

inforum

INFORMATIONSFORUM des Rechenzentrums der Universität Münster

Nr. 3

Juli 1977

Inhalt

Papier sparen!	1
Warteschlangen für Jobs	2
Maschinenunabhängige FORTRAN-Programmierung	3
Druckausgabe mit personenbezogenen Daten	5
PASCAL	5
Schriftenreihe des Rechenzentrums	6
Dateien auf Magnetband	9
Leserforum	13
Lehrveranstaltungen	15
Sprechzeiten	16
Jobprofil	17

Papier sparen!

J. Ebert

Manche Benutzer (aber auch manche Compiler) sind sehr großzügig, was den Papierverbrauch (d.h. die durchschnittliche Anzahl der Zeilen pro Seite) betrifft. Gerade in der Testphase von Programmen kann man bestimmt davon ausgehen, daß die Druckausgabe nur zur Fehlersuche angeschaut wird. Dann ist es aber nicht nötig, so viel Platz aufzuwenden.

Um die Möglichkeit zu schaffen, etwas sparsamer mit dem immer teurer werdenden Papier umzugehen, wurde die /*JOBPARM-Karte eingeführt, mit deren Hilfe man den Seitenvorschub unterdrücken kann. Hierzu muß - möglichst direkt hinter der Job-Karte - eine Karte folgender Form eingefügt werden:

```
/*JOBPARM NOEJECT
```

Dies bewirkt, daß bei dem gesamten Job-Output - und das gilt auch für Batch-Läufe - anstelle eines Vorschubs auf eine neue Seite nur drei Leerzeilen eingeschoben werden.

Verwendet man katalogisierte Prozeduren, die einen Compiler aufrufen, so ist durch die Angabe CPRINT=DUMMY auf der EXEC-Karte die Möglichkeit gegeben, den Output des Compilers völlig zu unterdrücken. Dies sollte zum Beispiel immer dann genutzt werden, wenn ein Programm wiederholt unverändert übersetzt wird; denn dann ist eine Auflistung des Compiler-Outputs überflüssig.

Ist es zu Dokumentations-Zwecken oder zur Fehlersuche nötig, ausführliche Informationen über die Übersetzung zu drucken, so kann dies bekanntlich durch die Angabe von EDIT= auf der EXEC-Karte geschehen. Die Benutzung dieses Parameters sollte aber die Ausnahme bleiben und nicht aus Bequemlichkeit beim nächsten Einlesen im Job belassen werden, was leider nur zu häufig geschieht!

Außerdem soll an dieser Stelle auch noch einmal darauf hingewiesen werden, daß jeder Programmierer den von seinen Programmen erzeugten Output so entwerfen sollte, daß alle Seiten möglichst gut ausgenutzt werden.

Warteschlangen für Jobs

B. Neukäter

Jeder Auftrag zur Ausführung von Programmen (kurz: Job), der dem Rechner übergeben wird, muß vom Betriebssystem verwaltet werden. Diese Aufgabe übernimmt bei uns die Betriebssystem-Komponente HASP (Houston Automatic Spooling Priority System). HASP liest die Jobs von den Kartenlesern im Rechenzentrum und an den Terminals. Außerdem verwaltet HASP Jobs, die über CRJE-Schreibmaschinen und -Bildschirme eingegeben werden. Jeder Job wird in eine der folgenden Warteschlangen nach CPU-Zeit- und Speicherplatz-Bedarf einsortiert:

Warteschlange	Zeitbedarf (min)	Speicherplatzbedarf (KBytes)
A	1 - 4	81K - 120K
C	> 4	81K - 120K
E	1 - 4	41K - 80K
F	> 4	bis 80K
G	1 - 4	121K - 160K
H	> 4	121K - 160K
K	1 - 4	161K - 200K
M	> 4	161K - 200K
N	1 - 4	bis 40K

Darüberhinaus gibt es für jeden Batch-Compiler eine besondere Warteschlange.

Man kann demnach - wenn man den Speicherbedarf einmal außer acht läßt - die Jobs in drei Gruppen einteilen:

- I. Jobs, die einen Batch-Compiler benutzen (Batch-Jobs),
- II. Jobs mit einem Zeitbedarf von 1 bis 4 Minuten,
- III. Langläufe.

Der Benutzer kann Zeit- und Speicherplatzbedarf seines Jobs auf der JOB-Karte angeben. Die JOB-Karte

```
//ABC99XY JOB (ABC99,0020,Z23),A.USER,REGION=155K
z.B. Fordert 20 Minuten CPU-Zeit in 155 KBytes Hauptspeicher an.
HASP sortiert den Job demnach in Warteschlange H ein. Die
gleiche Wirkung hat CLASS=H anstatt REGION=155K.
```

Innerhalb der einzelnen Warteschlangen sind die Jobs nach Prioritäten geordnet. Dabei ist die Priorität eines Jobs umso höher

- je kürzer die angegebene Rechenzeit,
- je weniger Jobs der gleichen Gruppe (I, II oder III) vom gleichen Benutzer eingelesen wurden,
- je länger der Job warten mußte.

Die Priorität eines Jobs - anfänglich je nach Rechenzeit aus einer Tabelle bestimmt - wird für jeden in der gleichen Gruppe schon vorhandenen Job dieses Benutzers um eine Stufe vermindert. Danach wächst die Priorität mit zunehmender Wartezeit. Nach zwei Stunden hat sich die Priorität eines Jobs der Gruppe II z.B. um eine Stufe erhöht.

Das Vermindern der Prioritäten bei gleichem Benutzer hat zur Folge, daß es nicht sinnvoll ist, einen Zwei-Minuten-Job z.B. in zwei Ein-Minuten-Jobs zu zerlegen: der zweite Ein-Minuten-Job würde nicht eher bearbeitet als der Zwei-Minuten-Job. Da zwei Jobs mehr Betriebsmittel erfordern als ein Job, unterstützt diese Regelung den Benutzer in seinem Bestreben, die Rechenanlage optimal zu nutzen.

Maschinenunabhängige FORTRAN-Programmierung

R. Schmitt

Diese Überschrift hat nur einen Sinn, wenn es auch eine maschinenabhängige FORTRAN-Programmierung gibt. Das ist tatsächlich der Fall, auch wenn FORTRAN durchaus zu Recht als problemorientierte Programmiersprache bezeichnet wird, die es erlaubt, Probleme weitgehend maschinenunabhängig zu formulieren. Will man FORTRAN-Programme unverändert auf beliebigen EDV-Anlagen ausführen, muß man alle Maschinenabhängigkeiten vermeiden. Hier werden - ohne Anspruch auf Vollständigkeit - einige der gebräuchlichsten maschinenabhängigen Sprachelemente und -formulierungen genannt und Tips zur maschinenunabhängigen Umformulierung gegeben.

- 1) Nicht jede EDV-Anlage arbeitet mit Bytes zu 8 Bit. Längenangaben wie REAL*8 und INTEGER*2 sind daher maschinenabhängig.

Umformulierung: DOUBLE PRECISION statt REAL*8;
INTEGER*2 und LOGICAL*1 vermeiden.

- 2) Überlagerungen von Variablen unterschiedlichen Typs (durch EQUIVALENCE oder COMMON) sind zumindest dann maschinenabhängig, wenn man unterstellt, daß eine bestimmte Wertzuweisung an eine Variable einer darüberliegenden

Variablen anderen Typs einen vorhersehbaren Wert gibt.

Beispiel: EQUIVALENCE (I,X)
 I=0

Damit erhält X nicht bei allen EDV-Anlagen den Wert 0.

3) Bei Text- und Zeichenverarbeitungsaufgaben ist besondere Vorsicht geboten. Spezielle Sprachelemente, wie etwa bei WATFIV, sind nicht allgemein verbreitet. Selbst bei Beschränkung auf den Sprachstandard sind aber nahezu alle Formulierungen maschinenabhängig, insbesondere:

- die interne Verschlüsselung, d.h. die Entsprechung zwischen einer INTEGER- oder REAL-Variablen als Textzeichen und ihrem numerischen Wert.

Zuweisungen von Zeichengrößen an Variablen sollten daher nie in Kenntnis einer bestimmten Verschlüsselung als Zahlenwert erfolgen, sondern mit einer DATA-Anweisung oder durch einen READ-Befehl (A-Format).

Die interne Verschlüsselung hat insbesondere Konsequenzen für die Sortierfolge.

- die Anzahl der Zeichen, die einer einzigen INTEGER- oder REAL-Variablen zugewiesen werden können.

Die erfolversprechendste Strategie ist diejenige, die unter Verzicht auf ökonomische Speicherplatzausnutzung jeder Variablen nur ein Zeichen zuweist (READ und WRITE nach dem Format A).

Es sollte jeweils vor der Erstellung eines textverarbeitenden Programmes geprüft werden, ob FORTRAN hierzu die angemessene Sprache ist.

4) WRITE-Befehle ohne FORMAT-Anweisung erzeugen maschinenabhängige Datei-Formate.

Will man Dateien mit FORTRAN-Programmen erzeugen und auf anderen EDV-Anlagen, wiederum mit FORTRAN-Programmen, verarbeiten (Magnetband- oder Lochkartendateien), läßt sich bei Beachtung der folgenden Punkte der Umstellungsaufwand minimieren:

- READ und WRITE nur mit Format-Anweisung;
- A-Format ausschließlich für Zeichen und Texte, F- und E- bzw. D-Format nur für numerische Daten;
- Eingabeformate möglichst gleich den bei der Erzeugung benutzten Ausgabeformaten.

Das Problem maschinenabhängiger FORTRAN-Programmierung besteht an allen EDV-Anlagen und erschwert nicht nur die Übertragung unserer Programme auf andere EDV-Anlagen, sondern insbesondere die Übernahme von Programmen, die für andere EDV-Anlagen geschrieben wurden. Über Erfahrungsberichte würden wir uns freuen.

Neben maschinenabhängigen Ausdrucksmöglichkeiten enthält jede FORTRAN-Version noch Spracherweiterungen oder -einschränkungen, die eine Programmübernahme auf beliebige EDV-Anlagen erschweren können. Dieses Problem sprengt aber den Rahmen dieses Artikels.

Druckausgabe mit personenbezogenen Daten

K. Reichel

Das Rechenzentrum stellt ab Juli 1977 abschließbare Stahlböcher für Druckausgabe mit personenbezogenen Daten als Ergänzung zu den bestehenden Ausgabeböchern zur Verfügung. Fachnummern können für jeweils ein Kalenderjahr im Rechenzentrum (K. Reichel, Tel.: 2481) beantragt werden. Nähere Einzelheiten über technische und organisatorische Maßnahmen zum Schutz personenbezogener Daten (BDSG §6 Abs.1 Satz 1) können dort erfragt werden.

P_A_S_C_A_L

H. Stenzel

Die Programmiersprache PASCAL von N. Wirth findet in letzter Zeit immer mehr Beachtung. Insbesondere von zwei Seiten wird die Aufmerksamkeit auf diese Sprache gelenkt, nämlich im Zusammenhang mit der Diskussion um die Methoden und das Vermitteln der Strukturierten Programmierung und bei der Suche nach höheren Programmiersprachen, die sich zur Programmierung und Implementierung auf Mikroprozessoren eignen.

Im Rechenzentrum steht ein an der State University of New York in Stony Brook entwickelter Compiler zur Verfügung. Er stützt sich auf den Revised Report der Sprache PASCAL von 1973. Vergleichbar ist dieses System in mancher Hinsicht mit Algol W, zum Beispiel sind ähnliche Fehlerbehandlungs- und Programmtesteinrichtungen in ihm enthalten. Die Ziele, die bei der Definition der Sprache im Vordergrund standen, sind unter anderem (Zitat aus dem Report):

- 1) eine Sprache zur Verfügung zu stellen, mit deren Hilfe das Programmieren als systematische Disziplin gelehrt werden kann, und die auf gewissen fundamentalen Konzepten aufbaut, die sich auf klare und natürliche Art in der Sprache widerspiegeln (wie zum Beispiel Strukturierte Programmierung).
- 2) Implementierungen dieser Sprache zu entwickeln, die sowohl zuverlässig als auch effizient auf augenblicklich verfügbaren Rechnern sind. Dadurch ist PASCAL sowohl eine Sprache zur Vermittlung von Programmierkenntnissen und -konzepten als auch ein mächtiges Hilfsmittel (im Hinblick auf Möglichkeiten und Ausführungszeiten) für theoretische Informatiker und Programmierer.

Die Rechenzentrumsbenutzer mit Interesse an dieser Sprache werden gebeten, diesen Compiler zu erproben und uns über Ihre Erfahrungen zu berichten. Ein typisches Programm könnte wie folgt aussehen :

```

//ABC00PAS JOB (ABC00,TEST,W66),NAME,CLASS=G
//          EXEC PASCAL
//COMPILE.SYSIN DD *
PROGRAM BEISPIEL (INPUT, OUTPUT) ;
VAR CH : CHAR ;
    COUNT : ARRAY ('A' .. 'Z') OF INTEGER ;
    LETTER : SET OF 'A' .. 'Z' ;
BEGIN
    LETTER := ('A' .. 'Z'.) ;
    FOR CH := 'A' TO 'Z' DO COUNT (CH) := 0 ;
    WHILE NOT EOF DO
    BEGIN
        WHILE NOT EOLN DO
        BEGIN READ (CH) ; WRITE (CH) ;
            IF CH IN LETTER THEN COUNT (CH) := COUNT (CH) + 1 ;
        END ;
        WRITELN ; READLN
    END
END .
%EOF
DIESER TEXT WIRD VON OBIGEM PROGRAMM GELISTET, UND DIE
BUCHSTABEN WERDEN GEZAEHLT.
//

```

Weitere Hinweise zu dieser Implementierung und Ergänzungen zum Report, die diesen Compiler betreffen, können von mir angefordert werden. Aus der immer umfangreicher werdenden Literatur über PASCAL möchte ich nur zwei 'Klassiker' nennen :

- K.Jensen, N.Wirth, PASCAL User Manual and Report, Springer 1974, Lecture Notes in Computer Science 18, ca. DM 20,-.
- N.Wirth, Systematisches Programmieren, Teubner Studienbücher Informatik, 1975, ca. DM 20,-.

Schriftenreihe des Rechenzentrums

W.A. Slaby

Mit seiner Schriftenreihe, die seit März 1974 herausgegeben wird, verfolgt das Rechenzentrum die Absicht, insbesondere Berichte über Forschungsarbeiten aber auch Vorlesungsausarbeitungen und Dokumentationen größerer Programmsysteme, die von Mitarbeitern des Rechenzentrums oder mit ihrer Beteiligung erstellt werden, dem an Problemen der Datenverarbeitung und Informatik interessierten Leser zugänglich zu machen.

Inzwischen sind in dieser Reihe 25 Titel erschienen bzw. befinden sich kurz vor dem Erscheinen, wie Sie der nachfolgenden Liste entnehmen können (die mit einem * gekennzeichneten Nummern sind leider bereits vergriffen). Die einzelnen Titel können im Sekretariat des Rechenzentrums bei Frau E. Geisler (Zi. 101) zum angegebenen Selbstkostenpreis käuflich erworben werden.

Zusammenstellung der Schriftenreihe des Rechenzentrums

- Nr. 1 - Möglichkeiten interdisziplinärer Zusammenarbeit am Beispiel von Mathematik und Medizin
von Helmut Werner, März 1974; 2,- DM (13 Seiten)
- Nr. 2 - Automatische Übersetzung in Blindenkurzschrift
von Wolfgang A. Slaby, Mai 1974; 2,- DM (16 Seiten)
- Nr. 3* - Einführung in das Arbeiten mit C P S (Conversational Programming System)
von Walter Bosse, Mai 1974; 5,- DM (82 Seiten)
- Nr. 4 - Formeln zur Berechnung der optischen Größen, die bei den mit Brillen und Haftschalen korrigierten Augen auftreten
von H. Werner, H. Gernet, H. Ostholt, G. Neuser, Juni 1974; 2,- DM (26 Seiten)
- Nr. 5* - Implementierung eines Verfahrens zur Approximation empirischer Daten mit einer Summe von gewichteten Gaußverteilungen
von Maike Bestehorn, Juni 1974; 5,- DM (82 Seiten)
- Nr. 6* - Nichtlineare Anpassung von Meßdaten nach der Methode der kleinsten Quadrate
von Dietrich Braess, Juni 1974; 2,- DM (14 Seiten)
- Nr. 7* - FORMAC (Formula Manipulation Compiler) Eine Einführung in die PL/I-FORMAC-Sprache
von Hilmar Pudlitz, Oktober 1974; 5,- DM (76 Seiten)
- Nr. 8 - The P/9M Interactive Graphic System
von M. Bestehorn, G.G. Pieroni, H. Stenzel, Januar 1975; 4,- DM (54 Seiten)
- Nr. 9 - Computerized Braille Production
Proceedings of the 1. International Workshop in Münster (Germany), March 1973
von R.A.J. Gildea, G. Hübner, H. Werner, Dezember 1974; 5,-DM (95 Seiten)
- Nr. 10 - Linguistische Probleme bei der automatischen Produktion der deutschen Blindenkurzschrift
von Jochen Splett, Dezember 1974; 3,- DM (31 Seiten)
- Nr. 11* - Die Konstruktion bester Tschebyscheff-Approximationen mit Splines bei freien Knoten
von Albrecht Baumann, Dezember 1974; 5,- DM (82 Seiten)

- Nr. 12 - Plotter-Software - Erweiterte Anwendungssoftware
von Horst Stenzel, Bernd Schulze, Januar 1975;
2. verbesserte und erweiterte Auflage, September 1976;
3,- DM (41 Seiten)
- Nr. 13 - Informatik - Ein Überblick
von K.B. Mertz, B. Neukäter, W.A. Slaby,
September 1975; 5,- DM (107 Seiten)
- Nr. 14* - Einführung in das Programmieren mit ALGOL W
von Walter Bosse, September 1975; 7,50 DM (232 Seiten)
- Nr. 15 - DMP (Data Management Program)
Ein Programmsystem zur Verwaltung strukturierter Daten
von Hans-Werner Kisker, erscheint in Kürze
- Nr. 16 - GEOMAP - ein FORTRAN-Programm zur Erzeugung von
Choroplethen- und Isolinienkarten auf dem
Schnelldrucker
von Hilmar Pudlitz, Januar 1976; 2,- DM (23 Seiten)
- Nr. 18 - Clusteranalyse großer Datenmengen - effektive
Verfahren und Programme zur Clusterung quantitativer
Daten
von Detlef Steinhausen, erscheint in Kürze
- Nr. 19 - Optimale Auslegung von Bestrahlungsplänen
von Udo Ebert, März 1976; 4,- DM (68 Seiten)
- Nr. 20 - Algorithmen für nichtlineare Optimierungsaufgaben ohne
Nebenbedingungen
von Ulrich Hornung, März 1976; 2,- DM (27 Seiten)
- Nr. 21 - I M S - Benutzerhandbuch
Hinweise für den Einsatz von IMS
von Roland Schmitt, April 1976; 5,- DM (75 Seiten)
- Nr. 22 - Exponentialapproximation mit polynomialen Exponenten
in der Tschebyscheff-Norm
von Maike Bestehorn, erscheint in Kürze
- Nr. 23 - Eine Regularisierung der Splines mit freien Knoten
von Jan Brink-Spalink, April 1977; 4,- DM (56 Seiten)
- Nr. 24 - Automatische Erzeugung formaler Übersetzungssysteme
aus endlichen Mengen von Beispielen
von Wolfgang A. Slaby, erscheint im Juli 1977
- Nr. 25 - Operator-Algorithmen auf Matrizen-Darstellungen von
Graphen
von Jürgen Ebert, erscheint in Kürze

Dateien_auf_Magnetband

W. Bosse

Dieser Artikel soll eine einführende Übersicht bezüglich der Beschreibung von Dateien geben, die auf Magnetband gespeichert sind. Es werden die zum Anlegen und Auffinden solcher Dateien notwendigen Parameter beschrieben, die in der Job-Kontrollsprache (JCL) für die Dateiverwaltungsroutinen des Betriebssystems zu formulieren sind. In Verbindung damit werden auch einige grundlegende Begriffe erläutert.

Die von einem Programm zu verarbeitenden Informationen (Daten) werden bei Batch-Anwendungen häufig auf Lochkarten bereitgestellt, können aber ebenso auf anderen Datenträgern vorliegen, z.B. auf einem Magnetband. Magnetbänder eignen sich besonders zur sequentiellen Speicherung und Verarbeitung großer Datenmengen.

Das Magnetband ist einem Tonband vergleichbar; die Beschriftung erfolgt in mehreren, in der Längsrichtung des Magnetbandes parallel zueinander liegenden Spuren, für die jeweils ein Schreib-/Lesekopf in der Magnetbandeinheit vorhanden ist. Dadurch ist es möglich, in jeder Position des Bandes alle zur Darstellung eines Zeichens notwendigen Bits (und ein zusätzliches Prüfbit zum Schutz vor Übertragungsfehlern) gleichzeitig zu schreiben bzw. zu lesen. Die Schreibdichte, d.h. die Anzahl der Schreib-Positionen auf dem Magnetband pro Längeneinheit, hängt von den technischen Gegebenheiten der Geräte zur Magnetbandverarbeitung ab.

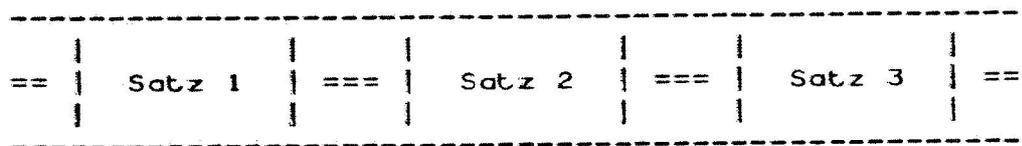
Entsprechend der Zeichenverschlüsselung im EBCDIC (Extended Binary Coded Decimal Interchange Code) erfolgt bei allen Magnetbandlaufwerken des Rechenzentrums die Aufzeichnung der Daten in 9 Spuren (8 Bits und 1 Prüfbit); die übliche Schreibdichte beträgt 1600 bpi (bytes per inch), d.h. 640 Zeichen/cm. Obwohl die Magnetbänder natürlich verschieden lang sein können, sind 730 m als Bandlänge üblich.

Charakteristisch für Magnetbandgeräte ist der Start-Stop-Betrieb und die sich daraus ergebende Gruppierung der Informationen in sogenannte Blöcke. Das Band wird jeweils zum Lesen oder Schreiben eines Blocks gestartet und am Ende des Blocks wieder gestoppt. Dadurch entsteht zwischen je zwei Blöcken eine Lücke (gap) von 1,5 cm Länge. Wenn z.B. immer der Inhalt einer Lochkarte (80 Zeichen) auf ein Magnetband übertragen wird, entstehen - bei einer Schreibdichte von 1600 bpi - Blöcke von 0,125 cm Länge; dies bedingt ein ungünstiges Längenverhältnis zwischen Block und Lücke.

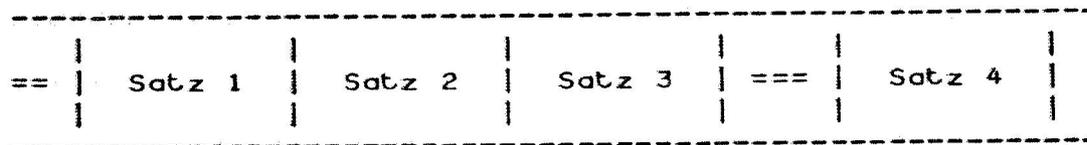
Deshalb faßt man im allgemeinen mehrere, von einem Programm zu verarbeitende, logische Sätze (logical records) zu einem Block zusammen, der im Pufferbereich des Hauptspeichers aufgebaut und dann zusammenhängend übertragen wird (Schreiben) bzw. nach der Übertragung vom Magnetband im Puffer entsprechend aufgeteilt wird (Lesen). Die Verwaltung der Blöcke im Puffer

erfolgt dabei automatisch durch Routinen des Betriebssystems. Extrem große Blocklängen sind allerdings nicht zu empfehlen, da entsprechend viel Platz im Pufferbereich des Hauptspeichers reserviert werden muß. (Eine Blocklänge von ca. 4000 Bytes kann als günstig angesehen werden.) Wenn jeder Block nur einen logischen Satz enthält, nennt man diese Speicherung ungeblockt (andernfalls: geblockt).

ungeblockt:



geblockt:



Unter einer Datei versteht man eine Datenstruktur (als logisch zusammenhängende Menge von Daten), die auf einem externen Speichermedium - z.B. einem Magnetband - vorliegt. Ein Programm kann immer nur Teile davon im Hauptspeicher verarbeiten, nämlich die logischen Sätze. Diese beschreiben zueinander ähnliche Unterstrukturen und nehmen im einfachsten Fall jeweils einen Block des externen Speichers ein (ungeblockte Speicherung).

Jede Datei, die innerhalb eines Jobs angesprochen werden soll, muß der Dateiverwaltung des Betriebssystems beschrieben werden; dies geschieht durch ein DD-Statement (Data Definition Statement). Diese Statements der Job-Kontrollsprache müssen jeweils dem Step zugeordnet werden, der die Verarbeitung der beschriebenen Datei durchführt.

Zur Beschreibung von Magnetband-Dateien sind folgende Parameter des DD-Statements wichtig:

Parameter	Abkürzung	Bedeutung/Verwendung
DSNAME	DSN	Name der Datei
DISP		Verfügbarkeit
UNIT		Eingabe-/Ausgabe-Einheit
VOLUME	VOL	Bezeichnung des Bandes
LABEL		Position auf dem Band
DCB		Datei-Attribute

Auf einem Magnetband werden die Dateien hintereinander gespeichert. Daraus ergibt sich, daß nur die jeweils letzte Datei modifiziert werden kann, ohne nachfolgende Dateien zu beeinträchtigen. Eine neue Datei sollte grundsätzlich hinter allen bereits vorhandenen angelegt werden. Löschen einer Datei geschieht durch Überschreiben bzw. Anlegen einer neuen Datei in derselben Position. Dabei ist allerdings Vorsicht geboten, weil anschließend alle nachfolgenden Dateien ebenfalls nicht mehr verfügbar sind.

Im folgenden muß also nur noch beschrieben werden, mittels welcher Parameter eine neue Datei auf Magnetband angelegt werden kann und wie sich eine bereits vorhandene Datei wieder auffinden läßt. Klein geschriebene Angaben beziehen sich auf Eintragungen, die entsprechend den sich an die Darstellung der DD-Statements anschließenden Erläuterungen vorzunehmen sind; die geschweiften Klammern [] gestatten eine Auswahl der untereinander stehenden Alternativen. Da in der Regel alle Angaben blank-frei, d.h. ohne Leerzeichen, gemacht werden müssen, wird durch „_“ explizit angegeben, wo ein oder mehrere Leerzeichen stehen müssen.

Anlegen einer Datei:

```
//ddname_DD_DSN=dsname,DISP=(NEW,[PASS]),
                               [KEEP]
//_UNIT=TAPE,VOL=SER=tape#,LABEL=position,
//_DCB=(RECFM=format,LRECL=record,BLKSIZE=block)
```

Auffinden einer Datei:

```
//ddname_DD_DSN=dsname,DISP=( [NEW] , [PASS] ),
                               [MOD] [KEEP]
//_UNIT=TAPE,VOL=SER=tape#,LABEL=position
```

Spezialfall:

Die Datei wurde bereits in einem früheren Step desselben Jobs verwendet und durch PASS gekennzeichnet.

```
//ddname_DD_DSN=dsname,DISP=( [NEW] , [PASS] )
                               [MOD] [KEEP]
```

Erläuterungen:

ddname Dieser Name darf aus bis zu 8 Zeichen (Buchstaben,#,\$,@,Ziffern) bestehen, aber nicht mit einer Ziffer beginnen; er macht die in dem DD-Statement gegebene Dateibeschreibung innerhalb des Programms verfügbar. Für FORTRAN-IV-Programme muß ddname die spezielle Form FTxxF001 haben, wobei xx jeweils die im Programm verwendete Datei-Nummer angibt, z.B. 03. Jedes einem Step zugeordnete DD-Statement muß einen eindeutig gewählten ddname haben. Wenn eine katalogisierte

Prozedur aufgerufen wird, muß die Dateibeschreibung einem Step der Prozedur zugeordnet werden, indem der Name des Steps (procstepname) dem dsname vorangestellt wird:

```

procstepname.dsname
Beispiele:   BAND
             EXECUTE.DATEN
             EXECUTE.FT03F001

```

dsname bezeichnet den Namen einer physischen Datei auf Magnetband; er soll durch die Verrechnungsnummer des Benutzers qualifiziert sein und kann anschließend aus bis zu 8 Zeichen bestehen (vgl. dsname).
Beispiel: UUU01.WERT200X

NEW Die Datei soll neu angelegt werden.

OLD Die Datei existiert bereits.

MOD Die bestehende Datei soll modifiziert, d.h. am Ende erweitert werden.

PASS Die Datei wird in einem folgenden Step desselben Jobs nochmals verwendet. Da in diesem Fall das Magnetband nach Abschluß der Datei nicht automatisch zurückgespult wird, empfiehlt sich diese Angabe auch für den Fall, daß eine auf dem Band folgende Datei verarbeitet werden soll.

KEEP Die Datei wird abgeschlossen, und das Magnetband wird zurückgespult.

TAPE Diese Angabe beschreibt die Magnetbandlaufwerke. Das Betriebssystem wählt automatisch eines davon aus.

tape# bezeichnet die Nummer des zugeteilten Magnetbandes.

position gibt die Position der Datei auf dem Magnetband an (implizit: 1).

format beschreibt das Format, das für die Speicherung der Sätze verwendet werden soll (record format). Alle Sätze einer Datei müssen dasselbe Format haben.

```

F      - Sätze fester Länge, ungeblockt
FB     - Sätze fester Länge, geblockt
V      - Sätze variabler Länge, ungeblockt
VB     - Sätze variabler Länge, geblockt

```

record legt die Länge eines logischen Satzes (logical record length) in Bytes fest. Wenn als Satzformat (format) V oder VB angegeben wird, muß record die maximale Satzlänge + 4 beschreiben.

block gibt die Länge eines physischen Blocks (block size) in Bytes an. Beim Satzformat (Format) V oder VB muß für block die maximale Blocklänge + 4 (= n*record+4) angegeben werden, wobei n die Anzahl der Sätze pro Block beschreibt.

Beispiele:

a) zum Anlegen einer Datei

```
//EXECUTE.DATEN DD DSN=UUU01.WERT200X,DISP=(NEW,KEEP),
// UNIT=TAPE,VOL=SER=000077,LABEL=2,
// DCB=(RECFM=FB,LRECL=126,BLKSIZE=3780)
```

b) zum Auffinden einer Datei

```
//EXECUTE.INPUT DD DSN=UUU01.WERT200X,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=SER=000077,LABEL=2
```

Als Richtwert für die Nutzung von Magnetbändern (730 m) können maximal 16 Dateien pro Band mit einem durchschnittlichen Umfang von 400 bis 800 Blöcken angegeben werden, wobei ca. 4000 Bytes als günstige Blocklänge anzusehen sind. Wesentlich größere Dateien sollten nach Möglichkeit als erste oder letzte Datei auf einem Band stehen.

Leserforum

Ich bin über die Systemprogramme und Programmbibliothek nicht informiert und würde daher gerne mehr über deren Anwendungsmöglichkeiten in FORTRAN IV erfahren.

H. Amend

Informationen über Systemprogramme vermittelt die regelmäßig stattfindende Veranstaltung "Kontrollsprachen des Betriebssystems".

Als FORTRAN-Programmbibliothek empfehlen wir die NAG-Bibliothek (inforum Nr. 2). Beratung hierzu wird mo-fr 11-12, in Zimmer 108 (Roxeler Straße 60) durchgeführt.

R. Schmitt

Jedem Programmierer kann es passieren, daß ihm bei der Erstellung des Programms ein Fehler unterläuft, der sich verheerend beim Drucken seines Outputs auswirkt. Es kann ja durchaus sein, daß versehentlich eine große Anzahl von Seiten ausgedruckt wird, die durch einen Programmierfehler nur eine

Zeile pro Seite enthält. Ich schlage hiermit vor, daß die Systemgruppe des RZ im HASP eine Änderung vornimmt, damit so etwas ohne den Eingriff eines Operators unterdrückt werden kann. Als Regelung könnte man 5 Seiten mit ≤ 3 Zeilen zulassen. Die restlichen Informationen sollten dann direkt anschließend mit einem Hinweis von HASP ausgedruckt werden.

M. van Os

Inst. f. Med. Informatik
u. Biomathematik

Seit Aufstellung der Rechanlage beobachten wir das oben beschriebene Problem, da jedoch eine algorithmische Entscheidung über Absicht oder Fehler bei der Erstellung einer Liste sehr schwierig ist (z.B. besteht ein typischer FORTRAN-Fehler darin, daß nach zwei oder drei korrekten Seiten die erste Ziffer ("1") einer Ausgabezeile als Vorschubzeichen interpretiert wird und dann eine große Zahl fast weißer Blätter erzeugt), haben wir im vergangenen Jahr die /*JOBPARM-Kontrollkarte mit dem Parameter NOEJECT (oder EJECT=NO) eingeführt. Diese Einrichtung sollte in der Test-Phase eines Programms genutzt werden, um die Auswirkung der beschriebenen Fehler zu reduzieren.

Wir verstehen die Anregung als Anstoß zu der Überlegung, ob nicht als Standard-Aktion der Vorschub wie bei "NOEJECT" behandelt werden sollte und erst für die Produktions-Jobs vom Benutzer eine Kontrollkarte mit der Eintragung /*JOBPARM EJECT in den Job einzufügen wäre, und wären allen Benutzern für Kommentare in dieser Angelegenheit dankbar.

H. Meyer

IMPRESSUM

Redaktion inforum

W. Bosse	(Tel.: 490-2476)
J. Ebert	(Tel.: 490-2607)
P. Lommel	(Tel.: 490-2686)
R. Schmitt	(Tel.: 490-2475)

Rechenzentrum der Universität
Roxeler Str. 60
4400 Münster

Auflage dieser Ausgabe: 500
Redaktionsschluß der nächsten Ausgabe: 30.9.1977

Lehrveranstaltungen

In der vorlesungsfreien Zeit vor Beginn des WS 1977/78 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 27.6. bis 15.7.1977
bei H. Mecke im Dispatch des Rechenzentrums (Raum 07) in die Anmelde Listen einzutragen.

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

5.9. - 23.9.1977: FORTRAN IV

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

Dozenten: Eickenscheidt, Reichel, Zörkendörfer
Vorl.-Nr.: 220020, 220035, 220040
Hörsäle: M2, M3, M4
Beginn: 5.9.1977 um 9 Uhr c.t. im M2

26.9. - 14.10.1977: ALGOL W

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

Dozent: Wessels
Vorl.-Nr.: 220088
Hörsaal: M6
Beginn: 26.9.1977 um 9 Uhr c.t. im M6

29.9. - 14.10.1977: 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: Bestehorn, Pudlatz, Sturm
Vorl.-Nr.: 220054, 220069, 220073
Hörsäle: M2, M3, M4, M5
Beginn: 29.9.1977 um 9 Uhr c.t. im M2

Die Teilnehmer der Kurse werden gebeten, diese im WS 77/78 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 27.6. bis 12.7.1977

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

Sprechzeiten

Da die Mitarbeiter des Rechenzentrums in der Beratung und Unterstützung der Benutzer tätig sind, darüberhinaus aber auch verschiedene andere Aufgaben im Rechenzentrum wahrzunehmen haben, werden im folgenden regelmäßige Zeiten für die Kontaktaufnahme angegeben, zu denen sie persönlich oder telefonisch erreichbar sind.

Name	Zimmer	Tel.	Zeit
Ahrens	R 112	2672	mo-fr 14-16
Dr. Bestehorn	H 02	2483	di 10-12
Blanke	H 31	2678	di 14-16
Bosse	H 26	2476	do 9-11
Ebert, J.	H 25	2607	mo 9-11
Dr. Ebert, U.	H 13	2682	fr 10-11
Eickenscheidt	R 109	2673	n. V.
Goorkotte	R 112	2672	mo-fr 13.30-17
Dr. Kamp	H 11	2474	di,do 11-12
Kisker	H 13	2682	do 10-12
Lommel	H 34	2686	mi 9-11
Mecke	R 07	2466	mo-fr 9-12,14-16
Dr. Mertz	H 16	2683	mi 11-12
Meyer	H 21	2685	do 9-11
Nabrotzki	H 14	2468	di 9-11
Neukäter	H 22	2477	mi 9-11
Dr. Pudlitz	H 15	2472	di 11-12
Reichel	R 04	2481	mi 9-11
Schmitt	H 33	2475	di 9-11
Slaby	H 12	2473	n. V.
Dr. Steinhausen	H 36	2464	mo 10-12
Stenzel	H 35	2608	mo 9-11
Sturm	H 32	2609	mi 9-11
Wessels	H 05	2681	mi 10-12
Dr. Zörkendörfer	H 01	2471	di,do 11
Zwick	R 110	3799	n. V.

Gebäude:

H = Hittorfstraße 27

R = Roxeler Straße 60

Jobprofil

J. Ebert, W. Schafmann

Die folgenden Tabellen zeigen (anhand der Daten des Monats März 1977), wie sich die Benutzer-Jobs auf die Klassen A bis N und die "Batch"-Compiler aufteilen.

Teilt man die bearbeiteten Jobs nach ihrer Klasse und der im Account-Feld geschätzten Zeit auf, so lag im März 1977 zum Beispiel folgende Verteilung vor:

Verteilung nach Zeit-Schätzungen (min)

CLASS	0<T<= 2	2<T<= 4	4<T<=10	10<T<=30	30<T<=60	60<T
A	2477	1076	0	0	0	0
C	0	0	681	112	7	0
E	3952	559	0	0	0	0
F	0	0	182	22	1	0
G	1063	315	0	0	0	0
H	0	0	230	49	0	0
K	214	246	0	0	0	0
M	0	0	255	60	7	0
N	434	12	0	0	0	0
ALGW	895	0	0	0	0	0
WATF	3105	0	0	0	0	0
PL/C	2763	0	0	0	0	0
LIST	373	0	0	0	0	0
DUPL	453	0	0	0	0	0

Es zeigt sich also, daß die kleinen Jobs bei weitem den größten Anteil haben. Noch deutlicher wird diese Tatsache, wenn man die Jobs nach ihrem wirklichen Zeitverbrauch gruppiert. Dann wird klar, daß für Produktionszwecke (mindestens 10 Minuten CPU-Zeit) kaum Rechenkapazität vorhanden ist.

Verteilung nach Zeit-Verbrauch (min)

CLASS	0<T<= 2	2<T<= 4	4<T<=10	10<T<=30	30<T<=60	60<T
A	3286	259	0	0	0	0
C	230	112	379	71	1	0
E	4350	151	0	0	0	0
F	43	35	113	13	1	0
G	1304	73	0	0	0	0
H	74	50	122	31	0	0
K	419	41	0	0	0	0
M	106	62	122	25	7	0
N	445	1	0	0	0	0

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)