

# inforum

---

-----  
INFORMATIONSFORUM des Rechenzentrums der Universität Münster

Jahrgang 3, Nr. 4

Oktober 1979  
-----

## Inhalt

Verlängerung der Rechenerlaubnis	2
Arbeiten am Betriebssystem	3
Automatisches Auslagern von VSPC-Dateien	3
Neue JCL-Prozeduren	5
Speicherung privater Dateien auf Magnetplatten	5
Prozeduren für VSPC-Benutzer	7
Änderungen bei der Verarbeitung von Disketten	9
NAG-Bibliothek: Erweiterte Version (Mark 7)	10
Benutzerhandbuch	11
Beschreibung der Plottersoftware	11
Schriftenreihe des Rechenzentrums	12
Bekannte Fallstricke im VSPC	13
Plot-to-Print	14
Personalia	15
Lehrangebot des Rechenzentrums im WS 1979/1980	16
RUMSERV - Ein neues Dienstprogrammkonzept	21
RUMSERV-Befehle zur Verwaltung privater Magnetplattendateien	28
Bearbeitungsreihenfolge von Rechenaufträgen	35
Datenstationen an der Rechenanlage	37
Turnaround-Zeiten (Juni bis August)	40

**Verlängerung der Rechenerlaubnis**      P. Janßen, S. Zörkendörfer

Um eine zweckentsprechende Nutzung der Rechenanlage und Betriebsmittel des Rechenzentrums im Sinne der Benutzungsordnung zu gewährleisten, werden einige Zuteilungen des Rechenzentrums nur befristet ausgesprochen und jeweils bei Bedarf auf Antrag neu vergeben und verlängert. So erlöschen zum Jahresende alle Rechenerlaubnisse und Aufgabennummern, abgesehen von jenen, die im Rahmen einer Lehrveranstaltung des WS 79/80 vergeben wurden oder nach dem 1.11.79 für ein neu beantragtes Rechenvorhaben erteilt wurden. Damit erlischt zum Jahresende auch die mit einer Aufgabennummer verbundene Genehmigung für eine VSPC-Benutzung. Ferner sind die Zuteilung der Magnetbänder und ihrer Stellplätze, der Ausgabe-Schließfächer, der Ablage-Schließfächer im Keller sowie der Lochkartenschränke und die Vorhaltung von Archiv-Dateien bis zum Jahresende befristet.

Obwohl die Verwaltung der Benutzerdatenbank durch zwangsweise vorübergehende Abmietung von Software (vgl. *inforum* Jg. 3, Nr. 2 (April 1979)) stark beeinträchtigt ist, wird das Rechenzentrum auch dieses Jahr bemüht sein, den Benutzern das Verfahren der Verlängerung möglichst einfach zu gestalten. Wir werden die betroffenen Benutzer Mitte November über die jeweilige Hochschuleinrichtung, auf die sich die drei Kennbuchstaben der Aufgabennummer beziehen, anschreiben:

- \* Sie erhalten ein (zum Teil von uns vorausgefülltes) Formular zur Verlängerung der Aufgabennummer und gegebenenfalls der VSPC-Kennung, sowie ein Formular für die Zusammenstellung weiterhin benötigter Magnetbänder.
- \* Alle Benutzer, die 1979 eine gültige Rechenerlaubnis hatten, werden aufgefordert, der Benutzungsordnung entsprechend einen Jahresbericht einzureichen.

Jahresberichte und - sofern gewünscht - Verlängerungsanträge für Aufgabennummern, VSPC-Kennungen und Magnetbänder sollten dann vollständig ausgefüllt und unterschrieben bis zum 15.12.1979 im Rechenzentrum eingegangen sein. Das Rechenzentrum schreibt die betroffenen Benutzer - wieder über die jeweilige Hochschuleinrichtung - zum Jahresende an und informiert dabei über die erfolgte oder nichterfolgte Verlängerung der Rechenerlaubnis.

Aus den Erfahrungen der letzten Jahre heraus scheint es uns angezeigt, Sie auf folgende Punkte aufmerksam zu machen:

- \* Das Absenden der Formulare wird im Rechenzentrum durch Aushang bekanntgegeben; Fragen Sie im Sekretariat Ihrer Hochschuleinrichtung gegebenenfalls nach!
- \* Achten Sie auf das vollständige Ausfüllen der unteren Hälfte des Verlängerungsformulars (Angabe zur Finanzierung, Aufgabengruppe gemäß Benutzungsordnung, Name des

Unterzeichners, Institutsstempel, Unterschrift); bei unvollständig ausgefüllten Anträgen kann die Rechenerlaubnis nicht verlängert werden, da die Erklärung über die Einordnung des Vorhabens nach der Benutzungsordnung unbedingt erforderlich ist.

\* Auch das Vorliegen des Jahresberichts für 1979 ist Voraussetzung für eine Verlängerung.

\* Halten Sie bitte den Termin 15.12.1979 ein, ansonsten kann eine Verlängerung zum 1.1.1980 und damit eine unterbrechungslose Benutzung der Rechanlage für Ihr Rechenvorhaben nicht gewährleistet werden.

#### Arbeiten am Betriebssystem

P. Janßen

Aufgrund der neuen Betriebssystemkonzeption können Betriebssystemarbeiten zum größten Teil während des normalen Rechenbetriebes abgewickelt werden. In den vergangenen Monaten sind Systemarbeiten, die zu Unterbrechungen führen können, mit Rücksicht auf die Arbeiten der Benutzer in den Abendstunden oder am Wochenende durchgeführt worden. Solche Systemarbeiten, die die gesamte Anlage beanspruchen, sind in letzter Zeit seltener geworden und nur noch gelegentlich notwendig gewesen, wie die Benutzer wohl festgestellt haben.

Da auch in Zukunft nicht beabsichtigt ist, eine feste Zeitspanne für derartige Systemarbeiten zu reservieren, in der dann kein allgemeiner Rechenbetrieb stattfinden würde, werden gelegentlich Arbeiten, die zu Störungen führen können, auch weiterhin in den Abendstunden durchgeführt werden. Wir betrachten dies als einen tragbaren Kompromiß zwischen dem Wunsch des Benutzers nach einem ungestörten Rechenbetrieb während seiner Dienstzeit und dem Wunsch unserer Mitarbeiter, ihre Tätigkeiten weitgehend innerhalb der normalen Dienststunden auszuüben. Systemarbeiten, die mit höherer Wahrscheinlichkeit zu Beeinträchtigungen des Rechenbetriebs führen, sollen jedoch zukünftig nur noch mittwochs abends durchgeführt werden.

#### Automatisches Auslagern von VSPC-Dateien

R. Többicke

Da die VSPC-Systemdatei in der letzten Zeit bedrohlich voll geworden ist, werden ab dem 15. Oktober vom Rechenzentrum Bereinigungen der Datei durchgeführt. Hierzu werden VSPC-Benutzerdateien (Files), die mehr als 4 Wochen lang nicht gelesen wurden, auf Magnetband ausgelagert und aus der Bibliothek des Benutzers gelöscht.

Werden so ausgelagerte Dateien von ihrem Eigner innerhalb von drei Monaten zurückgefordert, so werden sie zum nächstmöglichen Zeitpunkt (etwa einen Tag nach der Zurückforderung) wieder verfügbar sein.

Der Workspace 10 RECLAIM bietet jedem Benutzer die Möglichkeit, Auskunft über ihm gehörende ausgelagerte VSPC-Dateien zu erhalten und weiterhin benötigte Dateien zurückzufordern.

Der Workspace wird geladen durch die Kommandofolge

```
ENTER APL
)LOAD 10 RECLAIM
```

Das Arbeiten im Workspace selbst wird von einer APL-Funktion kontrolliert, die beim Laden des Workspace aufgerufen wird. Alle weiterhin notwendigen Kommandos werden von dieser Funktion erfragt und können sowohl in Groß- als auch in Kleinschreibung eingegeben werden.

Zu beachten ist, daß das Kommando )LOAD 10 RECLAIM sowie das zur Beendigung der APL-Sitzung notwendige Kommando )OFF HOLD in gültigen APL-Zeichen eingegeben werden müssen.

Abhängig vom Terminal-Typ bedeutet dies folgendes:

IBM 3276, 3277, 3278:

es dürfen nur Großbuchstaben verwendet werden.

VD15, falls beim Logon auch upshift-2 eingegeben wurde:  
APL/ASCII - Taste auf APL stellen.

VD15, falls beim Logon nicht upshift-2 eingegeben wurde:  
LOCK-Taste drücken, somit in Großbuchstaben schreiben.

IBM 2741:

entweder einen APL-Schreibkopf verwenden oder in Kleinbuchstaben schreiben und das Zeichen ) durch das Zeichen " ersetzen.

Bei der Benutzung eines oben nicht angegebenen Terminals sollte man die aufgeführten Möglichkeiten ausprobieren und bei weiteren Problemen in der Beratung des Rechenzentrums (Tel.: 2485) nachfragen.

Treten bei der Benutzung des Workspace Unterbrechungen auf (APL-Interpreter Meldungen wie z.B. SYNTAX ERROR, INTERRUPT o.ä.), so empfiehlt es sich, den Workspace durch )LOAD 10 RECLAIM neu zu laden oder mit )OFF HOLD die APL-Sitzung zu beenden. Bei anderen Meldungen sollte das Rechenzentrum benachrichtigt werden.

**Neue JCL-Prozeduren**

W. Bosse

**1) PL/I Checkout Compiler:**

In Ergänzung der bereits in der Januar-Ausgabe von *inforum* (Jg. 3, Nr.1) angegebenen Prozeduren steht jetzt auch die katalogisierte Prozedur PLCC zur Verfügung, bei der es in Verbindung mit der Parameterangabe PARMC=RUN auch möglich ist, Übersetzen und Ausführen der Programme in einem Step durchzuführen. Zur Kennzeichnung der Eingabedatei muß man die Anweisung

```
//SYSIN DD *
```

oder

```
//COMPILE.SYSIN DD *
```

verwenden.

**2) VS BASIC Compiler:**

Seit dem 1. Oktober 1979 steht der Übersetzer für BASIC wieder zur Verfügung. Dazu sind die katalogisierten Prozeduren

```
BASICC BASICD
```

```
BASICCE
```

```
BASICF BASICX
```

bereitgestellt worden.

**3) SPSS:**

Wie in der letzten *inforum*-Ausgabe angekündigt, ist die katalogisierte Prozedur SPSS abgeändert worden, so daß jetzt automatisch die neue Version 8 aufgerufen wird. Bis zum Jahresende wird es möglich sein, die alte Version 6 durch den Aufruf

```
// EXEC SPSS,VERSION=6
```

zu benutzen. Nach Möglichkeit sollte aber schon jetzt die neue Version eingesetzt werden.

Die neue Prozedur SPSSORT ermöglicht von SPSS aus die Verwendung des Sortierprogramms.

**Speicherung privater Dateien auf Magnetplatten**

H. Meyer

Zur Speicherung privater Benutzerdateien werden vom 1. November 1979 an auch Magnetplatten zur Verfügung gestellt. Es ist daran gedacht, daß kleinere, häufig benutzte Magnetbanddateien, speziell Dateien mit direktem Zugriff und Zwischenergebnisse eines Programms für einen Wiederanlauf desselben Programms oder die Weiterverarbeitung durch ein Folgeprogramm direkt ansprechbar sein sollen.

Für die Belegung von Plattenplatz gelten vorerst (d.h. bis genügend Erfahrungen vorliegen) folgende einschränkende Bedingungen:

- a) Benutzer, die private Dateien während der Bearbeitung eines Problems auf Magnetplatten anlegen wollen, beantragen dies auf einem entsprechenden Formular unter Angabe der Verrechnungsnummer, der zu bearbeitenden Aufgabe und des Bearbeitungszeitraums. Dieses Formular wird im Dispatch des Rechenzentrums (Zi. 02) ausgegeben.
- b) Nach Eintrag der Verrechnungsnummer in entsprechende Systemdateien können Plattendateien angelegt werden. Dabei darf eine einzelne Datei nicht mehr als 100 Spuren mit einer Kapazität von jeweils maximal 6200 Bytes (das sind fünf "2314"-Zylinder) umfassen, je Benutzer können höchstens 200 Spuren belegt werden.
- c) Es können sequentielle Dateien und Dateien mit direktem Zugriff (PL/I: REGIONAL; FORTRAN: DEFINE FILE) angelegt werden; falls andere Dateitypen benötigt werden, ist die Datenorganisation mit dem Rechenzentrum abzusprechen. ISAM-Dateien können nicht eingerichtet werden, da die Unterstützung ausläuft.
- d) Die Benutzerdateien müssen grundsätzlich mit dem neuen Dienstprogramm des Rechenzentrums RUMSERV angelegt und gepflegt werden (vgl. den Artikel "RUMSERV-Befehle zur Verwaltung privater Magnetplattendateien" in diesem inforum); sie erhalten Namen, die mit der Verrechnungsnummer als erstem Index qualifiziert sind, und werden katalogisiert, so daß ein späterer Zugriff nur über die Angaben

```
//ddname DD DISP=SHR,DSN=userid.dsname
```

erfolgen kann. Gelöscht werden solche Dateien mit dem Dienstprogramm.

- e) Für eine gute Nutzung des Plattenplatzes auf allen verfügbaren Magnetplattentypen sollten die Daten in möglichst großen Blöcken mit einer BLKSIZE  $\leq$  6200 gespeichert werden.
- f) Dateien, die nicht nach den Regeln der Punkte a)-e) angelegt wurden oder die sieben Kalendertage unbenutzt geblieben sind oder die vier Kalendertage nach Erstellung noch über 30% freien Platz enthalten, werden automatisch gelöscht; falls zum Zeitpunkt der regelmäßigen Plattenkontrolle durch das Rechenzentrum von einem Benutzer mehr als 200 "2314"-Spuren belegt sind, werden so viele Dateien freigegeben, bis dieses Limit nicht mehr überschritten ist. Nach Ablauf einer Rechnerlaubnis werden sämtliche Daten gelöscht, die zur zugehörigen Verrechnungsnummer gehören.

- g) Von jeder Datei sollte nach der erfolgreichen vollständigen Erzeugung eine Kopie auf Magnetband übertragen werden, um einen Schutz gegen Plattenfehler zu erreichen und um Dateien, die längere Zeit nicht mehr benötigt werden, jederzeit löschen zu können. Das Rechenzentrum entwickelt im Rahmen von RUMSERV Funktionen für die unkomplizierte Auslagerung privater Plattendateien auf private Magnetbänder und für die problemlose Wiederherstellung der ursprünglichen Dateien auf Magnetplatten.

### Prozeduren für VSPC-Benutzer

W. Bosse

In der VSPC-Modellbibliothek 10 sind unter den Namen VSPCPRT, VSPCLIST und VSPCSAVE Muster für den Aufruf der katalogisierten Prozeduren VSPCPRT und VSPCSAVE gespeichert, die durch ein LOAD- oder MERGE-Kommando in den eigenen Arbeitsspeicher eingefügt werden können.

Beispiel:

```
LOAD 10 VSPCPRT
```

Mittels dieser Prozeduren kann ein VSPC-Benutzer eine VSPC-Datei auf Papier oder auf Magnetband ausgeben sowie eine Übersicht über alle in seiner Bibliothek befindlichen Dateien ausdrucken.

#### a) Übersicht über eine VSPC-Bibliothek

Eine Zusammenstellung der Namen und Attribute aller in der eigenen VSPC-Bibliothek gespeicherten Dateien läßt sich, wie in dem Muster 10 VSPCLIST beschrieben, drucken.

Beispiel:

Für die Benutzerkennung 219937 mit dem aktuellen Kennwort GEHEIM sind dazu folgende Angaben nötig.

```
//VSPCLIST EXEC VSPCPRT,EPRINT=A
//SYSIN DD *
AUTH 219937/GEHEIM
LIST ALL
```

Auf dem VSPC-Terminal erhält man eine Übersicht der Namen bekanntlich durch das Kommando QUERY LIBRARY und kann die Attribute einer Datei mittels des Kommandos QUERY FILE erfragen.

#### b) Drucken von VSPC-Dateien

Gemäß dem Muster 10 VSPCPRT lassen sich eine oder mehrere VSPC-Dateien ausdrucken. Da keine automatische Trennung der Dateien vorgenommen wird, empfiehlt es sich, jeweils zwischen zwei Dateien die Datei 10 VSPCEND anzugeben. Von jeder Zeile

werden maximal 132 Zeichen ausgegeben; längere Zeilen werden abgeschnitten.

**Beispiel:**

Für die Benutzerkennung 219937 mit dem aktuellen Kennwort GEHEIM sollen die beiden Dateien PROBE und TEST9 ausgedruckt werden.

```
//VSPCPRC EXEC VSPCPRC
//SYSIN DD *
AUTH 219937/GEHEIM
EXPORT PROBE TO(SYSOUT)
EXPORT 10 VSPCEND TO(SYSOUT)
EXPORT TEST9 TO(SYSOUT)
```

Um zu vermeiden, daß auch über den Falz gedruckt wird, kann die Anzahl der zu druckenden Zeilen pro Seite über den Parameter LINECT der JOBPARM-Anweisung festgelegt werden, z.B.

```
/*JOBPARM LINECT=66
```

Solange keine Zeile einer VSPC-Datei mehr als 80 Zeichen enthält, kann diese auch durch einen Job vom Typ LIST oder mittels der Prozedur LIST aufgelistet werden, indem man die Datei in einen entsprechenden Job einfügt.

**c) Auslagern von VSPC-Dateien auf Magnetband**

Eine oder mehrere VSPC-Dateien können als eine Magnetbanddatei ausgelagert werden, indem die katalogisierte Prozedur VSPCSAVE - wie in der Modelldatei 10 VSPCSAVE beschrieben - aufgerufen wird. Der Dateiname auf Magnetband muß sich dabei aus der Verrrechnungsnummer des Benutzers und dem bis zu acht Zeichen langen Namen zusammensetzen. Wenn einzelne Zeilen der angesprochenen VSPC-Dateien mehr als 80 Zeichen enthalten, muß der Parameter RECSIZE beim Aufruf die maximale Zeilenlänge angeben, weil sonst jede längere Zeile einfach abgeschnitten wird.

**Beispiel:**

Für die Benutzerkennung 219937 mit dem aktuellen Kennwort GEHEIM sollen die beiden VSPC-Dateien DATEN und DATENX (in dieser Reihenfolge) auf das Magnetband 999999 als zweite Datei unter dem Namen UUU01.DATEN79 ausgelagert werden; die maximale Zeilenlänge beträgt 125 Zeichen.

```
//VSPCSAVE EXEC VSPCSAVE,DSNAME='UUU01.DATEN79',
// VOLSER=999999,LABEL=2,RECSIZE=125
//SYSIN DD *
AUTH 219937/GEHEIM
EXPORT DATEN TO(SAVETAPE)
EXPORT DATENX TO(SAVETAPE)
```

Änderungen bei der Verarbeitung von Disketten

K. Elix

Seit dem Erscheinen des ersten Artikels über Disketten im Informations Jg. 3, Nr. 2 (April 1979) haben sich einige Änderungen in der Verarbeitung von Disketten ergeben, die zusätzliche Anmerkungen notwendig machen.

Das Verfahren der Dateneingabe von Diskette ist unverändert geblieben. Jedoch empfiehlt es sich, auf der Jobkarte MSGCLASS=A anzugeben, um einen Papierausdruck des auf der Diskette codierten Jobs zu erhalten. Diesem Papierausdruck entnimmt man auch Informationen darüber, ob der Job erfolgreich war oder nicht bzw. etwaige Fehlermeldungen.

Zur Datenausgabe auf Diskette muß der Job, der die Daten ausgibt, eine DD-Anweisung der folgenden Form enthalten:

```
//ddname DD SYSOUT=0,DSID=dsid,
//          DCB=(LRECL=lrecl,BLKSIZE=lrecl,RECFM=FB)
```

Dabei bedeutet:

**dsid** Name der Datei auf der Diskette  
**lrecl** Satzlänge der Ausgabedatei

Nachdem dieser Job erfolgreich war, kann die Ausgabediskette bespielt werden. Das Bespielen der Ausgabedisketten erfolgt vorläufig nach Rücksprache des Benutzers mit dem Operateur. Ein Benutzer, der eine mit seinen Daten bespielte Diskette in Empfang nehmen möchte, muß dabei den Jobnamen des von ihm gestarteten, erfolgreichen Jobs und den Namen der Datei auf der Diskette angeben. Außerdem muß er eine andere Diskette bei den Operateuren abgeben, um den Rückfluß der Disketten zum Rechenzentrum zu gewährleisten. Für den Datenaustausch mit dem Rechenzentrum können Instituten auf Antrag einige Disketten zur Verfügung gestellt werden.

Der Benutzer erhält dann eine mit seinen Daten bespielte Diskette. Als Ausgabediskette dienen ausschließlich vom Rechenzentrum vorformatierte Disketten. Die Diskette mit der Ausgabedatei muß binnen einer Woche nach Ablauf des zugehörigen Jobs bei den Operateuren abgeholt werden.

### NAG-Bibliothek: Erweiterte Version (Mark 7) und Beispielprogramme

U. Ebert

Die jetzige Version der NAG-Bibliothek (Mark 6) wird zum 1. November 1979 durch eine neue, erweiterte Version (Mark 7) abgelöst. Sie enthält - wie im Manual schon angekündigt - die folgenden 32 Routinen aus Mark 6 nicht mehr, für die aber Ersatzroutinen zur Verfügung stehen. Bei Schwierigkeiten gibt die Programmierberatung Auskunft.

E04AAF E04BAF E04CDF E04CEF  
E04DCF E04FBF E04HAF F01CCF  
G05AAF bis G05BBF

und die entsprechenden Routinen in einfacher Genauigkeit (E04AAE, ..., G05BBE).

Darüber hinaus sind bei der Bibliothek in einfacher und doppelter Genauigkeit jeweils 82 neue Routinen hinzugefügt und jeweils 11 gründlich umgearbeitet worden. Die neuen Routinen stammen aus den Gebieten:

C05 Nullstellen transzendenter Gleichungen  
D01 Integration  
D02 Gewöhnliche Differentialgleichungen  
D03 Partielle Differentialgleichungen  
E02 Kurvenanpassung  
E04 Optimierung (Minimierung) einer Funktion  
F01 Matrix-Operationen  
F02 Eigenwerte und Eigenvektoren  
F04 Lineare Gleichungssysteme  
G01 Statistik  
S Spezielle Funktionen  
X02 Maschinenkonstanten  
X04 Ein- und Ausgabe

Dabei liegt der Schwerpunkt der Erweiterung auf den Gebieten: Gewöhnliche Differentialgleichungen, Partielle Differentialgleichungen, Kurvenanpassung, Minimierung, Matrix-Operationen und dünn besetzte Matrizen.

Die gerade beschriebenen Änderungen spiegeln sich natürlich auch in der Dokumentation wider. Für Mark 7 gibt es ein neues Mini-Manual, und die bestehenden Library-Manuals (bisher 4) werden durch einen fünften Band erweitert. Um einerseits Komplikationen wegen der in Mark 7 nicht mehr vorhandenen Routinen zu vermeiden und um andererseits die neuen Möglichkeiten, die Mark 7 bietet, voll ausschöpfen zu können, ist es notwendig, die vorhandenen Mini-Manuals zu ersetzen und die vorhandenen Library-Manuals durch einen Update (Band 5 und weitere Korrekturen) auf den aktuellen Stand zu bringen. Das Rechenzentrum hat allen, die bisher schon NAG-Unterlagen bezogen haben, schon im Juli eine Sammelbestellung angeboten.

Ebenfalls ab 1. November werden die in den Library-Manuals abgedruckten Beispielprogramme (in doppelter Genauigkeit) für die einzelnen Routinen maschinenlesbar zur Verfügung stehen. Sie befinden sich in der untergliederten Datei CMP1.NAGSAMP. Jedes Member enthält ein Beispielprogramm für eine bestimmte Routine, die dazugehörigen Daten und Steueranweisungen für den WATFIV-Compiler. Auf diese Weise kann in einem einfachen Job-Step das Beispielprogramm ausgeführt werden. Jeder Benutzer kann sich dann an diesem Beispiel für sein eigenes Programm orientieren.

Allgemeine Form des Aufrufs:

```
// EXEC WATFIV  
//SYSIN DD DSN=CMP1.NAGSAMP(name),DISP=SHR
```

(name ist der Name der entsprechenden NAG-Routine, z.B. G05DKF)

#### **Benutzerhandbuch**

W. Bosse, U. Ebert

Das Manuskript einer vollständig neuen Version des Benutzerhandbuchs ist im wesentlichen fertiggestellt. Zur Zeit erfolgt die Erfassung des Textes in maschinenlesbarer Form, an die sich die Korrekturphase unmittelbar anschließen wird.

Neben den neu zusammengestellten Beschreibungen zum Einsatz der Betriebssystemfunktionen wird das Benutzerhandbuch eine ausführliche Beschreibung der Kommandos des Dialogsystems VSPC sowie Angaben über betriebliche Regelungen des Rechenzentrums enthalten.

Nach den umfangreichen Vorbereitungen (einschließlich Erstellen eines Registers) wird das Benutzerhandbuch in wenigen Wochen in Druck gehen. Das Erscheinen wird durch Aushang im Rechenzentrum und durch die Hot\_News bekanntgegeben.

#### **Beschreibung der Plottersoftware**

Ab sofort liegt die Software-Information Nr. 3 "Plotter-Software" von Bernd Schulze und Horst Stenzel in der Version vom August 1979 vor. Sie kann zum Preis von 5.- DM bei Frau M. Luth, Zi. 101, dienstags und donnerstags in der Zeit von 10.30 bis 11.30 Uhr käuflich erworben werden.

Obwohl es sich hierbei bereits um die dritte, verbesserte und erweiterte Auflage der Nr. 12 in der Schriftenreihe des Rechenzentrums handelt, sind Anregungen und Korrekturhinweise stets willkommen. Diese können direkt an die Autoren gerichtet werden.

**Schriftenreihe des Rechenzentrums**

W.A. Slaby

Seit der letzten Übersicht über Neuerscheinungen im **inforum** Nr.1 des laufenden Jahrgangs sind in der **Schriftenreihe** des Rechenzentrums folgende Beiträge veröffentlicht worden und können bei Frau M. Luth dienstags und donnerstags in der Zeit von 10.30 bis 11.30 Uhr käuflich erworben werden:

- Nr.35 - Lösung von gewöhnlichen Differentialgleichungen mit nicht-linearen Splines  
von H. Anndt, März 1979; 29 Seiten; 3.-DM
- Nr.36 - Programmpaket zur automatischen Klassifikation  
- Verfahren zur Clusterung quantitativer und qualitativer Daten  
von N. de Lange und D. Steinhausen, April 1979;  
53 Seiten; 5.-DM
- Nr.37 - Ein Code zur Approximation von Hyperflächen im  $R_n$   
von B. Schulze, Juli 1979; 70 Seiten; 6.-DM
- Nr.38 - First and second order sufficient optimality conditions  
in mathematical programming and optimal control  
von H. Mauren, August 1979; 33 Seiten; 3.-DM
- Nr.39 - WK Version 1 1979  
Ein Programmpaket zur automatischen Erzeugung  
sprachgeographischer Karten  
von W. Westerhoff, Juli 1979; 156 Seiten; 9.-DM
- Nr.40 - Darstellung von Histogrammen durch Splinefunktionen  
von W. Herden, September 1979;

**Impressum**Redaktion **inforum**

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

Satz: U. Kaiser  
Druck: H. Mecke

Rechenzentrum der Universität  
Roxelner Straße 60  
4400 Münster

Auflage dieser Ausgabe: 600  
Redaktionsschluß der nächsten Ausgabe: 31.12.1979

**Bekannte Fallstricke im VSPC**

W. Bosse

Die in der inForum-Ausgabe Jg. 3, Nr. 1 (Januar 1979) begonnene Aufzählung einiger Fallstricke im Dialogsystem VSPC wird hiermit fortgesetzt.

Bitte teilen Sie uns Ihre Erfahrungen mit VSPC durch Ihren Beitrag zum Leserforum, durch "Einwurf" in den "Briefkasten" 4400 oder auch telefonisch mit (siehe Impressum).

- 7) Sie freuen sich über die Möglichkeit, mit Hilfe der Prozedur VSPCSAVE eine VSPC-Datei auf Magnetband auslagern zu können. Wie in der Modelldatei 10 VSPCSAVE beschrieben, bereiten Sie den Aufruf vor, stellen einen Job zusammen (TAPES=1 in der JOBPARM-Anweisung nicht vergessen!) und übergeben diesen durch das SUBMIT-Kommando an das Betriebssystem. Dann löschen Sie beruhigt die VSPC-Datei mittels PURGE in Ihrer Bibliothek ...

Spätestens nach Erhalt der Druckausgabe des Jobs ist es mit Ihrer Ruhe vorbei, denn die VSPC-Datei konnte nicht auf Magnetband ausgelagert werden, weil sie sich nicht mehr in der VSPC-Bibliothek befand. Das Auslagern erfolgt nämlich nicht durch Ausführen des SUBMIT-Kommandos, sondern durch den Übermittlung des Jobs, der erst einige Zeit später gerechnet wird. Löschen Sie eine auszulagernde VSPC-Datei deshalb erst, nachdem Sie sich über den fehlerfreien Ablauf des entsprechenden Jobs informiert haben.

Übrigens -- für das Drucken von VSPC-Dateien gilt sinngemäß dasselbe.

- 8) Sie haben sich längst daran gewöhnt, VSPC-Kommandos in abgekürzter Form einzugeben (indem Sie z.B. die ersten drei Buchstaben verwenden). Jetzt suchen Sie, ausgehend von der aktuellen Zeile im Arbeitsspeicher diejenige Zeile, in der die Zeichenkette 'JOB' vorkommt; und da Sie wissen, daß man bei Zeichenketten, die nur aus Buchstaben und/oder Ziffern bestehen, in der Regel die Apostrophe weglassen kann, wollen Sie das LOCATE-Kommando in der Form

```
LOC JOB
```

eingeben. Doch im Eifer des Gefechts schreiben Sie

```
LOA JOB
```

und löschen damit den bearbeiteten Inhalt des Arbeitsspeichers, indem Sie die zufällig vorhandene Datei JOB laden ...

Fehlleistungen dieser Art entstehen dadurch, daß bestimmte Kommandos - wie in diesem Fall LOAD - sehr häufig verwendet werden und schon "wie im Schlaf" eingegeben werden. Relativ selten benutzte Kommandos sollten deshalb nicht abgekürzt werden.

So richtet der Schreibfehler  
 LOAATE JOB  
 keinen Schaden an und kann durch  
 LOCATE JOB  
 schnell korrigiert werden.

- 9) Sie möchten mit Hilfe des ?-Kommandos (AID-Funktion) bestimmte Erläuterungen erhalten, empfangen aber statt dessen nur die Mitteilung  
 INVALID CHARACTER '?'  
 Dann ist das kein Fehler von VSPC, sondern eine Konsequenz Ihrer Eingabedaten ...

Sie haben das Fragezeichen vorher als Zeichen in einem BACKSPACE-, NEWLINE- oder TABSET-Kommando definiert und damit jeder weiteren Verwendung entzogen. (Eine solche Vereinbarung wird oft automatisch durch die Kommandoliste PROFILE beim Logon getroffen, so daß man sich seines Fehlers nicht gleich bewußt ist.)

Besonders "beliebt" ist es, folgende Vereinbarungen zu treffen:

```
BACKSPACE 'B'
NEWLINE   '?'
TABSET    'D'
```

Dann können Sie anschließend kein neues BACKSPACE-Kommando ausführen, die AID-Funktion ist blockiert (s.o.) und ein Logoff ist auch unmöglich. Durch

```
NEWLINE ''
```

wird das ?-Kommando wieder verfügbar, und

```
TABSET ''
```

gibt das OFF-Kommando wieder frei. Letzteres wird auch dringend benötigt, denn das BACKSPACE-Kommando ist erst nach einem neuen Logon wieder einsatzbereit.

Achten Sie deshalb im VSPC darauf, daß Sie jederzeit  
 B, ? und D  
 sagen können.

### Plot-to-Print

H. Pudlatz

Für Testzwecke beim Plotten mit FORTRAN stelle ich das Unterprogramm PLOTS (Entries: PLOT, PLOTE, CHPEN, WHERE, STATE, SCALE, LINE, AXIS, LGAXIS) zur Verfügung, das wie folgt anstelle der Plottersoftware aufgerufen werden kann:

```
// EXEC FORTCE
//CJMPLE.SYSIN DD *
  [Fortran-Programm mit Plot-Aufrufen]
//EXECUTE.LINKIN DD DSN=SYSTEM.RM30BJ(URZ04PLT),DISP=SHR
```

Dabei wird die Zeichenfläche 68x100 cm auf eine Druckerseite abgebildet. Alle oben genannten Entries werden im

Rahmen der Darstellungsgenauigkeit auf dem Drucker voll unterstützt. Die Plotterroutinen SYMBOL und NUMBER werden ignoriert, ihre Verwendung am Schluß aber kommentiert. Ebenso kommentiert werden Fehler bei der Parameterübergabe und die in der Beschreibung der Plottersoftware (Stenzel/Schulze) genannten Plotfehler 0, 1, 2 und 4.

Die Aufrufreihenfolge "PLOTS - Übrige Plotterroutinen - PLOTE" muß eingehalten werden, da sonst kein Bild gedruckt wird. Diese Sequenz kann aber beliebig oft erscheinen.

Über die Möglichkeiten der Plottersoftware hinaus steht ein Common-Bereich

COMMON /WINDOW/ XMIN, XMAX, YMIN, YMAX

zur Verfügung, der ein "windowing" erlaubt, Falls

0. <= XMIN < XMAX <= 100.

0. <= YMIN < YMAX <= 68.

im aufrufenden Programm angegeben werden, anderenfalls wird die volle Zeichenfläche von 68x100 cm verwendet. Das Druckfenster nutzt in jedem Fall die volle Druckpapierfläche von 54x129 Druckpositionen, d.h. für Verzerrungen ist der Benutzer verantwortlich.

Das Programm gehört nicht zur vom Rechenzentrum gewarteten System-Software. Bei etwaigen Schwierigkeiten im Umgang mit dem Programm steht der Autor beratend zur Verfügung.

### Personelle

Zum 1.9.1979 hat sich der Kreis der studentischen Mitarbeiter durch Wiederbesetzung freigewordener Stellen vergrößert. Als neue Kollegen begrüßen wir die Herren

Christoph Bergmann,  
Thomas Engelbert,  
Andreas Heidrich,  
Jürgen Hölters,  
Hajo Müller,  
Thomas Papke,  
Markus Tacke.

Nach bestandenen Examen ist Herr Klaus Elix, bereits seit langem im Rechenzentrum tätig, nunmehr wissenschaftlicher Mitarbeiter.

Verlassen haben uns zum 31.8.1979 die Herren Johannes Grass und Wolfgang Herden, die längere Zeit, zunächst als studentische, später als wissenschaftliche Mitarbeiter dem Rechenzentrum angehörten, sowie aus dem Kreis der studentischen Mitarbeiter Herr Winfried Kerstiens und - nach bestandenen Examen - Herr Winfried Schafmann.

Ihnen wünschen wir auch weiterhin viel Erfolg.

Lehrangebot des Rechenzentrums im WS 1979/1980Einführung in die EDV

Die Vorlesung ist eine Einführung in die Grundlagen der elektronischen Datenverarbeitung. Sie richtet sich sowohl an die Hörer, die einen ersten Kontakt zur EDV suchen, als auch an die, die bereits eine Programmiersprache beherrschen und grundsätzliche Kenntnisse erwerben möchten. Ausgehend von den logischen, mathematischen und technischen Grundlagen werden die Grundkomponenten einer Rechanlage erläutert. Hierbei ist auch eine kurze Einführung in die Mikroprozessortechnik geplant. Anschließend werden komplexere Themen (z.B. Betriebssysteme) behandelt. Für die Dauer der Vorlesung steht den Hörern ein Assembler zum Austesten kleinerer Programme zur Verfügung. Vorkenntnisse werden nicht vorausgesetzt.

Dozent:            Kisker  
 Vorl.-Nr.:        220057  
 Hörsaal:           M4  
 Zeit:              Mo 13-15, Beginn: 22.10.1979

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:        Nienhaus, Sturm  
 Vorl.-Nr.:        220061, 220076  
 Hörsaal:           M5  
 Zeit:              Di 15-17, Beginn: 23.10.1979  
                   oder Mi 15-17, Beginn: 24.10.1979

Programmieren in PL/I (nichtnumerische Anwendungen)

In der Lehrveranstaltung wird eine allgemeine Einführung in die Programmiersprache PL/I gegeben. Dabei werden schwerpunktmäßig solche Sprachelemente und Programmiermethoden behandelt, die bei der Lösung von Aufgaben aus dem nichtnumerischen Bereich - etwa der Textverarbeitung - Anwendung finden. Spezielle Vorkenntnisse sind nicht erforderlich.

Dozent:           Pudlatz  
 Vorl.-Nr.:        220095  
 Hörsaal:           M4  
 Zeit:              Do 15-17, Beginn: 25.10.1979

Programmieren in PL/I (numerische Anwendungen)

Ziel der Vorlesung ist eine allgemeine Einführung in die Programmierung. Es soll die systematische Konstruktion von Algorithmen für vorgegebene Probleme sowie die Formulierung dieser Algorithmen in der Programmiersprache PL/I intensiv geübt werden. Die Beispiele werden aus dem naturwissenschaftlichen Bereich gewählt.

Spezielle Vorkenntnisse sind nicht erforderlich.

Literatur:

- W.-D. Mell / P. Preus / P. Sander  
Einführung in die Programmiersprache PL/I  
Bibliographisches Institut Mannheim/Wien/Zürich 1974
- R. Hahn / B. Nienhaber  
Probleme lösen mit dem Computer Teil I+II  
Neuser Verlag Tübingen 1978

Dozent: Benduhn  
Vorl.-Nr.: 220080  
Hörsaal: M6  
Zeit: Di 13.30 s.t.-15, Beginn: 23.10.1979  
+ 2 Std. Übungen

Programmieren in ALGOL W

Durch ihren einfachen und logischen Aufbau macht die Programmiersprache ALGOL W das Erlernen der grundsätzlichen Methoden und Techniken des strukturierten Programmierens auch für Anfänger leicht. Hauptsächliches Anwendungsgebiet sind numerische Probleme, aber auch für typische Fragestellungen der Informatik stehen geeignete Hilfsmittel zur Verfügung.

Dozent: Mertz  
Vorl.-Nr.: 220100  
Hörsaal: M6  
Zeit: Mi 15-17, Beginn: 17.10.1979  
+ 1 Std. Übungen n.V.

Statistische Datenanalyse mit dem Programmsystem SPSS

Vorge stellt wird das Programmsystem SPSS ("Statistical Package For the Social Sciences") in der am hiesigen Rechenzentrum implementierten Version 8. Mit diesem System stehen bequem aufzurufende Programme zu den gebräuchlichen univariaten und multivariaten statistischen Verfahren bereit. Als typisch für die Anwendungsfreundlichkeit des Systems sei angeführt, daß viele Kunden unseres Hauses die Auswertung ihrer Probleme - in der Regel die Auswertung von Fragebogenaktionen - ausschließlich mit den Prozeduren des SPSS durchführen.

In dieser Veranstaltung wird den Teilnehmern das programmiertechnische Rüstzeug zur Durchführung derartiger Auswertungen vermittelt; Vorkenntnisse in der Programmierung werden nicht vorausgesetzt, wohl aber sind solide Grundkenntnisse bezüglich der anzusprechenden statistischen

Verfahren wie Kenntnisse der Anwendungsmöglichkeiten dieser Verfahren im jeweiligen Fachgebiet erwünscht und bei den praktischen Übungen von großem Nutzen.

Literatur:

Nie, N.H. et al, SPSS, McGraw-Hill 1975, ISBN 0-07-046531-2

Hull, H.C. and N.H. Nie, SPSS UPDATE, McGraw-Hill 1979,  
ISBN 0-07-046534-7

Beutel, P. et al, SPSS7, Gustav Fischer Verlag 1978,  
ISBN 3-437-40065-7

Dozent: Steinhausen, Zörkendörfer  
Vorl.-Nr.: 220114  
Hörsaal: M2  
Zeit: Do 15-17, Beginn: 18.10.1979  
+ 2 Sd. Übungen Mo 15-17, Hörsaal M4

PL/I für Fortgeschrittene

Voraussetzung für die Teilnahme an dieser Veranstaltung ist der erfolgreiche Abschluß einer einführenden Vorlesung in ALGOL, FORTRAN oder P\_/I. Ziel der Vorlesung ist es, die Hörer mit fortgeschrittenen Programmier-Techniken bekannt zu machen. Zu diesem Zweck wird die Programmiersprache PL/I benutzt.

Die Vorlesung beginnt mit einer kurzen Zusammenfassung der Sprachelemente und Konzepte von PL/I, um auch dem ALGOL- oder FORTRAN-Programmierer einen Einstieg zu ermöglichen. Im Kern der Vorlesung werden Programmier-Techniken behandelt, die in den Einführungsvorlesungen nicht ausführlich besprochen werden können. Hierzu zählen Themen wie Blockstruktur, Ein/Ausgabe auf Bändern und Platten, Unterprogramm-Techniken, List Processing, Compile-Time Facilities, Multitasking. Der Hörer soll einen Einblick in die Möglichkeiten einer universellen Programmiersprache erhalten und in die Lage versetzt werden, effiziente und gut strukturierte Programme zu schreiben.

Dozent: Neukötter  
Vorl.-Nr.: 220129  
Hörsaal: M3  
Zeit: Di 15-17, Beginn: 16.10.1979

Simulation mit GPSS

GPSS (General Purpose Simulation System) ist eine allgemeine, in der Form knappe Sprache zur Simulation von diskreten Ereignisfolgen. Anhand vieler Beispiele werden die Elemente und Möglichkeiten von GPSS dargestellt und die mit Computersimulationen verbundenen Probleme diskutiert.

Dozent: Ebert  
Vorl.-Nr.: 220133  
Hörsaal: Seminarraum Rechenzentrum, Raxeler Str. 60  
Zeit: Mi 9-11, Beginn: 24.10.1979

Graphische Anwendungsprogrammierung

Im Rahmen dieser Veranstaltung werden die im Rechenzentrum vorhandenen Geräte für die Graphische Datenverarbeitung und die Programmsysteme zur Benutzung dieser Systeme vorgestellt. Es wird eine Einführung in die graphische Anwendungsprogrammierung gegeben.

Dozenten:           Schulze, Stenzel  
Vorl.-Nr.:           220148  
Hörsaal:            Seminarraum Rechenzentrum, Roxeler Str. 60  
Zeit:                DI 14-15.30, Beginn: 23.10.1979

Höhere Programmiersprachen: SNOBOL4

SNOBOL4 eignet sich besonders zur Bearbeitung nichtnumerischer Probleme (Manipulation von Zeichenketten, Listenverarbeitung, etc.) und unterscheidet sich in seinem Aufbau wesentlich von anderen Programmiersprachen wie ALGOL, FORTRAN und PL/I. Dadurch ergeben sich neue angenehme Ausdrucksmöglichkeiten für den Benutzer.

Programmierkenntnisse sind für die Teilnahme an dieser Lehrveranstaltung nützlich (eine Sprache).

Dozent:            Basse  
Vorl.-Nr.:           220152  
Hörsaal:            Seminarraum Rechenzentrum, Roxeler Str. 60  
Zeit:                DI 15.30 s.t.-17, Beginn: 23.10.1979

Informatik - ein Überblick

Ziel dieser Vorlesung ist es, Hörern aller Fachrichtungen einen Überblick über wichtige Teilgebiete der Informatik zu geben. Unter anderem sollen dabei Themen wie Algorithmen und Programme, Formale Sprachen und Programmiersprachen, Graphen- und Codierungstheorie behandelt werden.

Kenntnisse in einer höheren Programmiersprache (z.B. Pascal, PL/I, ALGOL) sind wünschenswert, werden aber nicht vorausgesetzt. Literatur:

Bauer/Goos: Informatik - Eine einführende Übersicht  
HTB 80, 91 Springer Verlag  
Mentz/Neukötter/Slaby: Informatik - Ein Überblick  
Schriftenreihe des Rechenzentrums Nr. 13

Dozent:            Slaby  
Vorl.-Nr.:           220167  
Hörsaal:            Seminarraum Rechenzentrum, Roxeler Str. 60  
Zeit:                MI 13.30 s.t.-15, Beginn: 17.10.1979

Gewöhnliche Differentialgleichungen

Elementare Methoden, Existenz- und Eindeutigkeitssätze für Systeme von Differentialgleichungen. Stetige und differenzierbare Abhängigkeit von Parametern. Verhalten der Lösungen im Komplexen. Klassifikation der Singularitäten. Stellen der Bestimmtheit.

Dozent:            Werner, H.  
Vorl.-Nr.:        (152694)  
Hörsaal:         M5  
Zeit:             Mo, Do 9-11, Beginn: 15.10.1979

Seminar über Fragen der angewandten Informatik

Die Strukturen spezialisierter Rechensysteme sollen in diesem Seminar behandelt werden, z.B. Prozessor, Hybridrechner, Array-Prozessoren, Feldrechner. Eine Vorbesprechung findet am 8.10.1979 im Seminarraum des Rechenzentrums statt. (Interessenten wollen sich direkt mit H. Stenzel in Verbindung setzen).

Dozenten:        Stenzel, Werner, H.  
Vorl.-Nr.:        220186  
Hörsaal:         Seminarraum Rechenzentrum, Roxeler Str. 60  
Zeit:             Mo 16-18, Beginn: 22.10.1979  
                    Vorbesprechung am 8.10.1979

Seminar über steife Systeme von Differentialgleichungen

Besprechung von Arbeiten zur numerischen Behandlung von Differentialgleichungen; insbesondere von steifen Systemen.

Dozent:            Helfrich, Maurer, Werner, H.  
Vorl.-Nr.:        (152680)  
Hörsaal:         Seminarraum 1  
Zeit:             Mo 17-19, Beginn: 15.10.1979  
                    (weitere Termine werden im Seminar bekannt gegeben)

Überseminar über Fragen der angewandten Mathematik

Besprechung neuerer Ergebnisse aus der Approximationstheorie und numerischen Mathematik.

Dozenten:        Haverkamp, Helfrich, Hornung, Maurer, Werner, H.  
Vorl.-Nr.:        (152675)  
Hörsaal:         Seminarraum 1  
Zeit:             Do 11-13, Beginn: 19.10.1979

**RUMSERV -- Ein neues Dienstprogrammkonzept**

R. Schmitt

RUMSERV ist ein am Rechenzentrum entwickeltes Programmsystem, das die Benutzung von Dienstprogrammen (Utilities) vereinfachen und vereinheitlichen soll. Unter Dienstprogrammen verstehen wir dabei im Rechenzentrum entwickelte oder vom Hersteller angebotene Programme, die betriebssystemnahe Standardaufgaben lösen.

**a) Funktionsübersicht**

Bisher wurde jedes Dienstprogramm mit einer eigenen Job-Control-Prozedur aufgerufen, wobei die gewünschte Leistung teils mit Job-Control-Parametern, teils durch Eingabe-Karten spezifiziert wurde.

Demgegenüber bietet RUMSERV einen einheitlichen Formalismus, verschiedene Dienstprogramme zu aktivieren. Jedes Dienstprogramm wird in Form eines **Befehls** an RUMSERV codiert. Ein solcher Befehl besteht aus einem ihn identifizierenden **Befehlsnamen**, gegebenenfalls gefolgt von einem oder mehreren Parametern, der **Parameterliste**. Für den generellen Aufbau der Parameterliste gibt es ein einheitliches Regelsystem, das als **Kontrollsprache** von RUMSERV gesondert beschrieben wird.

Anhand einer Liste erlaubter Befehlsnamen, der **Befehlsliste**, entscheidet RUMSERV, ob ein zu bearbeitender Befehl gültig ist. Für in diesem Sinne gültige Befehle erhält RUMSERV aus der Befehlsliste einen Verweis auf die für diesen Befehl gültige **Parametersyntax** und weiterhin den Namen des für die Bearbeitung des Befehls zuständigen Dienstprogrammes. RUMSERV interpretiert die codierten Parameter anhand dieser Parametersyntax, erstellt als Ergebnis dieser Interpretation eine **Werteliste** und aktiviert anschließend das zuständige Dienstprogramm, dem es die Werteliste übergibt.

Diese Verfahrensweise erlaubt es dem Rechenzentrum, ohne allzu großen Aufwand neue Befehle zu den bestehenden hinzuzufügen, falls neue Dienstprogramme verfügbar werden. Es muß lediglich die Parametersyntax des neuen Befehls definiert und sein Name in die Befehlsliste eingetragen werden. Die meisten bisher verfügbaren Dienstprogramme sind allerdings nicht unmittelbar an RUMSERV anschließbar. Es muß vielmehr zu jedem erst ein Programm erstellt werden, das die von RUMSERV erzeugte Werteliste in eine Form umsetzt, die das jeweilige Dienstprogramm versteht.

**a) Steuerung der Verarbeitung**

Nach dem oben geschilderten Funktionsschema haben die Befehle keine eigentliche Wirkung auf RUMSERV, sondern werden nach formaler Interpretation an Dienstprogramme weitergereicht. Das gilt nicht für einen einzigen Befehl, den RESET-Befehl. Dieser hat unmittelbare Wirkung auf die Arbeitsweise von

RUMSERV. Ohne bereits hier auf die noch zu beschreibende Parametersyntax allgemein einzugehen, seien seine Parameter vorgestellt. n ist im folgenden eine vorzeichenlose ganze Zahl.

COL1(n) oder BEGIN(n)      n gibt an, mit welchem Zeichen jedes Eingabesatzes der signifikante Teil des Befehls beginnt. Der Inhalt der ersten n-1 Zeichen wird nicht beachtet (implizit: n=2).

COL2(n) oder END(n)      n gibt die Nummer des letzten für die Interpretation des Befehls signifikanten Zeichens an. Zeichen hinter dem n-ten Zeichen werden nicht beachtet (implizit: n=72).

MAXCODE(n)      n ist eine obere Schranke für den RETURN-Code, den ein Dienstprogramm liefern kann, ohne daß RUMSERV die Befehlsausführung als fehlerhaft betrachtet (siehe NOGO-Schalter) (implizit: n=4).

GO-NOGO      löscht bzw. setzt den NOGO-Schalter. Bei gesetztem NOGO-Schalter untersucht RUMSERV die Befehle nur auf syntaktische Korrektheit, aktiviert aber kein Dienstprogramm. Der NOGO-Schalter wird gesetzt durch:

- ungültige Befehlsnamen;
- ungültige Parameterlisten (Syntaxfehler);
- abnormale Beendigung eines Dienstprogrammes (ABEND);
- RETURN-Code eines Dienstprogrammes, der größer ist als der derzeitige Wert von MAXCODE;
- NOGO-Parameter des RESET-Befehls.

PRINT-NOPRINT      löscht bzw. setzt den NOPRINT-Schalter. Bei gesetztem NOPRINT-Schalter werden die eingelesenen Befehle nicht aufgelistet. Dadurch kann bewirkt werden, daß vertrauliche Befehlsparameter (z.B. Kennworte) unsichtbar bleiben. Fehlermeldungen werden unabhängig vom NOPRINT-Schalter immer gedruckt.

Der RESET-Befehl ist unabhängig vom NOGO-Schalter immer wirksam. Mit seiner Hilfe kann der Benutzer von RUMSERV entscheiden, ob nach einer fehlerhaften Ausführung eines Befehls die nachfolgenden Befehle noch ausgeführt werden sollen. Außerdem kann bei gesetztem NOGO-Schalter eine Befehlsfolge zunächst auf syntaktische Korrektheit getestet werden, bevor sie zur Verarbeitung übergeben wird. Wegen dieser besonderen Bedeutung des RESET-Befehls werden Fehler in seiner Parameterliste von RUMSERV nicht in gleicher Weise behandelt wie bei anderen Befehlen (Setzen des NOGO-Schalters), sondern beenden die Verarbeitung unmittelbar.

**c) Job-Control**

RUMSERV benötigt nur DD-Statements für die Befehlseingabe (SYSIN) und für die Ausgabe der Ergebnisse (RUMPRINT und SYSPRINT).

SYSIN kann eine beliebige sequentielle Datei sein oder auch eine Verkettung (Concatenation) von sequentiellen Dateien. Vorzugsweise werden die Befehle über den Eingabe-Kartensatz eingegeben.

RUMPRINT ist vorzugsweise eine Druckdatei (SYSOUT). Auf ihr protokolliert RUMSERV die eingelesenen Befehle, schreibt Fehlermeldungen und informiert über die Befehlsausführung (RETURN- oder ABEND-Code des Dienstprogramms).

Die Druckausgabe der aufgerufenen Dienstprogramme erfolgt nicht über die Protokolliste (RUMPRINT), sondern über die Datei mit dem Namen SYSPRINT. Das hat seine Gründe darin, daß die Dienstprogramme von RUMSERV unabhängig sind. Sie werden zwar von RUMSERV aktiviert, aber nicht während der Ausführung kontrolliert. Insbesondere kann RUMSERV die Druckanforderungen der aufgerufenen Dienstprogramme (z.B. Seitenvorschub, Zeilenzahl pro Seite) nicht mit seinen eigenen Druckanforderungen koordinieren.

Damit ersichtlich ist, welche Druckzeilen auf SYSPRINT zu welchen der auf RUMPRINT protokollierten und nummerierten Befehle gehören, druckt jedes Dienstprogramm als erste Aktion eine Kopfzeile, in der der Name des Befehls steht, der dieses Dienstprogramm aufgerufen hat, sowie die laufende Nummer dieses Befehls in der Protokolliste.

**d) Aufruf**

RUMSERV steht im Stapel- und im Monitorbetrieb zur Verfügung.

Im Stapelbetrieb erfolgt der Aufruf mit der katalogisierten Prozedur SERVICE als TEST- oder PROD-Lauf in der Form:

```
//UUU01 JOB (UUU01,TEST,A01),SOMMER
// EXEC SERVICE[,Parameter]
//EXECUTE.SYSIN DD *
.
Befehle
.
//
```

Die möglichen Parameter sind:

CPRINT:    SYSOUT-Parameter für die Protokoll-Datei (RUMPRINT);  
Voreinstellung: CPRINT='\*'

EPRINT: SYSOUT-Parameter für die Ausgabe-Datfel der Dienstprogramme (SYSPRINT); Voreinstellung: EPRINT='\*'

PARME: Parameterliste für einen vor allen eingelesenen Befehlen auszuführenden RESET-Befehl.

Im Monitorbetrieb wird RUMSERV durch den neuen Job-Typ SERV aufgerufen:

```
//UUU02 JOB (UUU02,SERV,A02),WINTER
//SERVICE [Parameter]
```

•  
Befehle  
•

//

Die Parameter der SERVICE-Karte wirken als Parameterliste eines vor allen folgenden Befehlen auszuführenden RESET-Befehls.

Im Monitorbetrieb wird eine andere Befehlsliste als im Stapelbetrieb benutzt. Damit ist es möglich, gewisse Befehle nur im Stapelbetrieb zu erlauben.

### a) Kontrollsprache

RUMSERV benutzt existierende Software des Betriebssystems zur Definition einer Parametersyntax und zur Interpretation einer Parameterliste. Fügt also dem Sprachenwirrwarr existierender Dienstprogramme keine neue Sprache hinzu. Die Kontrollsprache ist dennoch für die Benutzer des RZ eine neue Sprache, da die Dienstprogramme, die bereits jetzt diese Sprache benutzen, nur dem Systemprogrammierer, nicht jedoch allgemein verfügbar sind. Sie ist mit einigen Einschränkungen eine Übermenge der von VSPC (Virtual Storage Personal Computing) verwendeten Sprache.

Wie bereits in a) ausgeführt, besteht ein Befehl aus dem Befehlsnamen, gegebenenfalls gefolgt von einer wechselnden Zahl von Parametern, der Parameterliste.

Eine beliebige Zahl von Trennzeichen (delimiter) trennt das Befehlswort von dem ersten Parameter sowie die einzelnen Parameter untereinander.

Fehler in den Parametern, soweit sie durch RUMSERV erkannt werden, bezeichnen wir als syntaktische Fehler. Die verschiedenen Möglichkeiten für eine Parametersyntax werden nachfolgend erläutert. Hingegen sind Fehler in den Parametern, die erst von dem aufgerufenen Dienstprogramm entdeckt werden, Verstöße gegen eine mit dem jeweiligen Dienstprogramm festzulegende Semantik.

Art und Anzahl der Parameter werden in der Syntaxbeschreibung jedes Befehls definiert. Die einzelnen Parameter sind dabei entweder Positionsparameter (positional) oder Schlüsselwortparameter (keyword), sie sind entweder immer erforderlich (required) oder können weggelassen werden (optional). In letzterem Fall kann in der Syntaxbeschreibung eine Voreinstellung (default) angegeben werden, die dann angenommen wird, wenn für diesen Parameter in der Parameterliste kein Wert spezifiziert wurde.

Die Positionsparameter eines Befehls haben eine wohldefinierte Reihenfolge. In die Parameterliste eingesetzte Werte werden in der spezifizierten Reihenfolge von links nach rechts den Positionsparametern zugeordnet, die der jeweiligen Position entsprechen. Es ist erlaubt, in einer Parameterliste weniger Werte anzugeben als Positionsparameter für diesen Befehl definiert sind, vorausgesetzt, die weggelassenen Parameter haben in der Syntaxbeschreibung das Attribut "optional". Es werden dann aber immer von links nach rechts so vielen Positionsparametern Werte zugeordnet, wie Werte spezifiziert wurden. Es gibt keine Möglichkeit, Positionsparameter zu überspringen, d.h. Werte so zu spezifizieren, daß etwa der erste Wert dem ersten Positionsparameter, der zweite aber nicht dem zweiten, sondern dem dritten zugeordnet werden soll.

Schlüsselwortparameter sind reihenfolgeunabhängige Parameter. Die Zuordnung eines in der Parameterliste spezifizierten Wertes zu einem Schlüsselwortparameter geschieht dadurch, daß ein Schlüsselwort (nicht nur ein Wert) in der Parameterliste angegeben werden muß. Falls eine Befehlssyntax Positions- und Schlüsselwortparameter verwendet, gehen die Positionsparameter den Schlüsselwortparametern voraus. Schlüsselworte dürfen in diesem Fall in der Parameterliste nur dann angegeben werden, wenn bereits für alle Positionsparameter Werte eingesetzt worden sind. Andernfalls nähme das angegebene Schlüsselwort in der Parameterliste die Position eines Positionsparameters ein und würde nicht als Schlüsselwort interpretiert, sondern als Wert für diesen Positionsparameter. Schlüsselwortparameter haben im übrigen immer das Attribut "optional". Die Auslassung eines Schlüsselwortparameters ist also nie ein Syntaxfehler im oben definierten Sinne.

Ein in der Syntaxbeschreibung definierter Schlüsselwortparameter kann mehrere Schlüsselworte als gültig erkennen. Die auf der Parameterliste angegebene Information wird dahingehend ausgewertet, ob ein Schlüsselwort und gegebenenfalls welches aus einer Reihe von erlaubten Schlüsselworten für diesen bestimmten Parameter angegeben wurde. Zusammen mit einem Schlüsselwort können aber auch Werte spezifiziert werden. Dies geschieht dadurch, daß hinter dem Schlüsselwort, in Klammern eingeschlossen, sogenannte Unterpparameter angegeben werden können. Eine solche

Unterparameterliste gehorcht den gleichen Regeln wie die Parameterliste selbst, kann also ihrerseits aus Positions- und/oder Schlüsselwortparametern bestehen.

Befehlsnamen dürfen bei der gewählten Implementierung bis zu 8 Zeichen, Schlüsselworte bis zu 31 Zeichen lang sein. Es dürfen zur Bildung dieser Worte nur Buchstaben und Ziffern verwendet werden und das erste Zeichen muß immer ein Buchstabe sein. Befehlsnamen und Schlüsselworte dürfen durch Weglassen von Endbuchstaben abgekürzt werden, solange die Eindeutigkeit dabei nicht verloren geht. Vor allzu intensiver Ausnutzung dieser Möglichkeit wird aber gewarnt, da durch Hinzufügen neuer Befehlsnamen oder neuer Schlüsselworte zu einer Befehlssyntax eine vorher bestehende Eindeutigkeit verloren gehen kann.

Als Trennzeichen dienen Leerzeichen (blanks), Kommata oder Kommentare. Letztere sind wie PL/I-Kommentare gebaut, d.h. sie beginnen mit /\* und enden mit \*/. Die Eingabe ist nicht kartengebunden. Ein Befehl kann über beliebig viele Sätze (Lochkarten, Terminalzellen, etc.) fortgesetzt werden, wobei die Fortsetzung zwischen Parametern oder auch innerhalb eines Parameters erfolgen kann. Ein Satz eines Befehls gilt als unvollständig, wenn das letzte Zeichen, das kein Leerzeichen ist, entweder ein Plus- oder ein Minuszeichen ist. In letzterem Fall wird der nachfolgende Satz als Fortsetzung verwendet, indem er das Minuszeichen ersetzt. Bei Fortsetzung durch ein Pluszeichen wird ebenso verfahren, doch werden im Fortsetzungssatz alle führenden Leerzeichen weggelassen. Ein Semikolon schließlich beendet die Anweisung auch dann, wenn dahinter noch weitere Informationen folgen.

Es werden in dem Artikel "RUMSERV-Befehle zur Verwaltung privater Magnetplattendateien" in dieser Ausgabe von inforum die derzeit existierenden Befehle vorgestellt, weshalb sich hier die Angabe von Beispielen erübrigt.

## 2) RUMSERV-Fehlermeldungen

Meldungen haben die allgemeine Form:

RUM7nnx Text

Dabei ist nn eine Folgenummer, x kann folgende Werte haben:

- x=I: Information, keine Fehlermeldung
- x=W: leichte Fehler (Warning); es wird ein RETURN-Code 4 erzeugt, die Verarbeitung geht aber weiter.
- x=E: schwere Fehler (Errors); es wird ein RETURN-Code 8 erzeugt und der NOGO-Schalter gesetzt.

x=S: schwere Fehler (Severe Errors); die Verarbeitung wird sofort mit RETURN-Code 12 beendet.

Der jeder Meldung mitgegebene Text beschreibt die nähere Ursache des Fehlers.

#### g) Die Befehle NEWPAGE und NEWLINE

Wie bereits ausgeführt, hat RUMSERV keinen Einfluß auf die Druckliste der Dienstprogramme (SYSPRINT). Auch die einzelnen Dienstprogramme "wissen" nicht, wieviel Zeilen auf der laufenden Seite bereits durch das vorangegangene Dienstprogramm gedruckt wurden, wann also eine neue Seite begonnen werden soll.

Mit dem NEWPAGE-Befehl hat der Benutzer von RUMSERV die Möglichkeit, gezielt einen Seitenvorschub auf dieser Druckliste vorzunehmen und auf der neuen Seite eine Kopfzeile mit erklärendem Text und wahlweise einer Seitennummer zu drucken. Der Text der Überschrift wird als Unterparameter des Schlüsselwortes TITLE wie folgt angegeben:

```
TITLE('TEXT')
```

Hochkommata, die Bestandteil des Textes sind, müssen, wie üblich, doppelt angegeben werden. Wird das Schlüsselwort TITLE weggelassen, druckt NEWPAGE einen Standardtext, der den Jobnamen und die Nummer enthält, den der NEWPAGE-Befehl in der Protokollliste erhalten hat. Will man keinen Text in der Überschrift, kann man dies durch TITLE('') bewirken. Eine Seitennummer n auf der neuen Seite wird durch

```
PAGECOUNT(n)
```

bewirkt.

Der NEWLINE-Befehl wirkt wie der NEWPAGE-Befehl, beginnt jedoch keine neue Seite, sondern druckt eine Zwischenzeile, die zwei Zeilen von der zuletzt gedruckten Zeile abgesetzt ist. Der PAGECOUNT-Parameter ist hier zwar erlaubt, aber ohne Wirkung.

Beide Befehle kennen noch als weiteren Parameter das Schlüsselwort FILE (auch DDNAME). Dies ist eine Vorsorge für den Fall, daß zukünftige Dienstprogramme weitere Druckdateien benötigen. Angabe eines DD-Namens im FILE-Parameter bewirkt die NEWPAGE- bzw. NEWLINE-Aktion statt auf der Druckdatei SYSPRINT auf der mit diesem DD-Namen bezeichnete Druckdatei. Für Dateien, die keine Druckdateien (SYSOUT) sind, ist NEWPAGE bzw. NEWLINE unwirksam.

### RUMSERV-Befehle zur Verwaltung privater Magnetplattendateien

R. Schmitt

Die Regeln, nach denen private Magnetplattendateien eingerichtet werden können, sind in dem Artikel "Speicherung privater Dateien auf Magnetplatten" in dieser Ausgabe von inForum beschrieben. Der folgende Artikel beschreibt die RUMSERV-Befehle, die zum Anlegen, Löschen und Abfragen solcher Dateien dienen.

#### 1) Ergänzung zur RUMSERV-Kontrollsprache

Die RUMSERV-Kontrollsprache kennt zahlreiche Typen von Positionsparametern, von denen in diesem Zusammenhang die folgenden am wichtigsten sind:

##### a) STRING

Dies ist eine Zeichenkette, bestehend aus Buchstaben, Ziffern sowie den Zeichen \$, # und @. Die maximale Länge wird in der Syntaxbeschreibung festgelegt, ebenso, ob nur Ziffern, nur Buchstaben oder beides verwendet werden dürfen. Letztere Festlegung kann für das Anfangszeichen abweichend von den restlichen Zeichen getroffen werden.

##### b) QSTRING

Dies ist eine Zeichenkette, die beliebige Zeichen enthalten kann. Sie muß in Hochkommata eingeschlossen werden, und zwar auch dann, wenn sie keine Sonderzeichen enthält. Die begrenzenden Hochkommata sind nicht Teil der Zeichenkette. Falls ein Hochkomma Teil der Zeichenkette sein soll, muß es durch zwei aufeinander folgende Hochkommata repräsentiert werden.

##### c) DSNAME

Dieser Parameter ist für die Angabe von Dateinamen vorgesehen. Er ist in der einfachsten Form eine Zeichenkette aus maximal 8 Zeichen (STRING), deren erstes ein Buchstabe sein muß. Erlaubt sind auch mehrstufige Namen, deren einzelne Stufen durch Punkte voneinander getrennt sind und aus maximal 8 Zeichen, beginnend mit einem Buchstaben, bestehen dürfen. RUMSERV bildet aus den spezifizierten Namen einen Dateinamen, indem es ihnen die Vernrechnungsnummer (die ersten 5 Zeichen des Jobnamens), gefolgt von einem Punkt, voransetzt.

Beispiel: Die Vernrechnungsnummer sei XYZ01. Aus einem Parameterwert SOURCE wird dann der Dateiname XYZ01.SOURCE.

Es besteht auch die Möglichkeit, Dateinamen anzugeben, die nicht mit der Vernrechnungsnummer qualifiziert sind. Diese müssen in Form eines QSTRINGs, also in Hochkommata eingeschlossen, angegeben werden.

Die DSNAME-Syntax erlaubt außerdem die Angabe eines Member-Namens in Klammern hinter dem Dateinamen. So bezeichnet 'SYSTEM.SOURCE(XYZ01ABC)' das Member XYZ01ABC in der Bibliothek SYSTEM.SOURCE.

d) Werteliste

Ein Positionsparameter kann so definiert werden, daß für ihn in der Parameterliste mehrere Werte angegeben werden können. Eine solche Werteliste besteht aus beliebig vielen Werten, zusammengefaßt durch ein Klammernpaar. Die einzelnen Werte müssen einheitlich von dem Typ (STRING, DSNAME usw.) sein, der für diesen Positionsparameter festgelegt ist. Falls die Werteliste nur einen einzigen Wert enthält, kann das umschließende Klammernpaar weggelassen werden.

2) Anlegen von Dateien (ALLOCATE)

Im ALLOCATE-Befehl werden Name, Platzbedarf und Attribute einer neu anzulegenden Datei angegeben. Die Syntax lautet:

```

                (BYTES(...))  ] (MODEL(...))  ]
                (TRACKS(...)) ] (LIKE(...))   ]
ALLOCATE dsname (RECORDS(...)) ] (TYPE(...))  ]
                (BLOCKS(...)) ] (ATTRIBUTES(...))
                (DEFINEFILE(...))
    
```

(...) bedeutet, daß dieser Parameter Unterparameter hat.

"dsname" ist ein Positionsparameter vom Typ DSNAME und gibt den Dateinamen an. Ein Member-Name darf nicht angegeben werden. Es können nur Dateien angelegt werden, die mit der Verrechnungsnummer des Benutzers beginnen. Nach erfolgreicher Ausführung des ALLOCATE-Befehls ist die Datei angelegt und katalogisiert.

Die Attribute einer Datei können auf drei verschiedene Arten angegeben werden:

- a) explizit durch das Schlüsselwort ATTRIBUTES,
- b) durch Angabe eines Dateityps mit dem Schlüsselwort TYPE,
- c) durch Bezugnahme auf eine existierende Datei mit einem der beiden bedeutungsgleichen Schlüsselworte MODEL und LIKE.

Zu a):

Die Unterparameterliste des ATTRIBUTES-Schlüsselwortes hat die Gestalt (Voreinstellungen sind unterstrichen):

```

(SEQUENTIAL        ) (FIXED                )
(DIRECT            ) (UNDEFINED           )
(PARTITIONED(...)) (VARIABLE            ) RECSIZE(...)
(MEMBERS(...))     ) (SPANNEDVARIABLE)
                      ) (STANDARDFIXED )

```

```

                      ) (NOCTL )
KEYLENGTH(...) (BLOCKED(...)) (CTLASA)
                      ) (BLKSIZE(...)) (CTL360)

```

Der erstgenannte Parameter legt die Dateiorganisation fest (DSORG-Parameter der DD-Anweisung). Neben den sich selbst erklärenden Schlüsselworten SEQUENTIAL und DIRECT kann eine Bibliothek (partitioned dataset) erzeugt werden durch die beiden bedeutungsgleichen Schlüsselworte PARTITIONED und MEMBERS. Der (einzige) Unterparameter dieser Schlüsselworte muß eine vorzeichenlose ganze Zahl sein und gibt die Höchstzahl der Eintragungen (Member) an. Index-sequentielle Dateien (ISAM) können nicht erzeugt werden.

Für den zweiten Parameter, das Satzformat (RECFM-Parameter der DD-Anweisung), ist keine Voreinstellung vorhanden, der Parameter muß also immer angegeben werden. Die Schlüsselworte erklären sich selbst.

Die Angabe der Satzlänge mit dem Schlüsselwort RECSIZE ist immer erforderlich und erfolgt ebenfalls als vorzeichenlose ganze Zahl. Als Satzlänge muß die Länge der tatsächlichen Information angegeben werden. Die für einige Funktionen erforderlichen Zuschläge (4 Bytes für variables Satzformat, 1 Byte für Vorschubsteuerung) nimmt das ALLOCATE-Programm automatisch vor. Die Satzlänge darf, außer beim Format SPANNEDVARIABLE, den Wert 6200 nicht überschreiten.

Falls erforderlich, können Dateien mit Schlüsseln (Keys) erzeugt werden. Die Länge des Schlüssels wird durch den Parameter KEYLENGTH spezifiziert. Auch sie muß eine vorzeichenlose ganze Zahl sein und darf den Wert 255 nicht überschreiten. Schlüssel sind nur für ungeblockte Sätze erlaubt.

Die Blockung wird, anders als bei der Job-Control-Sprache, nicht als Eigenschaft des Satzformates angesehen, sondern durch die Schlüsselworte UNBLOCKED, BLOCKED oder BLKSIZE gesondert angegeben. Sätze werden geblockt, wenn einer der Parameter BLOCKED oder BLKSIZE angegeben wurde. Der jeweilige Unterparameter muß wiederum eine vorzeichenlose ganze Zahl sein. Er gibt bei BLKSIZE die gewählte Länge, bei BLOCKED den Blockungsfaktor, d.h. die Anzahl der Sätze pro Block, an. Inkonsistente Blocklängen (zu groß, kein Vielfaches der Satzlänge) werden von ALLOCATE auf den nächst kleineren korrekten Wert reduziert.

Der letztgenannte Parameter schließlich gibt die Vorschubsteuerung an. NOCTL bedeutet, daß keine Vorschubzeichen verwendet werden, CTLASA bezeichnet ASA-Steuerzeichen (RECFM=xxA), CTL360 gibt Maschinensteuerzeichen (RECFM=xxM) an.

Zu b):

Es ist vorgesehen, für Standardaufgaben Dateltypen einzuführen, deren Attribute auf die lokalen Gegebenheiten abgestimmt sind. Im TYPE-Schlüsselwort wird der Typname als ein Parameter vom Typ STRING (maximal 8 Zeichen) angegeben.

Bisher sind zwei Typen verfügbar:

CARDS bezeichnet eine sequentielle Kartendatei, NOFORMAT eine sequentielle Datei, die als Standard für unformatierte Ein-/Ausgabe in FORTRAN-Programmen benutzt werden kann. Weitere Typen werden nach Bedarf definiert.

Zu c):

Bei Angabe des Namens einer existierenden Datei als Unterparameter der Schlüsselworte MODEL oder LIKE werden die Attribute dieser Datei für die neu einzurichtende Datei benutzt. Es können nur die Attribute einer dem Benutzer eigenen Datei übernommen werden.

Es ist möglich, eine neue Datei gemäß einem Typ oder einer Modelldatei anzulegen, ihr aber eine vom Vorbild abweichende Dateiorganisation zu geben. Dies wird bewirkt durch Angabe der Dateiorganisation (DIRECT, SEQUENTIAL oder PARTITIONED(...)) als zweiten Unterparameter des TYPE- bzw. MODEL-Schlüsselwortes.

Der Platzbedarf kann angegeben werden als Anzahl von Zeichen (BYTES), Sätzen (RECORDS), Blöcken (BLOCKS) oder (logischen) Spuren (TRACKS). Die Anzahl selbst wird als Unterparameter in Form einer vorzeichenlosen ganzen Zahl angegeben. Diese Angabe ist immer erforderlich. Ein weiterer Unterparameter, ebenfalls eine vorzeichenlose ganze Zahl, gibt den Platz an, der - bis zu fünfzehnmal - zusätzlich angelegt wird, falls der ursprünglich angelegte Platz nicht ausreicht. Falls diese Angabe fehlt, wird kein zusätzlicher Platz angelegt. Für DIRECT-Dateien wird dieser Parameter ignoriert.

Eine logische Spur ist 6200 Bytes lang und hat die Eigenschaft, daß auf eine wirkliche Spur beim kleinsten Plattentyp (2314) eine logische Spur paßt, beim nächstgrößeren (3330) zwei und bei Platten vom Typ 3350 deren drei. Falls die Platzangabe in TRACKS erfolgt, können die Attribute weggelassen werden.

Die übrigen Schlüsselworte können nur verwendet werden, wenn dem ALLOCATE-Programm die Dateiattribute bekannt sind. Die Angabe wird in Blöcke umgerechnet und die Datei wird angelegt mit der erforderlichen Anzahl von Blöcken.

Eine Besonderheit stellen die Direct-Access-Dateien der FORTRAN-Sprache dar. Eine solche Datei kann nur benutzt werden, wenn sie mit Hilfe eines DEFINE FILE-Befehls (FORTRAN) initialisiert wurde. Die Unterparameter des DEFINEFILE-Schlüsselwortes des ALLOCATE-Befehls haben die gleiche Bedeutung wie die ersten drei Parameter des DEFINE FILE-Befehls der FORTRAN-Sprache (Anzahl der Sätze, Satzlänge, Format). Bei Verwendung dieses Schlüsselwortes wird die Datei angelegt und initialisiert. Zusätzlich angegebene Attribute werden ignoriert. Das DEFINEFILE-Schlüsselwort ist die einzige Möglichkeit, eine solche Datei anzulegen. Insbesondere kann das MODEL-Schlüsselwort nicht verwendet werden, um die Charakteristik einer existierenden Direct-Access-Datei zu übertragen.

### 3) Behandlung existierender Dateien (VERIFY)

Für Dateien, die mit dem ALLOCATE-Befehl angelegt wurden, verwaltet das Betriebssystem das Datum der letzten Benutzung. Darin unterscheiden sich solche Dateien von Dateien, die mittels DD-Anweisungen angelegt wurden. Der VERIFY-Befehl nimmt bei Dateien, die nicht mit dem ALLOCATE-Befehl angelegt wurden, Eintragungen vor, die diese Dateien als mit dem ALLOCATE-Befehl angelegt erscheinen lassen.

Der Befehl hat folgenden Aufbau:

```
VERIFY Dateinamenliste VOLUME(...)
```

Es dürfen nur Dateien des jeweiligen Benutzers angegeben werden. Member-Namen sind syntaktisch erlaubt, werden aber ignoriert. Der einzige Unterparameter des Schlüsselwortes VOLUME gibt den Namen der Magnetplatte an, auf der die Dateien stehen. Dieser Parameter ist nur erforderlich, wenn die Dateien nicht bereits katalogisiert sind. Sie werden dann durch den VERIFY-Befehl katalogisiert.

### 4) Löschen von Dateien (SCRATCH)

Der SCRATCH-Befehl löscht Dateien oder Bibliothekseintragungen (Member). Die Dateien müssen katalogisiert sein und dem Benutzer gehören. Sie brauchen nicht notwendigerweise mit dem ALLOCATE-Befehl angelegt worden sein.

Der Befehl hat folgenden Aufbau:

```
SCRATCH Dateinamenliste MEMBERS(...)
```

Die in der Dateinamenliste aufgeführten Dateien werden gelöscht. Falls bei einem Dateinamen ein Member-Name angegeben wurde, wird die Datei nicht gelöscht. Sie wird vielmehr als Bibliothek angesehen und der angegebene Eintrag (Member) wird gelöscht. In diesem Fall können auch Eintragungen aus den mit SYSTEM beginnenden Benutzerdateien gelöscht werden, sofern ihre Namen mit der Verrachnungsnummer des jeweiligen Benutzers beginnen.

Bei Verwendung des Schlüsselwortes MEMBERS darf die Dateinamenliste nur einen Dateinamen mit oder ohne Membernamen enthalten. Der Unterparameter des Schlüsselwortes MEMBERS ist eine Liste weiterer Member-Namen, die aus der angegebenen Datei gelöscht werden. Die Angabe mehrerer Member-Namen in dem Klammerausdruck hinter einem Dateinamen ist zwar naheliegend, nach der eingangs gegebenen Beschreibung des Parametertyps DSNAME aber nicht zulässig.

#### 5) Umbenennen von Dateien (RENAME)

Der RENAME-Befehl hat den Aufbau:

```
RENAME dsname NEWNAME(dsname)
```

Die durch den Positionsparameter bezeichnete Datei wird umbenannt und erhält als neuen Namen den Namen, der als Unterparameter des NEWNAME-Schlüsselwortes angegeben ist. Beide Namen müssen der Namenskonvention entsprechend mit der Verrachnungsnummer qualifiziert sein.

Falls neben dem Namen der umzubenennenden Datei ein Member-Name angegeben ist, wird nicht die Datei, sondern dieser Member-Eintrag umbenannt. Der Unterparameter des NEWNAME-Schlüsselwortes bezeichnet in diesem Fall keinen Dateinamen, sondern den neuen Member-Namen. Member aus den mit SYSTEM beginnenden Benutzerdateien können ebenfalls auf diese Art umbenannt werden.

#### 6) Freigabe unbenutzten Platzes (RELEASE)

Der einzige Parameter des RELEASE-Befehls ist eine Liste von Dateinamen. Member-Namen werden ignoriert. Unbenutzter Platz in den Dateien der Liste wird freigegeben, sofern es sich nicht um die Dateilorganisation DIRECT handelt. In diesem Fall ist der RELEASE-Befehl wirkungslos.

Freigegeben werden können nur völlig unbenutzte (physische) Spuren am Ende der Datei. Insbesondere also wird der durch das Löschen von Eintragungen in einer Bibliothek freigewordene Platz durch den RELEASE-Befehl nicht freigegeben.

## 7) Listen von Datei-Information (LISTDS)

Der Aufbau dieses Befehls ist:

LISTDS Dateinamenliste MEMBERS

Im Unterschied zu den bisher genannten Befehlen kann der LISTDS-Befehl auch ohne Parameter benutzt werden. In dieser Form werden die wesentlichen Informationen über alle Dateien des Benutzers ausgegeben. Dateien, die nach den geltenden Konventionen (siehe den Artikel "Speicherung privater Dateien auf Magnetplatten" in dieser Ausgabe von *inforum*) ungültig sind, werden in der Liste entsprechend markiert.

Ohne Angabe des Schlüsselwortes MEMBERS werden die Datei-Informationen in gleicher Weise erstellt wie beim LISTDS-Befehl ohne Parameter, es werden aber nur Informationen über die in der Liste genannten Dateien erstellt.

Bei Verwendung des Schlüsselwortes MEMBERS werden die Dateien der Liste als Bibliotheken aufgefaßt und es werden die Namen der Eintragungen (Member) aufgelistet. In diesem Fall dürfen auch SYSTEM.-Dateien angegeben werden, es werden aus ihnen aber nur die dem Benutzer gehörenden Eintragungen aufgelistet.

## 8) Fehlermeldungen

Es wurde angestrebt, alle Fehlersituationen, gleichgültig, ob durch falsche Benutzung der Befehle oder ohne Verschulden des Benutzers aufgetreten, durch sich selbst erklärende Fehlermeldungen zu beantworten. Verbesserungsvorschläge sind selbstverständlich willkommen.

## 9) Beispiele

Die Verrechnungsnummer sei ABC05.

- a) Anlegen einer Datei des Typs CARDS, bestehend aus 1000 Sätzen:

```
ALLOCATE NEU BLOCKS(1000) TYPE(CARDS)
```

Die Datei erhält den Namen ABC05.NEU.

- b) Löschen der Eintragung ABC05UV aus SYSTEM.SOURCE und ABC05AB aus SYSTEM.OBJECT:

```
SCRATCH ('SYSTEM.SOURCE(ABC05UV)', 'SYSTEM.OBJECT(ABC05AB)')
```

- c) Löschen der Eintragungen ABC05P1, ABC05P2 und ABC05P7 aus SYSTEM.OBJECT:

```
SCRATCH 'SYSTEM.OBJECT' MEMBERS(ABC05P1 ABC05P2 ABC05P7)
```

**Beachte:** Der Unterparameter des Schlüsselwortes MEMBERS ist eine Liste von Member-Namen, muß also gemäß den für Wertelisten geltenden Regeln in Klammern eingeschlossen werden, sofern mehr als ein Wert angegeben wurde. Daher muß korrekterweise ein zweifaches Klammersymbol angegeben werden. Da aber das Schlüsselwort MEMBERS nur einen einzigen Unterparameter kennt, kann ein Klammersymbol weggelassen werden.

- d) Anlegen einer Datei namens ABC05.PRINT mit den Attributen, die bisher durch den Job-Control-Parameter DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689) angegeben wurden. Der Platzbedarf sei etwa 100000 Bytes, es soll aber eine Überschreitung dieses Kontingents um jeweils 10000 Bytes möglich sein:

```
ALLOCATE PRINT BYTES(100000 10000) -
      ATTRIBUTES(VARIABLE BLOCKED(5) RECSIZE(132) CTLASA)
```

**Beachte:** Die Berechnung der Satzlänge erfolgt als:  
 eigentliche Information (132) + Satzbeschreibung (4) +  
 Vorschubzeichen (1) = 137.  
 Die Blocklänge wird zu 5\*Satzlänge+4 = 689 errechnet.

- e) Listen aller Eintragungen aus SYSTEM.OBJECT:

```
LISTDS 'SYSTEM.OBJECT' MEMBERS
```

### Bearbeitungsreihenfolge von Rechenaufträgen

B. Neukäter

Das Betriebssystem einer Rechenanlage ist ein kompliziertes Programm mit mannigfachen Aufgaben. Eine wesentliche Aufgabe besteht darin, jedem Rechenauftrag - kurz Job genannt - Betriebsmittel zu seiner Abwicklung zuzuordnen. Unter Betriebsmittel versteht man in diesem Zusammenhang Rechenzeit, Speicherplatz im Hauptspeicher und auf Magnetplatten, Magnetbandgeräte, Magnetbänder und ähnliches. Da die Betriebsmittel begrenzt sind, muß das Betriebssystem dafür sorgen, daß für jeden initialisierten Job während seiner Ausführungszeit die benötigten Betriebsmittel in genügendem Umfang verfügbar sind. Um den Gesamtdurchsatz möglichst groß zu halten, muß das Betriebssystem ebenfalls darauf achten, daß die Betriebsmittel gleichmäßig ausgelastet sind. Hinzu kommen die Forderungen, die der Benutzer der Anlage an die Verweilzeit seiner Rechenaufträge stellt.

Aus der Sicht eines Benutzers kommt es darauf an, die Verweilzeit seiner Rechenaufträge zu minimieren. Er gerät notwendigerweise in einen Interessenkonflikt mit den anderen Benutzern. Das Rechenzentrum muß die Forderungen aller Benutzer

auf großen Gesamtdurchsatz im Auge behalten und trotzdem die Verweilzeit einzelner Jobs in erträglichen Grenzen halten.

Dies gilt insbesondere für Jobs, die der Programmentwicklung dienen. Zu diesem Zweck werden besondere Jobtypen angeboten. Das sind die Jobtypen ALGW, PL/C, SNOB, SPAS, WATF, durch die schnelle Compiler aufgerufen werden, die für Testzwecke besonders geeignet sind. Diese Jobs werden praktisch sofort nach der Eingabe bearbeitet.

Zum anderen gibt es den Jobtyp TEST. Die Jobs dieses Typs werden in eine TEST-Warteschlange eingereiht. Die Position innerhalb der TEST-Warteschlange wird bestimmt durch die geschätzte CPU-Zeit (TIME-Parameter in der JOBPARM-Anweisung): Je kleiner die CPU-Zeit, desto größer die Priorität, desto günstiger ist der Platz in der Warteschlange.

Zur Zeit gilt folgende Tabelle:

Zeitbedarf (sec)	Priorität
1 - 4	10
5 - 10	8
11 - 20	7

Für Jobs, die mehr als 20 Sekunden CPU-Zeit benötigen oder verhältnismäßig viel Output erzeugen, ist der Jobtyp PROD eingerichtet. Diese Jobs gelangen in eine PROD-Warteschlange. Die Prioritäten innerhalb der PROD-Warteschlange können der folgenden Tabelle entnommen werden:

Zeitbedarf (sec)	Priorität
21 - 40	6
41 - 60	5
61 - 120	4
121 - 300	3
301 - 600	2
601 - 1000	1

Die Priorität eines Jobs steigt mit seiner Verweilzeit. Dadurch bekommt ein Job mit geringerer Priorität auch dann eine Chance, einmal initialisiert zu werden, wenn eine große Anzahl Jobs mit geringem Rechenbedarf auf Initialisierung wartet.

**Datenstationen an der Rechenanlage**

P. Janßen

Die folgende Liste gibt eine Übersicht über die Datenstationen, die derzeit an der Rechenanlage für Anwendungen angeschlossen sind oder in Kürze angeschlossen sein werden. Die Geräte sind teilweise für besondere Aufgaben bestimmt. Im Übrigen wird der Zugang von der jeweiligen Hochschuleinrichtung am Aufstellungsort geregelt.  
 (Geräteart: D=Datensichtgerät, G=graphischer Bildschirm, S=Dialogschreibmaschine, R=RJE-Station)

Anzahl	Art	Hersteller	Gerätetyp
--------	-----	------------	-----------

**Fachbereich 1:  
 Institut für neutestamentliche Textforschung**

1	D	Tandberg	TDV 2114
---	---	----------	----------

**Fachbereich 4:  
 Universitäts-Str. 14-16**

1	R	Data 100	100-76
1	D	IBM	3276
4	D	IBM	3278
2	D	Video Data	VD15
2	S	IBM	2471

**Institut für Ökonometrie**

1	D	IBM	3275
---	---	-----	------

**Institut für Siedlungswesen**

1	S	IBM	2741
---	---	-----	------

**Sonderforschungsbereich 26**

1	S	IBM	2741
1	G	Tektronix	4013

**Fachbereich 5:  
 Verwaltung der Kliniken**

1	D	IBM	3275
---	---	-----	------

Anzahl	Art	Hersteller	Gerätetyp
<b>Fachbereich 6:</b>			
<b>Medizinische Klinik</b>			
1	R	IBM	3780
<b>Medizinische Informatik</b>			
1	R, Rechner	Varian	V77-600
8	D	Hazeltine	Mod. one
<b>Fachbereich 8:</b>			
<b>Psychologisches Institut</b>			
1	G	Tektronix	4051
1	S	DEC	LA37
<b>Fachbereich 10:</b>			
<b>Sonderforschungsbereich 7</b>			
1	D	Tandberg	TDV2114
<b>Sonderforschungsbereich 164</b>			
1	D	Tandberg	TDV2114
<b>Fachbereich 11:</b>			
<b>Germanistisches Institut</b>			
1	D	Tandberg	TDV2116
<b>Fachbereich 12:</b>			
<b>Englisches Seminar</b>			
1	D	Tandberg	TDV2114
<b>Fachbereich 15:</b>			
<b>Institut für Numerische Mathematik</b>			
1	D	Video Data	VD20
1	G	Tektronix	4051

Anzahl	Art	Hersteller	Gerätetyp
<b>Fachbereich 16:</b>			
<b>Institut für Kernphysik/ Theoretische Physik</b>			
1	R	Data 100	100-71
1	D	IBM	3276
1	D	IBM	3278
<b>Institut für Geophysik/ Mineralogie</b>			
1	R	IBM	3780
1	G	Tektronix	4013
1	S	DEC	LA37
<b>Institut für Angewandte Physik</b>			
1	G	Tektronix	4051
<b>Fachbereich 17:</b>			
<b>Institut für Mineralogie</b>			
vgl. Fachbereich 16			
<b>Institut für anorganische Chemie</b>			
1	D	Video Data	VD15
<b>Institut für Organische Chemie</b>			
1	D	Video Data	VD15
1	D	IBM	3275
<b>Institut für Physikalische Chemie</b>			
1	D	Video Data	VD20
<b>Zentrale Universitätsverwaltung</b>			
<b>Dezernat 2</b>			
1	R	Data 100	100-76
2	D	IBM	3275
<b>Fachhochschule Münster</b>			
1	R, Rechner	Prime	300
<b>Pädagogische Hochschule Münster</b>			
1	G	Tektronix	4013
1	D	Video Data	VD20

Anzahl	Art	Hersteller	Gerätetyp
Rechenzentrum			
1	D	IBM	3276
2	D	IBM	3277
8	D	IBM	3278
6	D	Video Data	VD15
1	D	Tandberg	TDV2114
1	G	Tektronix	4015
1	G	Tektronix	4013
2	S	IBM	2741
1	R	IBM	3780

Turnaround-Zeiten (Juni bis August)

K. Reichel

In der folgenden Tabelle sind für verschiedene Jobtypen die Zeiten zwischen Einlesen und Ausdrucken in der Form "Stunde:Minute" dargestellt. Die mitausgedruckten Prozentzahlen beziehen sich auf die Anzahl der Jobs, die keine Monitor-Läufe sind (ca. 50% aller Jobs).

Monat	Jobtyp					
	MONITOR	TEST		PROD		LONG
		o. Band	m. Band	o. Band	m. Band	
Juni	0:01	0:06 65.9%	0:49 16.6%	4:38 11.6%	2:01 5.8%	12:08 0.1%
Juli	0:01	0:03 67.3%	0:06 15.6%	3:38 10.9%	0:50 6.1%	22:18 0.1%
August	0:01	0:02 64.2%	0:08 16.9%	3:01 11.5%	1:12 7.3%	18:16 0.1%

Die längere Turnaround-Zeit für PROD-Läufe ohne Band im Verhältnis zu denen mit Band liegt zum Teil daran, daß Läufe mit Bändern in geringerer Zahl vorliegen. Wie sehr die Turnaround-Zeiten von der Anzahl der Jobs in der betreffenden Warteschlange abhängen, zeigen die Zeiten für Läufe vom Typ PROD in den letzten Wochen, die nicht mehr so günstig sind wie in der Tabelle, da die Anzahl der Jobs sehr stark gestiegen ist.

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 Straße 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)