYBER 73-74

NUMMER 15

DEZEMBER 1975



L.M.O.T.

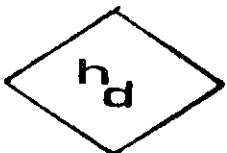
IM BRENNPUNKT

1 NEUER SIMULA - COMPILER (SEITE 15)

2 MNF - COMPILER (SEITE 19)

3 NEUE PROGRAMMSAMMLUNGEN (SEITE 34)

4 TAPEDUMP - UTILITY (SEITE 53)



*Eigentümer, Herausgeber und Verleger:
EDV-Zentren UNI - TU Wien, Cyber 73-74.
Für UNI-Wien: Universitätsstraße 7, 1010 Wien
Für TU-Wien: Gußhausstraße 27-29, 1040 Wien
Für den Inhalt verantwortlich: Anton Roza.
Druck: Österr. Hochschülerschaft der TU-Wien.*

ACHTUNG!! WEIHNACHTSBETRIEB 1975 !

Montag (1975-12-22) und NORMALER BETRIEB
Dienstag (1975-12-23)

Ab Mittwoch (1975-12-24) bis
Dienstag (1976-01-06) KEIN BETRIEB

Ab Mittwoch (1976-01-07) wieder NORMALER BETRIEB

CYBER 73: AB 08.00 UHR

CYBER 74: AB 08.30 UHR

Frohe Weihnachten

allen Benutzern !



KURSE

TERMIN VON BIS	KURSBE- ZEICHNUNG	KURSTITEL	VORTRAGENDER
1975-12-15 1975-12-17	RD3	Benutzung des Systems CYBER 70	Dipl.Ing. H. MASTAL
1975-12-18 1975-12-19	RD5	File-Handling	Dipl.Ing. H. MASTAL

1976-02-16 1976-02-27	RD1	Einführung in das Programmieren	Dipl.Ing. R. BRAUN
1976-03-17 1976-04-06	RD2	FORTRAN	Ing. G. SCHMITT
1976-06-28 1976-06-30	RD3	Benutzung des Systems CYBER 70	Dipl.Ing. H. MASTAL
1976-07-01 1976-07-02	RD5	File-Handling	Dipl.Ing. H. MASTAL

RD4 nach Bedarf (Terminal-Benutzung)



DEZEMBER 1975

KURSE

TERMIN VON BIS	KURSTITEL	VORTRAGENDER
1976-01-12 1976-01-14	SELCOM Version 2	Dr. P. RASTL
1976-04 (erste Woche)	C O M P A S S (Assemblersprache der CDC)	Wolfgang ALTFAHRT
1976-04-26 1976-05-08	F O R T R A N (Einführung in das Programmieren)	Karl PECHTER
1976-05-10 1976-05-14	I N T E R C O M (Einführung in die Terminalbenutzung)	Wolfgang ALTFAHRT
1976-05-17 1976-05-21	STATISTISCHE PROGRAMMPAKETE	Rudolf WYTEK



RD1: EINFÜHRUNG IN DAS PROGRAMMIEREN (PASCAL)

Inhalt:

Dieser Kurs ist nicht zu verwechseln mit der gleichnamigen Vorlesung! Er ist speziell für jene gedacht, die in die Programmierung elektronischer Rechenanlagen einsteigen wollen oder müssen, dafür aber relativ wenig Lernzeit haben oder aufwenden wollen.

Nach einer allgemeinen Einführung konzentriert sich der Kurs auf die Problemanalyse, d.h. auf die Formulierung von realen Problemen in Algorithmen, die dann relativ einfach in einer problemorientierten Sprache (im Kurs verwendete Sprache PASCAL) formuliert werden können.

Zur praktischen Anwendung und Vertiefung des vorgetragenen und erarbeiteten Stoffes haben die Teilnehmer nach jedem Vortrag Gelegenheit, vorgegebene Aufgaben mit Hilfe der Rechenanlage zu lösen, wobei fachkundige Beratung jederzeit in Anspruch genommen werden kann.

Dauer: 10 Tage zu je 3 Std. Vortrag und 3 Std. Übungen

Kursziel: Fähigkeit, reale Probleme mit Hilfe einer Rechenanlage zu lösen bzw. sich kurzfristig Kenntnisse anderer Programmiersprachen im Selbststudium anzueignen.

Kursleiter: Dipl.Ing.Rudolf BRAUN

RD2: FORTRAN

Dieser Kurs ist prinzipiell für Benutzer gedacht, die bereits Kenntnisse in einer Programmiersprache haben, doch ist in den ersten drei Kurstagen eine Wiederholung der wichtigsten Grundbegriffe (Flußdiagramm, Steuerkarten, arithmetische Ausdrücke, Files) vorgesehen. Der Hauptteil des Kurses behandelt Standard-FORTRAN, wie es von den Compilern MNF und FTN verarbeitet werden kann. Die letzten vier Tage befassen sich mit Spezialgebieten wie Overlays,

Segmentierung, Normung, Terminalbenutzung, Random access files, Sortiersystemen u.ä. Während des Kurses haben die Teilnehmer Gelegenheit, praktische Erfahrungen an der Rechenanlage zu sammeln.

Dauer: 15 Tage zu je 2 Std.

Kursziel: Beherrschung der Programmiersprache FORTRAN und deren praktische Anwendung.

Kursleiter: Ing. Gerhard SCHMITT

RD3: BENUTZUNG DES SYSTEMS CYBER 70

Dieser Kurs erfordert praktische Programmiererfahrung in mindestens einer Programmiersprache.

Inhalt: Es werden die wesentlichen Möglichkeiten des Großrechners gezeigt.

Kursziel: Die Fähigkeit, durch ausreichende Kenntnisse und mittels Handbücher auch komplexere Funktionen des Systems, wie zum Beispiel Aufbau von eigenen Programmbibliotheken, Bearbeitung von Dateien oder rechnerunterstützte Änderungen in großen Programmsystemen zu benutzen.

Dauer: 3 Vormittage

Kursleiter: Dipl.Ing. Helmut MASTAL

RD4: TERMINALBENUTZUNG

Inhalt:

Dieser Kurs ist für jene gedacht, die via interaktives Terminal (Bildschirm, schreibend) die Rechenanlage benutzen wollen.

Hauptaugenmerk wird auf Erstellung, Modifikation, interaktive Ausführung von Jobs bzw. ihre Übermittlung von und zur Rechenanlage



gelegt. Der Kurs ist zweigeteilt: am Vormittag Vortrag, am Nachmittag praktische Übungen an etlichen Terminals verschiedener Hersteller.

Dauer: 3 Tage zu je 3 Std. Vortrag und 1 Std. Übungsmöglichkeit. Außerdem stehen zusätzlich noch zwei ganze Übungstage zur Verfügung.

Kursziel: Fähigkeit, via Terminal sinnvoll mit einer Rechenanlage zu arbeiten.

Kursleiter: Dipl.Ing. Rudolf BRAUN

RD5: FILE-HANDLING

Dieser Kurs setzt Vorkenntnisse - Besuch des Kurses RD3 - Benutzung des Systems CYBER 70 - voraus.

Inhalt: Dateien am System CYBER 70 - Strukturen und Zugriff

Dauer: 2 Vormittage, 1 Nachmittag (3 Std. täglich)

Kursziel: Fähigkeit, für individuelle Probleme die geeignete Speicherorganisation anzuwenden.

Kursleiter: Dipl.Ing. Helmut MASTAL



Das Interfakultäre Rechenzentrum der Universität Wien
veranstaltet im Sommersemester 1976 folgende Kurse:

1. FORTRAN (EINFÜHRUNG IN DAS PROGRAMMIEREN)

KURSGEBÜHR: S 200,-

Zeit:*) 1976 04 26 - 05 08

Inhalt:

Einführung in die Programmierung in der Programmiersprache FORTRAN. Vorkenntnisse sind nicht erforderlich. Dieser Kurs soll Anfänger mit den nötigen Kenntnissen ausstatten, für einfache Probleme FORTRAN-Programme zu erstellen und von dieser Grundlage aus selbständig weitergehende Programmiererfahrungen zu erwerben. Bei diesem Kurs werden regelmäßig Übungen an den Einrichtungen des Rechenzentrums durchgeführt.

KURSLEITUNG: Karl PECHTER

2. INTERCOM (EINFÜHRUNG IN DIE TERMINALBENÜTZUNG)

KURSGEBÜHR: keine

Zeit:*) 1976 05 10 - 05 14

Inhalt:

Terminalbetrieb unter Benützung des Betriebssystems INTERCOM. Vorausgesetzt werden Kenntnisse über SCOPE in dem Ausmaß, wie sie der Kurs SCOPE I, bzw. die entsprechende Broschüre bieten. Dieser Kurs bietet die Kenntnisse, die als Voraussetzung für eine Genehmigung zur Benutzung der Terminals am Rechenzentrum verlangt werden.

KURSLEITUNG: Wolfgang ALTFAHRT

3. STATISTISCHE PROGRAMMPAKETE

KURSGEBÜHR: keine

Zeit:*) 1976 05 17 - 05 21

Inhalt:

Anwendung der statistischen Programmpakete des Interfakul-

tären Rechenzentrums:

SSPS, BMD, SSP, CLUSTAN, SELCOM.

KURSLEITUNG: Rudolf WYTEK

4. COMPASS (ASSEMBLERSPRACHE DER CDC)

KURSGEBÜHR: keine

Zeit: *) 1976 04 (erste Woche)

Inhalt:

Einführung in COMPASS, die Assemblersprache der CDC-CYBER. Der Kurs wird nicht PP-COMPASS behandeln.

KURSLEITUNG: Wolfgang ALTFAHRT

5. SEMINAR SELCOM VERSION 2

KURSGEBÜHR: keine

Zeit: *) 1976 01 12 - 14, vormittags

Inhalt:

Einführung in die erweiterten Möglichkeiten der neuen Version des statistischen Programmsystems SELCOM.

KURSLEITUNG: Peter RASTL

Weitere Kurse nach Bedarf.

Ferner findet einmal monatlich (jeweils Freitag 9-12 Uhr) in der Programmierstube des Rechenzentrums UNI, Stiege 3, 1. Keller, eine Einführung in die Einrichtungen des Rechenzentrums statt, die allen Interessenten und neuen Rechenzentrumsbenutzern empfohlen wird.

Die nächsten Termine sind: 1976

10. Jänner, 07. Feber, 06. März, 03. April, 08. Mai, 05. Juni

KURSLEITUNG: Peter RASTL

*) Ort und genaue Zeiten für sämtliche Kurse werden rechtzeitig bekanntgeben. Anmeldung im Sekretariat des IRZ-UNI, Tel.Nr.: 43-61-11/Kl.16 D.



E 3 HANDBÜCHER

Die Informationsschriften und Handbücher der Rechenzentren UNI und TU sind an beiden Rechenzentren erhältlich (UNI: Programmberatung, TU: Sekretariat und Programmberatung).

INFORMATIONSSCHRIFTEN UNI:

Preis öS:

SPSS	10,-
UTILITY-PROGRAMME	10,-
INTERCOM	10,-
LOCHERBESCHREIBUNG	10,-
SELCOM	10,-
ALGOL	10,-
UPDATE	10,-
SERIE BYTE	10,-
SCOPE	10,-
FORTTRAN-UMSTELLUNG	10,-

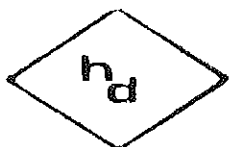
INFORMATIONSSCHRIFTEN TU:

SCOPE 3.4	Handbuch	(92 Seiten)	30,-
SCOPE 3.4	Kurzfassung	(22 Seiten)	10,-
RD1	Kursunterlage (als PASCAL- Unterlage ver- wendbar)	(52 Seiten)	20,-
BENUTZUNG DER PRO- GRAMMBIBLIOTHEK	Kurzfassung (2. Auflage)	(23 Seiten)	20,-
ANLEITUNG ZUR BENUT- ZUNG VON MAGNETBÄNDERN	Kurzfassung	(5 Seiten)	5,-
ALGOL	Handbuch (in Vorbereitung)		
COBOL UMSTELLUNGSKURS	Kursunterlage	(22 Seiten)	10,-
INTERCOM	Handbuch	(ca. 40 S.)	20,-
FORTTRAN	Handbuch (3. Auflage in Vorbereitung)	(120 Seiten)	45,-
FORTTRAN	Fehlermeldungen	(40 Seiten)	15,-
PASCAL	Handbuch (2. Auflage)	(50 Seiten)	20,-
RATFOR	Handbuch (in Vorbereitung)		
MNF	Auszug aus engl. Manual	(80 Seiten)	30,-



Preisliste der CDC Manuals

No.	Rev.	Titel	Preis
60359400	C	8-Bit Subroutines Version 1.0 Ref Man	68.04
60384700	A	Algol Version 4 Ref Man	113.40
19980300	A	Basic Language Version 2 Ref Man	102.06
60328400	B	Cobol Version 4 Instant	39.69
60384100	E	Cobol Version 4 Ref Man	238.14
60361000	E	Compass Version 3 Instant	30.61
60361700	B	Compass Version 3 Instruction Card	11.34
60360900	E	Compass Version 3 Ref Man	153.09
60359200	C	Data Description Language for the Query Update Sub-Schema (QU DDL) Ref Man	45.36
60307000	F	Form Version 1 Ref Man	56.70
60357900	B	Fortran Extended Version 4 Instant	30.61
60305600	G	Fortran Extended Version 4 Ref Man	181.44
60384600	B	Intercom Interactive Command Summary	11.34
60385700	C	Intercom Interactive Guide for Users of Cobol	79.38
60359700	B	Intercom Interactive Guide for Users of Fortran Extended	79.38
60307100	E	Intercom Version 4 Ref Man	30.72
60372200	A	Loader Instant	39.69
60344200	G	Loader Ref Man	102.06
60385500	B	Query Update Version 2 Mini User Guide	45.36
60384900	B	Query Update Version 2 Ref Man (vols 1,2)	136.08
60385300	A	Record Manager Guide for Users of Cobol	68.04
60385200	A	Record Manager Guide for Users of Fortran Extended	56.70
60359600	B	Record Manager Version 1 File Organization User's Guide	179.17
60307300	E	Record Manager Version 1 Ref Man	113.40
60307200	H	Scope Version 3.4 Ref Man	187.11
60358700	A	Scope Version 3.4 User's Guide	145.15
60358500	D	Simscrip Version 3 Ref Man	90.72
60343900	G	Sort/Merge Versions 4 and 1 Ref Man	68.04
60328800	B	Sympl Version 1 Ref Man	68.04
60360200	B	Update Instant	45.36
60342500	E	Update Ref Man	45.36



D 4,7 LINK

LINK - FILE-COMPRESS-ROUTINE AN DER CYBER 74

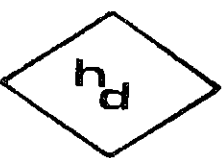
Es ist bekannt, daß unter dem Betriebssystem SCOPE 3.4 Bibliotheken nur in beschränktem Maße eingesetzt werden können. Zum Beispiel kann EDITLIB COBOL-Hauptprogramme nicht verarbeiten und UPDATE ist relativ kompliziert und nicht für Binärfiles zu verwenden. Außerdem müssen Direktiven verwendet werden.

Um die Lücke der Software zu schließen, wurde am RZ-TU ein Programm LINK entwickelt, das auf einfache Weise (ohne Direktiven) das Speichern von fast beliebig strukturierten sequentiellen Dateien (Plattenfiles) in einer Bibliothek ermöglicht. (Diese Bibliotheksform hat nichts mit dem in SCOPE verwendeten Begriff LIBRARY zu tun).

Das Programm LINK kann beliebige Recordtypen verarbeiten. Daneben wird aber vor allem Speicherplatz gespart, da die minimale Größe einer Partition nur 2 PRU's beträgt. Bei Speicherung als Einzelfiles werden jeweils mindestens 1 RB belegt! Es können bis ca. 30 Partitions in einem Record Block komprimiert werden.

Es wird dringend ersucht, vom Programm LINK Gebrauch zu machen, da der Speicherplatz auf den Magnetplatten sehr knapp ist!

Eine detaillierte Beschreibung von LINK ist in der Programmberatung zum Preise von S 2.- zu haben.



LINK - DIRECTORY OF DATALIB - DATE= 1975.07.31 SIZE= 152/ 3

PARTITION	DATE	VERSION	PRUS	LEVEL	FLAG
*****	75/07/28	8	256		
SELECT	75/05/26	0	11		M
LIST	75/07/22	2	18	1	M
STUD	75/06/08	0	52		::
PROF	75/06/23	1	10		M
CCOPY	75/06/20	0	15		M
PLOT	75/07/14	3	3	3	P
JOB	75/07/26	1	1		C
DATA	75/07/28	0	1		D

Einsparung: 8 Partitions benötigen nur 3 Record Blocks,
statt 8 RB bei konventioneller Speicherung!

Das Programm LINK hat folgende Verarbeitungsmöglichkeiten:

1. Eine (LINK-) Bibliothek kann eröffnet oder reorganisiert werden.
2. Ein File kann als Partition in die Bibliothek eingespeichert werden.
3. Eine schon gespeicherte Partition kann wieder extrahiert und als Local File bereitgestellt werden.
4. Eine Partition kann gelöscht werden.
5. Das Inhaltsverzeichnis (Directory) kann aufgelistet werden.
6. Gespeicherte Steuerkarten können direkt ausgeführt werden, ohne daß (wie bei der ROUTE-Karte) ein neuer Job kreiert wird.

Roman KAISER



D 5.6 SIMULA

NEUER SIMULA-COMPILER

Mitte Oktober wurde der neue SIMULA-Compiler an der CYBER 74 installiert. Er wurde vom NDRE (Norwegian Defence Research Establishment) entwickelt und ist in PASCAL geschrieben.

Neben geringen Einschränkungen (Typübereinstimmung von aktuellen und formalen Parametern, keine Systemprozeduren als aktuelle Parameter) bietet er folgende Vorteile gegenüber der Standard CDC-Version, die nicht mehr verfügbar ist:

- (.) Ca. 1.3mal schnellere Übersetzung;
(CM 55000 für ein 3000 Zeilen-Programm).
- (.) Ca. 2-3mal schnellere Ausführungszeit
bei halber Feldlänge.
- (.) Bei Laufzeitfehlern werden alle Files
geschlossen und ein Post-Mortem-Dump
ausgegeben.
- (.) Die DATASET-Karte wurde entfernt.
- (.) Implementierung der Prozedur CALL.

Verwendung des neuen NDRE-SIMULA-COMPILERS

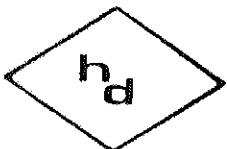
⋮
ATTACH, SIMULA.
ATTACH, SIMLIB.
SIMULA, L.
LGO.
⋮

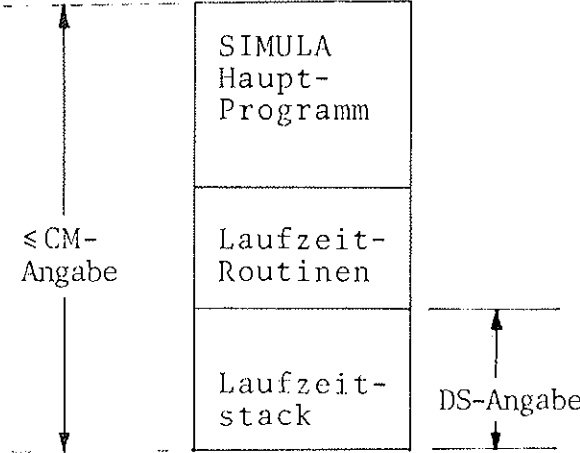
Rudolf BRAUN



COMPILE - TIME OPTIONS

Option	Bedeutung	Default
I=lfm I	Source-Programm steht auf File lfn - " - COMPILE	I=INPUT
L=lfm L	Source Listing kommt auf File lfn - " - OUTPUT	
A	Listung des entsprechenden Assembler-Codes	
N	<u>Ausschaltung</u> der Indexüberprüfung	nicht ge- geben (=Indexüber- prüfung ein- geschaltet).
LS=i	Jede i-te Zeilennummer wird abge- speichert (zur Lokalisierung von Laufzeitfehlern)	LS=1
C	Ausgabe einer Cross-Referenz-Liste	nicht ge- setzt.
B=lfm	Binärprogramm kommt auf File lfn	B=LGO.
D	Debug-Option (in Zusammenhang mit den Debug-Pro- zeduren benötigt)	
SIMULA.≡ SIMULA,I=INPUT,LS=1,B=LGO. (Beachte, daß standardmäßig <u>kein</u> Source-Listing erzeugt wird.)		



Option	Bedeutung	Default
I=1fn	SIMULA-Standard-Eingabe-File SYSIN wird durch File <u>1fn</u> ersetzt	I=INPUT
O=1fn	SIMULA-Standard-Ausgabe-File SYSOUT wird durch File <u>1fn</u> ersetzt	O=OUTPUT
DS=nnnnn ₈	<p>Gibt die Größe des Laufzeitstacks an (<u>Nur</u> bei eingeschaltetem Reducemode von Bedeutung). Beachte auch die EFL-Option.</p> 	DS=6000 ₈
<p>EFL=ON</p> <p>=PART</p> <p>=OFF</p>	<p>Der Stack breitet sich bis maxfl (Job-CM) aus</p> <p>Der Stack breitet sich nur bei Bedarf aus (höchstens bis maxfl), sodaß "STACK OVERFLOW" meistens vermieden werden kann.</p> <p>Der Stack hat die konstante Größe DS</p> <p>(Die EFL-Option ist nur bei eingeschalteten Reduce-Mode von Bedeutung!)</p>	EFL=ON
D=code	<p>Debug-Option</p> <p>code=C_nC_{n-1}...C₂C₁;</p> <p>C_i=i-te Ziffer; 1≤i≤5</p> <p>(Code-Ziffer, die von der Prozedur SIMDUMP verwendet wird)</p>	nicht gesetzt
LGO. ≡ LGO, I=INPUT, O=OUTPUT, DS=6000, EFL=ON.		

MNF (MINNESOTA-FORTRAN) - COMPILER

An den Rechenanlagen CYBER 73 und CYBER 74 ist eine neue Version des MNF-Compilers installiert worden. Die Verwendung dieses FORTRAN-Compilers ist sehr zu empfehlen, da er gegenüber anderen Compilern folgende Vorteile bietet:

- a) bessere Fehlerdiagnostik bei der Compilation
- b) in vielen Fällen (bes. in kleinen Programmen) Erzeugung von besserem Code, d.h. raschere Programmausführung
- c) einfach zu handhabende Möglichkeiten zur Fehlersuche (Debug-Anweisungen)
- d) äußerst genüge Übersetzungszeit

Im folgenden sollen die wichtigsten Eigenschaften dieses Compilers kurz besprochen werden - für weitergehende Informationen stehen MNF-Reference Manuals in der Programmberatung zur Verfügung. In nächster Zeit wird in beiden Rechenzentren auch eine kurze Broschüre über MNF erhältlich sein.

1. AUFRUF DES MNF-COMPILERS

Die Steuerkarte zum Aufruf des MNF-Compilers ist:

MNF.

oder

MNF (p₁, p₂, ..., p_n)

bzw. damit gleichbedeutend

MNF, p₁, p₂, ..., p_n.

Dabei stellen die p_i Parameter dar, die entweder die

Form a oder a = b



annehmen, wobei a einen der Buchstaben:

A, B, C, D, E, F, G, I, J, L, O, P, R, T, Y, Z

und b eine Ziffer oder einen Dateinamen darstellt.

Die Anweisung MNF ist gleichbedeutend mit

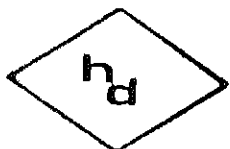
MNF(B=LGO,C=FTN,I=INPUT,L=OUTPUT,E=2,P=5000,R=0)

Die Bedeutung dieser Parameter ist:

B = LGO binärer Output auf der Datei LGO
 C = FTN FTN-Calling Sequence
 I = INPUT ... Quellenprogramm vom File INPUT
 L = OUTPUT .. Liste des Quellenprogramms auf das
 File OUTPUT
 E = 2 Fehlermeldung vom Typ COMMENT und NOTE
 werden unterdrückt, Fehlermeldungen vom Typ
 CAUTION, WARNING und FATAL werden geschrieben.
 R = 0 keine Cross Reference Map wird ge-
 schrieben.
 P = 5000 Zur Ausführungszeit werden maximal 5000
 Zeilen auf das File OUTPUT geschrieben.

Die Default-Werte für die Parameter wurden im allgemeinen
 gleich wie beim FTN-Compiler gewählt. Eine wichtige Aus-
 nahme ist:

	MNF,I.
Dies bedeutet	MNF(I=INPUT),
während	FTN,I.
gleichbedeutend mit	FTN(I=COMPILE)
ist.	



2. ERWEITERUNGEN DER FORTRAN-SPRACHE

MNF ist mit ANSI-Standard-FORTRAN vollständig und mit FTN-Extended Fortran weitgehend kompatibel. Zusätzlich zu den Möglichkeiten des FTN-Compilers gibt es noch folgende interessante Spracherweiterungen:

a) END- und ERR-Parameter im READ-Statement

`READ (n,f,END=a, ERR=b) liste`
ist zulässig, wobei a und b Anweisungsnummern sind, zu denen bei Dateiende oder bei Eingabefehler verzweigt werden soll. (IBM kompatibel) Der Parameter END=a stellt eine Alternative dar zu der ebenfalls zulässigen Abfrage des Wertes von EOF, z.B.

`IF(EOF(n).NE.Ø) GOTO a`

b) Arithmetische Ausdrücke im DO-Statement

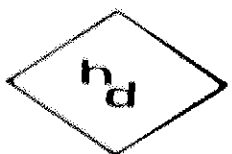
`DO a i=n1,n2,n3 oder DO a i=n1,n2`
wobei n₁,n₂,n₃ auch arithmetische Ausdrücke mit ganzzahligem Wert (im dafür erlaubten Wertbereich) sein dürfen.

c) FOR-Statement als Ergänzung zur DO-Schleife

`FOR a i=n1,n2,n3 oder FOR a i=n1,n2`
Diese Anweisung bietet die Möglichkeit, den Schleifenindex von größeren Werten zu kleineren hinlaufen zu lassen.
z.B.

`FOR 13 I=K+1,1`

Die Variable I nimmt die Werte K+1, K, K-1, ..., 3, 2, 1 an.



- d) Das Program Statement ist in der üblichen Form (abgesehen von einer Einschränkung) zulässig, die Default-Annahme ist jedoch anders als bei FTN:

```
PROGRAM ST. (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
```

Das Program Statement kann also weggelassen werden, wenn man die Einheitsnummern 5 und 6 für die Standard-Ein- und Ausgabedateien INPUT und OUTPUT verwendet (IBM Standard).

- e) Einige zusätzliche Darstellungsmöglichkeiten für spezielle Zahlen und Operatoren, z.B.

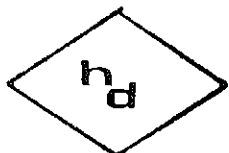
```
.INF. oder .R. für "Unendlich"
.IND. oder .I. für "Undefiniert"
.XOR. oder .X. für "Ausschließendes Oder"
                in logischen Maskenausdrücken
```

ferner > und < als Alternative zu .GT. und .LT. usw.

- f) Zusätzliche eingebaute Funktionen

```
CPTIME .... CP-Zeit seit Beginn des Jobs
IUEXP ..... Exponent einer reellen Zahl
COUNT ..... Anzahl der Einser-Bits in einem Wort
EQV ..... logische Äquivalenz
```

usw.



3. EINSCHRÄNKUNGEN GEGENÜBER FTN-COMPILER

Einige FTN-Anweisungen, die jedoch keine ANSI-Standard-FORTRAN-Anweisungen darstellen, sind in der MNF-FORTRAN Version nicht erlaubt.

- a) Außergewöhnlicher Rücksprung aus einem SUBROUTINE-Unterprogramm

Die Konstruktion

```
SUBROUTINE name (parameter),RETURNS(i1,i2,...)  
:  
RETURN i
```

zum Rücksprung aus einem Unterprogramm zu einer im rufenden Programm definierten Anweisungsnummer ist nicht erlaubt (dies kann man aber leicht durch Einführen einer Hilfsvariablen umgehen).

- b) Angabe einer Recordlänge im PROGRAM-Statement

```
PROGRAM name (file = b/r)
```

Die Angabe der Pufferlänge b ist zulässig, hingegen nicht die Angabe der Recordlänge in der Form /r. Die Standard-Recordlänge ist 150 Zeichen. Ferner ist auch die Verwendung der Einheitennummer 0 (TAPE0) unzulässig.

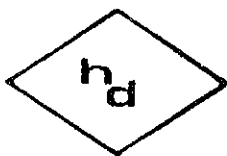
- c) Das arithmetische IF mit zwei Anweisungsnummern

```
IF (arithm. Ausdruck) m1,m2
```

ist nicht zulässig, hingegen ist das logische IF mit zwei Anweisungsnummern m₁,m₂

```
IF (logischer Ausdruck) m1,m2
```

erlaubt.



d) Anzahl der Fortsetzungskarten

Die Anzahl der Fortsetzungskarten (mit einem von Blank oder Null verschiedenen Zeichen in Spalte 6) ist vom Aufbau des Statements abhängig; im allgemeinen ist diese Zahl ≤ 19 , jedoch kann dieser Wert in Ausnahmefällen kleiner werden. Garantiert sind nur 9 Fortsetzungskarten.

4. UNTERPROGRAMME

Prüfung der Korrespondenz zwischen aktuellen und formalen Parametern.

- a) Der Computer prüft die Übereinstimmung dieser Parameter, falls die Unterprogramme gemeinsam compiliert werden. Es erfolgt eine Meldung, falls ein aktueller Parameter ein Ausdruck oder eine Konstante ist.
- b) Programme, die mit MNF (C=FTN) compiliert wurden, können Unterprogramme aufrufen, die vom FTN-Compiler übersetzt wurden, und umgekehrt. Jedoch müssen in jedem Fall die mit dem MNF übersetzten Programme zuerst geladen werden.

5. WEITERE UNTERSCHIEDE

Es treten noch einige sonstige Unterschiede zwischen FTN und MNF auf, z.B.:

a) Vorbesetzung nicht-initialisierter Variablen

Nichtinitialisierte Variable werden mit undefiniert vorbe-
setzt. Variable und Felder haben als Default-Wert
6000 000 0002 0040 0000B + Adresse, sofern sie nicht im
Blank COMMON sind. Im Blank COMMON werden Variable und Felder
mit -.IND. vorbe-
setzt.

Die Vorbesetzung von Feldern und Variablen außerhalb des



COMMONs kann mit dem MNF-Parameter Z auf Null geändert werden.

b) Arithmetische Ausdrücke mit gemischten Variablentypen

z.B.: wird X = I/J/R
wie X = FLOAT(I/J)/R
und nicht wie X = FLOAT(I)/FLOAT(J)/R (wie bei FTN)
übersetzt.

Ferner ist der Ausdruck A**B**C ohne Klammern verboten, da aufgrund der Standard-Regeln nicht klar ist, ob dies als A**(B**C) oder (A**B)**C zu übersetzen ist. FTN interpretiert A**B**C als (A**B)**C.

c) Die Literalkonstante "DON""T" wird (in der FORMAT-Anweisung) als die Zeichenkette DONT interpretiert. (Der FTN-Compiler hingegen faßt dies als die Zeichenkette DON"T auf.)

d) Die EQUIVALENCE-Anweisung behindert die Optimierung.

Wegen der Optimierung muß im MNF-Compileraufruf der Parameter F ("Forced Store") angegeben werden, wenn im Programm EQUIVALENCE-Anweisungen auftreten, durch die Variablen mit zwei verschiedenen Namen angesprochen werden.

Beispiel:

```
EQUIVALENCE(I,J)
J=I+1
K=I+1
```



In diesem Falle wird beim Weglassen des F-Parameters der Wert, der sich in der ersten Anweisung I+1 ergibt, nicht sofort nach J abgespeichert. Da aber I und J den gleichen Speicherplatz haben, geht I in die zweite Anweisung K=I+1 noch mit unverändertem Wert ein. Das kann man vermeiden, indem man entweder statt K=I+1 in der zweiten Anweisung K=J+1 schreibt oder indem man MNF,F angibt. Gleiches gilt, wenn in einem Unterprogramm ein- und derselbe Aktualparameter durch zwei Formalparameter dargestellt wird, z.B.

```
CALL UP(A,I,I)
.
.
SUBROUTINE UP(A,I,J)
DIMENSION A(I)
J=I+1
K=I+1
A(K)=5Ø.
```

e) Listengesteuerte Ein- und Ausgabe

Die Anweisung READ (n,*) liste
bzw. READ*, liste

liest Variablen im freien Format von Eingaberecords, wobei bei jedem READ-Befehl ein neues Record gelesen wird. (Bei Verwendung des FTN-Compilers erfolgt kein automatischer Übergang zu einem neuen Record).

Die Anweisungen

PRINT*, liste und PRINT fmt, liste

existieren, nicht aber die analogen FTN-Erweiterungen

WRITE*, liste und WRITE fmt, liste



6. DEBUG-ANWEISUNGEN

Die Debug-Anweisungen beginnen mit

C\$ in den Spalten 1 und 2,

unterscheiden sich ansonsten jedoch vollständig von den FTN-Debug-Anweisungen.

Bei den MNF-Debug-Anweisungen handelt es sich um normale, ausführbare FORTRAN-Statements, die jedoch nur beim Vorliegen des D-Parameters mitcompiliert werden.

Beispiel:

```
C$ PRINT*,X,Y
C$ CALL SUB(X,Y)
```

usw.

Darüber hinaus gibt es die TRACE-Anweisung, z.B.:

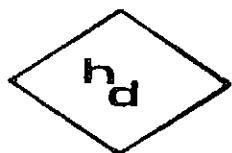
C\$	TRACE	SUBSCRIPTS	Indexüberprüfung
C\$	TRACE	STATEMENT NUMBER	Statistik über Anweisungsnummer
C\$	TRACE	variablenliste	Überwachung der Wertzuweisung
C\$	TRACE	ARITHMETIC	Überwachung des Wertebereichs

Eine umfassende Überprüfung kann durch die Anweisung

```
C$ TRACE
```

bzw. durch die T-Option auf der MNF-Karte erreicht werden.

Dabei werden folgende TRACE-Options aktiviert:



DOLOOPING	}	Kontrollausdruck nur bei Fehlern!
FORMATIO		
SUBSCRIPTS		
TRANSFERS	}	Kontrollausdruck immer!
CALLS		
STATEMENT NUMBERS		

MNF,T. ermöglicht daher eine sehr effektive Programmüberwachung ohne Änderung des Programms.

Die TRACE-Anweisungen wirken statisch bis ans Ende der Programmeinheit (END-Anweisung) oder bis sie mit einer

C\$ NO TRACE ...

-Anweisung ausgeschaltet werden.

Während die TRACE-Anweisungen

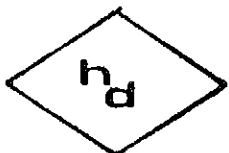
SUBSCRIPTS, ARITHMETIC, DOLOOPING, FORMATIO, TRANSFERS

mögliche Fehler aufdecken, liefern die TRACE-Anweisungen

STATEMENT NUMBERS, SUBPROGRAM CALLS, SUBPROGRAM FLOW

statistische Angaben über den Programmablauf.

Karl PECHTER
 Gerhard SCHMITT
 Willy WEISZ



DER FORTRAN-PREPROCESSOR RATFOR

FORTRAN ist sicher nicht als ideale Programmiersprache zu bezeichnen. Daß sie trotzdem die größte Verbreitung von allen Programmiersprachen findet, hat mehrere Ursachen, die hier stichwortartig zitiert seien:

Universalität:

Es gibt praktisch für jede Rechenanlage (selbst Minicomputer) einen FORTRAN-Übersetzer.

Kompatibilität:

Durch die seit 1966 bestehende Standardisierung hohe Kompatibilität zwischen Rechenanlagen verschiedener Hersteller. Außerdem durch die einfachen Schnittstellenkonzepte (COMMON-Blöcke, Unterprogrammparameter) weitgehende Kompatibilität mit anderen Programmiersprachen (ASSEMBLER, COBOL, ALGOL, PASCAL,...).

Effizienz des erzeugten Objektcodes:

Infolge des einfachen Sprachkonzepts und langjähriger Verbesserungen der Übersetzer wird teilweise bereits die Effizienz von guten Assembler-Programmen erreicht.

Alle diese Vorzüge können allerdings nicht über die Tatsache hinwegtäuschen, daß FORTRAN eine veraltete, unkomfortable Sprache ist. Das ist besonders daraus ersichtlich, daß als Grundelement für die Steuerung des Programmablaufs nur der Sprungbefehl und entsprechende Marken verwendet werden. Außerdem gibt es keinerlei Möglichkeit, Gruppen von Anweisungen (etwa



hinter einem IF) syntaktisch zusammenzufassen. So bleibt selbst dem besten Programmierer nichts anderes übrig, als unzählige Sprünge und Marken zu verwenden, die das entstehende Programm sehr schwer lesbar, schwer verständlich und äußerst unübersichtlich machen. Fehlersuche und Änderungen des Programms sind daher sehr langwierig. Eine weitere Unannehmlichkeit von FORTRAN sind viele unlogische und unnötig starre syntaktische Regeln (Position von Marken und Anweisungen, Fortsetzungskarten, etc.).

Eine Methode, die angeführten Mängel zu beseitigen oder wenigstens zu mildern, besteht darin, entsprechende Erweiterungen der Sprache zu definieren, die mehr Komfort bieten. Die Spracherweiterungen werden mit Hilfe eines Preprocessors in normales FORTRAN übersetzt. Diese Methode bietet sich besonders an, da die Erweiterungen eher oberflächliche Änderungen von Syntax und Semantik beinhalten.

Die Verwendung von Preprozessoren fand besonders in letzter Zeit weite Verbreitung. Eine vor kurzem durchgeführte Erfassung von FORTRAN-Preprozessoren ergab über 50 derartige Entwicklungen auf der ganzen Welt. In den letzten Monaten wurden einige dieser Produkte vom EDV-Zentrum der TU Wien beschafft, eines der leistungsfähigsten adaptiert und in erweiterter Form implementiert. Es handelt sich dabei um den Processor RATFOR, der von Brian W. Kernighan in den Bell Laboratories entwickelt worden ist.

Es folgt eine kurz gefaßte Aufzählung der Möglichkeiten von RATFOR mit selbsterklärenden Beispielen:



Freies Format beim Programmieren ("Syntactic sugar")

-mehrere Anweisungen pro Zeile

```
X = 0.0; Y = 0.0; Z = 0.0
```

-automatische Fortsetzung

```
X = A+B+
      C*(D+E)
```

-Kommentar an beliebigen Stellen

```
A = 1.0 # ANFANGSWERT SETZEN
```

-Einfach- und Doppelapostroph als String-Begrenzer

```
FORMAT (1X, 'WERT = ',I5, "DON'T USE")
```

-Verwendung von <, >, == etc. als Vergleichsoperatoren

```
IF(A<=B) ...
```

Syntaktische Erweiterungen

-Zusammenfassen mehrerer Anweisungen

```
IF(A<0.0) [A = A+B; C = 0
           B = B+1.0]
```

-IF-ELSE

```
IF(K<0) J = K
ELSE    J = 0
```

-WHILE

```
WHILE(CHAR(I) == BLANK) I = I+1
```

-FOR

```
FOR(I = N; I>0; I = I-1) ...
```

-DO

```
DO I = 1,N
   A(I) = 0.0
```

-BREAK zum Verlassen von Schleifen

```
FOR(I = 80; I > 0; I = I-1)
  IF(CHAR(I) .NE. BLANK) BREAK
```



-NEXT (Fortsetzen beim nächsten Schleifelement
in DO, FOR, WHILE)
DO I = 1, N
IF(X(I) < 0.0) NEXT
... Verarbeitung der positiven Elemente ...
]

Makro-Möglichkeiten

-DEFINE für symbolische Parameter
DEFINE FELDGROESSE 100
DIMENSION A (FELDGROESSE), B (FELDGROESSE)

-INCLUDE für Programmstücke, die auf getrennten
Dateien stehen
SUBROUTINE TEST(X)
INCLUDE COMMONPART

-Gesteuertes Ein- und Ausschalten von bestimmten
Programmteilen

DETAILS DER IMPLEMENTIERUNG AN DER CYBER 74

Aufruf:

ATTACH, RATFOR.
RATFOR, in, out, error, list.

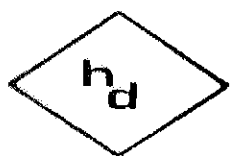
Kernspeicherbedarf:

etwa 35 000₈ Worte

Laufzeit: in der Größenordnung des
FTN-Übersetzers (OPT = 2)

Dokumentation:

RATFOR-Handbuch
(Programmberatung, Sekretariat)

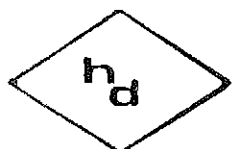


Am Mittwoch, dem 10. Dezember, 18.00 Uhr, findet im Seminarraum des Instituts für Technische Mathematik (Neues Institutsgebäude, Gußhausstraße 27-29, 4. Stock) ein Vortrag über das Thema:

'FORTRAN-Preprozessoren'

statt, in dem vorwiegend RATFOR besprochen und außerdem ein Ausblick auf andere Preprozessoren gegeben wird.

Dieter SCHORNBÖCK



PROGRAMMSAMMLUNGEN AN DER CYBER 73

Mehrere Programmsammlungen stehen an der CYBER 73 zur Verfügung. Neben den Programmen jener Programmsammlungen, welche bereits seit Jahren am EDV-Zentrum der Universität verwendet werden (SSP, BMD, CERN), können auch die Programme der neuerworbenen Harwell-Subroutine-Library und der Guttman-Lingoes-Program-Series von UPDATE-Bibliotheken auf Magnetbändern aufgerufen werden. Es ist jedoch zu beachten, daß ein großer Teil dieser Programme noch nicht an der CDC CYBER 73 adaptiert ist.

Ab Anfang 1976 wird die FORTRAN-Version der NAG-Bibliothek auch an der CYBER 73 zur Verfügung stehen, einer reichhaltigen Programm-bibliothek von ausgetesteten und gut dokumentierten Programmen. Darüber hinaus liegen für interessierte Benutzer Programm-listen und Beschreibungen der CACM-Algorithmen auf.

Ein alphabetisches Schlagwortverzeichnis (KWIC-Index der beschreibenden Programmtitel) der SSP-, CERN-, BMD- und Harwell-Programme, sowie einiger eigener Programme, kann bei der Programmberatung eingesehen werden.

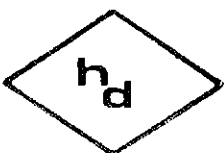
Ein typisches Beispiel für das Aufrufen und Compilieren eines Programms aus einer der fünf unten angegebenen Programmsammlungen ist folgender Job:

```

job,NT1,T100.
Accountkarte
LABEL,OLDPL,R,L=label*), NORING,VSN=vsn*).
UPDATE,Q,L=1.
FTN,I.
eor
*COMPILE programmname
eof

```

*) LABEL und VSN des Magnetbandes sind bei den Programmsammlungen angegeben.



1. SSP - SCIENTIFIC SUBROUTINE LIBRARY

label=SSP, vsn=4A17A1

Eine genaue Beschreibung der SSP-Programmbibliothek finden Sie im Heißen Draht Nr. 13 (Juni 1975).

2. BMD - BIOMEDICAL COMPUTER PROGRAMS

(Sammlung statistischer Programme)

label=BMDLIB, vsn=4D11A1

Neben dieser UPDATE-Bibliothek, welche alle BMD-Programme enthält (BMDX-, BMDP- und BMDØ-Serie), gibt es noch eine UPDATE-Bibliothek mit den für den RUN-Compiler adaptierten Programmen der BMDØ-Serie (label=BMD, vsn=4A18A1). Bei diesen Programmen ist das Program-Statement durch ein anderes ohne Angaben von Puffergrößen zu ersetzen.

Programmname: der sechsstellige Programmname

3. GUTTMAN - LINGOES - NONMETRIC - PROGRAM - SERIES

Eine Programmsammlung zur nicht-metrischen multidimensionalen Skalierung (Analyse von Ähnlichkeits- und Unähnlichkeitsmatrizen; Anwendung: vor allem in den Sozialwissenschaften).

label=GUTTMAN, vsn=4D09A1

Programmname: dreistellige Programmkennzeichnung

(Spalte 73-75 im Ausdruck)

(eventuell mit nachgestelltem A oder B)

4. CERN-PROGRAMME

(Numerische Methoden, Anwendung in den Naturwissenschaften)

label=UVPL, vsn=4D10A1

Programmname: vierstelliger Programmcode

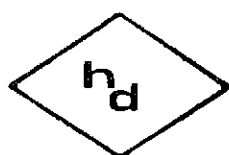
Ein großer Teil der FORTRAN-Programme der CERN-Programmlibrary befindet sich zusammen mit einer Reihe anderer Programme auf einer UPDATE-Bibliothek. Programmlisten der FORTRAN-Quellenprogramme jener CERN-Programme, welche auf der UPDATE-Bibliothek nicht enthalten sind, liegen im Programmberatungszimmer auf.



5. HSL - HARWELL SUBROUTINE LIBRARY
(Numerische Methoden, Anwendung in den Naturwissenschaften)
label=HSL, vsn=4A19A1
Programmname: der in der Programmbeschreibung angegebene
Name mit einem vorgesetzten A (bzw. B, wenn
auch eine neue Version dieses Programms
vorhanden ist).

Die Beschreibungen und Programmlisten der FORTRAN-Quellenprogramme aller dieser Programmsammlungen befinden sich bei der Programmberatung und können dort während der Dienstzeiten eingesehen werden. Für weitergehende Informationen über die Verwendung dieser UPDATE-Bibliotheken steht Ihnen die Programmberatung gerne zur Verfügung.

Dieter KÖBERL



D 6.1 PROGRAMMBIBLIOTHEKEN

PROGRAMMSAMMLUNGEN AN DER CYBER 74

Sammlung	Gebiete	Art	Sprache
THLIB1	Math + nnDV + Ut	U	A + F
IMSL	Math + Stat	U	F
NAG	Math + Stat	U	A + F
SSP	Math + Stat	U + H	F
CERN	Math + Ut	U	F
BMD	Stat	H	F
SPSS	Stat	S	

Math ... numerische Mathematik
 Stat ... Statistik
 nnDV ... nichtnumerische Datenverarbeitung
 Ut Utility (Hilfsprogramme)

U ... Unterprogramme A ... ALGOL 60
 H ... Hauptprogramm F ... FORTRAN
 S ... Programmsystem

Hinweise über die Anwendung der Programmbibliotheken sind im Handbuch "Benutzung der Programmbibliothek" (2. Auflage) enthalten.



PROGRAMMBIBLIOTHEK T H L I B I

NUMMER	NAME	T I T E L	KLASSE
LSF001	GAUL	Loesung eines linearen Gleichungssystems	F4.F
LSF002	AGAUL	Loesung eines linearen Gleichungssystems	F4.A
LSF003	ITGAUL	Loesung eines linearen Gleichungssystems mit Nachiteration	F4.F
LSF004	SOLIM	Linearer Ausgleich mit Hilfe der Zerlegung in singulaere Werte	F4.F
LSF005	FROMBG	Numerische Integration (Rombergverfahren)	D1.F
LSF006	AROMBG	Numerische Integration (Rombergverfahren)	D1.A
LSF007	FSMPSN	Numerische Integration (Simpsonregel)	D1.F
LSF008	ASIMPSN	Numerische Integration (Simpsonregel)	D1.A
LSF009	REGLEI	Polynomnullstellen (reelle Koeffizienten)	C2.A
LSF010	ALGLEI	Polynomnullstellen (komplexe Nullstellen)	C2.A
LSF011	OREIG	Eigenwerte einer reellen Matrix	F2.F
LTF012	KURVE	Kurvenzeichnen am Schnelldrucker	J5.F
LTF013	PLOTT1	Kurvenzeichnen am Schnelldrucker	J5.F
LSF014	STOERF	Rationale Interpolation	E1.F
LSF015	STOER	Rationale Interpolation	E1.A
LSF016	LAGRAN	Interpolation mit Polynomen	E1.F
LTF017	GETIT	Formatloses Einlesen in Fortran	14.F
LSF018	FHORN	Auswertung eines Polynoms mittels Horner-Algorithmus	C1.F
LSF019	ALHORN	Auswertung eines Polynoms mittels Horner-Algorithmus	C1.A
LSF020	NULL2	Nullstellen einer beliebigen Funktion	C5.A
LSF021	GBS	Integration eines Systems gewoehnlicher Differenzialgleichungen erster Ordnung	D2.F
LSF022	MAKINT MOVFCH MOVCHR MOVSTR KOMPFC ISEKCH ISKTC ISEKST	Character Handling Routines	M4.F
LSF023	QIKS	Sortieren im Kernspeicher	M1.F
LSF024	FNEVIL	Neville- Algorithmus	E1.F
LSF025	NEVIL	Neville- Algorithmus	E1.A
LSF026	ASPL	Ausgleichende Spline- Interpolation	E2.F
LTF027	ALKURV	Kurvenzeichnen am Schnelldrucker	J5.A
LTF028	APLOTT	Kurvenzeichnen am Schnelldrucker	J5.A
LSF029	ZZAHLA	Gleichverteilte Zufallszahlen	G5.A
LSF030	ZZAHLF	Gleichverteilte Zufallszahlen	G5.F
LSF031	EVSPL	Auswertung einer Splinefunktion	C1.F
LSF032	ASPLA	Ausgleichende Splineinterpolation	E2.A
LSF033	AVSPLA	Auswertung einer Splinefunktion	C1.A
LSF034	QSORTA	Sortieren von Gleitkommazahlen	M1.A
LSF035	QSORTE	Sortieren von Gleitkommazahlen	M1.F



PROGRAMMBIBLIOTHEK T H L I B I

NUMMER	NAME	T I T E L	KLASSE
LSF036	BISECA	Nullstelle einer beliebigen Funktion	C5.A
LSF037	BISECF	Nullstelle einer beliebigen Funktion	C5.F
LTF038	POLYRT	Polynomnullstellen (komplexe Koeff.)	C2.F
LSF039	GBSA	Integration eines Systems gewoehnlicher Differentialgleichungen erster Ordnung	D2.A
LSF040	AITGAU	Loesung eines linearen Gleichungssystems	F4.A
LSF041	UNSRAY	Berechnung der Eigenwerte und Eigenvektoren einer unsymmetr. Bandmatrix	F2.A F2.A
LSF042	SUSBE	Loesung eines linearen Gleichungssystems (unsymmetrische Bandmatrix)	F4.A
LSF043	ISUSBE	Zeilenweise Eingabe von unsymmetrischen Bandmatrizen	U3.A
LSF044	SYMRAY	Berechnung der Eigenwerte und Eigenvektoren einer symmetrischen Bandmatrix	F2.A
LSF045	SSYBE	Loesung eines linearen Gleichungssystems (symmetrische Bandmatrix)	F4.A
LSF046	ISSYBE	Zeilenweise Eingabe von symmetrischen Bandmatrizen	U3.A
LSF047	AQUA1	Numerische Integration	D1.F
LSF048	AQUA2	Numerische Integration	D1.F
LSF049	AQUA3	Numerische Integration	D1.F
LSF050	AQUA4	Numerische Integration	D1.F
LSF051	NULL2F	Nullstelle einer beliebigen Funktion	C5.F
LSF052	RSPL	Rationale Splineinterpolation	E2.F
LSF053	EVRSP	Auswertung einer rationalen Splinefunktion	C1.F
LSF054	RSPLA	Rationale Splineinterpolation	E2.A
LSF055	AVRSP	Auswertung einer rationalen Splinefunktion	C1.A
LSF056	HYPNCH	Stanzten von Binaerkarten fuer Hybridrechner	J1.F
LSF057	ASCI64	Umwandlung von Display- in A S C I I Zeichen	M2.F
LSF058	STZEI	Erzeugen von A S C I I- Steuerzeichen	M2.F
LSF059	FULASC	Umwandlung von Display- in A S C I I- Zeichen mit wahlweiser Gross- oder Kleinschreibung von Buchstaben	M2.F

PROGRAMMBIBLIOTHEK T H M I S C

NUMMER	NAME	T I T E L	KLASSE
MSM001	MIX269	Umwandlung von im 026 und 029 gelochten Programmen in 029 Lochcode. Zusaetzlich Apostrophumwandlung zur Behebung der Incompatibilitaet zwischen CDC Compilern und denen anderer Hersteller	L4.
MSM003	LISTAUD	Ausgabe des Audit - Listing in sortierter und kompakter Form	Q5.
MSM004	LIBLIST	Ausgabe des Editlib - Listings in sortierter und kompakter Form	Q5.
MSM005	MATOPT	Programmpaket zur mathematischen Optimierung nach Kuenzi, Tzschach und Zehnder	H1. H2.



NEUE PROGRAMMSAMMLUNGEN

Seit Anfang November 1975 stehen allen Benutzern der CYBER 74 zwei neue Programmsammlungen zur Verfügung, die IMSL-Library und die NAG-Library. Sie ersetzen die bisherige MSL-Bibliothek. Diese wurde mit 15. November 1975 abgemietet und kann somit nicht mehr verwendet werden.

I M S L

IMSL - INTERNATIONAL MATHEMATICAL & STATISTICAL LIBRARIES INC. aus Houston, USA - entwickelt und vertreibt seit 1972 eine Bibliothek von FORTRAN-Programmen für die Gebiete der Numerischen Mathematik und der Statistik, geeignet für die Maschinen aller wichtigen Hersteller. Die Bibliothek umfaßt jetzt an die 400 Programme und ist bei Industriefirmen, Regierungsstellen und Hochschulen in den USA und in Europa verbreitet. IMSL hat sich folgende Ziele gesetzt:

- hochwertige und zuverlässige Programme zu vertreiben, wobei die Erreichung einer ausreichenden Genauigkeit vor der Ersparnis von Speicher und Laufzeit geht;
- Benutzerfreundlichkeit, insbesondere durch eine einheitliche Dokumentation, die alle notwendigen Angaben in kurzer, prägnanter Form enthält;
- schnelle Kommunikation mit den Benutzern zur raschen Behebung eventueller Fehler und um Benutzerwünsche bei der Entwicklung neuer Programme berücksichtigen zu können;
- ständige Weiterentwicklung der Bibliothek, sodaß der Benutzer in den Genuß des raschen Fortschritts in der Entwicklung neuer Algorithmen und Programmiertechniken kommt.



Zur Erreichung dieser Ziele hat sich IMSL die Mitarbeit führender Experten der verschiedenen Gebiete der angewandten Mathematik, der Statistik und der Computerscience gesichert. Als Quellen für die Programme dienen die einschlägigen Zeitschriften (Numerische Mathematik, CACM, TOMS, u.ä.), sowie wissenschaftliche Arbeiten verschiedener Universitäten. Die dort veröffentlichten Programme werden sorgfältig geprüft, getestet, verbessert und den Richtlinien der Bibliothek angepaßt, soweit entsprechend dokumentiert. Jedes Jahr erscheint eine neue Version 'edition' der Library, die die Verbesserungen bestehender Programme sowie Erweiterungen auf neue Anwendungsgebiete enthält. Zur Zeit wird für CDC-Installationen (= "Library 3") Edition 4 vertrieben, Edition 5 erscheint im Dezember.

N A G

Ursprünglich um 1970 von sechs englischen Universitätsrechenzentren ("Nottingham Algorithms Group") zum Zwecke der Koordination der Entwicklung mathematischer Software für ICL 1906A Maschinen gegründet, umfaßt NAG nun einen Großteil der englischen Universitäten und andere interessierte Forschungsinstitute ("Numerical Algorithms Group"). NAG hat sich folgende Ziele gesetzt:



- Entwicklung einer hochwertigen Allzweckbibliothek für numerische und seit neuestem auch für statistische Anwendungen;
- Aufnahme nur von zuverlässigen, sehr gut ausgetesteten Programmen;
- Ausführliche und einheitliche Dokumentation der Programme;
- Bereitstellung aller Programme, sowohl für FORTRAN als auch für ALGOL;
- Einheitliche Bibliothek für die Maschinen der verschiedenen Hersteller, die bei den Mitgliedern installiert sind;
- Ständige Weiterentwicklung der Programme und des Umfangs der Bibliothek.

Die einzelnen Gebiete der angewandten Mathematik und Statistik sind auf die Mitglieder aufgeteilt. Jedes Mitglied ist auf seinem Gebiet für Sammlung, Sichtung, Auswahl, Tests, Verbesserungen und Dokumentation der Programme verantwortlich. Die Freigabe eines Programms erfolgt erst nach einer ausführlichen Prüfung durch ein anderes, unabhängiges Mitglied der Gruppe. Ungefähr einmal pro Jahr erscheint eine neue, als "Mark" bezeichnete Version der Bibliothek, die alle Verbesserungen und Erweiterungen enthält. Zur Zeit wird für CDC-Installationen Mark 4 vertrieben, für Frühjahr 1976 ist Mark 5 angekündigt.

NAG hat bis jetzt nur den FORTRAN-Teil der Bibliothek geliefert, die Verfügbarkeit des ALGOL-Teils wird über das "SYSBULL" bekanntgegeben.



DOKUMENTATION

An Dokumentation sind für beide Bibliotheken Librarymanuals erhältlich. Sie enthalten die ausführliche Beschreibung für jedes einzelne Programm sowie für die einzelnen Anwendungsgebiete eine Zusammenfassung und Hinweise für die Auswahl der Programme für einen speziellen Anwendungsfall. Bestellungen für diese Manuals werden in der Programmberatung entgegengenommen. Hier liegen die Manuals auch zur Einsicht auf. Es können Kopien der Beschreibung von einzelnen Programmen angefertigt werden, für am häufigsten verwendete Programme sind diese Beschreibungen schon fertig kopiert erhältlich.

VERFÜGBARKEIT

CYBER_74:

Alle Programme sind als SCOPE-Libraries auf Permanentfile abgespeichert.

An Steuerkarten sind die folgenden notwendig:

ATTACH,IMSLIB.

ATTACH,NAGLIB.

LDSET,LIB=IMSLIB.

LDSET,LIB=NAGLIB.

LGO. oder sonstiger Loaderaufruf LGO.

CYBER_73:

Ab Anfang 1976 wird die FORTRAN-Version der NAG-Bibliothek auch an der CYBER 73 zur Verfügung stehen. Beschreibungen der Programme (nicht die FORTRAN-Quellenprogramme) können bereits jetzt im Programmberatungszimmer eingesehen werden.

Dieter KÖBERL

Rudolf WELSER



M P O S

MPOS (= Multi Purpose Optimization System), ein Programmsystem für Probleme der linearen, ganzzahligen und quadratischen Programmierung, kann seit kurzem an der CYBER 73 verwendet werden. Es wurde von Mitgliedern des Vogelback Computing Center's (Northwestern University, Illinois) für Benutzer an Universitäten mit kleinen bis mittelgroßen Problemen und ohne tiefere Kenntnisse in der Mathematischen Programmierung konzipiert und zeichnet sich durch eine sehr einfache Art der Eingabe aus.

Die Lösung eines typischen Problems erfordert die Angaben des zu verwendenden Algorithmus (acht Algorithmen stehen zur Verfügung), die Angabe der Namen der Variablen, der Zielfunktion und der Nebenbedingungen und schließlich ein OPTIMIZE-Command. Die Obergrenzen der Anzahl der Variablen und jener Nebenbedingungen sind:

lineare Programmierung: 90 Variable, 128 Nebenbedingungen
quadratische Programmierung: 72 Variable, 66 Nebenbedingungen.

Eine genaue Beschreibung der Eingabedaten und des Aufrufs ist im MPOS - User's Guide (Programmberatung) angegeben.

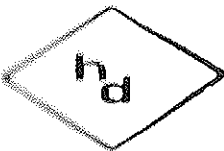
Dieter KÖBERL

SELCOM - VERSION 2

Seit kurzem steht Interessenten eine vorläufige Testversion der erweiterten Fassung des Programmsystems SELCOM auf der CYBER 73 zur Verfügung. SELCOM ist ein leistungsfähiges System zur Auswertung von Fragebögen und zur Häufigkeitsanalyse von Daten. Die neue Version 2 dieses Systems ist eine beträchtlich erweiterte und verbesserte Fassung des alten SELCOM, das schon seit geraumer Zeit von vielen Benutzern verwendet wird. Über die zusätzlichen Möglichkeiten der Version 2 gibt ein dreitägiges Seminar (12.-14.01.1976, vormittags) am IRZ-UNI Auskunft, das umso mehr allen potentiellen Benutzern empfohlen wird, als es zur Zeit noch keine ausreichenden Manuals gibt. Für Informationen stehe ich gerne zur Verfügung.

Peter RASTL

DEZEMBER 1975



F E H L E R

CPU-HARDWAREFEHLER CYBER 73

Der Hardwarefehler "Bitverlust in der CPU" wurde inzwischen vom Interfakultären Rechenzentrum lokalisiert und von den CDC-Technikern behoben. Die Fehlersuche gestaltete sich deswegen so schwierig, weil die Hardware-Testprogramme der CDC den Fehler nicht aufzeigten und das Interfakultäre Rechenzentrum erst eigene Hardware-Testprogramme entwickeln mußte.

Wir bitten die Benützer, Ergebnisse von Programmen, die in der Zeit vom 13. Oktober bis 03. November 1975 ermittelt wurden, die keine eindeutig erklärbaren Fehler aufweisen, durchzusehen bzw. erforderlichenfalls nochmals zu rechnen.

Johann PECLINOVSKY

FEHLER MIT DEM UNTERPROGRAMM WRITMS

Mit Einführung von SCOPE 3.4.3 trat bei der Routine WRITMS folgende Diskrepanz zum FTN-Manual auf:

Beim Aufruf

CALL WRITMS (u,fwa,n, k,r,s) siehe FTN Ref. Man III-7-2
darf der Parameter r, der angibt, ob geänderte Records die ursprüngliche Information überschreiben sollen oder nicht, nicht weggelassen werden, falls der Default-Wert 0 gewünscht wird. Dies führte bei einigen Jobs dazu, daß beim darauf folgenden READMS nichts gelesen wurde, d.h. der Zielbereich des Kernspeichers nicht überschrieben wurde.

Willy WEISZ



DEZEMBER 1975

AUSGABE AM DATENSTATIONSDRUCKER DER CYBER 73

Da die Datenstation an der CYBER 73 nicht über INTERCOM angeschlossen ist, müssen einige Punkte beachtet werden, wenn der OUTPUT eines Jobs, der nicht an der CYBER 73-Datenstation eingelesen wurde, (zB. Eingabe über Terminal) dennoch am Drucker dieser Datenstation ausgegeben werden soll. In diesem Fall ist das ROUTE-Kommando erforderlich. (siehe HD Nr.12-MAI 75)

Beispiel:

Ein am Intercom-Terminal im Editor erstelltes Programm, das am Datenstationsdrucker der CYBER 73 ausgedruckt werden soll, muß folgendermaßen aussehen:

```
JOB, ....  
ACCOUNT, ... .  
ROUTE(OUTPUT,DEF,FC=LL)  
.  
.  
.
```

Dieser am Terminal erstellte Job wird mit dem Befehl

```
ROUTE(lfn,DC=IN,TID=C)
```

in die INPUT-Queue dirigiert. (lfn ist die Datei, auf die der EDIT-File kopiert wurde.) (save ...)

Willy WEISZ
Wolfgang ALTFAHRT

BESCHREIBUNG DES UNTERPROGRAMMES P L A K A T

An der CYBER 73 ist in die FORTRAN-Bibliothek ein Unterprogramm neu aufgenommen worden, welches dazu dient, bis zu 10 Zeichen Text in JANUS-ähnlicher Schriftform auszugeben. Es können alle DISPLAY-Code Zeichen außer:

: < > @ \ ^ ;

ausgegeben werden.

Die Zeichenhöhe beträgt 12 Zeilen (+2 Leerzeilen, die vorher ausgegeben werden). Die Zeichenbreite beträgt 10 Schreibstellen (+2 Leerstellen, die vor jedem Zeichen ausgegeben werden). In eine Zeile können maximal 10 Zeichen ausgegeben werden.

Nach der Ausgabe erfolgt kein Zeilenvorschub.

Aufruf:

```
CALL PLAKAT (text)
           oder
CALL PLAKAT (text,unit)
```

dabei bedeutet:

text ein Wort mit den 10 Ausgabezeichen (als Hollerith-Literalkonstante oder Variable) für eine Zeile.

unit die Nummer (bzw. den Namen im L-Format) der Ausgabe-datei, falls die Ausgabe nicht auf das File OUTPUT erfolgen soll. Das entsprechende File muß natürlich in der PROGRAM-Karte des Hauptprogramms definiert sein.

PLAKAT kann nur in Programmen aufgerufen werden, die unter FTN kompiliert worden sind.

Wolfgang ALTFAHRT
Erwin HALPERN



<p>שנת 1975</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1976</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1977</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1978</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1979</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1980</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1981</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1982</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1983</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1984</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1985</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>	<p>שנת 1986</p> <p>1 ינואר</p> <p>2 פברואר</p> <p>3 מרץ</p> <p>4 אפריל</p> <p>5 מאי</p> <p>6 יוני</p> <p>7 יולי</p> <p>8 אוגוסט</p> <p>9 ספטמבר</p> <p>10 אוקטובר</p> <p>11 נובמבר</p> <p>12 דצמבר</p>
--	--	--	--	--	--	--	--	--	--	--	--



S P S S - NACHRICHTEN

Mit dieser Folge des HD ist SPSS (Statistical Package for the Social Sciences) nunmehr auch am IRZ-UNI verfügbar. Es steht somit SPSS Version 6.0 an beiden Rechenanlagen CYBER 73 und CYBER 74 zur Verfügung.

Der Aufruf erfolgt, analog zur schon bisher am IRZ-TU verwendeten Steuerkartensequenz, so:

```

Jobkarte          (ohne CM-Angabe, ohne Jobklasse)
Accountkarte
ATTACH,SPSS.
SPSS.            (oder: SPSS, parlist.)
eor
                SPSS-Programm
eof

```

Grundlage für die Verwendung von SPSS ist die Neuauflage des SPSS-Handbuches:

N.H. Nie, C.H. Hull u.a.
 SPSS - Statistical Package for the Social Sciences
 Second Edition, 1975, 675 Seiten
 McGraw-Hill Verlag

In den Programmberatungen des IRZ-TU ist ein kurzes deutsches Manual über die Verwendung von SPSS 6.0 erhältlich.

ACHTUNG! Zusätzlich zu den im englischen Handbuch beschriebenen Auswertungsprozeduren sind an CDC-Installationen von SPSS 6.0 die vier folgenden verwendbar:

NONPAR TESTS	parameterfreie statistische Tests
TETRACHORIC	tetrachorische Korrelation
RELIABILITY	Reliabilitäten von Skalen etc.
NONLINEAR	nicht-lineare Modelle

Diese Programme werden in späteren Nummern des HD noch genau beschrieben werden.

Rudolf WYTEK

H Y P N C H

Übertragen von Daten von der CDC CYBER 74 auf die Hybridrechenanlage EAI-PACER 600.

Da eine echte Datenverbindung zwischen Digital- und Hybridrechenanlage noch fehlt, wurde die folgende Möglichkeit zur Übertragung kleinerer Datenmengen mit Hilfe von Binärkarten geschaffen. Es ist unter anderem dadurch möglich, den an die EAI-PACER angeschlossenen Plotter zu verwenden.

An der CYBER 74 steht ein FORTRAN-Unterprogramm HYPNCH zur Verfügung, das Gleitkommazahlen eines Datenfeldes in das EAI-Datenformat umwandelt und die Werte mit der höchsten an der EAI verwendbaren Genauigkeit abstanzt.

An der Hybridanlage gibt es ein entsprechendes Einleseprogramm HYREAD. Beschreibungen sind in der Programmberatung erhältlich.

Rudolf WELSER

TTY — INTERCOM

Da es bei manchen schreibenden Terminals immer wieder zu Problemen kommt, indem das erste Zeichen nach einem Carriage Return verschluckt wird, möchte ich folgendes feststellen:

Die Driveroutine im INTERCOM sendet nach einem Carriage Return drei ASCII Nullcharacter, um dem Wagen am Terminal den Rücklauf zu ermöglichen. Diese Zeit reicht im allgemeinen aus. Beim Ankauf eines Terminals sollte jedoch diese Tatsache berücksichtigt werden, da von einigen Firmen sechs ASCII Nullcharacter verlangt werden. Dies führt letztlich zu dem Verschwinden des ersten Zeichens.

Herbert KRAUTSCHNEIDER



TELEFON - DIENST

CYBER 73

Ab sofort steht unseren Benutzern ein automatischer Telefondienst "rund um die Uhr" zur Verfügung:

1. Über die Telefon-Seriennummer 43-61-11*12 können mit Hilfe eines Anrufbeantworters Wünsche an das EDV-Zentrum deponiert werden. Die Gesprächsaufnahme wird regelmäßig abgehört und die zuständigen Mitarbeiter werden sofort verständigt. Auch bei Nicht-melden einer direkt angewählten Klappe wird nach 8mal Läuten in die Kanzlei (16) oder auf die Klappe 12 automatisch umgeschaltet.
2. Die Datentelefonleitungen der CYBER 73 sind ebenfalls mit einem Anrufbeantworter bestückt. Ist kein Terminal-Anschluß mehr frei oder kein INTERCOM-Betrieb, so meldet sich der Anrufbeantworter und gibt Näheres bekannt. (Tel. Nr. 43-89-71).

Gerhard NÖHRER

CYBER 74

Seit Jänner 75 steht unseren Kunden ein automatischer Telefonanrufbeantworter zur Verfügung. Das Band wird nur bei Hard- oder Software Störungen - die den ganzen Betrieb beeinflussen - besprochen. Es werden die genauen Angaben über die Art und mögliche Zeitdauer der Störung bekanntgegeben. Wir bitten um Verständnis, wenn die Zeitdauer der Störung nicht genau angegeben werden kann und der Betriebsbeginn erneut verschoben werden muß.

Während des normalen Betriebes steht dieses Telefon N U R bei technischen Schwierigkeiten den Kunden zur Verfügung. (Keine Programmberatung!!!).

Zur Erinnerung geben wir noch einmal die Telefonnummer und die Anrufzeiten bekannt:

65-43-50 Montag - Freitag von 09.00 - 22.00 Uhr

Die genaue Anrufzeit erfahren Sie auch aus dem "SYSBULL".

Anton ROZA



TAPEDUMP-UTILITY

1. BESCHREIBUNG DER ROUTINE-TPDUMP

Die TAPEDUMP-Routine ermöglicht Dumps von Datenbeständen, die sich auf Magnetbändern befinden. Die Verwendung von TPDUMP ist vor allem empfehlenswert, wenn keine ausreichenden Informationen über die Magnetband-Daten vorliegen. Es besteht die Möglichkeit des Dumps ab einem bestimmten Record in einem bestimmten File, die Beendigung des Dumps nach einem bestimmten File oder einer bestimmten Anzahl von Records sowie die Möglichkeit eines SHORT-Dumps zur Erfassung der Struktur der Banddaten. Der Dump erfolgt wahlweise hexadezimal oder oktal. Es besteht die Möglichkeit einer Codeumwandlung für ASCII, EBCDIC, BCD und DISPLAY. Maximale Recordlänge ist 20000 Byte a 6 Bit oder 15000 Byte a 8 Bit.

2. AUFRUF

Der Dump erfolgt von dem logischen File DUMTAPE. Der Steuerkartenrecord sollte eine VSN-Karte enthalten:

VSN,DUMTAPE=vsn.

Der Aufruf erfolgt durch die Steuerkarte:

TPDUMP,parameterliste.

Das Programm steuert folgende REQUEST-Karte bei:

REQUEST,DUMTAPE,NT,PE,NS,L,NORING,VSN=vsn.

2.1 MÖGLICHE PARAMETER

ALL Der Dump beginnt bei Record 1 im 1. File und endet nach dem 10000. File.

FILE=n Das Startfile wird durch n spezifiziert. Der Dump endet nach diesem File. Folgen zwei EOF-Marken direkt aufeinander, so wird das leere File dazwischen ebenfalls mitgezählt.



- REC=n Der Startrecord wird durch n spezifiziert.
Hat das File weniger Records als angegeben,
wird nicht gedumt.
- END=n Der Parameter spezifiziert das File, nachdem
der Dump abgebrochen werden soll.
- ASCII Es wird ASCII-Code angenommen und im literal-
dump dementsprechend gelistet.
- EBCDIC Es wird EBCDIC-Code angenommen und im literal-
dump dementsprechend gelistet.
- BCD Es wird BCD-Code angenommen und im literal-
dump dementsprechend gelistet.
- DISPLAY Es wird DISPLAY-Code angenommen und im literal-
dump dementsprechend gelistet. Außerdem erfolgt
der Dump oktal.
- SHORT SHORT=n bei mehreren Records gleicher Länge
werden nur n gelistet, die restlichen werden nur mehr
gezählt und als Summe ausgegeben. (Defaultwert
für n beträgt 3); n=0 liefert nur die Struktur
der Daten, aber keine Daten.
- LIST=n LIST=n gibt die Anzahl der Records an, die ge-
listet werden sollen. (Defaultwert für n be-
trägt 3).
- MAX Am Fileende wird die Länge des längsten Records
des Files angegeben.
- NOREQ Das Programm verlangt das Band nicht selbst;
d.h. Der Steuerkartenrecord muß eine REQUEST-
Karte enthalten. Folgende Parameter sollen
unbedingt angegeben werden:
REQUEST,DUMTAPE,NORING,NS,VSN=vsn.
- LET Dieser Parameter bewirkt, daß das gedumpte
Band am Jobende nicht ausgespannt wird, wo-
durch leichter zwei dumps als einzelne Jobs
aneinandergesetzt werden können. Zu Beginn
jedes Dumps befindet sich das Band am Anfang.



- PL=n Die Zahl n gibt die Anzahl der Zeilen an, die maximal ausgegeben werden können. (Defaultwert für n beträgt 5000).
- L=lfm Der Dump wird auf dem File lfm gelistet. (Defaultannahme ist OUTPUT).
- VSN=vsf vsf ist die volume serial number des Bandes, das gedumt werden soll. (Defaultwert ist 000000).

2.2 DEFAULT-PARAMETERWERTE

Der Aufruf:

TPDUMP.

ist äquivalent zum Aufruf:

TPDUMP,FILE=1,REC=1,END=1,ASCII,L=OUTPUT,PL=5000,VSN=0.

3. BEISPIELE

3.1 VOM BAND MIT DER VSN=TESTBD

SOLL DAS 2. UND 3. FILE IM EBCDIC-CODE GEDUMPT WERDEN

JOB,NT1, ...
ACCOUNT, ...
VSN,DUMTAPE=TESTBD.
TPDUMP,FILE=2,END=3,EBCDIC.
6/7/8/9

3.2 DIE FILESTRUKTUR DES OBIGEN BANDES

SOLL ERUIERT WERDEN

JOB,NT1, ...
ACCOUNT, ...
VSN,DUMTAPE=TESTBD.
TPDUMP,ALL,SHORT=0.

3.3 VOM 4. FILE DES OBIGEN BANDES

SOLLEN RECORD 477-481 IM DISPLAYCODE GEDUMPT WERDEN

TPDUMP,FILE=4,DISPLAY,REC=477,LIST=5.



3.4 EIN SIEBENSPIRBAND MIT VSN=TSEVEN SOLL GEDUMPT WERDEN

```
JOB,MT1, ...  
ACCOUNT, ...  
REQUEST,DUMTAPE,NORING,NS,HI,MT,L,VSN=TSEVEN.  
TPDUMP,NOREQ.
```

4. HINWEISE

4.1

SCOPE-Standardbänder können nur bis EOF gedumpt werden. Wegen eines Mangels der Systemsoftware endet das Programm abnormal, d.h. etwaige Steuerkarten nach dem Aufruf von TPDUMP werden in diesem Fall nicht mehr verarbeitet.

4.2

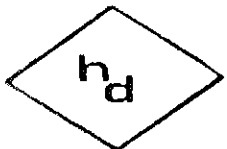
Falls Sie genau wissen, daß der längste Record, der sich auf dem Band befindet, 5120 Byte a 6 Bit oder 3840 Byte a 8 Bit nicht überschreitet, sollten Sie eine eigene REQUEST-Karte folgender Form angeben:

```
REQUEST,DUMTAPE,NORING,NS,NT,PE,S,VSN=vsn.
```

Diese Vorgangsweise beschleunigt das Programm um ein Mehrfaches!

ACHTUNG: Den NOREQ-PARAMETER in diesem Fall nicht vergessen!

Wolfgang ALTFAHRT



DEZEMBER 1975

