

inforum

INFORMATIONSFORUM des Rechenzentrums der Universität Münster

Jahrgang 2, Nr. 1

Januar 1978

Inhalt

Beschaffung einer neuen Rechanlage	1
Änderung der JCL-Prozeduren	3
Personalia	3
Magnetbandbetrieb	4
In eigener Sache	5
Benutzerhandbuch	6
Software Informationen	6
Leserforum	6
Lehrveranstaltungen	7
Der alte und der neue Plotter	8
Geänderte Plotter-Grundsoftware	10
Ein neues Sortier-/Misch-Programm	13
Literarische und linguistische Datenverarbeitung	15
NAG-Bibliothek	19
Struktogramme	20
FORTRAN-Standards	23
Aufruf von FORTRAN-Programmen in PL/I-Programmen	25
Sequentielle Verarbeitung von Dateien	27
Verbrauchsdaten 1977	32

Beschaffung einer neuen Rechenanlage

P. Janßen

Am 27.12.77 traf die schriftliche Nachricht ein, daß der Minister für Wissenschaft und Forschung des Landes NRW den letzten Antrag auf Erweiterung und teilweise Erneuerung der technischen Anlagen des Rechenzentrums vom Mai 1977 nach einer auferlegten reduzierenden Abänderung genehmigt hat. Damit haben jahrelange Bemühungen, die Rechnerkapazität im Rechenzentrum dem Bedarf entsprechend zu erhöhen, nun endlich Erfolg gezeitigt: die Anfänge dieser Bemühungen gehen bereits auf das Jahr 1973 zurück. Inzwischen ist der Kapazitätsbedarf, wie jeder Benutzer weiß, auf ein Vielfaches der vorhandenen Rechnerleistung

gestiegen, die allein im Vergleich zur derzeitigen bundesdurchschnittlichen Ausstattung von Hochschulrechenzentren der Studentenzahl nach nur etwa 1/20 der Leistung beträgt, die an der Universität vorhanden sein müßte. Die neue Rechnerzentraleinheit IBM 3032 wird gegenüber der heutigen IBM /360-50 eine etwas mehr als 15-fache Geschwindigkeit besitzen. Die Installation ist im Juli dieses Jahres vorgesehen, wahrscheinlich unmittelbar nach Beginn der Semesterferien. Allerdings bereitet uns große Sorge, ob die dazu nötigen Umbauten in den Maschinenräumen, die bereits seit mehr als einem Jahr abzusehen und vom Rechenzentrum angefordert sind, bisher aber immer noch nicht in Angriff genommen wurden, rechtzeitig fertig sein werden.

Neben der neuen Rechnerzentraleinheit sind in dem genehmigten Beschaffungsvorhaben vor allem Erweiterungen an Magnetplattenspeicher und Geräte zur Datenfernverarbeitung enthalten. Nähere Einzelheiten über die sich nach diesen Erweiterungen ergebende Konfiguration der gesamten Rechanlage und über den vorgesehenen Rechenbetrieb werden demnächst durch Aushang oder Rundschreiben bekannt gegeben. Das Rechenzentrum legt Benutzern mit sehr hohem Rechenbedarf nahe, noch vor Installation der neuen Rechanlage gemeinsam mit dem Rechenzentrum Vorbereitungen zu treffen, um den erheblichen Nachholbedarf an Rechen-Arbeiten nach Aufnahme des neuen Rechenbetriebs möglichst schnell abzubauen, soweit diese Arbeiten nach den mehrjährigen Verzögerungen noch nachgeholt werden können. Erfreulicherweise ist die vom Rechenzentrum im Beschaffungsantrag mit Nachdruck betonte und begründete Forderung, daß die Nachfolgeanlage mit der vorhandenen kompatibel sein müsse, auf allseitiges Verständnis gestoßen, so daß in dieser Hinsicht für den bisherigen und zukünftigen Benutzer keine Programmumstellungen erforderlich sein werden.

IMPRESSUM

Redaktion **INFORM**

W. Bosse	(Tel.: 490-2476)
J. Ebert	(Tel.: 490-2607)
R. Schmitt	(Tel.: 490-2475)
W. A. Slaby	(Tel.: 490-2473)

Satz: U. Kaiser
Druck: H. Mecke

Rechenzentrum der Universität
Roxeler Str. 50
4400 Münster

Auflage dieser Ausgabe: 500
Redaktionsschluß der nächsten Ausgabe: 31.3.1978

Änderung der JCL-Prozeduren

J. Ebert

Am Freitag, dem 27.1.1978, soll die Prozeduren-Bibliothek neu generiert werden. Dabei sind folgende Änderungen beabsichtigt:

1. Da der alte CIL-Plotter nicht mehr verfügbar ist, werden alle Prozeduren mit einem P im Suffix abgeschafft. Zum Plotten mit Hilfe der neuen Plot-Software (siehe H. Stenzel, Seite 10) auf dem neuen Plotter sind dann mit von den FORTRAN-Compilern und dem PL/I-Optimizing-Compiler (Präfixe: FORT, FORTH, PLO) übersetzten Programmen keine besonderen Prozeduren mehr nötig. Die bisher vorhandene Praxis, Plotprogramme durch Verwendung der normalen Prozeduren (Suffix ohne P) zu testen, wird dann nicht mehr fortgeführt werden können. Als Ersatz hierfür ist Plot-Preview anzusehen. Das Plotten mit dem PL/I-(F)-Compiler ist nicht mehr möglich, und wegen der geplanten Abschaffung dieses Compilers wird die notwendige Software hierfür auch nicht erstellt werden.
2. Für die Ausführung von FORTRAN-Unterprogrammen in PL/I-Optimizer-Programmen (siehe U. Ebert, Seite 25) wird ein entsprechender Satz von Prozeduren zur Verfügung gestellt. Zu ihrem Aufruf ist an den Namen der entsprechenden Prozedur ein F (für "FORTRAN") anzuhängen, d.h. es wird die Prozeduren PLOCFF, PLOCLEF, PLOCFL, PLOCMF, PLOEF, PLOLEF, PLOLF, PLOMF und PLOXF geben.
3. Für ALGOL 68 wird ein vollständiger Satz von Prozeduren eingeführt. Hierfür ist das Präfix A68 vorgesehen. Die Prozeduren sind den bei uns üblichen Konventionen angepaßt. Insbesondere lauten die Stepnamen COMPILE, LINKAGE und EXECUTE. Damit stehen dann die Prozeduren A68C, A68CE, A68CL, A68CLE, A68CM, A68E, A68L, A68LE, A68M, A68D und A68X zur Verfügung.
4. In den PL/I-Optimizer-Sort-Prozeduren (Präfix: PLO; Suffix mit S) wird als Sortier-Software das Programmpaket CA-SORT zur Verfügung gestellt (siehe H. Meyer, Seite 13).

Personalie

Nach 13-monatiger Tätigkeit bei uns kehrte Daniel Zwick Ende September 1977 wieder in die USA zurück, und Bernd Schulze verließ uns Ende November nach mehrjähriger Tätigkeit als studentische Hilfskraft.

Als wissenschaftliche Mitarbeiter kamen am 1.11.77 Klaus Langer und am 1.12.77 Wolfgang Kaspar zu uns, und am 1.12.77 kehrte Klaus Dieter Thomalla nach anderthalbjähriger Beurlaubung zur Weiterbildung wieder als Operateur zu uns zurück.

Magnetbandbetrieb

W. Bosse

Nachdem die organisatorischen Vorbereitungen jetzt abgeschlossen worden sind, werden bezüglich der Magnetbänder ab sofort die folgenden Änderungen im Rechenbetrieb eingeführt, deren Ziel ein sowohl für den Benutzer als auch für den Operateur erleichtertes Umgang mit Magnetbändern ist.

/*JOBPARM-Karte:

Ab sofort entfällt die Bedingung, alle Magnetbänder, die in einem Job verwendet werden, auf der /*SETUP-Karte zu spezifizieren, wodurch der Job zunächst in den Hold-Status versetzt und erst nach einem Freigabe-Kommando des Operateurs bearbeitet wird. Jobs, die vom Rechenzentrum ausgegebene Magnetbänder benötigen, können jetzt im normalen Betrieb ohne diese Kennzeichnung eingelesen und abgewickelt werden.

Um aber die Bearbeitung der Jobs, die Magnetbänder verwenden, besser koordinieren zu können, muß auf der /*JOBPARM-Karte die für den Job benötigte Anzahl verschiedener Magnetbandgeräte (nicht Magnetbänder!) angegeben werden. Dadurch kann vermieden werden, daß sich Jobs aufgrund der begrenzten Anzahl von 6 vorhandenen Magnetbandgeräten gegenseitig behindern. Die /*JOBPARM-Karte muß direkt hinter der JOB-Karte liegen und hat - sofern nicht andere Parameter gleichzeitig gesetzt werden - die Gestalt:

```
/*JOBPARM TAPES=anzahl
```

anzahl steht dabei für eine vorzeichenlose ganze Zahl die den Bedarf an Magnetbandgeräten angibt.

(Eine ausführliche Beschreibung der /*JOBPARM-Karte ist im Benutzerhandbuch, Version 3.3, auf S.10 zu finden).

Die "Schanzeit" zum Eingewöhnen in den Gebrauch des Parameters TAPES der /*JOBPARM-Karte wird am 1. Februar 1978 ablaufen. Nach diesem Termin wird die Bearbeitung jedes Jobs, der mehr Magnetbandgeräte belegt, als auf der /*JOBPARM-Karte angegeben worden sind, sofort abgebrochen.

/*SETUP-Karte:

Die Verwendung der /*SETUP-Karte ist immer noch für solche Jobs obligatorisch, die Nicht-Standard-Bänder (z.B. No-Label-Bänder, private Bänder) benötigen. Dadurch wird auch der Ausnahmecharakter derartiger Bearbeitungen deutlich.

Diese speziellen Magnetbänder sollen durch eine Bandnummer (Volume Serial Number; bei No-Label-Bändern: als externes Label) gekennzeichnet sein, die nach folgendem Muster aufgebaut ist:

ppiii

Die ersten drei Zeichen (ppp) enthalten private Information und die letzten drei Zeichen (iii) sind für die vom Rechenzentrum vergebenen Buchstaben zur Kennzeichnung des Instituts (vgl. die Buchstaben in der Verrechnungsnummer) vorgesehen.

Band-Archiv:

Um die dem Benutzer zur Verfügung stehende Magnetbandkapazität zu vergrößern, wird mit der Einführung eines Band-Archivs die Möglichkeit geboten, Magnetband-Dateien, die zur Zeit nicht im laufenden Betrieb verwendet werden, auszulagern und damit Speicherplatz auf den vom Rechenzentrum zur Verfügung gestellten Magnetbändern zu gewinnen. Dabei sei noch auf die Wahl der Dateinamen (dsname) hingewiesen: sie müssen durch die Verrechnungsnummer des Benutzers qualifiziert sein (z.B. UUU01.DATGEN01).

Dateien, die in das Band-Archiv aufgenommen werden sollen, müssen H. Goorkotte (Zi. 112, Tel. 490-2672) genannt werden. Wenn eine archivierte Magnetband-Datei wieder benötigt wird, so kann der Benutzer diese natürlich auch wieder auf eines seiner Magnetbänder kopieren lassen.

In eigener Sache:

In seiner nunmehr einjährigen Existenz hat inforum erfreulich viele Abonnenten gefunden. Leider bringt die wachsende Zahl auch ein wachsendes Problem, nämlich die Portokosten. Wir sind nach wie vor bestrebt, inforum auf Anforderung kostenlos zuzusenden, bitten aber um Ihr Verständnis und Ihre Unterstützung bei unseren Bemühungen, unsere Kosten gering zu halten.

Wenn Sie als Universitätsangehöriger inforum bestellen, geben Sie bitte die Adresse des Instituts an, bei dem Sie beschäftigt oder erreichbar sind. Zustellungen an Ihre Privatadresse können wir in diesem Fall nicht vornehmen. Aber auch als Angehöriger von Institutionen, die nicht durch die Universitätspost beliefert werden, helfen Sie uns, wenn Sie Ihre Dienstadresse angeben, so daß wir für den Fall, daß für Ihre Dienststelle mehrere Bestellungen vorliegen, die Exemplare gesammelt versenden können.

Wir freuen uns über Ihr Interesse an inforum, auch wenn Sie dem Benutzerkreis des Rechenzentrums nicht mehr angehören. Sollte Ihr Interesse aber nach einiger Zeit nachlassen, zögern Sie bitte nicht, uns dies mitzuteilen. Ein ungelesenes Exemplar in unserer Ablage kostet uns weniger als in Ihrer.

Benutzerhandbuch

W. Bosse

Eine neue Version des Benutzerhandbuchs - "Hinweise zur Benutzung der Rechanlage, Version 3.3" - ist erschienen und kann im Sekretariat des Rechenzentrums bei Frau E. Geisler (Zi. 101) zum Preis von 4,- DM erworben werden.

Software-Informationen

W. Bosse

Die Reihe der Software-Informationen des Rechenzentrums wurde im Juli 1975 begründet, um den Benutzern der Rechanlage Detail-Informationen zum Einsatz verschiedener Software-Komponenten zu eröffnen. Dies betrifft einerseits die Herausgabe von Beschreibungen, die in Verbindung mit gekauften bzw. gemieteten Programmsystemen zum freien Nachdruck geliefert werden, und schließt andererseits auch Neuauflagen von in der Schriftenreihe des Rechenzentrums erschienenen Dokumentationen vorhandener Software sowie technische Beschreibungen ein. Die Software-Informationen können als Ergänzung zum Benutzerhandbuch angesehen werden.

Inzwischen sind in dieser Reihe die nachfolgend angegebenen Titel erschienen; diese können im Sekretariat des Rechenzentrums bei Frau E. Geisler (Zi. 101) zum angegebenen Selbstkostenpreis erworben werden.

Zusammenstellung der Software-Informationen des Rechenzentrums

- Nr. 1 - The FORMAT Manual, Release 5
von Gerald M. Berns; Stanford, Juli 1971
erschienen im Juli 1975; 4,- DM (64 Seiten)
- Nr. 2 - Interaktive graphische Datenverarbeitung mit dem
Sichtgerät TEKTRONIX 4015
von Detlef Steinhausen, Horst Stenzel, Eberhard Sturm
erschienen im November 1977; 3,- DM (30 Seiten)
- Nr. 3 - Plotter-Software - Erweiterte Anwendungssoftware
von Horst Stenzel, Bernd Schulze (verbesserte und
erweiterte Neuauflage von Nr. 12 der Schriftenreihe)
erschienen im September 1976; 3,- DM (41 Seiten)
- Nr. 4 - User's Guide to PL/C, The Cornell Compiler for PL/I,
Release 7.5
von PL/C Project; Ithaca, April 1976
erschienen im Mai 1977; 3,- DM (44 Seiten)

Leserforum

Das Leserforum bleibt auch weiterhin eine ständige Rubrik!

Lehrveranstaltungen

In der vorlesungsfreien Zeit am Ende des WS 1977/78 werden vom Rechenzentrum wieder einige ganztägige Kurse durchgeführt, in denen Stoffvermittlung und Übungen integriert sind. Diese Veranstaltungen sollen durch intensive Betreuung eigene Programmierübungen der Teilnehmer fördern. Das bedingt eine Begrenzung der Teilnehmerzahl der einzelnen Veranstaltungen. Interessenten werden deshalb gebeten, sich in der Zeit

vom 1. bis 17.2.1978

bei H. Mecke im Dispatch des Rechenzentrums (Zi. 07) in die Anmelde listen einzutragen.

Für alle nachfolgend beschriebenen Lehrveranstaltungen sind Vorkenntnisse in der Programmierung nicht erforderlich.

27.2. - 17.3.1978: FORTRAN IV

FORTRAN IV ist eine einfache, aus verhältnismäßig wenigen Sprachelementen bestehende und schnell zu erlernende Programmiersprache, die sich zur Bearbeitung numerischer Probleme insbesondere aus dem naturwissenschaftlichen Bereich eignet.

Dozenten: Eickenscheidt, Reichel, Mertz
Vorl.Nr.: 220016, 220020, 220035
Hörsäle: M4, M5, M6
Beginn: 27.2.1978 um 9 Uhr c.t. im M4

20.2. - 7.3.1978: ALGOL W

Der angenehm systematische Aufbau der Programmiersprache ALGOL W erleichtert das Erlernen von Methoden und Techniken des strukturierten Programmierens; die Sprache eignet sich neben der Bearbeitung numerischer Probleme auch für Fragestellungen der Informatik.

Dozent: Bestehorn
Vorl.Nr.: 220073
Hörsaal: Seminarraum des Rechenzentrums
Beginn: 20.2.1978 um 9 Uhr c.t.

6.3. - 23.3.1978: PL/I

PL/I ist eine Programmiersprache mit vielseitigen Anwendungsmöglichkeiten. Sie gestattet sowohl die Behandlung numerischer Probleme als auch die Manipulation von Texten und Datenstrukturen. Umfangreiche Ein- und Ausgabemöglichkeiten erlauben außerdem eine effiziente Bearbeitung großer Datenmengen.

Dozenten: Neukäter, Slaby, Stenzel
Vorl.Nr.: 220040, 220054, 220069
Hörsäle: M1, M2, M3
Beginn: 6.3.1978 um 9 Uhr c.t. im M1

**3.4. - 14.4.1978: Statistische Methoden und Datenanalyse
mit dem SPSS**

Das SPSS - Statistical Package for the Social Sciences - stellt eine Programmsammlung der gebräuchlichen univariaten und multivariaten statistischen Methoden dar. Für die Einführung in die Handhabung dieses Systems und in die zugrundeliegenden Verfahren sind Kenntnisse auf dem Gebiet der Statistik nützlich.

Dozenten: Steinhausen, Zörkendörfer
Vorl.Nr.: 220088
Hörsaal: M5
Beginn: 3.4.1978 um 9 Uhr c.t.

Die Teilnehmer der Kurse werden gebeten, diese im SS 1978 zu belegen. Dies ist neben der bestandenen Abschlußklausur Voraussetzung für die Aushändigung eines Scheines über die erfolgreiche Teilnahme.

Eine Beratung zum Lehrangebot des Rechenzentrums und insbesondere zu den hier angekündigten Veranstaltungen erfolgt in der Zeit

vom 13. bis 24.2.1978

durch W. Bosse (Tel.: 490-2476).

Der alte und der neue Plotter

H. Stenzel

Seit Ende letzten Jahres ist der CIL-Plotter des Rechenzentrums wegen defekter Schrittmotoren nicht mehr einsatzbereit. Inzwischen hat sich herausgestellt, daß eine Reparatur unangemessen teuer würde (ca. 20 % des Preises, den ein neuer Plotter mit ähnlichen Spezifikationen kosten würde) und auch im Hinblick auf die absehbare Einsatzbereitschaft eines neuen Plotters nicht mehr lohnt.

Acht Jahre lang war der Plotter im Rechenzentrum installiert. In dieser Zeit hat er, abgesehen von etlichen Ausfallzeiten, durchschnittlich 250 m Papier pro Monat mit Zeichnungen versehen; das entspricht einer Fläche von mehr als 2000 Quadratmetern. Die Plotterbenutzung ist ständig gestiegen: allein 1977 wurde der Plotter von ca. 5000 Jobs in Anspruch genommen, der Papierverbrauch ist auf mehr als 450 Meter pro Monat angewachsen und die beiden Zeichenfedern haben zusammen pro Monat zwischen 15 und 20 km "zurückgelegt", bei einer Einschaltzeit von im Mittel täglich 7 Stunden.

Gestatten Sie mir noch etwas mehr "Statistik": Wenn alles, was durch den Plotter gelaufen ist, aufgehoben worden wäre, und, auf DIN A4-Größe geschnitten, übereinandergestapelt würde, wäre der Turm fast so hoch wie das neben dem Rechenzentrum stehende Mathematikhochhaus. Und die Stift- und Papierbewegungen, die der Plotter in seinen acht Jahren gemacht hat, hätten ausgereicht,

um mit der dicksten bei uns verfügbaren Strichstärke in natürlicher Größe ein Abbild eines der Randstreifen der Autobahn zwischen Münster-Nord und Münster-Süd zu zeichnen, oder um auf jedes der eben erwähnten DIN A4-Blätter einen Klecks von ca. 5 cm Durchmesser zu malen.

Die Beschaffung eines neuen Plotters hat sich mit dem Zusammenbruch des alten überschnitten. Ich nehme an, daß nach vor Beginn des Sommersemesters der Betrieb auf dem neuen Gerät voll wieder aufgenommen werden kann.

Zur Vorstellung des neuen Plotters hier eine Gegenüberstellung einiger Daten :

	alt	neu
Typ	CIL 6011	BENSON 1222
Ansteuerung über	Magnetbandstation 5009	V24-Interface und F8 Mikroprozessor
Arbeitsweise	inkrementell	Interpolierend
kleinste Schrittgröße	0.1 mm	0.1 mm
Genauigkeit	< 0.1 mm	< 0.025 mm
max. Geschwindigkeit	6 cm/sec	15 cm/sec
Zeichenwerkzeuge	2	3
Standardpapierbreite	77 cm	32 cm (73 cm möglich)
Herkunftsland	GB	F

Für die Plotterbenutzer wird der hauptsächlichste Unterschied zwischen dem BENSON- und dem CIL-Gerät einerseits in der gleichzeitigen Verfügbarkeit von 3 gegenüber 2 Zeichenstiften und andererseits in dem Übergang von inkrementeller zu interpolierender Arbeitsweise deutlich werden. Mit dem neuen Plotter werden gerade Liniensegmente beliebiger Winkellage fast platt durchgezogen ("interpoliert"), so daß zu erwarten ist, daß trotz der gleich gebliebenen kleinsten ansprechbaren Schrittgröße die Qualität der Zeichnungen steigt.

Das Rechenzentrum versucht darüberhinaus, eine wegen vermutlich zunehmender Wahrnehmung der Möglichkeiten der graphischen Ausgabe zu erwartende Steigerung der Papierkosten abzufangen. Dies soll zum einen dadurch erreicht werden, daß das Testen von Programmen bis zur Produktionsreife ohne den Einsatz des Plotters mit dem Tektronix-Bildschirmgerät und Plot-Preview durchgeführt wird. Es wird auch gewünscht, daß Bilder, an die keine hohen Anforderungen in Bezug auf Größe und Qualität gestellt werden, mit der Hardcopy-Einrichtung des Bildschirmgeräts erstellt werden. Außerdem soll eine bessere Ausnutzung des teuren Plotterpapiers dadurch erreicht werden, daß alle Bilder, die mit einer Papierbreite von ca. 29 cm auskommen, auf schmales Papier gezeichnet und von den andern abgearbeitet werden.

Geänderte Plotter-Grundsoftware

H. Stenzel

Zusammen mit der in dieser Ausgabe von *Infocum* von J. Ebert angekündigten Umstellung der katalogisierten Prozeduren wird am 27.1.78 die Unterstützung des CIL-Plotters aufhören. Statt dessen werden zum gleichen Zeitpunkt Plot-Routinen in FORTRAN IV G und H sowie PL/I Optimizing Compiler Programmen verfügbar sein, die den bisher schon in WATFIV und PL/C benutzbaren Routinen für Plot-Preview entsprechen und die im Hinblick auf den neuen Plotter erweitert worden sind. Über die Veränderungen, die sich für die Benutzer der Plottersoftware aus verschiedenen Gründen ergeben, werde ich im Folgenden einige Angaben machen.

Die bislang im Rechenzentrum zur Verfügung stehende Plottersoftware war auf den alten Plotter zugeschnitten (z.B. maximale Ausdehnung einer Zeichnung, Zahl der Zeichenstifte). Die veränderte Hardware macht hier Änderungen nötig.

Da die Verfügbarkeit des PL/I (F) Compilers abläuft, wird für ihn die Entwicklung nicht fortgeführt. Stattdessen wurde mit den Routinen für den Optimizing Compiler ein neuer Anfang gemacht.

Das Tektronix-Bildschirmgerät zusammen mit Plot-Preview eröffnet die Möglichkeit, Bilder mit erheblich geringerem Aufwand als beim Plotter erforderlich sichtbar zu machen. Statt durch die katalogisierten Prozeduren zwischen Produktions- (z.B. FORTCEP) und Testläufen (z.B. FORTCEI) zu unterscheiden, um unvollständige oder fehlerhafte Bilder vom Plotter fern zu halten, erscheint es deshalb besser, soviel wie möglich an graphischer Ausgabe eines Programms bereits in der Testphase sichtbar zu machen.

Die Einschränkungen, die zum Abbruch der Bilderzeugung führen, sind andere geworden. Allgemein gilt, daß bei Erkennen eines Verstoßes durch die Plot-Routinen eine ausführliche Fehlermeldung erzeugt wird, das Bild aber immer noch mit Plot-Preview betrachtet werden kann (auch auf die Gefahr hin, daß es wenig aussagekräftig oder stark entstellt ist).

Wenn ein Bild aus sehr vielen Einzelstrichen besteht (mehr als 15000 Aufrufe von PLOT), kann es sein, daß es mit den neuen Routinen nicht mehr vollständig erzeugt werden kann, da seine Speicherplatzanforderungen zu groß werden. Die Folge ist PLOTFEHLER 3. Zur Vermeidung dieser Schwierigkeit ist bei der Umstellung alter Plotprogramme grundsätzlich zu prüfen, ob Zeichenketten durch PCHARS ausgegeben werden können und ob die Rasterung von Kurven vergrößert werden kann.

Es kann vorkommen, daß sich die CPU-Zeit durch die Umstellung stark ändert. Hier wird im allgemeinen eine Änderung zugunsten der neuen Routinen zu beobachten sein. Das gilt besonders für Bilder, die hauptsächlich aus längeren geraden

Strichen bestehen und die statt der älteren SYMBOL-Routinen PCHARS verwenden.

Ausgetestete und in Objekt- oder Ladebibliotheken stehende FORTRAN IV-Produktionsprogramme müssen zur Umstellung auf die neuen Routinen nicht neu übersetzt werden. Für Lademoduln genügt es, mit dem Linkage Editor die alten Routinen durch die neuen zu ersetzen. Die dafür nötigen Steuerkarten können bei der Beratung abgeholt werden.

Nach der Umstellung stehen die in der folgenden Tabelle aufgeführten Funktionen zur Erzeugung von Bildern auf dem Tektronix-Bildschirmgerät und, sobald einsatzbereit, auf dem neuen Plotter zur Verfügung.

	FORT G&H	PL/I QPT 1)	WATFIV 2)	PL/C 2)
PLOTS	x 3) 4)	x 4) 5)	x 4)	x 4)
PLOT	x	x	x	x
PLOTE	x 6)	x 6)	x	x
PLOTPV	x 7)	-	x 8)	- 9)
CHPEN	x 10)	x 10)	x 10)	x 10)
FACTOR, RPLOT, STATE, WHERE	x	x	x	x
PLOTF	na	- 11)	na	- 11)
CONDITION (MARGIN)	na	- 11)	na	- 11)
%INCLUDE PLOTDCI	na	- 12)	na	na
%INCLUDE PLOT, %INCLUDE PLOTPROC	na	x 12)	na	-
GPGNUM	x 13)	x 13)	x 13)	-
PCHARS	x	x	x	x
SYMBOL	x 14)	-	-	-
PLOT_STROKES, PLOT_TEXT, PLOT_SYM, TEXT_INIT, STROKES_INIT, STROKES_INIT, STROKES_SET, GREEK_SET	-	x 14)	-	-
BCNV	x	-	-	-
NUMBER	x	-	-	-
SCALE, PLOT_SCALE	x	x	-	-
AXIS, PLOT_AXIS	x	x	-	-
LINE, PLOT_LINE	x	x	-	-
CILRUM	x	-	-	-
AXIS_INIT, LINE_INIT	-	x	-	-
CREIS, PLOT_KREIS	x	x	-	-
PKxxxx-Routinen	x	- 15)	- 16)	- 16)

(x = ja, - = nein, na = nicht anwendbar)

Anmerkungen zur Tabelle

Für Details muß auf die "Beschreibung der Plottersoftware und der Anwendungssoftware" *Software-Information* Nr.3 verwiesen werden. Einige der in der obigen Tabelle erwähnten Funktionen dieser Software können erst in der nächsten Auflage dieser Schrift vollständig dokumentiert werden. Die folgenden Anmerkungen erläutern die Unterschiede und Änderungen in Bezug auf die jetzt vorgenommene Umstellung.

- 1) Die Deklarationen der Entries sind die gleichen wie für den PL/I-(F)-Compiler beschrieben.
- 2) Es ist nicht vorgesehen, mit den Test-Compilern WATFIV und PL/C erzeugte Bilder anders als über Plot-Preview auszugeben.
- 3) Angegebene Parameter werden ignoriert.
- 4) PLOTS muß immer ohne Parameter aufgerufen werden, denn für den Tektronix-Bildschirm erübrigt sich die Angabe von Zeichenstiften, und mit dem neuen Plotter werden immer die gleichen drei Zeichenstifte verfügbar sein.
- 5) Die Entries PLOTS0, PLOTS1 und PLOTS2 existieren nicht mehr und sind durch PLOTS ersetzt.
- 6) PLOTE signalisiert das Ende der Erzeugung eines Bildes und bewirkt seine Übernahme in die Plot-Spool-Datei. Es kann in einem Jobstep bis zu fünfmal erfolgreich aufgerufen werden.
- 7) Der Aufruf von PLOTPV ist aus Kompatibilitätsgründen zugelassen, er bleibt aber ohne jede Wirkung.
- 8) PLOTPV bewirkt die Übernahme eines Bildes in die Plot-Spool-Datei.
- 9) Die Übernahme eines Bildes in die Plot-Spool-Datei wird durch die Kontrollkarte *PLOT bzw. //PLOT bewirkt.
- 10) Hier sind die Werte 1, 2 und 3 zugelassen und bezeichnen auf dem Plotter Stifte in den Farben schwarz, rot oder blau und auf dem Tektronix-Bildschirmgerät ausgezogene, gepunktete oder gestrichelte Linien.
- 11) PL/I-Programmen stehen CONDITION (MARGIN) und ON FINISH CALL PLOTF nicht mehr zur Verfügung, da diese Bedingungen von der Systemsoftware anders gehandhabt werden müssen.
- 12) %INCLUDE PLOTDCI kopiert die für den PL/I-(F)-Compiler typischen Deklarationen und ist nicht mehr zu verwenden. An die Stelle sind zwei andere Member in RUMI.SOURCE getreten: Mit %INCLUDE PLOT werden die Routinen der Grundsoftware deklariert. Da der Optimizing Compiler die Möglichkeit dazu

ohne Verwendung der Macro-Phase bietet, kann dies für alle PL/I-Programme empfohlen werden.

%INCLUDE PLOTPROC kopiert die Preprocessor-Routine PLOT_ROUTINES, die wie bisher unter Einschaltung der Macro-Phase des Compilers (PARMC=M) die Deklarationen der übrigen Entries und Prozeduren erzeugt.

- 13) Nach CALL PLOTE (PLOTVP bei WATFIV) steht in einem externen Speicherbereich GPGNUM (COMMON /GPGNUM/ <*4-Variablen> bzw. DECL GPGNUM CHAR(4) EXTERNAL) eine 4 Byte lange Zeichenkette mit folgender Bedeutung: Wenn das Bild die Nummer nnn erhalten hat, steht dort '0nnn'. Wenn beim Übertragen in die Plot-Spool-Datei ein Fehler mit Code abb aufgetreten ist, ist der Inhalt 'Cabb'.
- 14) Die Ausgabe von Zeichenketten sollte vorzugsweise durch Aufruf von PCHARS erfolgen, da dadurch die in der modernen Hardware vorhandenen / Möglichkeiten genutzt werden können, um Speicherplatz und Ausführungszeit zu sparen.
- 15) Die PKxxxx-Routinen der Erweiterten Anwendungssoftware sollen demnächst auch in einer Version für den PL/I Optimizing Compiler fertiggestellt werden.
- 16) Es hat sich leider gezeigt, daß der in den Test-Compilern vorhandene Platz nicht ausreicht, um die PKxxxx-Routinen zur Verfügung zu stellen. Wir suchen noch nach einem Ausweg aus dieser Situation.

Im Interesse aller Benutzer der Plottersoftware bitte ich Sie dringend, mir alle beobachteten Abweichungen der Realität von der Beschreibung, Ihre Ergänzungs- und Verbesserungsvorschläge und mit den augenblicklichen Mitteln nicht lösbarer Probleme aus dem Gebiet der Bilderstellung mitzuteilen.

Ein neues Sortier-/Misch-Programm

H. Meyer

Mit "CA-SORT/OS" (Version 5.0) steht ab 27.1.78 ein Programm zur Verfügung, das über die Funktionen des bisher einsetzbaren SORT/MERGE-Programms (360S-SM-023) bei effizienterer Leistung (kürzere Rechenzeit, geringerer Platzbedarf für die Zwischenspeicherung) verfügt und das darüberhinaus neue Einsatzmöglichkeiten eröffnet wie die Änderung der Sortierfolge (ALTSEQ-Anweisung), die Sortierung ohne Arbeitsdateien (in-core-sort mit der Prozedur HSSORT) und den Aufruf von Sortierfunktionen in FORTRAN IV- und PL/I-Programmen analog zu normalen E/A-Arbeiten.

A) Sortieren in einem eigenen Jobstep (SORT, HSSORT)

Es stehen dann zwei katalogisierte Prozeduren zur Verfügung, von denen "SORT" wie bisher eingesetzt werden kann, während es HSSORT gestattet, ohne Verwendung von Zwischenspeichern allein im Hauptspeicher eine kleine Menge von Sätzen zu sortieren (ca. 1000 Karten in 120K Hauptspeicher) und dann sofort auszugeben:

```
//stepname EXEC proname[,PARM=Parameter für Optimierung]
//EXECUTE.SORTIN DD Angaben über die Eingabedatei
//EXECUTE.SORTOUT DD Angaben über die Ausgabedatei
//EXECUTE.SYSIN DD *
- Kontrollanweisungen -
```

mit den folgenden möglichen Kontrollanweisungen:

```
SORT FIELDS=((<Feld>,s[,<Feld>,s]...)[,FORMAT=f]
            [,FILSZ=m][,SKIPREC=n][,EQUALS|NOEQUALS])
ALTSEQ CODE=((<Folgedefinition>[,<Folgedefinition>]...))
[INCLUDE|OMIT] COND=(<Bedingung>)[,FORMAT=f]
```

wobei

```
<Feld> ::= p,l[,f]
<Folgedefinition> ::= xxuu
<Bedingung> ::= [<Vergleich>|<zusammengesetzte Bedingung>]
<Vergleich> ::= <Feld>,<Vergleichsoperator>,<Feld>[k]
<Vergleichsoperator> ::= {EQ|NE|GT|GE|LT|LE}
<zusammengesetzte Bedingung> ::= [( )<Bedingung>()],
    <logischer Operator>,[ ( )<Bedingung>()]
<logischer Operator> ::= {AND|&|OR||}
```

mit:

- p - Startposition des Feldes im Satz
- l - Länge des Feldes
- f - Formatcode {CH|BI|ZD|PD|FI}
 - (falls sämtliche Felder eines FIELDS- oder COND-Parameters das gleiche Format haben, kann f in der Feldbeschreibung fortfallen und mit dem FORMAT-Parameter angegeben werden)
- s - Anordnung {A: aufsteigend | D: fallend}
- m - Anzahl der zu sortierenden Sätze
- n - Anzahl der überspringenden Sätze in der Eingabedatei
- xx - Hexadezimaldarstellung eines Eingabezeichens
- uu - Hexadezimaldarstellung der neuen Position von xx in der Sortierfolge
 - (Die ALTSEQ-Anweisung gestattet eine Abänderung der Sortierfolge; die modifizierte Sortierfolge wird benutzt wenn als Formatcode AQ angegeben wird)
- k - 'Zeichenfolge', 'X'Hexverschlüsselung' oder Dezimalzahl

B) Sortieren von einem PL/I-Programm aus (PLISRT)

Wie bisher das alte läßt sich dieses Sortierprogramm von einem PL/I-Programm aus über die vier Prozeduren PLISRTA, PLISRTB, PLISRTC und PLISRTD mit unveränderten Parametern aufrufen. Dabei ist die Angabe der ALTSEQ- und der INCLUDE/OMIT-Anweisung nicht möglich, es ist jedoch geplant, eine abgeänderte Sortierfolge fest vorzusehen, die dann mit "FORMAT=AQ" benutzt werden kann.

C) FORTRAN IV- und PL/I-Unterprogramme für das Sortieren (SRAM: Sort Access Method)

Die Unterprogramme erlauben den Einsatz des Sortierprogramms anstelle normaler (WRITE-)Ausgabeeweisungen, und das (auch mehrfache) Wiedereinlesen der sortierten Sätze anstelle normaler (READ-)Eingabeeweisungen, wobei mehrere Sortierungen nach- oder nebeneinander durchgeführt werden können (etwa bei der parallelen Benutzung mehrerer Dateien). Für die Verwendung des Sortierprogramms wird zu jeder Sortierung ein SORT-Kontrollblock (SRTCB) definiert, der Informationen über die benutzte Arbeitsdatei, die übrige Sortierumgebung und die Sortierfelder enthält und der in den zugehörigen Unterprogrammaufrufen als Parameter dient. Die folgenden fünf Eingangspunkte in das Sortierprogramm sind zu benutzen (F: FORTRAN, P: PL/I):

- | | |
|--|---|
| 1. F: CAOPN(SRTCB)
P: SRTOPEN(SRTCB) | Initialisierung der
Sortierung |
| 2. F: CAFIL(SRTCB,area)
P: SRTFILL(SRTCB,area) | Übergabe der zu
sortierenden Sätze |
| 3. F: CAGET(SRTCB,area,&label)
P: SRTGETR(SRTCB,area) | Sequentielles Lesen
der sortierten Sätze |
| 4. F: CACLS(SRTCB)
P: SRTCLSE(SRTCB) | Schließen der Datei nach der
Sortierung (mindestens ein Satz
muß gelesen werden!) |
| 5. F: CAROP(SRTCB)
P: SRTROPN(SRTCB) | Öffnen der Datei für ein
erneutes Lesen der Sätze in
Sortierfolge |

Näheres ist nach der Installation des Sortierprogramms bei der Beratung zu erfahren oder einer Programmbeschreibung zu entnehmen, die noch erarbeitet wird.

Literarische und linguistische Datenverarbeitung W.A. Slaby

In diesem Artikel sollen Einsatzmöglichkeiten der automatisierten Datenverarbeitung im Bereich der Geisteswissenschaften, insbesondere in der Literaturwissenschaft und Linguistik aufgezeigt werden. Wegen der Vielfalt möglicher

Anwendungen kann dabei keine Vollständigkeit angestrebt werden; vielmehr ist mein Ziel, anhand von Projekten, die in Zusammenarbeit mit dem Rechenzentrum durchgeführt werden, einen exemplarischen Überblick über das weite Feld der literarischen und linguistischen Datenverarbeitung zu geben.

I. Kernbereiche der Computerlinguistik

a) automatische Erstellung von Wörterbüchern, Wortindizes und Konkordanzen

1. Wörterbücher

- Wörterbuch zu Auszügen aus "Stern" und "Die Zeit"
- Wörterbuch althochdeutscher Glossen
- rückläufiges Wörterbuch der koptischen Sprache
- Frequenzwörterbuch, grammatikalisches Wörterbuch und rückläufiges Wörterbuch zum griechischen Neuen Testament
- Erstellung eines Fachwortschatzes für ausländische Studierende
- Shakespeare Dictionary
- Verzeichnis der Schulbuch- und Schülersprache in der Primarstufe
- Lateinisch-Niederdeutsches Glossarium
- Ugaritisches Handwörterbuch

2. Indizes

(Im Unterschied zu Wörterbüchern treten zu den einzelnen Eintragungen weitere aus dem Text zu erschließende Angaben über Stelle, Sprecher o.ä. hinzu)

- Hapax Legomena zum griechischen Neuen Testament
- Bühnenanweisungen zu den Werken Shakespeares
- Wortindex zu den als "Bad Quartos" herausgegebenen Werken Shakespeares
- Index zum münsterischen Zeitschriftenverzeichnis

3. Konkordanzen

(Wortindizes mit Kontexten)

- Konkordanz zu den als "Bad Quartos" herausgegebenen Werken Shakespeares
- Konkordanz zum griechischen Neuen Testament
- Konkordanz zu den nachkonziliaren Texten der römisch-katholischen Liturgie

b) automatische Analyse von Zeichenketten

- automatische Lemmatisierung von Personen- und Flurnamen
- automatische Identifizierung von (Papyrus-)Fragmenten des griechischen Neuen Testaments

- computer-unterstützte Lemmatisierung beim Shakespeare Dictionary
- Identifizierung vorgegebener Buchstabenmuster in deutschen Wörtern
- Erkennen von Harmoniestrukturen in europäischer Musik

c) automatische Sprachübersetzung

- automatische Übersetzung von orthographisch repräsentierten Texten des Deutschen in Blindenkurzschrift
- automatische Phonemisierung von Texten des Deutschen und Englischen
- allgemeines lernendes Programmsystem zur automatischen Durchführung von Übersetzungen geringer Komplexität (z.B. Phonemisierung, Blindenkurzschrift, Silbentrennung)

Neben diesen Kernbereichen der Computerlinguistik, die ich an einer sicherlich nicht vollständigen Reihe von Projekten exemplifiziert habe, gibt es weitere Bereiche der Datenverarbeitung, die in den Dienst der Literatur- und Sprachwissenschaft gestellt werden können. Es handelt sich hierbei jedoch um Verfahren, deren Anwendung nicht auf dieses Wissenschaftsgebiet allein beschränkt sind.

II. Computerunterstützte Dokumentation

a) automatische Erstellung von Dokumentations-Listen (Vorteil des Einsatzes der EDV: ohne besonderen Mehraufwand lassen sich nach verschiedenen Kriterien sortierte Listen erstellen)

- Bibliographien zu Nusi und Ugarit
- MÜNSTERISCHES Zeitschriftenverzeichnis der Universitätsbibliothek
- Bibliotheksprogramm zur Unterstützung der Verwaltung einer Institutsbibliothek
- Schallplattenkataloge des Instituts für Musikwissenschaft
- Bearbeitung der Nomenklatur und des Indexkataloges des Historischen Wörterbuchs der Philosophie

b) Dokumenten-Datenbank mit Programmsystem zur Dokumentensuche (Vorteil gegenüber einem Katalog: hoher Aktualitätsgrad bei sich stark änderndem Dokumentenbestand)

- Datenbank aller Benutzer des Rechenzentrums
- Ausleih-Verwaltungssystem der Universitätsbibliothek

III. Computergesteuerter Satz

a) Programmsysteme zur Durchführung von Photo- und Lichtsatzaufgaben

b) Aufbereitung von Datenmaterial für ein automatisches Photo-/Lichtsatzverfahren

- Photosatz einer Shakespeare-Kondordanz
- Lichtsatz von Spezialübersichten (Frequenzwörterbuch, rückläufiges Wörterbuch, u.a.) zum griechischen Neuen Testament
- Lichtsatz einer Kondordanz zum griechischen Neuen Testament
- Lichtsatz eines rückläufigen Wörterbuchs der koptischen Sprache
- Photosatz von Parallelregistern mittelalterlicher Personennamen

IV. Anwendung mathematischer Verfahren

a) statistische Verfahren

1. in Verbindung mit Wörterbüchern, Wortindizes und Konkordanzen: Berechnung absoluter und relativer Worthäufigkeiten
2. bei der automatischen Analyse von Zeichenketten: Feststellung der absoluten und relativen Häufigkeiten bestimmter Buchstabenmuster in einer Klasse von Wörtern
3. statistische Verteilungen, Korrelationen
 - Harmonieverteilung in europäischer Musik
 - Sprachverhalten von Schülern der Primarstufe
 - Statistische Arbeiten über Struktur und Entwicklung mittelalterlicher Klosterkonvente

b) kombinatorische Verfahren

- Personenidentifizierung durch Gruppensuche in alten Klosterhandschriften
- Verwandtschaftsbeziehungen von Handschriften des griechischen Neuen Testaments

c) graphische Verfahren

- wort- und sprachgeographische Untersuchung des Deutschen
- automatische Bestimmung von Isoglossen

Dieser Überblick sollte ein Versuch sein, die vielfältigen Möglichkeiten des Einsatzes der elektronischen Datenverarbeitung in der Literatur- und Sprachwissenschaft anhand von konkreten, an unserer Universität durchgeführten Projekten aufzuzeigen. Wenn dabei ein Projekt aus diesem Bereich versehentlich nicht berücksichtigt worden sein sollte, so bitte ich um Nachsicht und würde mich über eine Zuschrift freuen.

NAG-Bibliothek

U. Ebert

Die Programme der NAG-Bibliothek sind jetzt sowohl in einfacher Genauigkeit wie in doppelter Genauigkeit vorhanden (bisher nur in doppelter Genauigkeit). An der vorhandenen Dokumentation ändert sich nichts, sie wird lediglich durch ein weiteres (gelbes) "Implementation Document" ergänzt.

Sucht man etwa ein Programm zur Matrixinvertierung, so muß man nach wie vor zuerst im Mini_Manual nachschauen, welches Programm für die entsprechende Fragestellung in Betracht kommt, z.B. F01AAF. Vor der Benutzung des Programms muß dessen Dokumentation im Library_Manual nachgelesen werden. Dort werden insbesondere auch die Parameter beschrieben.

Da F01AAF ein Programm in doppelter Genauigkeit ist, sind alle dort als "REAL" beschriebenen Größen als REAL*8 und "COMPLEX"-Größen als COMPLEX*16 zu wählen.

Möchte man stattdessen ein Programm in einfacher Genauigkeit benutzen, so muß man statt F01AAF das Programm E01AAE aufrufen (es ändert sich also nur der letzte Buchstabe; F = Doppelte Genauigkeit, E = Einfache Genauigkeit). In diesem Falle sind alle in der Dokumentation des Library Manuals als "REAL" beschriebenen Größen als REAL*4 und die "COMPLEX"-Größen als COMPLEX*8 zu wählen. Alles andere bleibt gleich.

In manchen Programmen werden Maschinenkonstanten benutzt bzw. erzeugt. Diese unterscheiden sich natürlich, je nachdem, ob es sich um Größen in einfacher oder doppelter Genauigkeit handelt. Details darüber befinden sich in den gelben "Implementation Documents" (ID B E Mk5 für einfache und ID B F Mk5 für doppelte Genauigkeit) in den Library Manuals des Rechenzentrums vorn in Band 1 und sollten zusätzlich konsultiert werden.

Beide Versionen der NAG-Bibliothek sind mit normalen FORTRAN-Prozeduren benutzbar. Es muß lediglich (wie bisher auch schon) ein zusätzlicher symbolischer Parameter angegeben werden:

```
//name EXEC FORTCE,SUBLIB=NAG
//COMPILE.SYSIN DD *
    Programme mit Aufruf von
    NAG-Unterprogrammen
//EXECUTE.SYSIN DD *
    Daten
```

In diesem Fall ist die NAG-Bibliothek verfügbar und nicht das SSP. Läßt man SUBLIB=NAG weg, so ist nur das SSP verfügbar.

Eine eingehende Beratung zur NAG-Bibliothek (Auswahl von Programmen, andere Schwierigkeiten) findet statt:

jeweils montags und donnerstags von 15 bis 16 Uhr
im Rechenzentrum, Roxeler Str. 60, Raum 108

Struktogramme:

J. Ebert

Seit einigen Jahren wird unter dem Stichwort "strukturierte Programmierung" die Auffassung propagiert, Programme seien unter Verwendung elementarer Kontrollstrukturen bausteinartig aufzubauen. Aber für die Sprachen, die eine Verwendung strukturierter Programmierung erlauben, wird trotzdem zur Dokumentation häufig auf das zur Beschreibung unstrukturierter Sprachen geschaffene Mittel der Flußdiagramme (Programmablaufpläne, DIN 66001) zurückgegriffen. Deshalb soll hier einmal auf eine modernere Form der Programmdokumentation aufmerksam gemacht werden, nämlich die Struktogramme.

Nach den Theoremen der strukturierten Programmierung reicht es aus und ist zu erstreben, zur Erstellung von Programmen mit den Kontrollstrukturen:

Statementzusammenfassung	(PL/I: <code>begin...end</code> <code>do...end</code>)
Wiederholung	(PL/I: <code>do while...</code> <code>do until...</code>)
Fallunterscheidung	(PL/I: <code>if...then...else...</code> <code>select...</code>)

auszukommen. Eventuell wird noch ein Notausgang (PL/I: `leave`) zugelassen.

Diesen Forderungen wird durch die Struktogramme Rechnung getragen. Sie lassen keine Dokumentation unzulässiger (d.h. nicht durch die obigen Strukturelemente begründeter) Sprünge zu und sind somit für die strukturierte Programmierung das angemessene Darstellungsmittel. Gleichzeitig sind die Struktogramme zu einer hierarchischen, d.h. schrittweise feiner werdenden Darstellung geeignet.

In Struktogrammen wird eine Anweisung grundsätzlich durch ein Rechteck repräsentiert:

Anweisung

Kontrollanweisungen d.h. solche Anweisungen, die andere als Teile enthalten, sind so dargestellt, daß in sie die entsprechenden Rechtecke eingefügt werden können.

Da es sich um eine Dokumentationstechnik handelt, also die Darstellung möglichst unabhängig von der verwendeten Sprache sein soll, ist es erlaubt, mehrere Anweisungen sprachlich zusammenzufassen und in einem Rechteck unterzubringen.

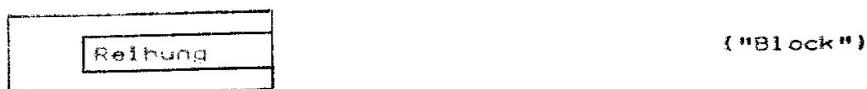
Beispiel:

MAX, MIN initialisieren
und X einlesen

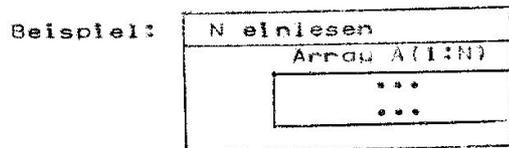
Eine Hintereinanderausführung von Anweisungen wird durch Untereinanderschreiben dargestellt:



Werden mehrere Anweisungen zu einem Block zusammenfaßt, so ist dies durch:



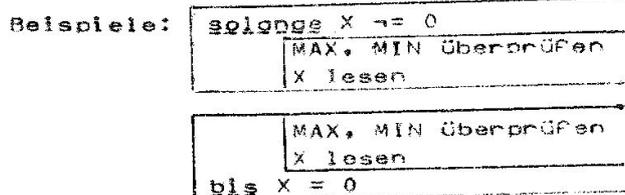
beschreibbar, wobei der obere Balken des äußeren Teils zur Angabe einer Überschrift oder zur Beschreibung von (für diesen Block) lokalen Größen verwendet werden kann.



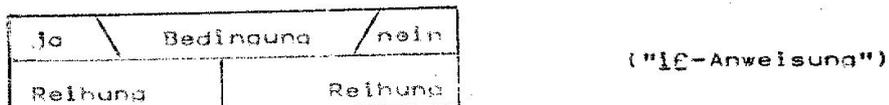
Für eine while-Schleife verwendet man:



und für eine until-Schleife:



Durch ein geteiltes Rechteck wird eine if-Anweisung dargestellt:

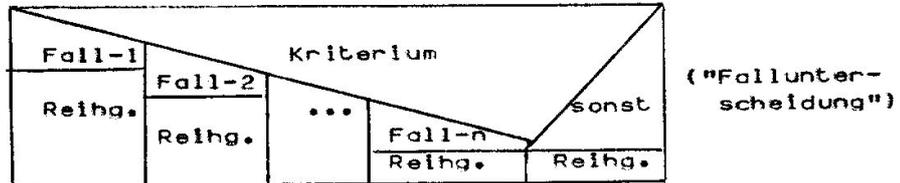


Dabei darf der Nein-Teil im Falle einer einfachen if-Anweisung aus Eindeutigkeitsrunden nicht fortgelassen werden.

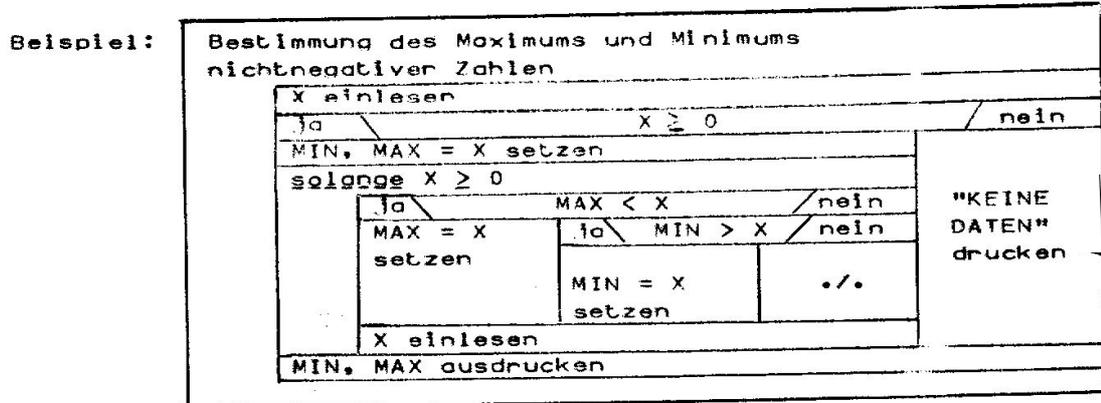
Beispiel:

ja	MIN > X	nein
MIN = X setzen		./.

Sind mehrere Fälle zu untersuchen, so ist diese Figur wie folgt zu verallgemeinern:



Entsprechend dem Aufbau strukturierter Programme ist dann ein Gesamtprogramm durch ineinanderfügen dieser Elemente zu dokumentieren.



Zur Darstellung eines Notausgangs kann noch das Symbol:



verwendet werden.

Weitere Anwendungen dieser Technik finden sich zum Beispiel in dem Buch
Schnupp/Floud: "Software", de Gruyter, 1976

FORTRAN-Standards

R. Schmitt

Bereits im Jahre 1966 wurde für die Programmiersprache FORTRAN ein Standard definiert (American National Standard FORTRAN, ANS X3.9-1966 siehe auch Comm. ACM 7-1964, 590 FF). Derzeit garantieren alle Hersteller von Großrechnern, daß in ihren FORTRAN-Compilern mindestens dieser Standard implementiert ist. Einhaltung der ANS-Standards garantiert also die problemlose Übertragbarkeit von Quellenprogrammen, was nicht unbedingt bedeutet, daß das Objektprogramm überall die gleichen Ergebnisse bringt (siehe *inCOrnum* Nr. 3, 1977).

Da FORTRAN anders als etwa ALGOL 60 zum Zeitpunkt der Definition der Standards bereits vielfach und seit langem implementiert war, wurde der Standard in gewissem Sinne der Durchschnitt der existierenden Versionen. Infolgedessen und insbesondere durch die um den Standard herumwuchernden Spracherweiterungen ist ANS-FORTRAN eine echte Teilmenge der meisten existierenden FORTRAN-Versionen. So erscheint dem Benutzer von FORTRAN IV auf IBM-Anlagen (im folgenden: IBM-FORTRAN) die Standardisierung der Sprache eher als deren Simplifizierung, eine Betrachtungsweise, die eine Einhaltung des Standards, meist sogar jegliche Beschäftigung mit ihm, wenig fördert.

Ohne den ANS-Standard überzubewerten, soll hier eine Lanze für die Standardisierung gebrochen werden. Insbesondere soll gezeigt werden, was wirklich verloren geht, wenn man den Standard einhält, und welchen Wert dieser Verlust hat. Selbstverständlich ist eine solche Wertung subjektiv und soll es auch sein.

Beginnen wir mit dem Alphabet: Daß das \$-Zeichen ein Buchstabe ist, und zwar der allererste (in der Sortierfolge), ist ein IBM-Aspekt, wenn auch kein exklusiver. Darauf kann man wohl verzichten.

Ein ganzes Bündel lieb gewordener Gewohnheiten muß man allerdings im Bereich der Typ-Spezifikation aufgeben. Da fehlt zunächst die IMPLICIT-Anweisung. Allerdings behalten die impliziten Voreinstellungen (I bis N: INTEGER; A bis H, O bis Z: REAL) ihre Gültigkeit. Auch gibt es im ANS-FORTRAN weniger Datentypen als im IBM-FORTRAN, nämlich neben REAL und INTEGER nur noch COMPLEX, DOUBLE PRECISION und LOGICAL sowie für Konstanten den Typ HOLLERITH (= LITERAL). Es fehlen also hexadezimale Konstanten und die Datentypen mit expliziter Längenangabe (z.B. REAL*8). Der einzig wirkliche Verlust ist hierbei das Fehlen doppelt genauer komplexer Zahlen, die restlichen Datentypen sind höchst maschinenabhängig. Schließlich ist es im ANS-FORTRAN nicht erlaubt, Variablen bereits in der Spezifikationsanweisung zu initialisieren. Dafür ist einzig die DATA-Anweisung zuständig. Wen das stört, der erinnere sich, daß IBM-FORTRAN eine Initialisierung in der DOUBLE-PRECISION-Anweisung auch nicht erlaubt. Die DATA-Anweisung ist im übrigen

vom ANS-Standard etwas vernachlässigt worden, erlaubt sie doch nicht die Initialisierung eines ganzen Arrays, sondern nur die einzelner Elemente. Das erfordert viel Schreibarbeit und deshalb gehen an dieser Stelle auch die meisten Compiler über den ANS-Standard hinaus.

Bleiben wir bei den Arrays: ANS-FORTRAN erlaubt nur maximal drei Dimensionen gegenüber sieben bei IBM-FORTRAN. Ferner sind zur Indizierung von Array-Elementen neben INTEGER-Variablen und -Konstanten höchstens Ausdrücke der Form:

$$C * V \uparrow K$$

erlaubt, wobei V eine INTEGER-Variablen ist und C und K INTEGER-Konstanten sind (zur Erinnerung: INTEGER-Konstanten sind vorzeichenlos)! Wer die recht große Freiheit kennt, die IBM-FORTRAN dem Programmierer an dieser Stelle läßt, wird diese Einschränkung sehr bedauern. Besonders dadurch, daß Array-Elemente und Funktionsaufrufe im ANS-Standard zur Indizierung nicht zugelassen sind, kann die Umstellung von Programmen, die diese Technik des IBM-FORTRAN benutzen, sehr mühsam werden. Hingegen wird der ANS-Programmierer die Fähigkeit des IBM-FORTRAN, auch REAL-Ausdrücke zur Indizierung zuzulassen, kaum vermissen.

Deutliche Einschränkungen gibt es auch bei arithmetischen Ausdrücken. Die unangenehmste ist wohl die, daß INTEGER-Ausdrücke nicht mit REAL-, DOUBLE-PRECISION- oder COMPLEX-Ausdrücken arithmetisch verknüpft oder verglichen werden dürfen; lediglich bei der Exponentiation ist ein Exponent vom Typ INTEGER unabhängig vom Datentyp der Basis erlaubt. Ferner ist zu beachten, daß die Reihenfolge der Abarbeitung von arithmetischen Ausdrücken im ANS-Standard nur soweit definiert ist, als es die Rangfolge der Rechenoperationen betrifft. Schließlich ist mehrfache Exponentiation ohne Klammerung verboten. Wirklich einschneidend ist keine dieser Einschränkungen. Mixed-Mode-Ausdrücke vermeidet man durch Verwendung der Transfer-Funktionen und die Klammerung bei der Exponentiation ist höchst empfehlenswert, zumal die implizite Abarbeitungsfolge solcher Ausdrücke im IBM-FORTRAN nicht unbedingt kanonisch ist. Bleibt noch zu vermerken, daß bei Wertzuweisungen beide Seiten durchaus von verschiedenem Datentyp sein dürfen. Die Konvertierung erfolgt nach den durch IBM-FORTRAN vertrauten Regeln.

Die Unterprogrammtechnik ist beim ANS-FORTRAN im wesentlichen dieselbe wie beim IBM-FORTRAN, sieht man von den folgenden drei Einschränkungen ab: Erstens fehlt die Möglichkeit, Anweisungsnummern als Parameter zu übergeben und im Unterprogramm für einen alternativen Rücksprung zu verwenden, dann gibt es keine ENTRY-Anweisung und schließlich sind formale Parameter der Form /A/ (call by location) nicht erlaubt. Letztere sind im IBM-FORTRAN ohnehin wenig bekannt und nur zur Unterstützung des DIRECT ACCESS sinnvoll, wogegen die ENTRY-Anweisung im IBM-FORTRAN häufig verwendet wird. Angesichts der höchst zweifelhaften Techniken, die im Zusammenhang mit diesem Befehl vom Hersteller und von einigen Lehrbüchern aufgezählt

werden, kann man das Fehlen dieser Anweisung im ANS-FORTRAN jedoch nur begrüßen. Wichtig zu wissen ist noch, daß nicht alle Standardunterprogramme des IBM-FORTRAN zum ANS-Standard gehören.

Bleibt das Kapitel Ein-/Ausgabe, das für eine Standardisierung naturgemäß das schwierigste ist. Zunächst muß man im ANS-FORTRAN auf NAMEDLIST, PRINT, PUNCH und - wie bereits angedeutet - auf DIRECT ACCESS (DEFINE FILE, FIND und zugehöriges READ/WRITE) verzichten. Der geringe Bekanntheitsgrad dieser Sprachmittel läßt diese Einschränkung unwesentlich erscheinen. Einige Umstellungsarbeit erfordert aber sicher, daß ANS-FORTRAN in der READ-Anweisung nicht die END-Klausel und - weniger wichtig - die ERR-Klausel kennt. Ansonsten ist aber das Konzept sowohl der formatierten als auch der unformatierten Ein-/Ausgabe im ANS-FORTRAN dasselbe wie im IBM-FORTRAN. Kleine Einschränkungen betreffen noch die FORMAT-Liste. Hier kennt der ANS-Standard konsequenterweise kein Z-Format, leider fehlt auch das T-Format, und Literale können nur mit dem H-Format angegeben werden (kein in Apostrophe eingeschlossener Text). Außerdem ist das G-Format nicht für INTEGER- und LOGICAL-Variablen zugelassen und schließlich dürfen nur solche Arraus als Object-Time-Formate verwendet werden, die keine Texte (H-Format) enthalten. Wer letzteres stört, der studiere einmal die entsprechenden Passagen bei IBM-FORTRAN (FORTRAN IV Language, GC28-6515).

Strikte Befolgung der Standards ist sicherlich nicht immer notwendig. Wenn es nicht stört, daß seine Programme nur auf Anlagen eines bestimmten Typs bei Verwendung eines bestimmten Compilers problemlos arbeiten, der mag unsere FORTRAN-Versionen mit allen bekannten und möglicherweise unbekanntem Tricks anwenden. Allen anderen hilft bei der Entscheidung für oder gegen ANS-Standard vielleicht folgende Gruppierung.

- Wer adhoc und privat ein kleines Problem mittels FORTRAN lösen will und wer sein Programm nach Erhalt der Resultate üblicherweise wegwirft, den braucht kein Standard zu kümmern.
- Wer Programme entwickelt, die nicht nur von ihm eingesetzt werden, sondern allgemein genaug sind, in eine Programm-bibliothek aufgenommen zu werden, sollte sich möglichst am ANS-Standard orientieren.
- Wer absehen kann, daß seine Programme auch auf anderen und andersartigen Rechenanlagen eingesetzt werden, der muß den ANS-Standard einhalten, will er nicht zeitraubende Umstellungsarbeiten riskieren.

Aufruf von FORTRAN-Programmen in PL/I-Programmen

U. Ebert

An zwei Beispielen werden hier die für den Aufruf von FORTRAN-Programmen in PL/I-Programmen wichtigen Fragen erörtert.

Anstelle der normalen PLO-Prozeduren (z.B. PLOCE) ist eine Prozedur mit einem zusätzlichen F am Ende (z.B. PLOCEF) zu verwenden (vgl. J. Ebert, S. 3, sie steht vom 27.1.1978 an zur Verfügung). Diese Prozeduren sorgen dafür, daß erstens die für

FORTRAN-Programme notwendigen Laufzeitroutinen (INPUT/OUTPUT, Fehlerbehandlung, SQRT, etc.) automatisch zur Verfügung gestellt werden (ca. 30K zusätzlicher Speicherplatz), und zweitens, daß die Programme der NAG-Bibliothek ohne zusätzliche Steuerkarten aufgerufen werden können.

Bei einem Aufruf eines FORTRAN-Programms von PL/I aus muß dann ein Teil der FORTRAN-Laufzeitroutinen initialisiert werden. Dies geschieht automatisch bei einem Aufruf eines FORTRAN-Programms. Damit es wegen der in PL/I vorhandenen dynamischen Programmstruktur nicht zu Komplikationen bzw. unnötig häufigen Initialisierungen kommt, sollte standardmäßig am Anfang der PL/I-Haupt-Prozedur (OPTIONS(MAIN)) das FORTRAN-Unterprogramm FORTRAN aufgerufen und deklariert werden:

```
DCL FORTRAN ENTRY EXT OPTIONS(FORTRAN);
CALL FORTRAN;
```

Das bewirkt die einmalige Initialisierung.

Am Beispiel des NAG-Programms C05ABF zur Nullstellenbestimmung werden die weiter notwendigen Vorkehrungen demonstriert. Es ist vollständig zu deklarieren:

```
DCL C05ABF EXT OPTIONS(FORTRAN)
ENTRY(DEC FLOAT(16),DEC FLOAT(16),/*INTERVALL*/
DEC FLOAT(16), /*GENAUIGKEIT*/
ENTRY(DEC FLOAT(16)) RETURNS(DEC FLOAT(16))
OPTIONS(FORTRAN), /*FUNKTION*/
DEC FLOAT(16), /*NULLSTELLE*/
BIN FIXED(31)); /*FEHLERPARAMETER*/
```

Der Aufruf

```
CALL C05ABF(A,B,EPS,FUNK,X,IFAIL);
```

bewirkt dann, daß im Intervall mit den Grenzen A und B für die Funktion FUNK eine Nullstelle X mit der Genauigkeit EPS gesucht wird. Obwohl das Programm C05ABF ein FORTRAN-Programm ist, kann die zu untersuchende Funktion FUNK eine PL/I-Prozedur sein (deren Name höchstens sieben Zeichen lang sein sollte). Sie muß nur entsprechend deklariert, als externe Prozedur übersetzt und mit der entsprechenden Option versehen werden:

```
DCL FUNK ENTRY(DEC FLOAT(16)) RETURNS(DEC FLOAT(16))
OPTIONS(FORTRAN) EXT;
...
*PROCESS;
FUNK: PROC(X) OPTIONS(FORTRAN) RETURNS(DEC FLOAT(16));
DCL X DEC FLOAT(16);
RETURN(...);
END;
```

Durch die Deklaration wird FUNK als eine Prozedur beschrieben, die die Gestalt eines FORTRAN-Programms hat. Die Spezifikation OPTIONS(FORTRAN) im Prozedur-Statement von FUNK sorgt dann

dafür, daß es diese Gestalt auch wirklich bekommt.

Bei der Übergabe von Feldern zwischen PL/I- und FORTRAN-Programmen hat man es mit dem Problem verschiedenartiger Abspelcherung in beiden Programmiersprachen zu tun. Bei eindimensionalen Feldern spielt das keine Rolle, aber bei mehrdimensionalen Feldern ist zu überlegen, ob eine Umspeicherung notwendig ist. Beim NAG-Programm F03AAE z.B. kann auf die Umspeicherung verzichtet werden, da die Determinante einer Matrix gleich der Determinante der transponierten ist:

DCL F03AAE ENTRY(...) OPTIONS(FORTRAN NOMAP);

Die Option NOMAP verhindert die Umspeicherung. So gewinnt man Rechenzeit und braucht weniger Speicherplatz.

Insgesamt sollte man sich beim Aufruf von FORTRAN-Programmen von PL/I-Programmen aus und umgekehrt bewußt sein, daß jeder Übergang von einer Programmiersprache in die andere einen gegenüber dem Aufruf zwischen Programmen derselben Programmiersprache erheblichen zusätzlichen Aufwand bedeutet. Sind es aber nur wenige Aufrufe, so kann man den Nachteil in Kauf nehmen. Außerdem sind die Vorteile zu betrachten, die sich daraus ergeben, daß man unter Umständen auf diese Weise auf vorhandene Programme zurückgreifen kann.

Sequentielle Verarbeitung von Dateien

W. Bosse

Dieser Artikel soll eine einführende Übersicht der wesentlichen Merkmale einer sequentiellen Verarbeitung von Magnetband-Dateien geben; es handelt sich dabei um eine Fortsetzung des Artikels "Dateien auf Magnetband" aus *inforum* Nr.3, 1977, S.9-13 (siehe auch die Korrektur in *inforum* Nr.4, 1977, S.11).

Dem Datenträger Magnetband entsprechend erfolgt die Speicherung von Dateien ausschließlich sequentiell, d.h. alle Sätze einer Datei werden, geblockt oder ungeblockt, fortlaufend hintereinandergeschrieben. Außerdem stehen auch die Dateien auf einem Magnetband hintereinander, so daß nur eine Veränderung der letzten Datei möglich ist, ohne andere Dateien zu beeinträchtigen. Daraus ergeben sich für die Verarbeitung solcher Dateien bestimmte Merkmale, die im Folgenden dargelegt werden.

Erzeugen einer Datei:

Alle Sätze werden in derselben Reihenfolge gespeichert, in der sie mittels eines Programms erzeugt und auf das gewünschte Magnetband geschrieben werden. Es empfiehlt sich deshalb, die Sätze vorher zu sortieren, falls sie einen Sortierbegriff enthalten.

Verarbeiten einer Datei:

Aufgrund der sequentiellen Speicherung einer Magnetband-Datei können die Sätze nur sequentiell verarbeitet werden, d.h. genau in der Reihenfolge, in der sie gespeichert worden sind. Um beispielsweise den letzten Satz einer Datei zu erreichen, müssen zunächst alle davor stehenden Sätze, vom ersten bis zum vorletzten, gelesen werden. Da in jedem Fall die gesamte Datei gelesen werden muß, erweist sich die Verarbeitung einer sequentiellen Datei umso vorteilhafter, je mehr Sätze tatsächlich bearbeitet werden müssen.

Fortschreiben einer Datei:

Als spezielle Form der Verarbeitung dient die Fortschreibung einer Datei (Update) dem Hinzufügen neuer Sätze (in Sortierfolge) sowie dem Löschen oder Modifizieren bereits vorhandener Sätze. Alle diese Veränderungen können aber wegen der sequentiellen Speicherung nicht in der Magnetband-Datei selbst vorgenommen werden; das Fortschreiben muß deshalb dadurch erfolgen, daß eine neue Datei als Kopie der alten mit den gewünschten Änderungen auf einem anderen Magnetband erzeugt wird.

Die Angaben über alle notwendigen Änderungen werden in einer Update-Datei zusammenfaßt, die häufig auf Lochkarten erstellt wird und die nach demselben Ordnungsbegriff sortiert sein muß wie die Fortzuschreibende Datei. Dann können beide Dateien parallel zueinander bearbeitet werden, wobei mittels des Ordnungsbegriffs und einer Kennzeichnung in der Update-Datei jeweils einer der folgenden Fälle zu bearbeiten ist:

- | | |
|--------------|--|
| Einfügen | - der Satz wird aus der Update-Datei als Zugang übernommen und (in Sortierfolge) in die neue Datei eingefügt; |
| Löschen | - der Satz wird nicht von der alten in die neue Datei kopiert; |
| Verändern | - der Satz wird nach entsprechender inhaltlicher Änderung in die neue Datei übernommen; |
| Kopieren | - der Satz wird unverändert übertragen, da es in der Update-Datei keinen Verweis auf ihn gibt; |
| Zurückweisen | - ein fehlerhafter Satz der Update-Datei wird ignoriert (nachdem eine entsprechende Mitteilung gedruckt worden ist). |

In einem Beispiel soll das dargestellt werden; dabei werden die Sätze allein durch ihren Ordnungsbegriff, eine sechsstellige Nummer, repräsentiert, neue Sätze werden unterstrichen und veränderte Sätze eingeklammert.

alter_Bestand (1. Magnetband):

```

-----
== | 000170 | 000171 | 000180 | 000185 | 000201 | 000203 | ==
-----
    
```

Update-Datell (Lochkarten):

Löschen 000171
 Einfügen 000174,000175
 Verändern 000185
 Löschen 000201

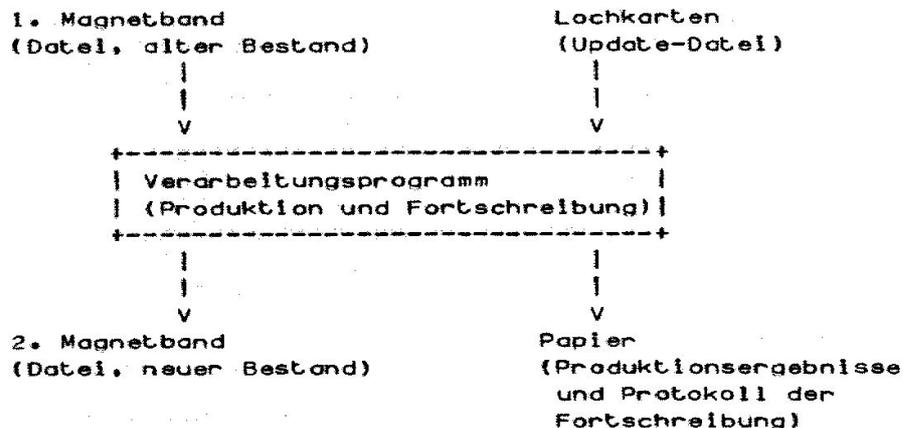
neuer_Bestand (2. Magnetband):

```

-----
== | 000170 | 000174 | 000175 | 000180 | (000185) | 000203 | ==
-----
    
```

Da die Fortschreibung einer Magnetband-Datell unabhängig von der Anzahl der vorzunehmenden Änderungen stets eine Bearbeitung der gesamten Datell erfordert, wird diese Aufgabe gerne mit einem Produktionslauf gekoppelt, bei dem alle Sätze verarbeitet werden müssen. Dadurch wird die Bearbeitung großer Datenmengen mit Magnetbändern wirtschaftlich.

Datenflussplan:



Datensicherheit:

Durch diese Form der Verarbeitung wird automatisch eine gewisse Datensicherheit bewirkt. Falls nämlich die Datell mit dem neuen Bestand verloren geht, so lässt sich aus dem alten Bestand in Verbindung mit der Update-Datell der Zustand leicht wieder

herstellen. Auf dieser Tatsache beruht das sogenannte Großvater-Vater-Sohn-Prinzip, bei dem drei Magnetbänder verwendet werden. Nach zwei Fortschreibungen ergibt sich:

1. Magnetband - alter Bestand ("Großvater")
2. Magnetband - neuer Bestand ("Vater")
3. Magnetband - neuester Bestand ("Sohn")

Bei der nächsten Fortschreibung wird dann zur Speicherung des aktuellen Bestandes der Datei ("Sohn") wieder das 1. Magnetband verwendet. Dadurch werden die sich auf dem 2. und 3. Magnetband befindenden Dateien zum "Großvater" bzw. zum "Vater".

Ein Beispiel:

Abschließend soll durch einen vollständigen Job die sequentielle Verarbeitung einer Magnetband-Datei dargestellt werden. Das Verarbeitungsprogramm ist in der Programmiersprache PL/I auf dem Sprachniveau des PL/I Optimizing Compilers geschrieben worden, der durch die katalogisierte Prozedur PLOCE aufgerufen wird. Auf die Verwendung der /*JOBPARM-Karte und die Wahl der Datei-Namen sei besonders hingewiesen.

Im Programm bezeichnen die File-Namen ALT und NEU den alten bzw. neuen Bestand der Magnetband-Datei, SYSIN die Update-Datei und SYSPRINT die Druck-Ausgabe. Der Ordnungsbegriff ist eine 6-stellige Nummer (SATZ.NR, KARTE.NR), die dem Inhalt des Satzes voraussetzt (SATZ.INHALT, KARTE.INHALT). Jede Karte der Update-Datei muß in Spalte 80 durch Blank (Einfügen), 'L' (Löschen) oder 'V' (Verändern) gekennzeichnet sein, wie das aus dem Format SYSIN_KARTE und dem Aufbau der SELECT-Anweisung zu entnehmen ist. Während die Fortschreibung der Magnetband-Datei ausführlich programmiert worden ist, wurden die Stellen des Programms, an denen eine weitere Verarbeitung der Sätze möglich ist, lediglich durch Kommentare hervorgehoben.

```
//UUU01TAP JOB (UUU01,NORM,B01),WINTER
/*JOBPARM TAPES=2
//BAND EXEC PLOCE
//COMPILE.SYSIN DD *
SEQ : PROC OPTIONS (MAIN) ;
/* BEISPIEL FUER SEQUENTIELLE DATEI-VERARBEITUNG */
DCL ( ALT INPUT, NEU OUTPUT ) FILE RECORD SEQL ;
DCL ( SYSIN INPUT, SYSPRINT PRINT ) FILE STREAM ;
DCL 1 SATZ, 10 NR CHAR ( 6),
      10 INHALT CHAR (70) ;
DCL 1 KARTE, 10 SATZ LIKE SATZ,
      10 ART CHAR ( 1) ;
DCL ( UNENDLICH, ALT_ENDE ) BIT (1) INIT ('0'B) ;
DCL FORTSCHREIBUNG BIT (1) INIT ('1'B) ;
SYSIN_KARTE : FORMAT ( A(6), A(70), X(3), A(1) ) ;
ON ENDFILE ( ALT )
BEGIN ; SATZ.NR = HIGH (1) ; ALT_ENDE = '1'B ; END ;
ON ENDFILE ( SYSIN ) FORTSCHREIBUNG = '0'B ;
```

```

/* ALLE SAETZE IN DEN DATEIEN ALT UND SYSIN          */
/* MUESSEN NACH NUMMERN (NR) AUFSTIEGEND SORTIERT SEIN */
GET FILE ( SYSIN ) EDIT ( KARTE )
      ( COL (1), R(SYSIN_KARTE) ) ;
DO UNTIL ( UNENDLICH ) ;
  READ FILE ( ALT ) INTO ( SATZ ) ;
  DO WHILE ( FORTSCHREIBUNG & KARTE.NR <= SATZ.NR ) ;
    SELECT ;
      WHEN ( ART = ' ' & KARTE.NR < SATZ.NR )
        /* ZUGANG UND EVENTUELL AUCH          */
        /* VERARBEITUNG VON KARTE.SATZ */
        WRITE FILE ( NEU ) FROM ( KARTE.SATZ ) ;
      WHEN ( ART = 'L' & KARTE.NR = SATZ.NR )
        /* LOESCHEN */
        READ FILE ( ALT ) INTO ( SATZ ) ;
      WHEN ( ART = 'V' & KARTE.NR = SATZ.NR )
        /* VERAENDERN */
        SATZ.INHALT = KARTE.INHALT ;
      OTHERWISE /* FEHLER */
        PUT FILE ( SYSPRINT ) EDIT
          ( '*** NICHT AKZEPTIERT' ) ( SKIP, A ) ;
    END ;
    PUT FILE ( SYSPRINT ) EDIT ( '*** UPDATE ***', KARTE )
      ( COL (22), A, X(1), R(SYSIN_KARTE) ) ;
    GET FILE ( SYSIN ) EDIT ( KARTE )
      ( COL (1), R(SYSIN_KARTE) ) ;
  END ;
  IF ALT_ENDE THEN LEAVE ;
  WRITE FILE ( NEU ) FROM ( SATZ ) ;
  /* EVENTUELL AUCH VERARBEITUNG VON SATZ */
END ;
PUT FILE ( SYSPRINT ) EDIT
  ( '*** ENDE DES PROGRAMMS ***' ) ( SKIP (2), A ) ;
END SEQ ;
//EXECUTE.ALT DD DSN=UUU01.DATGENA,DISP=(OLD,KEEP),
// UNIT=TAPE,VOL=SER=111111,LABEL=3
//EXECUTE.NEU DD DSN=UUU01.DATFORB,DISP=(NEW,KEEP),
// UNIT=TAPE,VOL=SER=222222,LABEL=2,
// DCB=(RECFM=FB,LRECL=76,BLKSIZE=3800)
//EXECUTE.SYSIN DD *
| [Update-Datel auf Lochkarten]
//

```

Das Programm SEQ kann auch zur Generierung einer Magnetband-Datel aus den auf Lochkarten vorliegenden Sätzen verwendet werden, indem das DD-Statement zum File ALT durch

```
//EXECUTE.ALT DD DUMMY
```

ersetzt wird. Die SYSIN zugeordneten Lochkarten müssen dann alle ein Blank in Spalte 80 aufweisen. Andererseits kopiert das Programm einfach den alten Bestand einer Magnetband-Datel, wenn die Update-Datel durch

```
//EXECUTE.SYSIN DD DUMMY
```

angegeben wird.

Verbrauchsdaten 1977

J. Ebert, W. Schafmann

	HASP- JOBS	BATCH- JOBS	SEITEN- ZAHL	GESTANZTE KARTEN	CPU- ZEIT

FB 1	204	0	2595	3663	7:42:08
FB 2	415	0	8326	6290	9:24:33
FB 4	22510	1332	251822	398843	405:50:25
FB 5	1602	154	17278	17866	34:28:35
FB 6	12782	160	278357	174183	305:31:45
FB 7	1023	4	7838	22375	27:31:44
FB 8	4186	363	46157	106783	78:12:17
FB 9	1411	23	21641	18753	16:58:58
FB 10	3766	358	42221	39603	71:18:50
FB 11	644	0	11928	8056	8:52:38
FB 12	1020	0	16593	3141	27:13:53
FB 15	3904	622	34135	35645	108:27:35
FB 16	28326	1166	250211	643902	1066:28:47
FB 17	14608	843	151901	316833	488:41:00
FB 18	1148	121	12380	29309	18:01:53
FB 19	6819	1243	60664	95396	109:34:43
IFL	924	128	10792	30032	8:56:27
RZ & UB	22697	3641	245024	427666	307:08:32
ZUV	10577	9	651999	269991	257:38:22
KURSE	32141	31450	146395	68329	
