

1000

inforum

INFORMATIONsforum des Rechenzentrums der Universität Münster

Jahrgang 5, Nr. 4

Oktober 1981

Inhalt	Seite
Anmerkungen zum Rechenbetrieb	2
Vortragsreihe über Mikrocomputer und Prozeßdatenverarbeitung	3
Lehrveranstaltungen im WS 81/82	4
Einführung in die Benutzung der Rechanlage	5
Computergesteuerter Satz wissenschaftlicher Texte	6
VS FORTRAN	9
Anträge auf Rechenerlaubnis für 1982	15
Sprechzeiten	16
Programme zur Multidimensionalen Skalierung (MDS)	17
Hardware-situation	18
Personalia	19
Jobverteilung nach Job-CPU-Zeit und Fachbereich	20

Anmerkungen zum Rechenbetrieb

W. Held

1. Betriebsstörungen

Den Benutzern unserer Rechenanlage mußten in den letzten drei Monaten zahlreiche Hardwareausfälle zugemutet werden. Für Ihre Geduld bedanken wir uns.

Nicht nur alte periphere Geräte sondern vor allem auch die moderne Zentraleinheit und die Plattenlaufwerke fielen oft und viel zu lange aus. Magnetbänder, die sonst als sichere Datenträger gelten, wurden gedehnt, Magnetplattenfehler führten zum Zusammenbruch des Betriebs, Fehler in der Zentraleinheit konnten erst nach Tagen beseitigt werden (zu Einzelheiten siehe Bericht von Herrn Reichel zur Hardwaresituation). Diese Instabilitäten der letzten Wochen entsprechen nicht dem Stand der Technik für Großrechner. Derartiges habe ich in meinem bisherigen Berufsleben noch nicht erlebt, und ich hoffe, daß sich der Betrieb in Kürze wieder stabilisiert. Die beiden mit der Instandhaltung und Instandsetzung beauftragten Firmen werden sich diesen Zustand nicht länger leisten können.

2. Engpässe im Dialogbetrieb

Besonders nachmittags treten häufig Engpässe und ungünstige Reaktionszeiten im Dialogbetrieb auf. Abhilfe ist vorläufig nur möglich, wenn verstärkt für die Arbeit am Terminal auf andere Tageszeiten ausgewichen wird. Die Rechenanlage ist auch am Abend und in der Nacht in Betrieb, das Rechenzentrum ist bis 22.00 Uhr geöffnet. Der Zugang zu den dezentral in Instituten aufgestellten Terminals sollte noch nach 19.00 Uhr möglich sein. Wir sind in der Lage, auf besonderen Antrag die Betriebsmittel z.B. in der Nacht etwas großzügiger zur Verfügung zu stellen als im Tagesbetrieb, wenn damit ein Abbau der Bedarfsspitzen erreicht werden kann. Bei Überlastungen des Nachtbetriebs können wir die Anlage auch am Wochenende eingeschaltet lassen.

3. Kürzungen der Haushaltsmittel

Die Haushaltsmittel der Universität (und damit auch des Rechenzentrums) werden im kommenden Jahr stark gekürzt werden. Wir hoffen, die nachteiligen Folgen möglichst in Grenzen halten zu können.

Nicht zu vermeiden ist eine Verschlechterung der Papierqualität. Dies ist wohl zumutbar, da ein Teil der Drucker-Ausgaben aus Testläufen stammt und ohnehin nicht länger aufbewahrt werden muß. Hochwertiges Papier wird nur noch in Ausnahmefällen (z.B. für Publikationen) bereitgestellt werden können.

Für verschiedene Geräte haben wir Wartungsverträge kündigen müssen. Dies gilt auch für einige Kartenlocher, die wir entsprechend gekennzeichnet haben. Bitte benutzen Sie die Locher nur, wenn keine anderen mehr frei sind.

Wir hoffen, daß wir im kommenden Jahr einige Arbeitsplätze im Rechenzentrum modernisieren können. Dabei denken wir in erster Linie an die Bereitstellung eines Mikro-Computersystems mit 8 - 10 Terminals, das zur Datenerfassung und für kleinere Programmläufe verwendet werden kann. Für größere Aufgaben wird ein Übergang auf den Großrechner geschaffen werden.

Auch durch diese Maßnahme wollen wir Kürzungen der Haushaltsmittel auffangen, denn es sollten gleichzeitig weitere Kartenlocher und andere alte periphere Geräte außer Betrieb gesetzt werden, für die die Folgekosten zu stark gestiegen sind.

Vortragsreihe über Mikrocomputer und Prozeßdatenverarbeitung

- | | | |
|-------------|---|-----------------------|
| 27. 10. 81 | Einführung in die Prozeßdatenverarbeitung
(Begriffe, Schwierigkeiten der Realtime-
verarbeitung) | Held |
| 3. 11. 81/ | Hardware für Mikrocomputer | Kisker/Schulze |
| 10. 11. 81 | (Schaltungsgrundlagen, Aufbau eines
Mikrocomputers, Ein/Ausgabebausteine) | |
| 24. 11. 81/ | Betriebssysteme und Compiler | Eickenscheidt/Pudlatz |
| 1. 12. 81 | (Datenverwaltung, Ein/Ausgabe-
organisation, Kommandosprache,
Programmiersprache PASCAL) | |
| 8. 12. 81/ | Ein-/Ausgabeschnittstellen und | Held/Schulze |
| 15. 12. 81 | Prozeßrechneranwendungen
(Start/Stop-Betrieb, BSC- und DIN 66019-
Protokolle, X.21- und X.25-Schnittstellen,
PDV-Bus, IEC-Bus, Camac-Schnittstelle,
Konfigurierung von Mikrocomputersystemen) | |
| 12. 1. 82 | Mikrocomputer und Textverarbeitung | Eickenscheidt |
| 19. 1. 82 | Prozeßrechner R30 des Rechenzentrums | Kisker |

Die Vorträge finden jeweils von 13.30 Uhr bis 15.00 Uhr im Hörsaal M4 des Fachbereichs Mathematik statt.

Lehrveranstaltungen im WS 81/82

Eine Übersicht mit kurzen Inhaltsangaben über das Lehrangebot des Rechenzentrums im WS 81/82 ist bereits in der Juli-Ausgabe von inforum erschienen.

Einführende Vorlesungen

- | | | |
|--------|---|--------------|
| 320015 | Einführung in die EDV
mo 13-15, Hörsaal M6 (Beginn: 26.10.81) | Kisker |
| 320049 | Programmieren in FORTRAN
mi 13.30-15, Hörsaal M5 (Beginn: 28.10.81) | Nienhaus |
| 320053 | Programmieren in FORTRAN
(für Mathematiker und Physiker)
mi 9-11, Hörsaal M5 (Beginn: 28.10.81) | Zörkendörfer |
| 320072 | Programmieren in PL/I
(numerische Anwendungen)
mi 15-17, Hörsaal M5 (Beginn: 28.10.81) | Neukäter |
| 320091 | Programmieren in PL/I
(nichtnumerische Anwendungen)
mi 11-13, Hörsaal M5 (Beginn: 21.10.81) | Kaspar |
| 320106 | Programmieren in PASCAL
di 15.30-17, Seminarraum Bibliothek
(Beginn: 27.10.81) | Bosse |

Weiterführende Vorlesungen

- | | | |
|--------|--|--|
| 320125 | Programmieren in PL/I für
Fortgeschrittene
mi 13-15, Hörsaal M6 (Beginn: 28.10.81) | Sturm |
| 320130 | Konzepte für Datenbank-Systeme
mi 11-13, Seminarraum Rechenzentrum
(Beginn: 28.10.81) | Meyer-Rinsche |
| 320144 | Kolloquium in Angewandter Mathematik
und Informatik
fr 15-17 oder 17-19,
Hörsaal M5 oder M6 | die wissen-
schaftlichen
Mitarbeiter des
Rechenzentrums |
| 320159 | Anleitung zum Einsatz der EDV bei
wissenschaftlichen Arbeiten | die wissen-
schaftlichen
Mitarbeiter des
Rechenzentrums |

Einführung in die Benutzung der Rechenanlage

Unter diesem Titel sind verschiedene Einzelveranstaltungen zusammengefaßt, die im WS 1981/82 jeweils montags von 13 bis 15 Uhr im Seminarraum der Bibliothek (Mathematik) stattfinden und allen interessierten Benutzern die Gelegenheit bieten, Kenntnisse für den praktischen Einsatz der Rechenanlage zu erwerben bzw. zu vertiefen. Eine Anmeldung ist in keinem Fall erforderlich.

Übersicht über die Termine und Einzelthemen:

- | | | |
|--------------------------|---|------------------|
| 26.10.1981 | Betrieb des Rechenzentrums
(maschinelle Ausstattung, betriebliche Regelungen, Bedienung von Geräten) | K. Reichel |
| 2.11.1981 | Aufbau eines Jobs für Standardaufgaben
(spezielle Monitorjobs, Aufruf katalogisierter Prozeduren, JOB-, JOBPARM-, ROUTE-, OUTPUT-, SETUP- und END-Anweisung) | R. Nienhaus |
| 9.11.1981 | Verwendung von katalogisierten Prozeduren
(EXEC-Anweisung, symbolische Parameter, Programm- und Benutzerbibliotheken) | K. Mertz |
| 16.11. und
23.11.1981 | Dateibeschreibung
(DD-Anweisung, Eingabedateien, Druck- und Stanzausgabe, Magnetband- und temporäre Magnetplattendateien) | K. Elix |
| 30.11.1981 | Dienstprogramm RUMSERV
(Anlegen von Magnetplattendateien, Aufruf einzelner Utilities) | R. Schmitt |
| 7.12. und
14.12.1981 | Dialogsystem VSPC
(Einsatzmöglichkeiten, einzelne Kommandos) | W. Kaspar |
| 21.12.1981 | Spezielle Dienstprogramme
(CA-SORT, Linkage Editor, Loader u. a.) | H. Meyer-Rinsche |

Computergesteuerter Satz wissenschaftlicher Texte

W. Kaspar

In Forschungsprojekten, die die EDV als Hilfsmittel einsetzen, steht man oft vor dem Problem, die zum Teil auf Datenträger gespeicherte Information auf möglichst einfache und kostengünstige Weise zu veröffentlichen. Der bisher meist gewählte Weg, die Daten mit einem Schnelldrucker zu drucken und anschließend von einem Setzer abschreiben zu lassen, ist meistens teuer und korrekturintensiv, insbesondere wenn es sich um größere Tabellenwerke etc. handelt. Um diese Nachteile zu umgehen, wurde in den letzten Jahren in Zusammenarbeit mit dem Rechenzentrum immer mehr der computergesteuerte Lichtsatz für die Veröffentlichung solcher Publikationen eingesetzt. Diese Technik ermöglicht es, die EDV-gemäß aufbereiteten Daten direkt, d.h. ohne die Arbeit eines Setzers, drucken zu können.

Die hierfür am Rechenzentrum zur Verfügung stehenden Programme können aber auch bei nahezu allen anderen Publikationen eingesetzt werden. Dadurch lassen sich die Satzkosten erheblich senken, und der Autor kann den Werdegang seiner Veröffentlichung über einen großen Zeitraum direkt mitverfolgen und beeinflussen.

Das Manuskript wird dann nicht mehr auf einer Schreibmaschine erstellt, sondern sofort an einem Datensichtgerät eingegeben. Der Autor erhält zur ersten Korrektur die Ausgabe eines z.B. an das Datensichtgerät angeschlossenen Druckers.

Zur Durchführung des computergesteuerten Umbruchs (d.h. Aufteilung des Textes in die zu druckenden Zeilen und Seiten) müssen in den Text Steuerzeichen eingefügt werden, die z. B. den Beginn von Kapiteln, Abschnitten, Absätzen etc. und den Anfang bzw. das Ende von Auszeichnungen und Anmerkungen kennzeichnen. Der umbrochene Text wird anschließend in simulierter Form auf einem Schnelldrucker ausgegeben. Dieser Ausdruck dient dem Autor zur inhaltlichen Korrektur aber auch zur Beurteilung des Zeilen- und Seitenaufbaus. Mit einfachen Änderungen von "Satzparametern" und Steuerzeichen kann in einem erneuten Umbruch die Gestaltung der Seiten beeinflusst werden.

Nimmt der Autor keine größeren Änderungen des Textes mehr vor, werden abschließend alle Trennungen, die vom Programm automatisch durchgeführt wurden, überprüft und ggf. berichtigt. Der korrigierte umbrochene Text wird dann so aufbereitet, daß er mit einer Lichtsatzanlage auf Film belichtet werden kann. Dieser Film wird i.a. an den jeweiligen Verlag weitergeleitet, der die Vervielfältigung und das Binden vornimmt.

Der eben skizzierte Arbeitsablauf und die zugehörigen Softwarepakete sind vom Rechenzentrum in Zusammenarbeit mit dem Sonderforschungsbereich 7 entwickelt worden, wobei viele automatisch gesteuerte Gestaltungsmöglichkeiten für die hohen satztechnischen Anforderungen der Publikationen des Sonderforschungsbereichs 7 entwickelt wurden. Seit dem letzten Jahr

wächst nun auch die Zahl von Autoren aus anderen Bereichen der Universität, die die Vorteile des computerunterstützten Satzes erkannt haben und die die vorhandenen Programmprodukte auch für ihre Veröffentlichungen einsetzen wollen.

Um alle Vorteile des am Rechenzentrum zur Verfügung stehenden Satzsystems ausschöpfen zu können, sollten von Seiten des Autors folgende Voraussetzungen erfüllt werden:

Zur Texterfassung sollte eine festangestellte Schreibkraft zur Verfügung stehen, die die Texte an Datensichtgeräten wie z. B. eines vom Typ IBM 3270 oder einem TANDBERG TDV 2114 bzw. SIEMENS 6.610 eingeben kann. Für die Textaufnahme können auch andere Datensichtgeräte eingesetzt werden, wenn sie entweder am VSPC angeschlossen oder mit ihnen Datenträger, z. B. Floppy-Disks, erstellt werden, die vom Rechenzentrum weiterverarbeitet werden können.

Weiterhin sollte der Autor konkrete Vorstellungen über die Gestaltung der Buchseiten haben, da er unter anderem entscheiden muß, wie groß die Höhe und Breite einer Seite (Satzspiegel) gewählt und mit welchen Schriften (Schriftfamilie, Schriftgrad) gedruckt werden soll. In Zusammenarbeit mit dem Rechenzentrum sollte dann geklärt werden, ob alle Anforderungen und Wünsche des Autors vom Satzsystem erfüllt werden können.

Der eigentliche rechnergesteuerte Satz wird vom Autor selbst oder einem seiner Mitarbeiter in Zusammenarbeit mit dem Rechenzentrum durchgeführt. Für die dabei anfallenden Arbeiten werden Kenntnisse der JCL, des Umganges mit größeren Datenbeständen (je nach Umfang der Publikation) und einiger Satzsteuerzeichen benötigt.

Da die Universität (noch) nicht über eine eigene Lichtsatzanlage verfügt, wird die Belichtung der fertig aufbereiteten Publikationen z. Zt. in Gütersloh bei der Firma Mohndruck auf einer DIGISET 40T2 durchgeführt. Die dadurch entstehenden Kosten für Filmmaterial und Arbeitszeit müssen vom Autor bzw. dem für die Veröffentlichung gewonnenen Verlag aufgebracht werden. Diese Kosten liegen i. a. erheblich unter denen für einen vergleichbaren Satz, der über einen Verlag abgewickelt wird.

Vergleicht man nun diese auf den ersten Blick vielleicht sehr umfangreich erscheinenden Anforderungen an den Autor und seine Mitarbeiter mit dem Arbeitsaufwand, der bei einer auf herkömmliche Weise veröffentlichten Publikation anfällt, ergeben sich für den computerunterstützten Satz am Rechenzentrum folgende Vorteile:

Der Text muß in seinen wesentlichen Teilen nur einmal geschrieben werden, d. h. Textänderungen und neue Abschnitte werden am bereits aufgenommenen Text durchgeführt, bzw. in

diesen eingefügt. Herkömmliche Schreibmaschinenmanuskripte mußten in solchen Fällen meistens komplett neu geschrieben werden. Diese Manuskripte mußten deshalb auch öfter korrekturegelesen werden, da beim erneuten Abschreiben wieder Schreibfehler auftreten konnten. Dieses wiederholte Korrekturlesen von nicht veränderten Textstellen entfällt bei Texten, die an Datensichtgeräten aufgenommen werden, da sie von Änderungen in anderen Teilen der Arbeit nicht berührt werden.

Alle aufgenommenen Texte können, soweit sie mit den erforderlichen Steuerzeichen versehen sind, zu jeder Zeit testweise umbrochen werden, d.h. der Autor erhält ein Schnelldruckerprotokoll, das die Gestaltung und den Text der Seiten, so wie sie später belichtet werden, simuliert. Der Autor kann also schon zu einem sehr frühen Zeitpunkt den Aufbau der Seiten, die Aufteilung des Textes und des Anmerkungssteils einer Seite, sowie den Seitenumfang seiner Publikation kontrollieren, ohne daß in diesem Stadium Kosten für einen Probesatz oder eine Probebelichtung entstehen.

Darüber hinaus können von ausgewählten Seiten auch Testbelichtungen angefertigt werden, die dem Autor ein exaktes Überprüfen der Seitengestaltung und der Schriftqualität ermöglichen.

Über ein flexibel zu handhabendes Steuerzeichensystem können jederzeit Änderungen am Satzspiegel und an den Schriftarten vorgenommen werden.

Ein wesentlicher Vorzug des computergesteuerten Satzsystems liegt darin, daß ähnlich wie bei der Textaufnahme der gesamte Text nur einmal inhaltlich überprüft werden muß, da Änderungen des Satzspiegels oder der Schrift keinen Einfluß auf den Inhalt der Veröffentlichung haben. Schreibfehler, die beim Handsatz durch wiederholtes Setzen von Textabschnitten erneut auftreten, sind hier nicht möglich. Dies erleichtert vor allem den Satz von Publikationen, deren fortlaufender Text erst durch ein DV-Verfahren erzeugt wurde (Wörterbuch, Konkordanz, rückläufiger Index, berechnete Tabelle, u.a.).

Insgesamt bietet der computergesteuerte Satz dem Autor die Möglichkeit, an einigen wesentlichen Produktionsabschnitten seine Vorstellungen über Inhalt und Gestaltung der Publikation direkt realisieren zu können.

VS FORTRAN

H. Pudlitz

VS FORTRAN ist eine IBM-Erweiterung der FORTRAN-Sprache, die 1977 als ANSI-Standard dokumentiert wurde. Sie enthält einige interessante neue Möglichkeiten, die über FORTRAN IV hinausgehen und darüber hinaus den Programmierer von einigen unlogischen Einschränkungen des 1966er Standards befreien. Obwohl VS FORTRAN mehr beinhaltet als der 77er-Standard, sollen im folgenden die Begriffe "VS FORTRAN" und "FORTRAN 77" synonym verwendet werden.

Seit August dieses Jahres ist VS FORTRAN in der Compiler-Version 1.1.0 am hiesigen Rechenzentrum implementiert (tatsächlich handelt es sich hierbei um eine "Null-Version", die im laufenden Benutzerbetrieb auf Fehler getestet werden soll). Für den Aufruf des Compilers bzw. die durch ihn übersetzten Programme stehen die von anderen Sprachen her bekannten Prozeduren (hier mit dem Sprachpräfix FORTV, vgl. Benutzerhandbuch, B3.2 Systematik der Sprachprozeduren) zur Verfügung. Da es sich - wie gesagt - um die erste Compiler-Version handelt, muß aus folgenden Gründen vor allzu großer Euphorie gewarnt werden:

Der Compiler hat einen relativ großen Speicherbedarf, so daß bei der Übersetzung sehr umfangreicher Programme Probleme auftreten können. In diesem Fall segmentiere man das Programm oder wende sich an die Beratung (Tel. 2486). Die Compiler-Version 1 enthält einige erkannte Fehler, um deren Behebung das Rechenzentrum ständig bemüht ist; aber auch bisher unbekannte Fehler können auftreten, die dem Rechenzentrum umgehend gemeldet werden sollten. Schließlich ist der von diesem Compiler erzeugte Code z.Zt. etwas weniger effizient als der des G1-Compilers.

Für die Optimierung des Codes sind mehrere Phasen vorgesehen, die aber in der Version 1 noch nicht implementiert sind. Deswegen sollte die im Kopfteil der Programmliste erscheinende und vom H-Extended-Compiler her bekannte Liste der Compiler-Optionen bzgl. der Option OPTIMIZE(0) vom Benutzer nicht modifiziert werden. Man probiere dagegen durch Setzen von PARMC=XREF in der EXEC-Karte die Ausgabe einer Cross-Reference-Liste der im Programm verwendeten Namen und Statementnummern. Von der Angabe PARMC=FREE (Gegensatz zu PARMC=FIXED) ist dagegen abzuraten, da diese Option zwar die Eingabe eines FORTRAN-Programmes im "freien" Format (d.h. nicht an die Spalten 7 - 72 gebundene Statements) erlaubt, was wegen der aber weiterhin erforderlichen Statementnummern, die sich dann im Programmtext nicht mehr deutlich absetzen, eher zu unübersichtlichen Quellprogrammlisten führt.

Bevor die neuen Sprachelemente von VS FORTRAN im Detail vorgestellt werden, sei noch auf einen Aspekt hingewiesen: FORTRAN ist nicht zuletzt deswegen die am weitesten verbreitete Sprache für naturwissenschaftlich-technische Anwendungen, weil für fast jede Hardware (sogar für Mikro-Rechner) Compiler vorhanden sind, die auf dem 1966er ANSI-Standard (FORTRAN IV) basieren, wohingegen es nur wenige Hersteller gibt, die mit ihren Anlagen

einen FORTRAN 77-Compiler anbieten. Anders gesagt: Bei bisherigen FORTRAN-Programmen konnte man davon ausgehen, daß die Übertragbarkeit (Portabilität) auf andere Rechenanlagen in hohem Maße gegeben war. Für in FORTRAN 77 geschriebene Programme, die u.U. extensiv von herstellerepezifischen Erweiterungen Gebrauch machen, beschränkt sich die Portabilität auf Anlagen dieses Herstellers (in unserem Falle: IBM). Benutzern unserer Anlage, die nicht daran denken, ihre Programme auf andere Rechenanlagen zu übertragen, kann man (z.Zt. unter obigem Vorbehalt) empfehlen, ihre Programme in VS FORTRAN zu schreiben, da einerseits mit verbesserten Nachfolgeversionen des Compilers gerechnet werden darf, andererseits die in FORTRAN 77 enthaltenen Möglichkeiten auch die des H-Extended-Compilers umfassen (dies trifft z.B. für die erweiterte Genauigkeit und die Erzeugung von Cross-Reference-Listen bereits zu und in Zukunft auch für die Code-Optimierung). Damit dürften die genannten Compiler (G1 und H-Extended) in absehbarer Zeit obsolet werden, zumal fast alle älteren FORTRAN-Sprachelemente in VS FORTRAN erhalten bleiben (Aufwärtskompatibilität).

Um mit der Vorstellung der neuen sprachlichen Ausdrucksmöglichkeiten von VS FORTRAN zu beginnen, betrachten wir die beiden folgenden Programme (links: FORTRAN IV, rechts: VS FORTRAN), die beide dasselbe leisten sollen:

```

REAL A(11)
READ (5,10) N, (A(I), I=1, N)
10 FORMAT (I3, 11F6.2)
X=-2.
20 IF (ABS(X).GT.1.) GOTO 30
Y=POLY(X, N, A)
GOTO 40
30 Y=0.
40 WRITE(6,50) X, Y
50 FORMAT (2F10.3)
X=X+.1
IF (X.LT.2.) GOTO 20
STOP
END

```

```

FUNCTION POLY(X, N, A)
REAL A(N)
S=0.
IF (N.LT.1) GOTO 20
DO 10 I=1, N
10 S=S*X+A(N+1-I)
20 POLY=S
RETURN
END

```

```

REAL A(0:10)
READ (5,10) N, (A(I), I=0, N)
10 FORMAT (I3, 11F6.2)
DO 20 X=-2., 2., .1
IF (ABS(X).LE.1.) THEN
Y=POLY(X, N, A)
ELSE
Y=0.
END IF
20 WRITE (6, '(2F10.3)') X, Y
END

```

```

FUNCTION POLY(X, N, A)
REAL A(0:*)
POLY=0.
DO 10 I=N, 0, -1
10 POLY=POLY*X+A(I)
END

```

Beide Programme tabellieren eine Funktion im Intervall $(-2, +2)$ in Schritten von 0.1. Die Funktion ist innerhalb von $(-1, +1)$ ein Polynom

$$p(x) = \sum_{i=0}^n a_i x^i$$

und außerhalb dieses Intervalls identisch Null. Die Koeffizienten des Polynoms, dessen Grad n maximal 10 sein soll, werden vorher eingelesen.

Im linken Programm (FORTRAN IV) muß der Koeffizientenvektor A deshalb mit der Dimension 11 deklariert werden und im folgenden READ-Statement muß zum Einlesen der 4 Koeffizienten eines Polynoms 3. Grades $N=4$ angegeben werden. Diese Umrechnung ist in VS FORTRAN nicht erforderlich, da Felder mit beliebigen ganzzahligen Indexbereichen durch "(Untergrenze : Obergrenze)" deklariert werden können. Falls die Untergrenze 1 ist, kann ihre Angabe und der Doppelpunkt (wie bisher allein üblich) entfallen. Unter- bzw. Obergrenze können auch negativ, jedoch darf die Obergrenze nicht kleiner als die Untergrenze sein. Dieser neuen Möglichkeit entsprechend fallen in VS FORTRAN eine Reihe von Einschränkungen bei den DO-Schleifen-Parametern weg. So dürfen Anfangs-, Endwert und Schrittweite auch negativ oder Null sein. Bei negativer Schrittweite wird die Laufvariable dekrementiert (vgl. das entsprechende Statement im Funktionsunterprogramm). Schließlich dürfen die DO-Schleifen-Parameter sogar arithmetische Ausdrücke sein. Ein weiterer wichtiger Unterschied bei der Abarbeitung von DO-Schleifen besteht darin, daß in VS FORTRAN die Abfrage, ob der Endwert bereits über- bzw. unterschritten ist, vor der Inkrementierung zu Beginn des Schleifendurchlaufs erfolgt. Dies hat zur Konsequenz, daß eine Schleife möglicherweise überhaupt nicht durchlaufen wird, wodurch sich entsprechende Abfragen vor Eintritt in die DO-Schleife in VS FORTRAN erübrigen (vgl. die beiden Versionen des obigen Funktionsunterprogramms). Im READ-Statement der VS FORTRAN-Version wird kein $A(I)$ eingelesen, wenn N negativ ist, während in der FORTRAN IV-Version mindestens $A(1)$ eingelesen wird.

Die im FORTRAN IV-Programm "ausprogrammierte" Schleife (X =Anfangswert, $X=X$ +Inkrement, $IF(X.LT.Endwert) GOTO Anfang$) kann in VS FORTRAN ebenfalls mit Hilfe des DO-Statements geschrieben werden. Kurz gesagt: Laufvariable und DO-Schleifen-Parameter dürfen vom Typ REAL sein, jedoch ist, was die Inkrementierung der Laufvariablen und das Erreichen eines Endwertes betrifft, Vorsicht geboten, da durch Abbruchfehler bei der internen Darstellung (Rundungsfehler) die vom Rechnen mit REAL-Größen her bekannten hardwarebedingten Probleme auftreten. In VS FORTRAN bleiben Schleifenvariable - anders als z.B. in PL/I oder PASCAL - auf die Typen INTEGER und REAL beschränkt, obwohl eine Erweiterung auf den in VS FORTRAN ebenfalls vorhandenen Typ CHARACTER denkbar gewesen wäre.

Variable vom Typ CHARACTER können Zeichenketten von 1 bis 500 Zeichen speichern. Die Länge einer Zeichenkettenvariable muß in einer Deklaration CHARACTER*n angegeben werden. Variable dieses Typs benötigen n Bytes Speicherplatz:

```

CHARACTER*80 KARTE,ZEILE
10 READ(5,20,END=30) KARTE
20 FORMAT (A80)
   ZEILE=' --->' // KARTE (40:50) // '<--- '
   WRITE (6,20) ZEILE
   GOTO 10
30 END

```

CHARACTER-Variable dürfen auch dimensioniert werden, jedoch ist eine Initialisierung (in dieser Compiler-Version?) nur in einer DATA-Anweisung, nicht aber in einer expliziten Typ-Spezifikation möglich. Konstante vom Typ CHARACTER*n werden in einfache Apostrophs eingeschlossen. (Die Formatangaben im ersten Beispiel haben die Form von CHARACTER-Konstanten). Für die Ein/Ausgabe ist das A-Format vorgesehen, das aber auch weiter für Variable anderen Typs verwendet werden darf. CHARACTER-Größen werden mit Hilfe des Verkettungsoperators (//) hintereinandergehängt. Ferner können Teilketten gebildet werden durch Angabe der Position des ersten und letzten Zeichens innerhalb einer gegebenen Kette, wobei diese beiden Angaben in Klammern und durch einen Doppelpunkt getrennt hinter die CHARACTER-Variable zu setzen sind. Im obigen Beispiel wird der Inhalt der Spalten 40 bis 50 jeder eingelesenen Lochkarte zwischen zwei Pfeilen ausgedruckt.

CHARACTER-Größen können durch die üblichen Vergleichsoperatoren miteinander verglichen werden, wobei ggfs. die kürzere durch Anhängen von Blanks auf die Länge der längeren gebracht wird. Insgesamt kann gesagt werden, daß durch diese Spracherweiterungen elementare Textverarbeitungsprobleme auch in FORTRAN ohne Schwierigkeiten lösbar sind. Zu diesem Beispiel sei abschließend vermerkt, daß das END-Statement als ausführbares Statement in VS FORTRAN gilt und somit eine Statementnummer haben darf. Bei fehlendem STOP übernimmt es auch die Funktion des STOP-Statements. Gleiches gilt für das END-Statement in Unterprogrammen, das die Funktion eines unmittelbar davorstehenden RETURN ersetzen kann. Beides erklärt das Fehlen von STOP und RETURN auch im ersten Beispiel.

Kommen wir zurück auf das erste Beispielprogramm: Im WRITE-Statement der VS FORTRAN-Version sieht man ferner, daß ein (nur einmal benötigtes) Format auch in Form einer Kettenkonstanten anstelle eines ein variables Format bezeichnenden Feldnamens geschrieben werden kann. Über erweiterte Möglichkeiten der Ein/Ausgabe in VS FORTRAN wird später berichtet werden.

Am Beispiel des Funktionsunterprogramms fällt auf, daß bei der (letzten) oberen Dimensionsangabe eines als formalen Parameter verwendeten Feldes ein Stern (*) stehen kann, da durch den

Übergabemechanismus "call by location" ohnehin auf dem Speicherplatz des im rufenden Programmsegment spezifizierten Feldes operiert wird. Die Prüfung auf Überschreitung der zulässigen Indexgrenze eines Feldes wird auch in VS FORTRAN nicht durchgeführt und bleibt wie bisher in der Verantwortung des Programmierers.

Für die alternative Wertzuweisung an die Variable Y, die in der FORTRAN IV-Version mit Sprunganweisungen (GOTO) realisiert ist, bietet sich in VS FORTRAN das "Block-IF"-Statement an. Die auf der folgenden Seite zuerst genannte Form ist als Erweiterung des logischen IF-Statements von FORTRAN IV aufzufassen und ist immer dann sinnvoll einzusetzen, wenn nach der IF-Klausel nicht nur ein, sondern mehrere Statements auszuführen sind. THEN und END IF haben hier klammernde Funktion. Ist der logische Ausdruck wahr, so werden die zwischen THEN und END IF stehenden Statements ausgeführt, sonst nicht.

<u>1. Form</u>	<u>2. Form</u>	<u>3. Form</u>
IF(log. Ausdr.) THEN	IF(log. Ausdr.) THEN	IF(log. Ausdr. 1) THEN
.	.	.
.	.	.
.	.	.
END IF	ELSE	ELSE IF(log. Ausdr. 2) THEN
	.	.
	.	.
	.	.
	END IF	ELSE
		.
		.
		.
		END IF

Mit der 2. Form ist eine einfache Alternative möglich, sie wird eingeleitet durch das Statement ELSE: Ist der logische Ausdruck wahr, so werden die zwischen THEN und ELSE stehenden Statements ausgeführt und dann mit dem auf END IF folgenden Statement fortgesetzt. Andernfalls werden nur die zwischen ELSE und END IF stehenden Statements ausgeführt.

In derselben Zeile wie ELSE darf ein einzelnes Statement nicht geschrieben werden, jedoch ist dort syntaktisch ein weiteres IF...THEN zulässig, das eine weitere Alternative einleitet (3. Form): Ist der erste logische Ausdruck wahr, so wird nur der zwischen IF...THEN und ELSE IF stehende Teilblock ausgeführt. Ist dies nicht der Fall, aber die auf ELSE IF folgende Abfrage, so wird nur der zwischen ELSE IF und ELSE stehende Teilblock ausgeführt, andernfalls nur der zwischen ELSE und END IF stehende Teilblock. Bei Bedarf können auf der gleichen Ebene mehrere ELSE IF-Statements mit ihren zugehörigen Teilblöcken folgen, und das letzte ELSE darf auch fehlen. Im letzteren Fall ist es dann aber möglich, daß keiner der im Block-IF enthaltenen Teilblöcke ausgeführt wird, nämlich dann, wenn keine der Abfragen wahr ist.

Aus einem Block-IF darf herausgesprungen oder innerhalb eines Teilblocks gesprungen werden, jedoch nicht von einem Teilblock in den anderen oder von außen in einen Block-IF hinein. Schließlich darf keines der Statements IF...THEN, ELSE IF, ELSE und END IF als letztes Statement einer DO-Schleife verwendet werden.

Durch das Block-IF-Statement werden die von anderen Sprachen her bekannten IF-THEN-ELSE-Konstruktionen möglich, womit auch in FORTRAN gewisse Aspekte der "Strukturierten Programmierung" Anwendung finden. Ein Vorteil der neuen Ausdrucksmöglichkeit ist offenkundig: die Anzahl der für Sprungziele benötigten Statementnummern kann drastisch reduziert werden; darüber hinaus werden Programme, besonders wenn noch DO-Schleifen-Bereiche oder Teilblöcke im Block-IF eingerückt werden, wesentlich besser lesbar als übliche FORTRAN-Programme mit zahlreichen GOTOS und an Spalte 7 "klebenden" Statements, obwohl die Möglichkeit des Einrückens logisch zusammenhängender Programmteile in jeder noch so alten FORTRAN-Version bestanden hat.

Das unscheinbare ABS(X) in Zeile 5 des ersten Beispiels sieht zwar genauso aus wie der entsprechende Funktionsaufruf in FORTRAN IV, es steckt aber etwas mehr dahinter: in FORTRAN IV ist man gezwungen, in Abhängigkeit vom Typ des Arguments der eingebauten Funktion die passende "spezifische" Funktion aus einer Familie von Funktionen zur Berechnung des Absolutbetrages einer reellen oder komplexen Zahl (ABS, IABS, DABS, CABS, CDABS) auszuwählen und diese u.U. in einem Spezifikationsstatement aufzuführen (z.B. REAL*8 DABS), falls der Funktionswert per vordefinierter Spezifikation als von dem tatsächlich berechneten Typ abweichender Typ interpretiert werden würde.

In VS FORTRAN reicht (obwohl auch das eben genannte Vorgehen möglich ist) die Angabe des "generischen" Funktionsnamens ABS für alle zulässigen Typen von Argumenten, wobei ggfs. auch noch eine von zwei weiteren Mitgliedern dieser Funktionenfamilie automatisch ausgewählt werden: QABS und CQABS. QABS steht für die Berechnung des Absolutbetrages einer "4-fach" genauen Gleitkommazahl, die durch REAL*16 deklariert werden kann und somit eine Mantisse von 30 Hexadezimalziffern, entsprechend 36 (!) Dezimalziffern, besitzt (analog gilt CQABS für komplexe Zahlen vom Typ COMPLEX*32). Derart erweiterte genaue REAL-Zahlen können z.B. als 1.57Q-5 angegeben werden mit einem Q (statt E bzw. D) zur Abtrennung des Exponenten; die Ein-/Ausgabe erfolgt entsprechend dem E- bzw. D-Format mit dem Q-Formatcode. Es sei jedoch darauf hingewiesen, daß es sich bei REAL*16 und COMPLEX*32 um eine IBM-spezifische Erweiterung handelt.

Das Konzept der generischen Funktionen gilt in VS FORTRAN für alle von FORTRAN IV bekannten Funktionsfamilien, so sind etwa MAX, MIN, SIN, COS, EXP, LOG usw. generische Namen. Leider ist dieses Konzept - anders als in PL/I - nicht voll durchgezogen: so kann man z.B. nicht LOG(I), sondern muß LOG(FLOAT(I)) für den natürlichen Logarithmus einer ganzen Zahl schreiben. Daneben existieren wieder die spezifischen Funktionen ALOG, DLOG, QLOG,

CLOG, CDLOG und CQLOG. Will man jedoch den dekadischen Logarithmus einer komplexen Zahl Z berechnen, so erleidet man mit LOG10(Z) Schiffbruch, da es zwar den generischen Namen LOG10 zu den spezifischen Namen ALOG10, DLOG10 und QLOG10 gibt, nicht aber zu CLOG10, CDLOG10 und CQLOG10, wobei die letzteren beiden wegen der überschrittenen, nach wie vor geltenden maximalen Namenlänge von 6 Zeichen sowieso anders hätten benannt werden müssen.

Betrachtet man die VS FORTRAN-Version des ersten Beispiels, so fällt auf, daß (abgesehen vom Format, das auch als CHARACTER-Konstante hätte ausgegeben werden können) die beiden einzigen Statementnummern lediglich zur Bezeichnung des Endes von DO-Schleifen benötigt werden: Hier hätte ein in die Sprache aufgenommenes "END DO" (ähnlich dem END IF) eine sauberere Lösung im neuen Sprachkonzept gebracht. Es mag ebenso bedauert werden, daß das DO-Statement in der Form der Zählschleife das einzige Schleifenstatement in VS FORTRAN geblieben ist, obwohl in anderen Sprachen längst WHILE- und REPEAT-Schleifen mit der Abfrage einer Bedingung am Anfang bzw. am Ende der Schleife zum täglichen Brot des Programmierers gehören. So kann man zwar in FORTRAN 77 einige GOTOs weniger schreiben, ohne sie kommt man aber weiterhin nicht aus.

Anträge auf Rechenerlaubnis für 1982

W. Bosse

Ein Blick in den Kalender zeigt es - nur noch wenige Wochen trennen uns vom 1.1.1982. Es wird deshalb Zeit, an verschiedene organisatorische Notwendigkeiten in Verbindung mit dem Zugang zur Rechenanlage zu erinnern.

Gemäß der Benutzungsordnung des Rechenzentrums hat jeder Benutzer die Pflicht, nach Ablauf der erteilten Rechenerlaubnis (spätestens am 31.12.1981 ist dies bei allen Benutzern der Fall) einen Jahresbericht für 1981 abzugeben über die unter jeder Verrechnungsnummer (=Benutzerkennung + Projektkennung) durchgeführten Arbeiten. Vielleicht erstellen Sie jetzt schon einmal einen Entwurf?

Um die Aktion der Verlängerung von bestehenden Verrechnungsnummern zeitlich etwas zu verteilen, wird das Rechenzentrum ab 16.11.1981 Anträge auf Rechenerlaubnis für 1982 bearbeiten, wobei die dazu bereitgestellten neuen Antragsformulare zu verwenden sind. Diese neuen Formulare werden Anfang November den Instituten zugesandt, werden aber auch im Dispatch des Rechenzentrums vorliegen. Obwohl keine grundsätzlichen Änderungen gegenüber dem Vorjahr vorgenommen wurden, sind die neuen Formulare aufgrund der gemachten Erfahrungen in einigen Teilen verändert (wir hoffen: verbessert) worden.

Außerdem müssen alle im Jahr 1982 weiterhin benötigten Magnetbänder, Stellplätze und archivierte Banddateien rechtzeitig, d.h. vor Jahresende, angemeldet werden.

Abschließend noch ein Hinweis: Die beantragte Dauer der Rechenerlaubnis sollte mit dem (geschätzten) Ende des im Antrag genannten Rechenvorhabens übereinstimmen, sofern dies vor dem 31.12.1982 liegt.

Sprechzeiten

Die Mitarbeiter des Rechenzentrums haben über ihre Beratungstätigkeit hinaus auch verschiedene andere Aufgaben im Rechenzentrum wahrzunehmen, die es mit sich bringen, daß sie nicht immer in ihrem Büro unter der angegebenen Telefonnummer erreichbar sind. Darum seien hier noch einmal - aber auch wegen der personellen Veränderungen der letzten Monate - Zeiten für eine erste Kontaktaufnahme genannt:

Name	Zimmer	Tel.	Zeit
Ahrens	H 25	2607	mo-do 9-11
Bendahn-Mertz	H 13	2682	di 15.30-17
Bosse	E 102	2461	n. V.
Eickenscheidt	H 31	2673	fr 14-15
Elix	E 112	2488	mo 11-12
Goorkotte	E 01	2672	mo-do 14-17
Dr. Exner	H 34	2686	mo 10-11
Dr. Held	E 104	3791	n. V.
Dr. Kamp	H 11	2474	di, do 11-12
Kaspar	H 14	2468	mi 11-12
Kisker	H 13	2682	do 10-12
Mecke	E 02	2466	mo-fr 9-12, 14-16
Dr. Mertz	H 16	2683	mi 11-12
Meyer-Rinsche	H 21	2685	do 9-11
Nabrotzki	H 14	2468	mi 8.30-11.30
Nienhaus	H 02	2483	di 13-14
Ost	E 112	2488	di 9-11
Dr. Pudlatz	H 15	2472	di 11-12, fr 10-11
Reichel	E 111	2481	mi 9.30-11.30
Schmitt	H 33	2475	di 9-11
Schulze	H 35	2608	do 10-11
Dr. Slaby	E 110	2681	n. V.
Spellmann	H 12	2473	mi 11-12
Dr. Steinhausen	H 36	2464	mo 10-11
Sturm	H 32	2609	mo 14-15
Dr. Zörkendörfer	H 01	2471	do 10-11

Gebäude:

H = Hittorfstraße 27

E = Einsteinstraße 60

Programme zur Multidimensionalen Skalierung (MDS) D. Steinhausen

Das an der University of Edinburgh entwickelte Programmpaket MDS (X) zur Multidimensionalen Skalierung: "The MDS(X) Series of Multidimensional Scaling Programs" steht seit kurzem am HRZ zur Verfügung. Es handelt sich dabei um eine Programm-Serie, die Programme zu verschiedenen Verfahren der MDS enthält. Diese Programme lassen sich dabei auf einfache Weise abrufen und werden durch eine SPSS-ähnliche Kontrollsprache mit den gewünschten Parametern versorgt. Im einzelnen sind folgende MDS-Programme und Verfahren in dem Paket enthalten:

CANDECOMP (CANonical DECOMPosition)
 HICLUS (HIERarchical CLUSTERing)
 INDSCAL-S (INDividual Differences SCALing)
 MDPREF (MultiDIMENSIONal PREFERence Scaling)
 MINICPA (Michigan-Israel-Nijmegen Integrated series: Conditional Proximity Analysis)
 MINIRSA (MINI Rectangular Smallest Space Analysis)
 MINISSA (Michigan-Israel-Nijmegen Integrated Smallest Space Analysis)
 MRSCAL (MetRIC SCALing)
 MVNDS (Maximum Variance Non-Dimensional Scaling)
 PARAMAP (PARAMetric MAPing)
 PINDIS (Procrustean INDividual Differences Scaling)
 PREFMAP (PREFERence MAPing)
 PROFIT (PROperty FITting)
 TRISOSCAL (TRIadic Similarities Ordinal SCALing)
 UNICON (UNIdimensional CONjoint measurement)

Nähere Auskünfte und Einsicht in das User Manual erhalten Sie bei D. Steinhausen (Tel. 2464).

ImpressumRedaktion inforum

W. Bosse (Tel. 83-2461)
 H. Pudlatz (Tel. 83-2472)
 W. A. Slaby (Tel. 83-2681)
 R. Schmitt (Tel. 83-2475)
 E. Sturm (Tel. 83-2609)

Satz: T. Engelbert, C. Icking
 Druck: H. Mecke

Rechenzentrum der Universität
 Einsteinstr. 60
 4400 Münster

Auflage dieser Ausgabe: 700
 Redaktionsschluß der nächsten Ausgabe: 31.12.1981

Hardwaresituation

K. Reichel

Seit Juli 1981 hat sich die Stabilität der Hardware spürbar verschlechtert. In folgender Tabelle sind die Ausfallzeit in Prozent der Betriebszeit, der mittlere Fehlerabstand in Stunden, die Anzahl der außerplanmäßigen IPL's und der Hauptfehlerverursacher aufgeführt:

Monat	Ausfall	Fehler- abstand	IPL	Ursachen
Juli	3,7 %	33 Std	15	Klimaanlage (3,5 Std) a) IBM-Director (12 Std) b) Memorex-Plattenfehler c)
August	4,4 %	37 Std	11	IBM-Director (14,5 Std) b) Memorex-Platten (6 Std) c)
September	13,3 %	14 Std	29	IBM-Stromversorgung (13 Std) d) IBM-Speicherfehler (51 Std) e) Memorex-Bandeneinheiten (2 Std) f)
Oktober (bis 15.)	5,0 %	11 Std	22	Memorex-Platten (11 Std) g)

Zu den einzelnen Fehlern und deren Beseitigung ist folgendes zu bemerken:

- Der Ausfall der Klimaanlage war bedingt durch im Laufe der Zeit verschlissene Motoren.
- Dieser Fehler trat regelmäßig nach dem Neustart am Wochenanfang auf. Zwei größere Reparaturen außerhalb der regulären Wartungszeit haben den Fehler verschwinden lassen.
- Systemneustarts wurden nötig, da vom System erwartete Rückmeldungen von den Platten ausblieben und zu Systemabstürzen führten. Der Fehler konnte bisher von der Firma Memorex noch nicht endgültig behoben werden.
- Im Stromversorgungsteil verschmorte eine Elektronikarte aus unbekannter Ursache.
- Intermittierende Fehler im Hauptspeicher ließen unter Hinweis auf (nicht vorhandene) Softwarefehler Programme unbegründet abbrechen. Dieser Fehler wurde durch die Firma IBM am Wochenende 3./4.10. nach mehrmaligen vorausgegangenen Fehlversuchen behoben und ist seitdem nicht mehr aufgetreten.
- Seit Mitte August wurden zunächst vereinzelt Bänder vor dem Bandanfang (vor den Datensätzen) gedehnt oder in der Mitte zerrissen. Anfang September häuften sich diese und andere Fehler (soeben beschriebene Bänder konnten nicht mehr gelesen werden), weshalb das Rechenzentrum unverzüglich einen Beschwerdebrief an die Firma Memorex verfaßte. Da sich der

- Zustand bis Mitte September verschlechterte, meldete das Rechenzentrum Regreßforderungen an. Seit dem 28.9. werden die Bandstationen repariert, schadhafte Teile ausgetauscht und Justagen vorgenommen. Seit Anfang Oktober schienen die Hauptfehler wie Bandzerreißen und Banddehnen behoben. Am 15.10. wurde ein weiteres Band zerrissen.
- g) Im Rahmen einer neuen Systemgenerierung wurde es nötig, einige Memorex-Platten (Wechselplatten von 200 MByte) auf den Einheiten umzulegen und neue Platten zu initialisieren. Durch Fehler an den Memorex-Platteneinheiten und Memorex-Plattenstapeln und Reparaturversuche erfolgten mehrere Systemabstürze. Die Fehlerbeseitigung durch die Firma Memorex und die Wiederherstellung defekter Platten durch das Rechenzentrum dauern noch an.

Personalia

Seit dem 16.7.1981 ist Herr Dr. H. Exner als wissenschaftlicher Mitarbeiter in der Abteilung Numerische Datenverarbeitung tätig. Herr Exner kommt aus Budapest und wird ein Jahr am Rechenzentrum arbeiten.

Zum 31.7.1981 hat Herr S. Hein seine Tätigkeit als studentischer Mitarbeiter des Rechenzentrums beendet.

Herr S. Ost, langjähriger studentischer Mitarbeiter des Rechenzentrums, hat nach erfolgreichem Abschluß seiner Diplomprüfung in Physik am 1.9.1981 eine Stelle als wissenschaftlicher Mitarbeiter in der Abteilung Systemsoftware übernommen.

Herr G. Plennis ist seit dem 1.7.1981 für seine Arbeit als Vorsitzender des Personalrats der nichtwissenschaftlichen Mitarbeiter der Universität beurlaubt.

Seit dem 1.9.1981 ist Herr R. Schmitt, der für eine Tätigkeit bei einer Bundesbehörde abgeordnet worden war, wieder als wissenschaftlicher Mitarbeiter in der Abteilung Systemsoftware tätig.

Neueste Meldung:

Wie wir soeben erfahren haben, schlossen unsere Kollegen Annette Benduhn und Dr. Klaus-Bolko Mertz am 20.10. den Bund fürs Leben, wozu wir beiden herzlich gratulieren.

Jobverteilung nach Job-CPU-Zeit und Fachbereich H. Schlattmann

Folgende Statistik schlüsselt die je Fachbereich gerechneten Jobs nach tatsächlich verbrauchter CPU-Zeit auf. Jobs, die keine CPU-Zeit verbraucht haben (z.B. wegen Password- oder JCL-Fehler) und Monitorläufe bleiben in der Tabelle unberücksichtigt, bei letzteren deshalb, weil die CPU-Zeit nicht den jeweiligen Benutzerkennungen zugeordnet werden kann.

Zeitraum: August - September 1981

Sekunden:	≤1	1-4	5-10	11-20	21-40	41-60
FB 1	8	10	5	0	0	0
FB 2	7	22	3	1	4	2
FB 3	0	0	0	0	0	0
FB 4	829	1470	998	603	262	58
FB 5	176	327	273	132	35	0
FB 6	952	1764	704	301	192	75
FB 7	9	76	5	0	0	0
FB 8	610	910	233	160	42	27
FB 9	858	1091	944	431	176	39
FB 10	622	1161	1192	385	396	137
FB 11	18	17	19	17	0	0
FB 12	136	124	49	9	11	0
FB 13	0	0	0	0	0	0
FB 14	1	1	6	0	0	0
FB 15	177	129	290	157	27	4
FB 16	647	2411	3337	1801	1290	225
FB 17	786	1845	755	432	141	55
FB 18	139	72	71	14	15	3
FB 19	348	821	390	471	121	42
FB 21-24	162	251	183	77	24	25
IFL	28	115	31	15	0	0
HRZ	2513	2484	1221	720	291	96
ZUV	357	1009	895	426	287	68
FH	0	1	0	3	0	0
Sonst.	544	986	161	34	1	0
	9927	17097	11765	6189	3315	856

Fortsetzung der Tabelle auf der folgenden Seite.

Sekunden:	60-120	120-300	300-600	600-1000	>1000	Summe
FB 1	0	0	0	0	0	23
FB 2	4	1	0	0	0	44
FB 3	0	0	0	0	0	0
FB 4	52	28	5	0	2	4307
FB 5	0	0	0	0	0	943
FB 6	45	49	15	8	0	4105
FB 7	0	0	0	0	0	90
FB 8	32	1	0	0	0	2015
FB 9	20	8	0	0	0	3567
FB 10	220	110	18	4	4	4249
FB 11	0	0	0	0	0	71
FB 12	0	1	8	0	0	338
FB 13	0	0	0	0	0	0
FB 14	0	0	0	0	0	8
FB 15	15	5	2	1	0	807
FB 16	361	342	394	418	56	11282
FB 17	41	30	15	25	14	4139
FB 18	0	0	0	0	0	314
FB 19	33	44	13	5	6	2294
FB 21-24	98	0	0	0	0	820
IFL	0	0	0	0	0	189
HRZ	127	112	45	46	20	7675
ZUV	53	7	0	0	0	3102
FH	0	0	0	0	0	4
Sonst.	0	0	0	0	0	1726
	1101	738	515	507	102	52112