

inforum

INFORMATIONSFORUM des Rechenzentrums der Universität Münster

Jahrgang 3, Nr. 1

Januar 1979

Inhalt

Ausfälle und Störungen in der Nahperipherie des Rechners	1
MVS - Ein neues Betriebssystem für die Rechenanlage	2
Kontrollanweisungen für einen Job im MVS	3
Virtuelle Maschinen	12
WATFIV-Version des NAG: Mark 6	14
Katalog der vorhandenen Programme	14
Lehrveranstaltungen	15
Personalia	16
Schriftenreihe des Rechenzentrums	17
Konfiguration der Rechenanlage	18
Führer durch das Rechenzentrum	20
Warum strukturierte Programmierung?	21
Numerische Datenverarbeitung im Rechenzentrum	24
Verbrauchsdaten 1978	26

Ausfälle und Störungen in der Nahperipherie des Rechners

P. Janßen

Ein wesentlicher Teil der Nahperipherie des neuen Rechners IBM 3032 im Rechenzentrum, nämlich alle Magnetbandgeräte und 90% der Magnetplattenspeicher, besteht aus Maschinen eines zweiten Herstellers; im einzelnen sind es:

- 6 Magnetbandgeräte TELEX 6420 mit zwei Steuereinheiten,
- 9 Magnetplattenspeicher TELEX 5312 mit einer Steuereinheit

und die Mitte dieses Jahres geliefert

- 4 Magnetplattenspeicher TELEX 6316 mit zwei Steuereinheiten,
- 2 Magnetplattenspeicher MEMOREX 3675 mit einer Steuereinheit.

Die europäische TELEX-Computer-Gesellschaft wurde im Frühjahr 1978 von der MEMOREX GmbH übernommen; die Firma MEMOREX ist seither für die Wartung der erstgenannten Geräte zuständig und lieferte auch die Ende 1977 bestellten Speicher TELEX 6316. Während die Mixed-Hardware-Verhältnisse unserer Rechenanlage in den vergangenen Jahren als zufriedenstellend beurteilt werden konnten, sind in den vergangenen Monaten, vor allem im Oktober, November und Dezember erhebliche Ausfälle und Störungen an jenen Geräten aufgetreten. Die Funktionsunfähigkeit jener Maschinen hat auch zu häufigen und längeren Ausfällen der Gesamtanlage geführt, wie jeder Benutzer gespürt hat. Während die neuen Speicher vom Typ MEMOREX 3675 die 30-tägige Funktionsprüfung nach Inbetriebnahme bestanden haben und zufriedenstellend funktionieren, sind die Plattenspeicher TELEX 6316/6833 4 1/2 Monate nach Installation noch nicht abgenommen worden. Vom Ausfall dieser Geräte sind die Benutzer zwar nicht unmittelbar betroffen worden. Da sich aber die Arbeiten zur Implementierung eines neuen Betriebssystems konzeptionell besonders auf diese Geräte stützen, haben sich hierbei erhebliche Verzögerungen ergeben, so daß nun erst Anfang Februar mit der Einführung des geplanten Betriebssystems gerechnet werden kann.

Mit der Firma MEMOREX sind über diese gravierenden Geräteausfälle mehrere Gespräche geführt worden; es bestehen offensichtlich Schwierigkeiten, nach der Übernahme der TELEX-Computer-Gesellschaft einen reibungslosen Betrieb jener Geräte sicherzustellen. Voraussichtlich werden die noch nicht abgenommenen Plattenspeicher in den nächsten Tagen ausgetauscht. Wegen der ansonsten für die nächste Zukunft zu befürchtenden Betriebsunsicherheit wird ferner erwogen, alsbald auch alle bisherigen TELEX-Geräte zu ersetzen. Kritisch könnte die Situation insbesondere bei den Magnetbandeinheiten werden, weil sie alle von diesem Typ sind.

MVS - Ein neues Betriebssystem für die Rechenanlage

H. Meyer

Voraussichtlich zwischen dem 9. und 12. Februar wird die Umstellung des Betriebssystems für die neue Rechenanlage erfolgen. Das Betriebssystem hört auf die Abkürzung MVS (für Multiple Virtual Storage) und nutzt die virtuelle Speichertechnik des Rechnerkomplexes IBM 3032.

Für den Rechenbetrieb im Bereich der Stapelverarbeitung bedeutet das die Weiterführung der bisherigen Organisation von Job-Abgabe und Ausgabe der gedruckten Ergebnisse; dabei wird als Nachfolger des bisherigen SPOOL-Programms HASP II das Job-

Eingabe-System JES2 eingesetzt, das einige Änderungen in den JOB- und JOBPARM-Anweisungen erforderlich macht. Auf diese Änderungen wird im Artikel "Kontrollanweisungen für einen Job im MVS" von W. Bosse in dieser Ausgabe von *inforum* eingegangen. Ferner wird sich das Aussehen der Systeminformationen in den Ausgabelisten ändern, die den Benutzer über verschiedene Anforderungen seines Jobs informieren. Eine Diskussion verschiedener Parameter wird erfolgen, sobald Erfahrungen mit dem neuen System vorliegen. Für Schwierigkeiten, die bei der Abwicklung von Programmen auftreten, die bisher ohne Beanstandungen bearbeitet wurden, wird eine intensive Beratung und Unterstützung eingerichtet werden.

Eine wesentliche Umstellung und Erweiterung erfährt die Dialogdatenverarbeitung: Das Betriebssystem verfügt über das Dialogsystem VSPC (Virtual Storage Personal Computing). Neben verschiedenen Datenstationen (Fernschreiber und Bildschirme) in einzelnen Instituten wird eine Reihe von Terminals im Rechenzentrum zur Verfügung stehen und bei der Programmentwicklung für alle Benutzer zugänglich sein. (Eine Übersicht über die räumliche Verteilung und die Gerätetypen soll in der nächsten Ausgabe von *inforum* erscheinen.) Einführungen in die Bedienung der Geräte und die Benutzung des Systems werden nach besonderer Ankündigung stattfinden.

Wichtige Komponenten des Dialogsystems sind die Kommandosprache zur Kommunikation mit dem System und zur Herstellung und Modifikation von Dateien (Files), der Interpreter für die Programmiersprache APL, die neben "Tischrechnerfunktionen" dem Benutzer der Datenstation die Erstellung einfacher Dialoge zur Lösung spezifischer Probleme gestattet, und ein Übersetzer für die Programmiersprache BASIC, die zur Erstellung von Vordergrundprogrammen einzusetzen ist.

Kontrollanweisungen für einen Job im MVS

W. Bosse

Der bevorstehende Übergang vom bisher verwendeten Betriebssystem MVT (Multiprogramming with a Variable Number of Tasks) zu dem neuen Betriebssystem MVS (Multiple Virtual Storage) wird aufgrund der Vorbereitungen des Rechenzentrums für den Benutzer in der Regel zu keinen größeren Umstellungsarbeiten führen, was den Aufbau der Jobs betrifft. Der gleichzeitige Wechsel von dem Job-Eingabe-System HASP (Houston Automatic Spooling Priority System) zu dessen Nachfolger JES2 (Job Entry System) bietet außerdem einige Annehmlichkeiten in bezug auf die Ausgabedateien.

Im folgenden werden die für den Aufbau eines Jobs wichtigen Änderungen aufgeführt; dies betrifft sowohl die Parameter der JCL-Anweisungen als auch die neuen Möglichkeiten in JES2.

JOB-Anweisung

Die JOB-Anweisung dient der Identifikation einer an die Rechenanlage gestellten Aufgabe (Job); sie muß deshalb auch weiterhin den Namen des Benutzers, seine ihm auf Antrag zugeteilte Verrechnungsnummer und die Bezeichnung des Ausgabe-fachs für die Druckausgabe enthalten. Außerdem wird in der JOB-Anweisung der Typ des Jobs festgelegt, der zu einer sinnvollen Klassifikation des Jobs innerhalb des Betriebssystems beiträgt.

Angaben über die gewünschte Rechenzeit (CPU-Zeit), die maximale Anzahl der zu druckenden Zeilen und der zu stanzenden Karten, spezielle Formulare, die Anzahl der Kopien, etc. können nicht mehr in der JOB-Anweisung erfolgen, sondern werden in der JOBPARM-Anweisung und teilweise auch in OUTPUT-Anweisungen für JES2 spezifiziert.

Der Parameter CLASS steht nicht mehr zur Verfügung, da über den JOB-Typ andersartige Aufteilungen der Jobs möglich sind und zur Zeit jeder Job bis zu 512 KBytes Hauptspeicherplatz benutzen kann.

Der Parameter MSGCLASS hat eine größere Bedeutung erhalten, da er jetzt auch die Ausgabe-kategorie bestimmter Ausgabedateien (SYSOUT=*) festlegt.

Format der JOB-Anweisung:

```
//jobnrxxx JOB (jobnr,type,box),name[,parameter]
```

jobnr die fünfstellige Verrechnungsnummer;

xxx 0 bis 3 weitere Zeichen (Buchstaben, Ziffern, @, #, \$), die in Verbindung mit **jobnr** den Job-Namen bilden;

type der Job-Typ, durch den bestimmte Anforderungen beschrieben werden; die zur Zeit definierten Typen sind:

LIST	Auflisten} der nachfolgenden Anweisungen
DUPL	Stanzen } und Daten;
SCAN	die JCL-Anweisungen des Jobs werden syntaktisch untersucht, ohne daß der Job zur Ausführung kommt;

ALGW	ALGOL-W-}
PL/C	PL/C- }
SNOB	SPITBOL-} Monitor-Lauf;
SPAS	SPASM- }
WATF	WATFIV- }

TEST	Testlauf;
PROD	Produktionslauf;

box die dreistellige Nummer des Ausgabefachs;

name der Name des Benutzers;

parameter die nachfolgend beschriebenen Parameter können je nach Bedarf die JOB-Anweisung ergänzen.

COND=((code,operator)[,...])

code eine Zahl zwischen 0 und 4095;

operator einer der Vergleichsoperatoren LT, LE, EQ, NE, GT.

Die Bearbeitung des Jobs wird abgebrochen, falls ein Programm einen Condition Code CC liefert, für den der Vergleich

code operator CC
zutrifft.

MSGCLASS=class

class ein Buchstabe zur Bezeichnung der Ausgabeklasse (implizit: A für den Drucker).

Dieser Parameter steuert die Ausgabe der System-Mitteilungen und der JCL-Anweisungen (wichtig für Benutzer eines Dialogsystems); außerdem werden alle Ausgabedateien, die durch **SYSOUT=*** beschrieben sind, der so angegebenen Ausgabeklasse zugeordnet.

MSGLEVEL=([jcl] [, alloc])

jcl Umfang der Liste mit den JCL-Anweisungen:

0 - nur die JOB-Anweisung,

1 - alle JCL-Anweisungen einschließlich der aufgerufenen Prozeduren,

2 - nur die vom Benutzer eingegebenen JCL-Anweisungen, implizit;

alloc Allocation/Termination-Mitteilungen sollen (1) oder sollen nicht (0, implizit) ausgegeben werden.

Unmittelbar hinter der JOB-Anweisung müssen die Kontrollanweisungen für das Eingabesystem JES2 stehen, deren allgemeine Form

/*Operation Operanden

ist und die mit dem Namen der Operation bezeichnet werden. Keine dieser Anweisungen darf über die Spalte 71 hinausgehen.

JOBPARM-Anweisung

Die JOBPARM-Anweisung gestattet es, für den Job bestimmte Grenzen festzulegen und Verarbeitungsbedingungen zu beschreiben. Viele der Angaben, die bei dem Betriebssystem MVT im Account-Feld erfolgten, sind jetzt auf die JOBPARM-Anweisung verlegt worden.

Die bisherigen Parameter EJECT/NOEJECT sowie RESTART/NOERSTART stehen nicht mehr zur Verfügung (für letztere gibt es aber eine modifizierte Form).

Format der JOBPARM-Anweisung:

```
/*JOBPARM,parameter[,...]
```

parameter die nachfolgend beschriebenen Angaben;
(mögliche Abkürzungen sind in Klammern angegeben).

TIME=sec (T)	die geschätzte Rechenzeit (CPU-Zeit) in sec; diese Angabe muß in den Grenzen liegen, die für den gewählten Job-Typ gemäß unten angegebener Tabelle festgelegt sind; es wird empfohlen, die Zeit möglichst genau anzugeben;
LINES=lines (L)	die geschätzte maximale Zeilenzahl (in 1000) der Druckausgabe des Jobs; siehe die unten angegebene Tabelle bzgl. der für jeden Job-Typ festgelegten Grenzen;
CARDS=cards (C)	die geschätzte maximale Anzahl der von dem Job zu stanzenden Karten; die unten angegebene Tabelle beschreibt die für jeden Job-Typ festgelegten Grenzen;
COPIES=copies (N)	die Anzahl der Exemplare der Druckausgabe des Jobs (implizit 1);
LINECT=count (K)	die Anzahl der Zeilen pro Seite für die Druckausgabe des Jobs (implizit 72);
NOLOG (J)	die Ausgabe des SYSTEM LOG (mit den Mitteilungen, die auf der Operatorkonsole erscheinen) wird unterdrückt;
FORMS=forms (F)	durch die Angabe einer vierstelligen Nummer können bestimmte Formulare für die Druckausgabe angefordert werden (die Verwendung einer solchen Nummer ist vorher mit dem Rechenzentrum abzustimmen);
TAPES=tapes	die Anzahl der für den Job benötigten Magnetbandlaufwerke (nicht Magnetbänder!);

RESTART=Y oder N die Bearbeitung des Jobs soll vollständig neu
(E) begonnen werden, falls vor seinem Abschluß das System abgestürzt ist und nach erfolgtem Systemstart weder mit einem Step noch bei einem Checkpoint die Bearbeitung fortgesetzt werden kann (Y, implizit); ein nicht vollständig bearbeiteter Job soll nicht automatisch neu gestartet werden (N);

PROCLIB=name kennzeichnet eine für den Job zu verwendende
(P) Bibliothek katalogisierter Prozeduren.

Jobtyp	Rechenzeit (sec)		Zeilen (in 1000)		Karten	
	implizit	maximal	implizit	maximal	implizit	maximal
TEST	10	20	1	10	0	500
PROD	60	1000	2	60	0	10000

Für die Job-Typen LIST, DUPL und SCAN gelten keine Einschränkungen; die Monitor-Job-Typen werden in ihren Grenzen durch die jeweils angesprochenen Monitorprogramme gesteuert.

ROUTE-Anweisung

Die ROUTE-Anweisung beschreibt, wo die Druckausgabe des Jobs erfolgen soll.

Format der ROUTE-Anweisung:

```
/*ROUTE,,PRINT,station
```

station die Druckausgabe des Jobs soll im Rechenzentrum (LOCAL oder R0) oder über eine durch die Ziffer n gekennzeichnete Remote-Station (REMDTEN oder Rn) erfolgen.

Die beiden Operanden müssen in Spalte 10 (PRINT) bzw. Spalte 16 (station) beginnen.

OUTPUT-Anweisung

Mittels der OUTPUT-Anweisung können für einzelne oder mehrere Ausgabedateien, die durch einen Code gekennzeichnet worden sind, spezielle Parameter gesetzt werden.

Format der OUTPUT-Anweisung:

```
/*OUTPUT_code_parameter[,...]
```

- code** ein aus bis zu vier Zeichen bestehender Name, durch den die Verbindung zu der Ausgabedatei hergestellt wird; in der die Ausgabedatei definierenden DD-Anweisung wird **code** über den Parameter **SYSOUT** angegeben;
- parameter** die nachfolgend angegebenen Parameter überschreiben für die betroffenen Dateien entsprechende Werte, die evtl. in der **JOBPARM-** bzw. **ROUTE-**Anweisung stehen; (mögliche Abkürzungen sind in Klammern angegeben).
- COPIES=copies** (N) die Anzahl der Exemplare der Druckausgabe der bezeichneten Dateien;
- FORMS=Forms** (F) durch die Angabe einer vierstelligen Nummer können bestimmte Formulare für die Druckausgabe der bezeichneten Dateien angefordert werden (die Verwendung einer solchen Nummer ist vorher mit dem Rechenzentrum abzustimmen);
- DEST=station** (D) die Ausgabe der bezeichneten Dateien soll im Rechenzentrum (R0) oder über eine durch die Ziffer n gekennzeichneten Remote-Station (Rn) erfolgen.

Die Beschreibung einzelner Verarbeitungseinheiten mit den dazugehörigen Programmen und Dateien geschieht über die JCL-Anweisungen **EXEC** und **DD**. Die dabei möglichen Parameter haben sich bis auf wenige Ausnahmen nicht geändert, so daß sich eine vollständige Beschreibung dieser Anweisungen erübrigt.

EXEC-Anweisung

In der **EXEC**-Anweisung beschreibt der Benutzer, welche katalogisierte Prozedur zur Lösung seiner Aufgabe aufgerufen werden soll. Es wird möglich sein, diese Prozeduren wie bisher zu verwenden, da das Rechenzentrum die notwendigen Umstellungen vornimmt. Allerdings können gewisse Schwierigkeiten beim Übergang auf das neue Betriebssystem **MVS** nicht ausgeschlossen werden, so daß die vom Rechenzentrum unterstützten Programme (u.a. **ALGOL W**, **APL**, **FORTRAN IV G1** und **H-Extended**, **PL/C**, **PL/I Optimizing Compiler**, **Plot-Software**, **Sort/Merge**, **SPSS** und **WATFIV**) mit Vorrang verfügbar gemacht werden; das betrifft auch alle **Utilities**, die durch katalogisierte Prozeduren aktivierbar sind.

Im **MVS** werden die Übersetzerprogramme **VS BASIC** und **PL/I Checkout Compiler** neu zur Verfügung stehen; sie können über folgende katalogisierte Prozeduren, die der Standardform entsprechen, verwendet werden:

BASICC	BASICO	
BASICCE		
BASICE	BASICX	
PLCCE	PLCCEF	PLCCES
PLCCL	PLCCLF	
PLCCLE	PLCCLEF	PLCCLES
PLCX	PLCXF	PLCXS

Im Zusammenhang mit der neuen Möglichkeit, die Ausgabe-Klasse von SYSOUT-Dateien über den MSGCLASS-Parameter der Job-Anweisung zu steuern, werden die symbolischen Parameter CPRINT und LPRINT in veränderter Form angegeben; für EXECUTE-Steps wird der entsprechende Parameter EPRINT neu eingeführt.

CPRINT=D oder '**
LPRINT=D oder '**
EPRINT=D oder '**

D die Ausgabe wird unterdrückt (entspricht DUMMY im MVT);

** die Druckausgabe erfolgt über die im MSGCLASS-Parameter bezeichnete Ausgabe-Klasse.

Alle anderen symbolischen Parameter werden unverändert übernommen.

Format der EXEC-Anweisung:

```
//[stepname]_EXEC_procedure[,parameter]...
```

stepname ein aus bis zu acht Zeichen bestehender Name;

procedure der Name einer katalogisierten Prozedur;

parameter die in der bezeichneten katalogisierten Prozedur definierten symbolischen Parameter können die EXEC-Anweisung ergänzen (außerdem die Parameter COND und PARM).

DD-Anweisung

Die DD-(Datei-Definition-)Anweisung gibt eine Beschreibung der in einem Job-Step verwendeten Dateien und spezifiziert deren Ein/Ausgabe-Möglichkeiten. Über den in dieser Anweisung angegebenen Namen (ddname) kann die Datei von dem ausführenden Programm angesprochen werden.

Format der DD-Anweisung:

```
//ddname DD parameter[,...]
```

Aus der Vielzahl der möglichen Angaben (**parameter**) sollen hier nur einige neue Parameter beschrieben und Hinweise zu bestimmten Werten gegeben werden.

Wie bereits in Verbindung mit der OUTPUT-Anweisung in JES2 erwähnt, kann für eine Ausgabedatei über den SYSOUT-Parameter ein Code definiert werden, der die Verbindung zu einer OUTPUT-Anweisung herstellt.

SYSOUT=(class[, ,code])

class ein Buchstabe zur Beschreibung der Ausgabe-Klasse (A - Drucker, B - Kartenstanzer) oder *, wenn die im MSGCLASS-Parameter der JOB-Anweisung spezifizierte Ausgabe-Klasse verwendet werden soll;

code ein aus bis zu vier Zeichen bestehender Name, durch den die Verbindung zu einer OUTPUT-Anweisung hergestellt wird.

OUTLIM=limit

limit die maximale Anzahl der Sätze, die in die Ausgabedatei aufgenommen werden sollen.

Dieser Parameter kann in Verbindung mit SYSOUT verwendet werden.

FREE=CLOSE oder END

Wenn eine Datei nicht für die gesamte Dauer des Job-Steps benötigt wird, so kann diese im Programm abgeschlossen und in dem Betriebssystem wieder frei verfügbar gemacht werden (CLOSE); implizit bleibt die Datei dem Programm für die Dauer des gesamten Steps zugeordnet (END). Nach Möglichkeit sollten Betriebsmittel nicht länger als nötig beansprucht werden.

Wenn der DCB-Parameter in Ergänzung zu SYSOUT verwendet wird, dann sollte keine Blockung angegeben werden, d.h. der Wert des DCB-Unterparameters RECFM enthält kein B.

Da zur (Zwischen-)Speicherung von Dateien auf Magnetplatten drei verschiedene IBM-Plattentypen (2314, 3330, 3350) zur Verfügung stehen, wird empfohlen, den DCB-Unterparameter BLKSIZE so zu setzen, daß unabhängig vom Plattentyp jede Spur möglichst optimal belegt wird. Für Sätze im Lochkartenformat (LRECL=80) bieten sich dazu die folgenden Blocklängen (BLKSIZE) an:

6160, 3120 oder 1680.

Bei anderen Satztlängen kann man anstreben, den Blockungsfaktor derart zu wählen, daß die angegebenen Werte (von unten) annähernd erreicht werden; in keinem Fall sollte ein Block mehr als 6233 Bytes enthalten (zur Orientierung: 6 K = 6144).

Um auch bei der Platzreservierung Schwierigkeiten mit den verschiedenen Plattentypen zu vermeiden, bietet es sich an, im SPACE-Parameter die durchschnittliche Blocklänge als Maß zu verwenden (statt TRK oder CYL). Gleichzeitig kann ROUND in der fünften Unterparameter-Position angegeben werden; das Betriebssystem berechnet dann den Speicherplatz in Zylindern des tatsächlich zugeordneten Plattentyps.

Beispiel: SPACE=(6160,70,,,ROUND)

Es werden 70 Blöcke der Länge 6160 benötigt (Kapazität für über 5000 Sätze im Lochkartenformat); durch ROUND ergibt sich folgende Reservierung:

Plattentyp	2314	3330 II	3350
Anz. der Zyl.	4	2	1

Eine temporäre Datei wird wie bisher durch den Parameter UNIT=WORK einer Magnetplatte des Typs 2314 zugeordnet - so sieht es der Benutzer. In Wirklichkeit wird eine dadurch gekennzeichnete Datei in den Paging-Platz aufgenommen, d.h. sie befindet sich ganz oder teilweise im Hauptspeicher; dadurch wird die Ein/Ausgabe von Sätzen etwa dem Zugriff auf die Elemente eines im Programm beschriebenen Feldes vergleichbar (virtuelle Ein/Ausgabe). Es erübrigt sich demnach, Zwischenergebnisse zwecks größerer Effizienz durch Ausnutzen des größeren Hauptspeicherbereiches selbst im Programm zu verwalten. Allerdings können index-sequentielle Dateien nicht auf diese Weise gespeichert werden. Durch die Angabe UNIT=DISK werden Magnetplatten vom Typ 2314 beschrieben (keine virtuelle Ein/Ausgabe).

Der UNIT-Unterparameter SEP zur Aufteilung mehrerer Dateien auf verschiedene Magnetplatteneinheiten wird in dem Betriebssystem MVS nicht mehr benötigt und deshalb ignoriert.

p.s. An einer neuen Version des Benutzerhandbuchs wird zur Zeit gearbeitet; mit der Fertigstellung ist erst nach Erscheinen der April-Ausgabe von inforum zu rechnen, da das Benutzerhandbuch Hinweise und Empfehlungen enthalten soll, die sich auf den praktischen Einsatz des Betriebssystems MVS mit JES2 sowie auf die Verwendung des Time-Sharing-Systems VSPC beziehen. Deshalb sind auch die bevorstehenden Änderungen in diesem Artikel ausführlich dargestellt worden. Weitere Hinweise wird die nächste Ausgabe von inforum enthalten.

Virтуelle Maschinen

H. Stenzel

Um ihre Großrechner effektiv nutzbar zu machen, werden von der IBM Betriebssysteme (Systemkontrollprogramme) angeboten, die in unterschiedlichem Ausmaß die real vorhandenen Betriebsmittel den Anwendungsprogrammen gegenüber "virtualisieren". Das bedeutet, daß der Rechner so genutzt werden kann, als seien mehr Betriebsmittel verfügbar als wirklich vorhanden.

Davon unterscheidet sich das bei uns bisher verwendete Multiprogramming. Mit ihm wird das Betriebsmittel CPU-Zeit mehreren, gleichzeitig aktiven Rechneranwendungen so zugeordnet, daß, soweit möglich, Ein/Ausgabe-Einheiten nicht die Nutzung der Zentraleinheit (CPU) limitieren.

Eine Virtualisierung der Papierperipherie wird beim bekannten Spooling mit HASP vorgenommen: nach Bedarf wird den Anwendungsprogrammen die Benutzung von Kartenleser und -stanzer und Schnelldrucker durch HASP vorgespiegelt, die Daten werden auf Plattenbereichen zwischengespeichert und die Ein/Ausgabe-Operationen werden, von den betrieblichen Gegebenheiten abhängig, zu anderer Zeit auf vorhandenen Geräten realisiert.

Die Hardware der /370-Rechner virtualisiert zusammen mit den Betriebssystemen das Betriebsmittel Hauptspeicher. Anwendungsprogramme verfügen über virtuelle Speicher (Adressräume), die durch dynamische Adreß-Umsetzung (DAT) auf den realen Speicher abgebildet werden. Darüber hinaus werden Speicherbereiche (Seiten), die von den Anwendungsprogrammen im Augenblick nicht benötigt werden oder die Platz belegen, der von Anwendungen mit höherer Priorität belegt werden soll, auf Magnetplatten ausgelagert (Paging).

Der Kontrollprogrammteil CP des VM370 (darüber hinaus beinhaltet VM noch weitere Komponenten, die ihrerseits in virtuellen Maschinen laufen) erreicht mit der Architektur des Systems /370 eine noch weiter gehende Virtualisierung, indem die gesamte maschinenorientierte Schnittstelle von Betriebssystemen vervielfacht wird. Jedem Benutzer (Betriebssystem) wird das funktionale Äquivalent eines vollständigen /370-er Systems zur Verfügung gestellt. Da diese Geräte nicht real vorhanden sind, spricht man von virtuellen Maschinen. Das bedeutet für CP, daß die Hauptspeicherseite 0, in der hardware- und peripherie-abhängige Daten stehen, die Kontrollregister der Zentraleinheit und die Zeitnahmeeinrichtungen für jede virtuelle Maschine simuliert werden müssen. Dadurch wird es VM370 möglich, auf einer Anlage mehrere, auch unterschiedliche Betriebssysteme parallel ablaufen zu lassen.

Die hauptsächlichlichen Komponenten einer virtuellen Maschine sind:

Virtuelle Konsole: Die Schalter und Bedienungselemente eines realen Systems werden für den Operateur auf einem Dialogterminal simuliert.

Virtueller Speicher: Dynamische Adressumsetzung und Paging simulieren für jede virtuelle Maschine einen eigenen, von den anderen Maschinen nicht beeinflussbaren Hauptspeicher.

Virtueller Prozessor: Ein Zeitscheibenalgorithmus gibt jeder konkurrierenden virtuellen Maschine aufgrund ihrer Priorität und des geschätzten Bedarfs einen möglichst fairen Anteil an CPU-Zeit.

Virtuelle Ein/Ausgabegeräte: Virtuellen Maschinen können, auch unter anderen Kanaladressen, reale Geräte zur alleinigen Benutzung zugeordnet werden (Dedizierte Geräte). Papierperipherie-Ein/Ausgabe kann von CP zwischengespeichert werden (Spooling). Für Magnetplattenspeicher und Dialogterminals existieren noch weitere flexible Zuordnungsmöglichkeiten.

Der zusätzliche Verwaltungsaufwand (Overhead) durch die gestiegene Komplexität des Systems wird dadurch in engen Grenzen gehalten, daß VM und Betriebssysteme der virtuellen Maschinen sich aufeinander einstellen (VM-VS-Handshaking) und so die Verdopplung von Funktionen vermieden wird.

Darüber hinaus kann VM bei der Ausführung von einigen, immer wiederkehrenden Funktionen mit speziellen Mikro-Instruktionen (VM Assist) durch die Hardware unterstützt werden.

Durch seine Reduzierung auf die Schaffung einer virtuellen Maschinenumgebung für die Betriebssysteme, unter denen die eigentlichen Anwendungsprogramme laufen, ist CP verhältnismäßig wenig umfangreich und kann besonders leicht an gegebene Systemkonfigurationen angepaßt werden.

Das Universitätsrechenzentrum ist durch die Installation von VM als Systemkontrollprogramm in der Lage, den unter MVT etablierten Betrieb beizubehalten und gleichzeitig die Möglichkeiten der neuen Hardware zu nutzen. Parallel zum Routinebetrieb können Modifikationen und Erweiterungen des Betriebssystems - insbesondere die Umstellung auf MVS - geplant, implementiert und getestet werden, ohne daß MVT-Benutzer oder -Operateure dadurch beeinträchtigt werden.

WATFIV-Version des NAG: Mark 6

U. Ebert

Mit der Einführung des neuen Betriebssystems Anfang Februar wird die alte WATFIV-Version der NAG-Bibliothek durch die neue erweiterte Version Mark 6 (einfache und doppelte Genauigkeit) ersetzt. Dadurch stehen die folgenden 17 Routinen auch für WATFIV-Benutzer nicht mehr zur Verfügung (vgl. inforum Nr. 4, Jahrgang 2):

D02AEF	E04CAF	F01BGF	F02AZF	S17ABF	S20AAF
E02AAF	F01ARF	F02ACF	S13ABF	S18AAF	S20ABF
E02ABF	F01ASF	F02ASF	S17AAF	S18ABF	

Gleichzeitig sind aber dafür alle 64 neuen Routinen, die die Version Mark 6 zusätzlich zur Version Mark 5 enthält auch bei Benutzung des WATFIV-Compilers aufrufbar. Somit ist der Umfang der NAG-Bibliothek für WATFIV-Benutzer und für Benutzer des GI- oder H-extended FORTRAN-Compilers identisch. Dadurch ist es wieder in jedem Fall möglich, FORTRAN-Programme, die Routinen aus dem NAG aufrufen, mit dem WATFIV-Compiler zu testen. Sollte der Zeitbedarf über die für einen Monitor-Job (Parameter WATF auf der JOB-Karte) zur Verfügung stehende Rechenzeit hinausgehen, so können diese Tests mit Hilfe der Prozedur WATFIV wie folgt durchgeführt werden:

```
//XYZ01WAT JJB (XYZ01,TEST,P20),NAME
// EXEC WATFIV
//EXECUTE.SYSIN DD DATA
//WATFIV TIME=10
  Programm
//DATA
  Daten
/*
//
```

ENDKARTE

Katalog der vorhandenen Programme

U. Ebert

Ab Ende Januar steht der Katalog der im Rechenzentrum benutzbaren Programme zur Verfügung. Er enthält alle Compiler, Dienstprogramme, Programmsysteme, Programmpakete und darüber hinaus einzelne Programme. Die Gliederung des Katalogs orientiert sich im großen und ganzen an der Share-Klassifikation. Pro Einheit des Katalogs (ein Programm oder ein anderes zusammenhängendes Stück Code) sind im einzelnen aufgeführt: eine Katalognummer, der Name, eine Kurzbeschreibung, die Sprache, der Typ (z.B. Haupt- oder Unterprogramm), ein Hinweis für den Zugriff und ein Verweis auf eine ausführliche Dokumentation.

Der Stand des Katalogs ist der vom 1.12.78. Er bezieht sich also auf das damals laufende Betriebssystem. Die durch die Neueinführung des Betriebssystems sich ergebenden Änderungen und

die weitere Fortschreibung des Katalogs werden Mitte 1979 vorgenommen. Es ist dann auch daran gedacht, den Katalog als Software-Information des Rechenzentrums herauszugeben. Bis dahin liegt je ein Katalog in der Programmierberatung des RZ und in der Bibliothek des RZ aus. Die Programmierberatung ist während der üblichen Zeiten auch zu Rückfragen und weitergehenden Erläuterungen bereit, insbesondere ist dort ein Großteil der ausführlichen Dokumentation einzusehen.

Lehrveranstaltungen

In der vorlesungsfreien Zeit am Ende des WS 1978/79 werden vom Rechenzentrum wieder einige ganztägige Intensivkurse 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 29.1. bis 16.2.1979

bei H. Mecke im Dispatch des Rechenzentrums (ZI 02) in die Anmelde Listen einzutragen.

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

Programmieren in 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: Herden, Reichel, Sturm

Vorl.-Nr.: 220015, 220020

Hörsäle: M1-M5

Zeit: 12.-23.2.1979, mo-fr,

Beginn: 12.2.1979, 9 Uhr c.t. im M1

Programmieren in 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 effektive Bearbeitung großer Datenmengen.

Dozenten: Ebert, Pudlatz

Vorl.-Nr.: 220034, 220049

Hörsäle: M2, M4-M6

Zeit: 5.-23.3.1979, mo-fr,

Beginn: 5.3.1979, 9 Uhr c.t. im M2

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 statistischen Verfahren sind Kenntnisse auf dem Gebiet der Statistik nützlich.

Dozenten: Steinhausen, Zörkendörfer
 Vorl.-Nr.: 220053
 Hörsaal: M3
 Zeit: 12.-27.3.1979, mo-fr.
 Beginn: 12.3.1979, 9 Uhr c.t. im M3

Die Teilnehmer der Kurse werden gebeten, diese im SS 1979 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 5. bis 16.2.1979

durch W. Bosse (Tel.: 83-2476).

Personalie

Folgende neue Mitarbeiter gehören dem Rechenzentrum an:

seit 1.11.1978

Herr R. Herrmann als studentischer Mitarbeiter.

seit 1.12.1978

Herr G. Plennis als Operateur.

seit 1.1.1979

Frau R. Nienhaus als wissenschaftliche Mitarbeiterin.

Wir heißen die Kollegin und Kollegen willkommen.

Ausgeschieden sind seit dem Erscheinen unserer letzten Ausgabe:

zum 31.10.1978

Herr Dr. K. Langer.

zum 30.11.1978

Herr K. Kämpers und

zum 31.12.1978

Herr H. Clausing

Wir wünschen Ihnen weiterhin viel Erfolg.

Nach bestandenem Examen, zu dem wir herzlich gratulieren, gehören dem Rechenzentrum seit 1.12.1978 Frau A. Benduhn und Herr W. Herden als wissenschaftliche Mitarbeiter an, die bisher beide als studentische Mitarbeiter bei uns tätig waren.

Schriftenreihe des Rechenzentrums

W.A. Slaby

Seit der letzten Übersicht über Neuerscheinungen im *inforum* Nr. 3, Jahrgang 2, sind in der Schriftenreihe des Rechenzentrums folgende Beiträge erschienen und können bei Frau M. Luth dienstags und donnerstags in der Zeit von 10.30 Uhr bis 11.30 Uhr käuflich erworben werden:

- Nr. 30 - Computerized Braille Production
Proceedings of the 2nd International Workshop in
Copenhagen (Denmark), September 1974
edited by H. Werner, June 1978; 69 Seiten; 6.-DM
- Nr. 31 - Die longitudinale Linienmethode für die ausgeartete
nichtlineare Fokker-Plancksche Differentialgleichung
von U. Hornung, September 1978; 95 Seiten; 7.-DM
- Nr. 32 - Simulation von Gleitkomma-Arithmetik mit beliebiger
Basis und Mantissenlänge
von U. Hornung und R. Többicke, Juni 1978; 16 Seiten;
2.-DM
- Nr. 33 - Interpolation by Smooth Functions under Restrictions
for the Derivatives
by U. Hornung, August 1978; 16 Seiten; 3.-DM
- Nr. 34 - Netzhautbildgröße und Kontaktlinsenüberkorrektion
von S. Zörkendörfer, Januar 1979; 40 Seiten; 5.-DM
- sowie der bereits angekündigte Beitrag
- Nr. 15 - DMP (Data Management Program)
Ein Programmsystem zur Verwaltung strukturierter Daten
von H.-W. Klöcker, Oktober 1978; 120 Seiten; 8.-DM

Außerdem sei darauf hingewiesen, daß inzwischen die Nummern 3-8, 10-12, 14, 16, 18, 20, 23 vergriffen sind; eine Neuauflage eines einzelnen dieser Beiträge ist nur bei entsprechender Nachfrage geplant.

Konfiguration der Rechenanlage

(I=IBM, T=TELEX, M=MEMOREX)

Anz.	Typ	Beschreibung
1	I 3032-06	Rechner mit 6 MBytes Hauptspeicher und 2 Kanal-gruppen von je 1MPX- und 5 BMPX-Kanälen
1	I 3027-01	Stromversorgungs- und Kühleinheit
1	I 3036-01	Konsoleinheit
4	T 6316-12	Magnetplattenlaufwerke mit je 317,5 MioBytes Speicherkapazität (SK), 1198000 Bytes/sec Übertragungsgeschwindigkeit (ÜG), 25 msec mittlere Zugriffszeit (mZZ)
2	M 6833	Steuereinheiten für 6316
2	M 3675	Magnetplattendoppelllaufwerke mit je 2 mal 200 MioBytes SK, 806000 Bytes/sec ÜG, 27 msec mZZ
1	M 3673	Steuermodul für 3675
1	M 3672	Steuereinheit für 3675
8	T 5312	Magnetplattenlaufwerke mit je 29 MioBytes SK, 312000 Bytes/sec ÜG, 30 msec mZZ
1	T 5328	Steuereinheit für 5312
8	I 2312/13/14	Magnetplattenlaufwerke mit je 29 MioBytes, 312000 Bytes/sec ÜG, 60 msec mZZ
6	T 6420-7	Magnetbandlaufwerke 320000 Bytes/sec ÜG, 1600 oder 800 bpi Aufzeichnungsdichte, 9 Spuren
2	T 6803-1	Steuereinheiten für 6420
3	I 1403-N1	Zeilendrucker mit 1100 Zeilen/min maximaler Druckleistung, 132 Druckpositionen/Zeile
1	I 2501-B2	Kartenleser mit 1000 Karten/min Lese- geschwindigkeit
1	I 2540-01	Kartenleser/-stanzer mit 1000 Karten/min Lese- geschwindigkeit; 300 Karten/min Stanz- geschwindigkeit
1	I 2520-B2	Kartenstanzer mit 500 Karten/min Stanz- geschwindigkeit
1	I 2671-01	Lochstreifenleser mit 1000 Zeichen/min Lese- geschwindigkeit
1	I 3540-B1	Diskettenleser/-beschrifter
1	I 2821-02	Steuereinheit für 1403
1	I 2821-05	Steuereinheit für 2 mal 1403 und 2540
1	I 2848-03	Steuereinheit für optische Anzeigen
1	I 3272-02	Steuereinheit für lokale Datensichtgeräte
1	I 3274-A01	Steuereinheit für lokale Datensichtgeräte
2	I 3705-E04	Datenfernverarbeitungssteuereinheiten

Führer durch das RechenzentrumRechenanlage:

siehe dazu den Artikel "Konfiguration der Rechananlage" in dieser Ausgabe von inforum

Einrichtungen und Zeiten:

- * Kartenlocher stehen im Durchgang zu den mathematischen Instituten und sind mo-fr 8.00-22.00 zugänglich. Längere Belegzeiten als eine Stunde und alle Störungen sind unbedingt sofort bei H. Mecke (Zi 02) bekannt zu machen. Bei Abwesenheit von H. Mecke wende man sich an die Operateure.
- * Job-Eingabe erfolgt durch die Benutzer. Der Kartenleser steht auf der rechten Seite vor dem Maschinensaal. Die Eingabe ist möglich mo, mi, do, fr 8.00-22.00 di 12.00-22.00.
- * Plot-Preview: der Bildschirm mit Hardcopy-Einrichtung steht im Raum 01. Plot-Preview läuft mo, mi, do, fr 8.00-16.00 di 12.00-16.00.
- * Dialogsysteme wird nach Einführung des neuen Betriebssystems bekanntgegeben.
- * Anmeldungen, Informationen bei H. Mecke (Zi 02) mo-fr 9.00-12.00, 14.00-16.00
- * Bibliothek enthält vorwiegend Informatik- und Systemliteratur; sie ist di, do 14.00-17.00 geöffnet und befindet sich in der Hittorfstr. 27 (Zi 23). Es ist nicht möglich, Bücher zu entleihen.
- * Beratung
 - Programmierberatung im Raum 06
 - mo, mi, do, fr 10.00-12.00
 - mo-fr 14.00-17.00
 - Problembberatung in der Hittorfstr. 27. Zur Kontaktaufnahme wende man sich an:
 - D. Steinhausen (Zi 36) mo 10.00-11.00
 - S. Zörkendörfer (Zi 01) mi 10.00-11.00
 - U. Ebert (Zi 31) fr 10.00-11.00

Systemberatung in der Hittorfstraße 27, II. Obergeschoß. Zur Kontaktaufnahme wende man sich an:
A. Ahrens (ZI 25)
mo-fr 9.00-11.00
Magnetbänder bei H. Goorkotte (ZI 111)
mo-do 14.00-17.00

* autom. Anrufbeantworter enthält aktuelle Informationen bei Unregelmäßigkeiten im Rechenbetrieb.
Tel. 2458 zu jeder Zeit.

Impressum

Redaktion inforum

W. Bosse (Tel.: 83-2476)
U. Ebert (Tel.: 83-2678)
R. Schmitt (Tel.: 83-2475)
W. A. Slaby (Tel.: 83-2473)
H. Wessels (Tel.: 83-2681)

Satz: U. Kaiser
Druck: H. Mecke

Rechenzentrum der Universität
Roxeler Straße 60
4400 Münster

Auflage dieser Ausgabe: 800
Redaktionsschluß der nächsten Ausgabe: 21.3.1979

Warum strukturierte Programmierung?

A. Benduhn, S. Stiller

Bereits im Beitrag von J. Ebert im inforum Nr. 1, Jahrgang 2, wurde auf die Eignung der Struktogramme zur Dokumentation strukturierter Programme hingewiesen. Hier soll versucht werden, den Begriff der "strukturierten Programmierung" etwas zu erhellen.

Es handelt sich dabei weder um eine Religion noch um eine Vermeidung von Sprunganweisungen ("GO TO"s). Man kann auch in FORTRAN strukturiert programmieren. Dies für alle, die, da sie nur FORTRAN kennen, jetzt bereits weiterblättern wollen. Wer allerdings gerne trickreich arbeitet und sich (auch heute noch) für Programme begeistert, die möglichst geheimnisvoll sind, nun, der soll hier nicht weiter seine Zeit verschwenden; denn strukturierte Programme beinhalten keine Geheimnisse und sind für jedermann lesbar! Ein möglicher Weg zur Erstellung solcher Programme soll hier kurz skizziert werden.

Am Anfang steht die gegebene Aufgabe, die meist eher zu kurz als zu ausführlich formuliert sein wird. Daraus wird eine genaue Beschreibung der eventuell vorhandenen Daten und des gewünschten Ergebnisses formuliert. Diese Aufgabenbeschreibung wird im Top-Down-Verfahren schrittweise verfeinert, d. h. das vorliegende komplexe Konzept wird durch mehrere einfache Teilkonzepte ersetzt. Handwerkszeug dazu sind die zum Entwurf der Struktogramme eingeführten Kontrollstrukturen: Anweisung, Anweisungsblock, Iteration und bedingte Anweisung. Aus einer Aufgabe entstehen dabei mehrere in sich abgeschlossene Teilaufgaben (siehe Beispiel am Ende des Artikels). Dabei ist den Schnittstellen dieser Teilaufgaben besondere Aufmerksamkeit zu schenken, da das Ergebnis der vorangegangenen Teilaufgabe die Daten der folgenden Aufgabe bildet. Daten und Ergebnisse der Teilaufgaben sollten also mit der gleichen Akribie behandelt werden wie die Eingangsbeschreibung der Gesamtaufgabe.

Diese Forderung mag dem, der seine Programme im Eigenbau erstellt, übertrieben erscheinen. Aber schon, wenn man an einer Stelle zur Lösung einer Teilaufgabe auf bestehende Unterprogramme aus Programmbibliotheken zugreift, gilt es, den Schnittstellen besondere Beachtung zu schenken. Es lohnt sich z. B. nicht, eine symmetrische Matrix zu erstellen, wenn das benutzte Unterprogramm so gut geschrieben ist, an dieser Stelle nur eine Dreiecksmatrix als Vektor gespeichert zu verlangen. Unbedingt notwendig wird die Beachtung der Schnittstellen, wenn größere Programme in Teamarbeit erstellt werden.

Der Feinheitegrad der Struktogramme sollte dabei so sein, daß jeder Programmierer, der sich eventuell später mit dem fertigen Programm beschäftigen muß, anhand der Struktogramme erkennt, was im Programm passiert.

Hier fällt dem aufmerksamen Leser auf, daß damit ein wesentlicher Teil der Programmdokumentation bereits gegeben ist, bevor das Programm überhaupt geschrieben wird.

Die Endstufe der Struktogramme wird dann in die gewählte Programmiersprache übertragen. Dabei sollte der Zusammenhang zwischen Programm und Struktogramm durch

1. reichliche Kommentare,
2. mnemotechnischen Code,
3. optisch klare Aufteilung mit Einrückungen,
4. Unterprogramme, die genau einer Teilaufgabe des Top-Down-Verfahrens entsprechen,

möglichst klar hervorgehoben werden.

Es ist klar, daß nicht alle Forderungen in allen Programmiersprachen beliebig gut erfüllt werden können. Der Mehraufwand, der sich beim Ablocken der Programme durch diese Forderungen ergibt, wird sich in der anschließenden Testphase und bei allen späteren Änderungen aber sicher bezahlt machen.

Außerdem können Sie Ihr Programm mit den Struktogrammen ohne langatmige Erklärungen der Beratung vorlegen, falls Sie Ihre Fehler einmal nicht selbst finden.

Literaturhinweise:

R.B. KIEBURTZ

Structured Programming and Problem solving with Algol W
Prentice-Hall Inc., New Jersey, 1975

W. JORDAN, H. URBAN

Strukturierte Programmierung;
Springer-Verlag; Berlin, Heidelberg, New York, 1978

GEWALD, HAAKE, PFADLER

Software Engineering, Grundlagen und Technik rationeller
Programmentwicklung
Reihe Datenverarbeitung

R. Oldenburg Verlag; München, Wien, 1977

SCHNUPP, FLOYD

Software Programmentwicklung und Projektorganisation
W. de Gruyter; Berlin, New York 1976

Beispiel für Top-Down-Entwurf anhand eines "programmierten Alltages":

Stufe 1

Beschreibung eines Tagesablaufs (= Aufgabenstellung)

Stufe 2

1. Beschr. des Morgens
2. Beschr. des Vormittags
3. Beschr. des Nachmittags
4. Beschr. des Abends

Stufe 3

1.1 Aufstehen
1.2 Frühstück
2.1 Arbeiten
2.2 Mittagessen

3.1 Arbeiten
3.2 Abendessen
4.1 ...
4.2 ...

Stufe 4

1.1.1 solange, bis wach	
Klingeln der vollautomatischen Weckuhr abwarten	
nervt das Klingeln?	
ja	nein
Weckautomatik kurz vor dem Ablaufen?	Augen öffnen "wach" sein
ja	
Wecker auf "nochmal Wecken" stellen	endgültig Augen öffnen und "wach" sein
1.1.2 schmutzig?	
ja	nein
duschen	↓
1.1.3 Zähne putzen	
1.1.4 anziehen	
1.2.1 Kater?	
ja	nein
Kaffee kochen	Tee kochen
1.2.2 Tisch decken	
1.2.3 essen	
1.2.4 abräumen	

Das Beispiel soll hier abgebrochen werden, da man bereits sieht, daß sich jede Aufgabe durch Verfeinerung beliebig genau erklären läßt.

Numerische Datenverarbeitung im Rechenzentrum

H. Wessels

In diesem Artikel soll über den Service des Rechenzentrums im Bereich numerischer Anwendungen berichtet werden. Es wird dabei Bezug genommen auf in Zusammenarbeit mit dem Rechenzentrum bearbeitete Projekte. Aufgrund der Vielfachheit möglicher Anwendungen wird dabei keine Vollständigkeit angestrebt.

I. Numerische Mathematik

- a) Beratung bei Auswahl und Einsatzmöglichkeiten von Unterprogrammen aus Programmbibliotheken.
- b) Beratung bei der Auswahl und Unterstützung bei der Programmierung geeigneter mathematischer Verfahren.
- c) Entwicklung und Dokumentation eigener Programme für spezielle Anwendungen im Bereich der numerischen Mathematik.

Anwendungen sind etwa:

- die Anpassung von Kurven an Meßwerte
- Berechnung von Eigenwerten bei großen Systemen
- Integration von Differentialgleichungen
- Berechnung von (Ausgleichs-) Splines
- Berechnung von Approximationen
- Lineare und nichtlineare Optimierung

II. Statistik

- a) Beratung und Unterstützung bei dem Einsatz von Programmsystemen wie SPSS.
- b) Beratung und Unterstützung bei der Interpretation der ermittelten Ergebnisse.
- c) Einführungsveranstaltungen zum Einsatz von Programmsystemen wie SPSS.
- d) Beratung bei der Auswahl geeigneter statistischer Verfahren.

Anwendungen in diesem Zusammenhang sind etwa:

- Auswertung von Fragebögen
- Aufstellung und Testen geeigneter Hypothesen
- Ermittlung empirischer Verteilungen
- Aufsuchen von Zusammenhängen z. B. zwischen Arbeitslosigkeit und gewissen Strukturmerkmalen der Bevölkerung

III. Medizinische Anwendungen

Über Anwendungen aus dem Bereich der Medizin soll in einem späteren Artikel ausführlicher berichtet werden.

IV. Spezielle Anwendungen

Für einige spezielle Anwendungen sind am Rechenzentrum Programme entwickelt worden.

- a) Computer-Kartographie
- b) Clusteranalyse

Verbrauchsdaten_1978

W. Schafmann

Diese Übersicht gibt eine Zusammenstellung der Verbrauchsdaten auf dem alten Rechner IBM /360-50 im ersten Halbjahr und auf der neuen Rechanlage IBM 3032 im zweiten Halbjahr.

	HASP- Jobs	Batch- Jobs	Seiten- Zahl	Gestanzte Karten	CPU- Zeit[%]

FB 1	719	31	6979	7234	0.7
FB 2	16	0	184	0	0.0
FB 4	18305	1165	235939	318032	12.5
FB 5	995	88	22367	26825	0.9
FB 6	11227	128	261314	64684	9.6
FB 7	1130	0	9274	47259	0.9
FB 8	1680	161	28667	37924	1.2
FB 9	1580	68	24614	22359	0.7
FB 10	4019	120	77385	55894	3.8
FB 11	365	0	7218	4461	0.2
FB 12	560	0	10244	5699	0.2
FB 14	5	0	23	0	0.0
FB 15	5175	967	56934	84968	3.5
FB 16	26272	1497	263939	605468	31.8
FB 17	9994	463	159580	258786	15.0
FB 18	929	202	10026	21023	0.7
FB 19	6508	874	76487	98757	2.8
IFL	695	22	18036	21294	0.2
RZ	14235	1997	167458	261807	6.4
ZUV	9863	2	698177	204153	6.9
KURSE	27257	26751	114857	49549	

Lieber Leser,

wenn Sie inforum regelmäßig beziehen wollen, bedienen Sie sich bitte des unten angefügten Abschnitts.

Hat sich Ihre Anschrift geändert oder sind Sie am weiteren Bezug von inforum nicht mehr interessiert, dann teilen Sie uns dies bitte auf dem vorbereiteten Abschnitt mit.

Bitte haben Sie Verständnis dafür, daß ein Versand außerhalb der Universität nur in begründeten Einzelfällen erfolgen kann.

Vielen Dank!

Redaktion inforum

An die
Redaktion inforum
Rechenzentrum
der Universität
Roxeler Str. 60
4400 Münster

Absender:
Name: _____
FB: _____ Institut: _____
Straße: _____
Außerhalb der Universität: _____

Ich bitte um Aufnahme in den Verteiler.

Bitte streichen Sie mich aus dem Verteiler.

Meine Anschrift hat sich geändert.
Alte Anschrift:

.....
(Datum)

.....
(Unterschrift)