

informum

INFormationsforum des Rechenzentrums der Universität Münster

Jahrgang 6, Nr. 2

April 1982

Inhalt	Seite
Schwierigkeiten beim Übergang zum neuen Betriebssystem	2
Betriebliche Mitteilungen	3
Hardware und Systemsoftware	4
Kleine Chronik der Fehler im neuen Betriebssystem	5
Leserforum	7
Personalia	8
Mitteilungen über die Rechentätigkeit 1981	9
Lehrveranstaltungen im SS 1982	10
Einführung einer neuen Version des JES 2	11
Dokumentation tut not	13
INTERLISP/370 Advanced System unter CMS	15
NAG-Bibliothek	16
SSP/PL1-Bibliothek	18
Zufallszahlen/Simulationen	19
Schriftenreihe des Rechenzentrums	20
Ein/Ausgabe in VS FORTRAN	21
informum-Quiz	26
Programmquerschnitt Januar - April	27

Schwierigkeiten beim Übergang zum neuen Betriebssystem

W. Held

Große Betriebssysteme werden vom Hersteller regelmäßig weiterentwickelt, um neue Anforderungen und Funktionen zu unterstützen und um Fehler zu beseitigen, die im Laufe der Zeit in einzelnen Rechenzentren bei bestimmten Konstellationen der Eingabeparameter auftreten. Nicht alle Änderungen sind für den eigenen Betrieb von Bedeutung, sie müssen also nicht immer sofort zur Nutzung bereitgestellt werden.

Allerdings kann man das Betriebssystem nicht völlig "einfrieren", da man dann vom Hersteller im Fehlerfall nicht mehr unterstützt wird (mit einem gewissen Recht wird von ihm jeweils nur die neueste Version gepflegt). Außerdem wird man von Zeit zu Zeit auch in Hochschulrechenzentren neue Leistungen bereitstellen müssen.

So hat uns das letzte Jahr eine große Zahl neuer Anforderungen gebracht, die vom Betriebssystem befriedigt werden müssen. Einige sollen kurz erwähnt werden: mehr Terminals und neue Terminaltypen, mehr Dialog-Betrieb, größere Plattenspeicheranforderungen, neue Softwarepakete, umfangreicherer Datenschutz, operatorloser Rechenbetrieb am Wochenende.

Bei der Untersuchung dieser Anforderungen stellten sich immer häufiger die Grenzen des bisherigen MVS-Systems heraus, und wir meinten, mit der neuen Version MVS/SP V1 R 3.0 eine gute Grundlage für den weiteren Ausbau des Rechenzentrums zu erhalten.

Nachdem andere Rechenzentren ihre Systeme bereits umgestellt hatten und unsere Systemgruppe die geplante Änderung mit großem Arbeitsaufwand und unermüdlichem persönlichen Einsatz vorbereitet hatte, wurde sie schließlich zum 1.3.82 vorgenommen.

Die Folge war ein über einen Zeitraum von etwa 3 Wochen nachhaltig gestörter Betrieb bei unerträglichem Antwortzeitverhalten, das auch jetzt noch weiter verbessert werden muß. Was war geschehen? Das neue System bot sich anfangs wie jede neue Version dar: Es gab einige Störungen, es wurden Fehler korrigiert und es wurden wieder neue sichtbar. Das ist der durchaus übliche Ablauf bei größeren Systemumstellungen. Von den meisten Fehlern merken i.a. nur einige wenige Benutzer und nicht alle etwas. Diesmal trat jedoch nach einigen Tagen immer wieder ein Fehler auf, der zum Absturz des Systems führte und sich weder reproduzieren noch eingrenzen ließ. Der "zentrale Service", mit dem die IBM eine telefonische Systemberatung anbietet, und die Mitarbeiter unserer Systemgruppe versuchten vergeblich, den Fehler in einem geeigneten Dump einzufangen. Da andere Rechenzentren diese Schwierigkeiten nicht gehabt hatten, mußte die Ursache in Münster gesucht werden, ein Rückgang auf das alte System war also nicht ratsam.

Früher wurde in derart verfahrenen Situationen die "lokale Programmunterstützung" der IBM zu Rate gezogen. Da dieser Service inzwischen kostenpflichtig geworden ist und unser Softwareetat seine Inanspruchnahme nicht mehr gestattet, konnten wir ohne vertragliche Grundlage nur langsam wirksame Hilfe vom Hersteller bekommen.

Schließlich gelang ein geeigneter Fehlerdump, der die Ursache erkennen ließ: Nicht das neue Betriebssystem arbeitete falsch, vielmehr lösten die Magnetplatten der Firma Memorex, die unter dem alten System noch störungsfrei gearbeitet hatten, den Fehler aus. Die IBM verwendete im neuen Betriebssystem Plattenzugriffsmethoden, die der Firma Memorex zwar bekannt waren, die sie aber noch nicht durch entsprechende Änderung des Mikroprogramms in unserer Magnetplattensteuereinheit implementiert hatte. Mixed Hardware!

Da Systemänderungen auch in Zukunft notwendig sein werden, stellt sich natürlich die Frage, wie man derart massive Störungen in Zukunft vermeiden kann. Diese Frage ist vor allem unter den Gesichtspunkten der Verteuerung des Softwareangebotes und der damit verbundenen Unterstützung durch die IBM auf der einen Seite und der Kürzung der dafür verfügbaren Haushaltsmittel auf der anderen Seite zu beantworten. Der vom Hersteller angebotene Dienst "zentraler Service", der bisher noch nicht über die Softwaregebühren hinaus bezahlt werden muß, reicht allein offensichtlich nicht aus, komplexe Fehlersituationen, wie sie bei Systemumstellungen auftreten können, schnell zu beheben. Erste Gespräche sind deshalb bereits mit Vertretern der Firma IBM geführt worden. Auch der Rechnerhersteller mußte trotz reduzierter Haushaltsmittel ein Interesse daran haben, daß seine Anlagen auch in Hochschulrechenzentren einwandfrei arbeiten.

Betriebliche Mitteilungen

Seit Anfang April können die Geräte des Rechenzentrums (Locher, Datensichtgeräte, Lochkartenleser, etc.) montags bis donnerstags auch über 22 Uhr hinaus genutzt werden. Die Zeiten für den Zugang zum Rechenzentrumsgebäude Einsteinstr. 60 werden davon nicht berührt; ein Verlassen des Gebäudes ist natürlich jederzeit möglich.

Hardware und Systemsoftware

H. Meyer-Rinsche

Im Verlaufe der Vorbereitungen für die Einführung neuer Systemsoftware sind in den vergangenen acht Monaten verschiedene Fehler aufgetreten, deren Erwähnung etwas Verständnis für die Arbeit des Rechenzentrums vermitteln soll.

Betriebssystemkomponenten und Systemdienstprogramme nutzen die Möglichkeiten vorhandener Hardware (Rechnerkomplex, Magnetplatteneinheiten und Magnetbandeinheiten) sehr viel umfassender als jedes anwendungsorientierte Benutzerprogramm. Darüber hinaus ist es dem Systemprogrammierer oftmals nicht möglich, durch Steuerparameter auf Schwachstellen in der Systemkonfiguration einzugehen und die Dienstprogramme so einzustellen, daß Checkpoints genommen oder Blockungsfaktoren angemessen eingestellt werden. Ferner arbeiten verschiedene Systemprogramme mit Privilegien, die eine totale Überwachung ausschalten. Hier seien fünf Beispiele beschrieben, die den Einsatz von Systemprogrammen stark beeinträchtigt haben.

1. Durch die spezielle Verdrahtung einer Magnetplattenkontroll-einheit, die für die Rechenanlage überflüssig ist und vom Rechenzentrum nicht erkannt geschweige denn gewünscht worden war, konnte das Betriebssystem nach bestimmten Kanalbefehlen in einigen Dienstprogrammen nicht mehr die korrekte Ausführung dieser Kommandos feststellen und stürzte ab.
2. Der Dump einer Magnetplatte auf Magnetband (das ist das Abschreiben sämtlicher Informationen einer Platte) benötigt sechs bis sieben Bandrollen. Bei der verheerenden Beschaffenheit der Magnetbandeinheiten im letzten Jahr, wobei immer wieder Magnetbänder zerissen wurden (ca. 1 von 10 aufgelegten Bändern), bestand kaum die Möglichkeit, eine Platte abzuschreiben und später wieder zu restaurieren.
3. Beim Dump einer Magnetplatte werden Blöcke geschrieben, die die Länge einer Spur der Platteneinheit haben können, das sind bei uns bis zu 20 000 Bytes. Zwischenzeitlich war es unsicher, daß einmal geschriebene Blöcke einwandfrei gelesen werden konnten, da - wie die Firma MEMOREX mitteilte - die Toleranzen der Schreib/Leseköpfe derart lange Blöcke nicht zuließen. Das Restaurieren von Magnetplatten zum Zwecke der Herstellung gesicherter Arbeitsstände wurde damit zum Abenteuer, das uns sechs bis acht Wochen Vorbereitungszeit für die Systemumstellung kostete.
4. Fehler in der Zentraleinheit führten dazu, daß lang laufende Systemdienstprogramme (bis zu 18 Stunden Realzeit) nicht mit der Chance auf vollständige Abwicklung gestartet werden konnten. Diese Programme verändern Dateien in einer Weise, daß nach einem Systemzusammenbruch der Inhalt dieser Dateien wertlos ist.
5. Nach Umstellung des Betriebssystems wurde ein wesentlich raffinierterer Mechanismus des Seitenwechsels eingesetzt. Die

Kanalbefehle für diesen Seitenwechsel überforderten die zu dieser Zeit eingesetzten Mikroprogramme in den Plattenkontrolleinheiten, so daß diese fehlerhaft reagierten und das Betriebssystem zum Aufgeben veranlaßten.

Kleine Chronik der Fehler im neuen Betriebssystem

H. Meyer-Rinsche

Die neue Systemsoftware wurde in der Zeit vom Dezember 1981 bis zum Februar 1982 eingerichtet und im Rahmen der Möglichkeiten der Systemgruppe ausgetestet. Die nach der Umstellung im März und April aufgetretenen Fehler waren leider von der Art, daß erst eine volle Dauernutzung des Systems diese zu Tage fördern konnte. Im folgenden sei eine kleine Auswahl aufgetretener Fehler und unserer Bemühungen, sie zu beheben, angegeben:

1. Ein Adressierungsfehler (S0C4) in verschiedenen Systemkomponenten, der zu Systemzusammenbrüchen führt, erweist sich als Fehler in einer Routine, die der Aufzeichnung von Hardware-Defekten dient und die bei einigen sporadisch auftretenden Gerätefehlern Systemvariablen überspeichert.
2. Die von einer Benutzergruppe verwendete Technik, in Jobs mit mehreren Steps JCL-Anweisungen mit Rückwärtsreferenzen der Form "DSN=*.stepname.procstep.ddname" zu verwenden, führt zu Fehlern in der Abarbeitung dieser Jobs; diese Art von Rückwärtsreferenzen kommt in keiner vom RZ zur Verfügung gestellten Prozedur vor, so daß der Fehler bei Tests nicht aufgetreten ist.
3. Das Betriebssystem sieht eine neue Technik der Organisation von VSAM-Dateien vor; der Einsatz dieser neuen Technik bei der Verwaltung der Informationen für das "RECLAIM" von VSPC-Files führt zu VSPC-Abstürzen und wird rückgängig gemacht.
4. Bildschirmgeräte, die mit der BSC-Leitungsprozedur an das System angeschlossen sind, werden durch die auf den letzten Wartungsstand gebrachte (jetzt aber dennoch nicht mehr gewartete) Version des Netzwerk-Steuerprogramms (ACF/VTAM Release 1) nicht korrekt bedient, so daß nach einem VSPC-OFF ein neues Logon nur nach Operatoreingriff möglich ist.
5. In der für die Systeminitialisierung (IPL) benötigten Datei SYS1.NUCLEUS werden hin und wieder einzelne Spuren gelöscht, das System ist dann nicht wieder ladbar und die Datei muß mühselig aus Sicherheitskopien rekonstruiert werden.
6. Ein besonders hartnäckiger Fehler, der das Betriebssystem drei Wochen lang immer wieder zum Absturz bringt, ist ein Paging-E/A-Fehler (S028), der die unterschiedlichsten Komponenten beeinträchtigt (JES2, VSPC, Converter). Intensive Untersuchungen der Fehleraufzeichnungen (Dumps), die aber den Zustand

des Systems erst kurze Zeit nach dem Fehler wiedergeben und so die eigentliche Ursache nicht mehr genau zeigen, ergeben, daß die MEMOREX-Magnetplatteneinheiten nicht korrekt (d.h. ebenso wie die entsprechenden IBM-Geräte) arbeiten. Dieser Fehler wird durch Lieferung eines neuen Mikroprogramms für die Plattenkontrolleinheit behoben, das einzusetzen die Firma MEMOREX bis dahin versäumt hatte.

7. Im Rahmen der Fehleranalyse zu 6. wird auch ein zur Systemsoftware gehörendes Programm zur Aufzeichnung des Inhalts des realen Speichers und der virtuellen Adreßräume aller aktiven Programme nach einem Zusammenbruch (Post-Mortem-Dump oder Stand-Alone-Dump) benötigt. Dieses Programm ist zunächst ebenfalls nicht einsetzbar.

8. Bei längerer Laufzeit des Betriebssystems nimmt die Zahl der verfügbaren Speicherrahmen (4 KBytes große Hauptspeicherblöcke) ab, und der Prozeß des Seitenwechsels (Paging) monopolisiert mehr und mehr die Programmausführung.

9. Die vom Meßprogramm RMF (resource management facility) angezeigten Werte über die Belegung des Hauptspeichers sind fehlerhaft (nämlich 0), so daß eine Analyse des Systemverhaltens zunächst nicht möglich ist.

Nachdem diese Fehler behoben oder umgangen waren, konnte eine nähere Untersuchung des Systemverhaltens einsetzen und eine optimale Einrichtung des neuen Systems in Angriff genommen werden.

Impressum

Redaktion inforum

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

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

Rechenzentrum der Universität
Einsteinstr. 60
4400 Münster

Auflage dieser Ausgabe: 700

Redaktionsschluß der nächsten Ausgabe: 30.06.1982

Leserforum

Nach einem frustrierenden Tag im Rechenzentrum möchte ich mir mit Hilfe eines Leserbriefes Luft machen!

2. März 82

- 8.30 Ankunft im Rechenzentrum, Maschine steht, Terminalraum geschlossen
- ca. 9.00 - 10.00 Maschine läuft, aber im Schneckentempo, VSPC-Befehle werden nach ca. 30 sec. ausgeführt
- 10.00 - 10.30 Maschinenstopp, in der "Pause" Plots abholen. Die eigentlich ästhetische Darstellung wird entstellt und fast ins Abstrakte verzerrt durch quer über die Zeichnung laufende (im Programm nicht vorgesehen!) Striche.
- 10.30 - 11.30 Maschine läuft sehr zufriedenstellend, 4 sec.-Jobs werden sofort gerechnet.
- ca. 11.30 Maschinenstopp. Nach Wiederaufnahme läuft mein (IBM)-Terminal nicht. Nach längerer Diskussion mit dem Operator (der sein bestes versuchte!) wurde ein anderes Terminal frei. Dann jedoch war es leider der Zeitpunkt, wo der FORTRAN-Kurs "Pause" hat und Studenten üben (wofür ich auch volles Verständnis habe). Doch leider dauern dann 4 sec.-Jobs ca. 30 Minuten! Ich hatte (mein Fehler) einen Buchstaben vergessen, nochmals submitter, dann
- ca. 13.30 (nach der Mensa), zwar gerechnet, aber immer noch ein Fehler drin, wieder abgeschickt, nach 1 Std. immer noch nicht gerechnet. Beim Herabsteigen zum Erdgeschoß stoße ich mich an diesem Gerüst, das nur "Einbahnverkehr" zuläßt. (Etwas ungünstiger Zeitpunkt, da zu dieser Zeit im RZ so viele Menschen auf und ab gehen!) Vergleiche zur HB-Werbung bieten sich an, Gründe zum "In die Luft gehen" gab es heute genügend.

Fragen:

Warum legt man die Einführung eines neuen Betriebssystems in die "Kurszeit", wo die Maschine eh' überbelastet ist?

Wie will man die Plotausgabe verbessern, denn Quer- und Längsstriche in den verschiedenen Farben, die nicht vorgesehen sind, kommen leider öfter vor?

Warum verbaut man den "Lebensnerv" des RZ (Weg vom Kartenlocher zum Kartenleser) in der Zeit des größten Betriebes?

Wie kann es passieren, daß RUMSERV (und andere Prozeduren, so wie z.B. SPSS) nach Einführen des neuen Betriebssystems nicht läuft, werden diese Dinge etwa nicht genügend ausgetestet?

Da mir an diesem Tag wahrscheinlich noch ein Magnetband gerissen, Plattendateien gelöscht oder Lochkarten hingefallen wären, gab ich für heute auf und hoffe auf bessere Zeiten.

Vielleicht lasse ich mir demnächst im "PROFILE" eine Biokurve ausrechnen, ob ich an dem Tag rechnen soll oder nicht, vorausgesetzt die Maschine läuft wieder...

Martin Hannemann, Geologisches Institut

Daß die erheblichen Engpässe nach der Systemumstellung diese und ähnliche Reaktionen bei unseren Benutzern hervorriefen, ist nur zu verständlich. Wir können sie nur voll bejahen, gaben sie uns doch Rückendeckung bei den Reklamationen gegenüber den beteiligten Firmen. Die Stellungnahmen des Rechenzentrums zu der angesprochenen Problematik entnehmen Sie bitte den vorangegangenen Artikeln.

Was die Renovierungsarbeiten am Treppenaufgang im Rechenzentrum betrifft, so handelte es sich hierbei um sofort erforderlich gewordene Brandschutzmaßnahmen, auf deren zeitliche Terminierung wir keinen Einfluß hatten. Das Gerüst blieb unnötig lange stehen, da zwischendurch Unstimmigkeiten wegen der Qualität der handwerklichen Ausführung durch die beauftragte Firma aufgetreten waren.

Die Verunzierungen von Plotterzeichnungen sind konstruktionsbedingte Schwächen der beiden im Einsatz befindlichen Plotter, die bei ständiger Kontrolle nahezu ausgeschaltet werden können, was bei Abarbeitung der Plotterwarteschlange während der Nachtschichten allerdings nicht garantiert werden kann. Gezielte Überwachung von Plotausgabe besonders bei rechenintensiven Langläufen ist jedoch nach Abgabe eines Jobbegleitzettels und einer speziellen Anforderung an das PLOTSERV-Programm möglich. Nähere Auskünfte erteilt Herr Sturm (Tel. 2609).

H. Pudlatz

Personalia

Seit dem 1.4.82 ist Herr M. Lauterbach als wissenschaftliche Hilfskraft in der Abteilung Numerische Datenverarbeitung des Rechenzentrums tätig.

Mitteilungen über die Rechentätigkeit 1981

W. Rosse

Vor einigen Wochen hat das Rechenzentrum jedem, der bei den Anträgen auf Rechenerlaubnis für 1981 die Angaben zur Finanzierung unterzeichnet hat (Hochschullehrer/Leiter der Einrichtung), die zu den daraufhin vergebenen Verrechnungsnummern abschließend erstellten Übersichten über die Rechentätigkeit im Kalenderjahr 1981 zugesandt. Gleichzeitig ist an diesen Personenkreis der für die Finanzierung Verantwortlichen die Bitte gerichtet worden, diese Übersichten auch dem jeweiligen Benutzer zur Kenntnis zu geben. Da nämlich eine verhältnismäßig große Zahl von Benutzern (u.a. Studenten mit Diplom-/Studienabschluß-Arbeiten) auf ihren Anträgen nur die Privatanschrift angegeben hat, ist ein Versenden solcher Übersichten an jeden einzelnen Benutzer von Seiten des Rechenzentrums aus Kostengründen leider nicht möglich.

Diese Übersichten enthalten die für jede Verrechnungsnummer ermittelten Gesamtangaben über die erfolgte Rechnernutzung - u.a. Anzahl der Jobs, Job-CPU-Zeit, gedruckte Zeilen, I/O-Service (d.h. Zugriffe auf externe Speicher), VSPC-Sitzungen.

Obwohl die Rechenerlaubnis für die Mehrzahl der Benutzer nach Ziffer 1 der Benutzungsordnung des Rechenzentrums in der Fassung vom 14.9.1978 antragsgemäß in der Aufgabengruppe a (unentgeltliche Benutzung) erteilt worden ist, ist es auch in diesen Fällen interessant festzustellen, wie hoch die Inanspruchnahme der DV-Anlagen des RZ kostenmäßig zu bewerten ist. Da am 25.3.1981 die Gebührenordnung für das Rechenzentrum in Kraft getreten ist, werden für jeden Benutzer auch die Verrechnungseinheiten ausgewiesen. Um nun diese Größe "begreifbar" zu machen, sind für jede Verrechnungsnummer die sich aus der Verrechnungseinheit bei den verschiedenen Kostenstufen (Betriebskosten/Selbstkosten Land/Gebühren) ergebenden DM-Beträge zur Information mit angegeben worden - angesichts eines wachsenden Kostenbewußtseins keine unwichtige Größe bei der persönlichen Einschätzung eines Rechenvorhabens.

Die meisten Betriebskosten umfassen lediglich die laufenden Kosten für Sachmittel und Betriebspersonal. Die Selbstkosten Land beinhalten die Sachkosten, die Kosten des gesamten Personals und die Amortisation der Investitionskosten für Geräte und Gebäude der Rechenzentrums. Die Gebühren werden in Anlehnung an die Preise gewerblicher Recheninstitute festgelegt.

Beginnend mit 1982 beabsichtigt das Rechenzentrum, solche Übersichten über die Rechentätigkeit quartalsweise zu erstellen und an die für die Finanzierung Verantwortlichen zu versenden.

Lehrveranstaltungen im SS 1982

Eine Übersicht mit kurzen Inhaltsangaben über das Lehrangebot des Rechenzentrums im SS 1982 ist bereits in der Januar-Ausgabe von inform erschienen. Hier folgen in Kurzform noch einmal die wichtigsten Angaben:

Einführende Vorlesungen

- | | | |
|--------|--|---------------|
| 320032 | Programmieren in FORTRAN
di 13-15, Hörsaal M5 (Beginn: 27.4.1982) | Reichel |
| 320047 | Programmieren in FORTRAN
mo 13-15, Hörsaal M3 (Beginn: 3.5.1982) | Steinhausen |
| 320070 | Programmieren in PL/I
(numerische Anwendungen)
di 13.30-15, Hörsaal M6 (Beginn: 27.4.1982)
+ 1 Std. Übungen | Mertz |
| 320085 | Programmieren in PL/I
(nichtnumerische Anwendungen)
mi 15-17, Hörsaal M6 (Beginn: 28.4.1982) | Sturm |
| 320090 | Programmieren in PASCAL
di 15-17, Hörsaal M4 (Beginn: 27.4.1982) | Pudlatz |
| 320123 | APL - Interaktive Programmierung und
interaktive Programme
mi 11-13, fr 11-13, Seminarraum RZ
(Beginn: 28.4.1982) | Meyer-Rinsche |

Weiterführende Vorlesungen

- | | | |
|--------|--|----------|
| 320119 | PL/I für Fortgeschrittene
mi 9-11, Hörsaal M6 (Beginn: 28.4.1982) | Kaspar |
| 320138 | Mikroprozessoren
mi 13-15, Hörsaal M6 (Beginn: 28.4.1982) | Kisker |
| 320142 | Betriebssysteme
mo, do 13-15, Hörsaal M5 (Beginn: 29.4.1982) | Held |
| 320157 | Informatik - Ein Überblick
mi 15-17, Hörsaal M5 (Beginn: 28.4.1982) | Neukäter |
| 320161 | Höhere Programmiersprachen: ADA
di 15.30-17, Seminarraum Bibliothek
(Beginn: 4.5.1982) | Bosse |
| 192245 | Thematische Computer-Kartographie
mi 14-17, Seminarraum IZ (Beginn 28.4.1982) | Pudlatz |

Einführung einer neuen Version des JES 2

K.-B. Mertz

Ab Anfang Mai soll eine neue Version des Job Entry Subsystem JES 2 eingesetzt werden, die den Möglichkeiten des seit dem 1. März benutzten Betriebssystems MVS/SP Rel 1.3 besser angepaßt ist als die bisher eingesetzte Version. JES 2 ist der Teil des Betriebssystems, der sämtliche Lochkartenleser, Lochkartenstanzer und Drucker verwaltet, über die Jobs ein- bzw. ausgegeben werden, und der außerdem die Reihenfolge der Bearbeitung der Jobs kontrolliert.

Für den Benutzer wird dadurch keine Umstellung erforderlich, außer daß jetzt auch beim Jobtyp KURS das PASSWORD angegeben werden muß, sofern der Jobname nicht mit einer speziell für einen Programmierkurs vergebenen Benutzerkennung beginnt. Ein kleiner Überblick über weitere Änderungen, die den Benutzer interessieren könnten:

- Die Prüfung der JOB-Anweisung wurde um formale Prüfungen des Jobnamens und der Fachnummer erweitert. Für den Fall eines Fehlers werden Meldungen vorgesehen, die den Grund für die Zurückweisung andeuten.
- Der Bildschirm beim Lochkartenleser im RZ wird jetzt alle Meldungen über Fehler, die beim Einlesen und bei der ersten Prüfung der Job-Kontroll-Karten bemerkt werden, anzeigen, soweit über diesen Kartenleser eingegebene Jobs betroffen sind. Dadurch ist eine schnellere Fehlerkorrektur möglich.
- Für die System-Meldungen eines Jobs (JES2-Job-Log, JCL-Liste, Allocation-Meldungen) kann man wie für andere Dateien spezielle FORMS usw. festlegen, indem man eine /*OUTPUT-Anweisung mit dem Code SYS und den gewünschten Parametern dem Job einfügt.
- Druck-Dateien, die zunächst gehalten waren (z.B. für VSPC) und dann freigegeben wurden (z.B. durch das VSPC-Kommando "route"), werden wie "normale" Druck-Dateien (SYSOUT=A oder MSGCLASS=A) nach Möglichkeit zu Datei-Gruppen zusammengefaßt. Dadurch werden zusätzliche Trennseiten eingespart.

Weitere Änderungen betreffen die Bedienung der Remote-Stationen. Sie werden hier nur kurz angedeutet, sind aber in einer neuen Version des Heftes "JES2-Kommandos für Remote-Stationen" berücksichtigt, die möglichst noch vor der Umstellung an die Institute mit eigenen Remote-Stationen versandt werden soll.

- Beim Kommando \$DF muß der Operand N nicht mehr als letzter angegeben werden. Wenn er angegeben wurde, erscheint eine Liste der Jobs, für die Output druckbereit vorliegt mit Angabe der Anzahl der Druckzeilen in der entsprechenden Datei-Gruppe.
- Neue Operanden bei den Kommandos \$L und \$C sowie ein neues Kommando \$TC erlauben es, einzelne Ausgabe-Dateien, einzelne Dateigruppen oder die gesamte druckbereite Ausgabe aufzulisten oder zu löschen bzw. deren Charakteristik (CLASS, DEST, FORMS usw.) nachträglich zu verändern.

Es mag den Leser verwundern, daß die jetzt angekündigte Umstellung des JES2 nicht schon vor zwei Monaten zugleich mit der Umstellung des Betriebssystems vorgenommen wurde. Diese Verzögerung hat ihren Grund in den zahlreichen Modifikationen, die erst in die mit dem neuen Betriebssystem gelieferte Version des JES2 eingebaut und getestet werden mußten.

Neben der Anpassung an die örtlichen Gegebenheiten wie z.B. die Bezeichnung der Ausgabefächer und die Möglichkeit des NOTIFY für VSPC-Benutzer gibt es im wesentlichen zwei Gründe für Modifikationen in der Größenordnung von ca. 7000 Lochkarten (davon ca. 4000 selbstgeschriebene und ca. 2000 aus dem Original kopierte Assembler-Instruktionen).

Zum einen konnte die Form der JOB-Anweisung vereinfacht und zugleich die Zugangskontrolle verbessert werden, indem zahlreiche Informationen über die Anforderungen an das System in einem einzigen Parameter, dem Jobtyp (TEST, PROD, WATF usw.), zusammengefaßt werden und weitere Angaben aus der vorgeschriebenen Form des Jobnamens entnommen werden. Das erscheint uns sehr wichtig bei einem so großen und heterogenen Benutzerkreis wie dem eines Universitätsrechenzentrums, das zahlreiche Anfänger in der EDV ausbilden muß.

Zum anderen wurden Verbesserungen für die Maschinenbedienung eingeführt, wie z.B. die Angabe der erforderlichen Magnetbandeinheiten, die Trennseite bzw. -karte bei der Druck- und Stanzausgabe sowie Erweiterungen und Beschränkungen einzelner Kommandos.

Zwar waren die meisten dieser Modifikationen bereits in der bisher benutzten Version des JES2 implementiert, doch wurden wir zu einer generellen Neu-Programmierung gezwungen durch eine wesentliche Änderung des JES2-Konzepts.

Dokumentation tut not.

W.A. Slaby

Sicherlich ist es Ihnen schon einmal ähnlich ergangen: Nach vielen Mühen und Geburtswehen (die zumeist durch Änderungen der Problemstellung hervorgerufen wurden) haben Sie Ihr Programmsystem zur Problemlösung erstellt und (mit mehrwöchiger Verspätung) das damit zusammenhängende DV-Projekt erfolgreich hinter sich gebracht; das einzige, was noch ansteht und von Ihnen verlangt wird, ist eine Dokumentation. Nun werden Sie sagen (insbesondere weil bereits das nächste DV-Projekt dringend auf eine Erledigung harrt), daß ein gut weil strukturiert geschriebenes Programm sich selbst dokumentiert, daß das Programm sowieso nicht geändert oder nur für das soeben abgeschlossene Projekt eingesetzt werden soll, daß niemand außer Ihnen selbst dieses Programm anwenden wird, daß Ihnen im Moment die Zeit dazu fehlt (sie wird Ihnen mit Sicherheit später ebenfalls fehlen), etc.

Wenn der Gedanke an eine Dokumentation erst beim Abschluß des DV-Projekts mit Nachdruck verfolgt wird, ist es zwar nicht zu spät, leichter wird die Angelegenheit jedoch dadurch auch nicht. Deshalb möchte ich im folgenden einige Anregungen geben, wie man sich das Erstellen einer Dokumentation erleichtern kann.

1. Dokumentation der Problemstellung

Bereits ganz zu Anfang, wenn Sie mit einem Problem konfrontiert werden, für das Sie eine DV-Lösung erarbeiten sollen, empfiehlt es sich, die Problemstellung schriftlich festzuhalten und gegebenenfalls vom Auftraggeber gegenzeichnen zu lassen. Auf diese Weise läßt sich die Gefahr verringern, daß erst zum Abschluß der Programmentwicklung festgestellt wird, daß das Programm zwar fehlerfrei arbeitet, aber mitnichten das leistet, was sich der Auftraggeber ursprünglich davon versprochen hat. Insbesondere bei Problemstellungen, die auf einem von den Denkweisen der Datenverarbeitung unberührten Boden gewachsen sind, ist die Gefahr des gegenseitigen Mißverstehens groß.

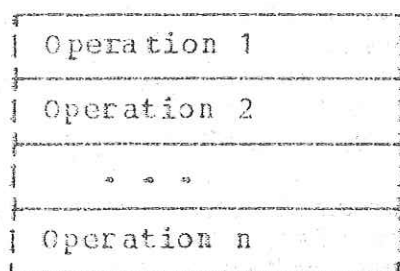
In einem (häufig dornenreichen) Iterationsprozeß wird deshalb die Problemstellung zumeist noch während der Programmentwicklung mehrfach modifiziert und präzisiert. Damit bei diesem Verfahren die Übersicht über den jeweils aktuellen Stand der Problemstellung nicht verlorenggeht, ist eine exakte Protokollierung jeder Modifikation mit Angabe des Änderungsdatums unumgänglich.

2. Dokumentationsfördernde Programmentwicklung

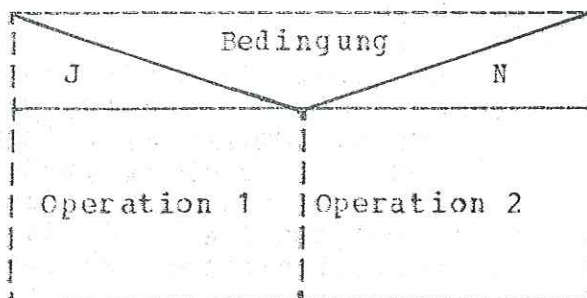
Mit den Methoden der Strukturierten Programmierung und des Programmierens durch schrittweise Verfeinerung haben sich gerade in den letzten Jahren Verfahrensweisen für die Programmentwicklung allgemein durchgesetzt, die für eine mit der Programmentwicklung simultan verlaufende Dokumentation sehr förderlich sind. Grundprinzip der Strukturierten Programmierung ist es dabei, Programme (im wesentlichen) aus drei normierten Grundstrukturen, der Sequenzstruktur, der Auswahlstruktur und der

Niederholungsstruktur aufzubauen. Losgelöst von einer speziellen Realisierung in irgendeiner Programmiersprache sind für diese Grundstrukturen und ihre Kombinationen zu einer Programmablaufstruktur mit den sogenannten Struktogrammen oder Nassi-Shneiderman-Diagrammen graphische Darstellungen entwickelt worden, die als wesentlicher Bestandteil einer Programmdokumentation angesehen werden können. Die Grundstrukturen haben dabei die folgende Gestalt:

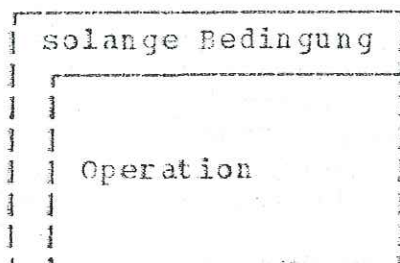
Sequenzstruktur



Auswahlstruktur



Wiederholungsstruktur



Struktogramme zur Darstellung eines Programmablaufs können nun induktiv dadurch aufgebaut werden, daß ausgehend von einer Grundstruktur jede einzelne Operation dieser Grundstruktur durch das Struktogramm ersetzt wird, welches diese Operation darstellt. Mit dieser Vorgehensweise haben wir bereits das zweite Standbein guter Programmiermethodik skizziert, die Methode des schrittweisen Verfeinerns, bei der eine zunächst nur global beschriebene, komplexe Operation als Sequenz-, Auswahl- oder

Wiederholungsstruktur erkannt und als solche beschrieben wird, deren Teiloperationen im nächsten Schritt auf die gleiche Weise weiter detailliert werden. Dieses Verfahren wird solange durchgeführt, bis im Struktogramm nur noch Operationen vorkommen, die unmittelbar in eine Anweisung der vorgesehenen Programmiersprache umsetzbar sind.

Sicherlich gehören zu einer Programmdokumentation noch weitere Beschreibungen wie die Darstellung der verwendeten Datenstrukturen und Dateien; mit diesen vier Bereichen

- detaillierte Problembeschreibung
- Struktogramm der Problemlösung
- Darstellung der verwendeten Datenstrukturen
- Datenflußplan

sind auch die wesentlichen Bausteine einer Dokumentation aufgezählt.

INTERLISP/370 Advanced System unter CMS

B. Neukäter

Für die Anwender der Programmiersprache LISP stellt das Rechenzentrum ein interaktives LISP-System unter CMS zur Verfügung. Es handelt sich hierbei um das von Bolt, Beranek and Newman, dem Xerox Palo Alto Research Center und dem Datenzentrum der Universität Uppsala entwickelte und vom Weizmann Institut, Israel, erweiterte INTERLISP/370 Advanced System. Es umfaßt Interpretierer, Übersetzer, Editor, E/A-System, Unterbrechungspaket, file system (prettydef), advisor, programmer's assistant, DWIM facility (do what I mean, automatische Fehlerkorrektur) und - für diejenigen, die keine besondere Vorliebe für die runden Klammern entwickelt haben - CLISP (Conversational LISP).

Da für das CMS nur in sehr beschränktem Umfang Bildschirme verfügbar sind, kann das System nur nach vorheriger Anmeldung benutzt werden. Interessenten mögen sich bitte an mich wenden (Tel. 2465).

NAG-Bibliothek

F.-P. Spellmann

Zwischenzeitlich wurden nach Vorschrift einer von der Numerical Algorithms Group herausgegebenen Fehler- und Korrekturliste die seit Einsatz der Version Mark 8 bekannt gewordenen Fehler korrigiert, die nach der bisher praktizierten Vorgehensweise erst in Mark 9 beseitigt worden wären. Zu warnen ist in diesem Zusammenhang vor einem nicht korrigierten fehlerhaften Verhalten der Routinen E04GDF, -GEF, -HEF und -HFF (sowie der zugehörigen einfach genauen Versionen), die fälschlicherweise IPAIL=0 zurückgeben, wenn die Matrix der zweiten partiellen Ableitungen der Zielfunktion im Endpunkt nicht positiv definit ist.

Die neue Version Mark 9 der NAG-Bibliothek ist in naher Zukunft verfügbar. Sie wird 17 neue Routinen enthalten, u.a. zur Lösung nichtlinearer Gleichungssysteme, zur Interpolation durch eine rationale Funktion und zur Analyse von Zeitreihen (in einem neuen Abschnitt G13 mit 8 neuen Routinen). Wie schon bei früheren Versionswechseln wird auch in diesem Fall eine Anzahl von Routinen entfallen, deren Namen bereits in der gegenwärtig noch gültigen NAG-Dokumentation publiziert sind. Soweit betroffene Routinen noch benutzt werden, sollte deshalb umgehend zur Verwendung der schon in der zur Zeit eingesetzten Version Mark 8 vorhandenen Ersatzroutinen übergegangen werden, die i.a. neben Verbesserungen auch alle Möglichkeiten der entsprechenden "alten" Routinen bieten.

Die folgende Liste beinhaltet die zur Aussonderung vorgesehenen sowie die zum Ersatz vorhandenen Routinen und zeigt Unterschiede zwischen "alten" und "neuen" Routinen auf. Durch die angegebenen Namen sind dabei jeweils auch die zugehörigen einfach genauen Programmversionen angesprochen (also z.B. mit C05AAF auch C05AAE):

alt	neu	Bemerkungen
---	---	-----
C05AAF, C05ABF, C05ACF	C05ADF	Verbesserter Algorithmus, zuverlässigere und robustere Routine (andere neue C05-Routinen wie C05AGF, -AJF, -AKF oder -AZF bieten zusätzlich neue Möglichkeiten)
C06AAF, C06ABF	C06ECF, C06EAF	Die neuen Routinen sind nicht auf Datenreihen beschränkt, deren Länge einer Potenz von 2 entspricht, und sind i.a. effizienter.
D01ACF, D01AGF	D01BDF, D01AJF	Verbesserte Algorithmen
D02ADF	D02HAF	Verbesserter Algorithmus und neu entworfene Software
D02AFF	D02TGF	Verbesserte Software, im Zusammenhang mit den "easy-to-use" Routinen D02JAF und D02JBF entworfen, die eine einzelne Gleichung bzw. ein lineares System bearbeiten

E01ADF	E01BAF +E02BBF	E01BAF verwendet eine zufriedenstellendere Form kubischer Splines und ist (in Verbindung mit E02BBF) effizienter, wenn die Daten an zahlreichen Punkten zu interpolieren sind.
F01BHF	F02WAF, F02WCF	Effizienterer Algorithmus und flexiblere Routine; Rechenzeit- und Speicherplatzbedarf sollte bei vielen Anwendungen geringer sein; der Fall $m < n$ wird durch die neue Routine F02WBF abgedeckt.
F01BJF	F01BWF	Bessere Leistung auf Maschinen mit virtuellem Speicher
F01BKF	F02WDF	Die neue Routine ermittelt in Fällen, in denen die QR-Zerlegung nicht zur Feststellung maximalen Rangs der Matrix führte, die singulären Werte, um eine zuverlässigere Bestimmung des Rangs zu erreichen.
F01BMF, F03ALF	F01LBF	Bessere Leistung auf Maschinen mit virtuellem Speicher
F02BMF	F01BWF +F02BFF	Effizientere Routinen
F04AUF	F04JGF	Routinen, die in Zusammenhang mit F01BKF und F02WDF zum Einsatz kommen können (s. o.)
F04AVF	F04LDF	Bessere Leistung auf Maschinen mit virtuellem Speicher
G01ACF	G04AEF	Die neue Routine bietet mehr Möglichkeiten und ist auf Konsistenz mit neuen G04-Routinen ausgelegt
H01AEF	H01BAF, H01ADF	H01AEF ist unter Umständen nicht zuverlässig. Die neue Routine H01BAF ist robust und stabil; H01ADF kann jedoch verglichen mit H01BAF weniger zeitaufwendig sein.

Die Routine D02AGF wird entgegen den Ankündigungen in der NAG-Literatur auch weiterhin zur Verfügung stehen, da nicht alle ihrer Möglichkeiten durch die Ersatzroutine D02HBF abgedeckt sind. Dennoch sollte - soweit möglich - D02HBF statt D02AGF eingesetzt werden.

SSP/PL1-Bibliothek

F.-P. Spellmann

Die in Form von Quellentexten angebotenen und dem Standard des PL/I-F-Compilers entsprechenden Unterprogramme der SSP/PL1-Bibliothek wurden durch geringfügige Modifikationen, die auf die eigentliche Programmfunktion keine Auswirkung haben sollten, an die Konventionen des PL/I-Optimizing Compilers angepaßt.

Darüber hinaus wurde der Zugriff zu den doppelt genauen Versionen der SSP/PL1-Programme, für die neben einer einfach genauen auch eine doppelt genaue Fassung vorliegt, neu geregelt.

Diese beiden Versionen eines Programms unterscheiden sich jeweils nur in einigen wenigen Programmzeilen. Deshalb erschien es sinnvoller, die Ableitung der Versionen aus jeweils nur einem Quellentext durch Aktivierung bzw. Deaktivierung der entsprechenden Programmzeilen mit Hilfe einer Compiler-Option (MARGINS) zu ermöglichen, als (wie bisher) zwei eigenständige Versionen zu speichern. Durch Einrücken des Programmtextes und geeignete Ergänzung von Kommentarbegrenzungszeichen wurden zu diesem Zweck die Texte der einfach genauen Programmversionen so modifiziert, daß - gesteuert durch den MARGINS-Parameter - drei verschiedene Programmfassungen erzeugt werden können, falls das Programm auch schon bisher in doppelter Genauigkeit existierte:

1. MARGINS(2,72,0)
einfach genaue Version (existiert bei allen Programmen);
Voreinstellung des Parameters MARGINS
2. MARGINS(5,72,0)
doppelt genaue Version; Programmname wie bei der einfach genauen Version (z.B. DET3)
3. MARGINS(4,72,0)
doppelt genaue Version; Programmname ergibt sich durch Vorstellen von "D" vor den Namen der einfach genauen Version (z.B. DDET3)

Die Einfügung der Kommentarzeichen geschieht dabei etwa in folgender Weise:

2 4	68 72	(Position)
/* */DET3:	/*S*/	aktiv bei MARGINS(2,72,0)
/*DDET3:	/*D*/	aktiv bei MARGINS(4,72,0) oder MARGINS(5,72,0)

Die dritte beschriebene Möglichkeit ist u.a. in den Fällen von Bedeutung, wo ein benutztes SSP/PL1-Programm in doppelter Genauigkeit ein weiteres SSP/PL1-Programm in doppelter Genauigkeit aufruft, da dieser Aufruf immer unter Verwendung des Programmnamens der einfach genauen Version geschieht (z.B. Aufruf von MINV in CANC oder DCANC).

Bei Programmen, die nur in einfacher Genauigkeit existieren, führen alle drei Parametersetzungen zum gleichen Programmtext.

Beim Einsatz von SSP/PL1-Programmen werden diese wie bisher durch %INCLUDE in das Benutzerprogramm integriert, wobei die Namen der einzuschließenden Routinen jedoch wie in der zugehörigen Dokumentation (u.a. einzusehen in der Programmierberatung oder in der Bibliothek des Rechenzentrums; Kurzbeschreibungen im Programmkatalog des RZ) ohne vorgestelltes "D" anzugeben sind. Der LIBRARY-Parameter der verwendeten katalogisierten Prozedur ist zu setzen und gegebenenfalls kann der MARGINS-Parameter spezifiziert werden, z.B.:

```
// EXEC PLOC,LIBRARY='CME1.PL1.SSPSRC',  
//      PARM='MARGINS(4,72,1)'
```

Der MARGINS-Parameter kann aber auch auf einer *PROCESS-Karte festgelegt werden, wobei die betroffenen Programme dann jedoch als externe Unterprogramme behandelt werden. Bei Gebrauch des PL/C-Compilers ist statt MARGINS(x,y,z) die Angabe SORMGIN=(x,y,z) auf der //PL/C-Anweisung zu codieren.

Die Ersetzung der alten SSP/PL1-Bibliothek durch die neue Fassung mit den beschriebenen Änderungen ist für den 1.6.82 vorgesehen.

Zufallszahlen/Simulation

Die Herren Prof. Dr. N. Schmitz (Institut für Mathematische Statistik, Tel. 3771) und Dr. H. Exner (HRZ, Tel. 2686) bereiten eine Bestandsaufnahme/Dokumentation und einen Vergleich der am HRZ verfügbaren bzw. benutzten (Pseudo-)Zufallszahlengeneratoren vor. Benutzer, die nicht auf die Standardpakete NAG, SPSS, SSP und IMSL zurückgreifen, werden deshalb gebeten, zur Information (bzw. auch Beratung) mit einem der Herren in Kontakt zu treten.

Schriftenreihe des Rechenzentrums

W. A. Slaby

Seit der letzten Übersicht über Neuerscheinungen im inforum, Jahrgang 5, Nr. 1 sind in der Schriftenreihe des Rechenzentrums folgende Beiträge veröffentlicht worden bzw. befinden sich im Druck und können (gegebenenfalls im Kürze) im Sekretariat bei Frau Luth montags und mittwochs von 14 bis 15 Uhr sowie dienstags und donnerstags von 10.30 bis 11.30 käuflich erworben werden:

- Nr. 48 - SPRG - Signal PROCESSing System Version 2.1
von St. Hein, M. Hoke, K. Langer, M. Lautsch, D. Steinhäusen; Januar 1982; 99 Seiten; 10,- DM
- Nr. 51 - Eine Spline-Methode zur numerischen Lösung von Systemen retardierter Differentialgleichungen mit zustandsabhängigen Verzögerungen und Anwendungen
von H.-J. Unger; Dezember 1980; 77 Seiten; 9,- DM
- Nr. 52 - Computer-Kartographie
Möglichkeiten des EDV-Einsatzes bei der Erzeugung topographischer und thematischer Karten
von H. Pudlatz; April 1982; 72 Seiten; 10,- DM
- Nr. 53 - Konvergenzverhalten bei der numerischen Lösung von Dirichlet-Problemen in Gebieten mit Ecken mittels eines Doppelschichtpotentialansatzes
von S. Stiller-Sieglitz; Juni 1981; 68 Seiten; 8,- DM
- Nr. 54 - Unscharfe Clusteranalyse quantitativer Daten
Theoretische Begründung, Implementierung und Beurteilung einiger Verfahren
von St. Hein; Januar 1982; 156 Seiten; 12,- DM
- Nr. 55 - Automatische Übersetzung von Texten des Deutschen in gesprochene Sprache
von F.-P. Spellmann; Dezember 1981; 109 Seiten; 10,- DM
- Nr. 56 - A Unilateral Boundary Value Problem for Unsteady Waterflow in Porous Media
by U. Hornung; Januar 1982; 43 Seiten; 7,- DM
- Nr. 57 - Ein allgemeiner Parser der Güte (n^3, n^2)
von L. Lasthaus-Freund; Januar 1982; 126 Seiten;
10,- DM

Ein/Ausgabe in VS FORTRAN

H. Pudlatz

In der inforum-Ausgabe Jg. 5, Nr. 4 wurden die auffälligsten Neuerungen von VS FORTRAN gegenüber FORTRAN IV beschrieben. Inzwischen wurde die Compiler-Version 1.1.1 implementiert, die eine Reihe von Fehlern der "Nullversion" nicht mehr enthält, so daß die Hoffnung besteht, daß nach Beseitigung weiterer Fehler und Ungereimtheiten (z.B. bei der Benutzung von CHARACTER-Größen als Parameter) VS FORTRAN der Standard am Rechenzentrum wird. Die älteren G1- und H-Extended-Compiler wurden deshalb zum 31.5.82 gekündigt.

VS FORTRAN ist weitgehend aufwärtskompatibel zu FORTRAN IV, so daß die meisten Benutzerprogramme ohne Änderung auch mit FORTV-Prozeduren für FORTRAN 77 weiterhin laufen. Was die hier näher zu besprechende Ein/Ausgabemöglichkeiten angeht, so entfallen die Statements DEFINE FILE, FIND und PUNCH, während die Statements READ und WRITE bei Direktzugriff zu Plattendateien anders zu schreiben sind. Darüber hinaus werden Literal-Konstanten in Formaten bei der Eingabe nicht mehr verändert.

Im oben genannten Artikel wurde bereits erwähnt, daß nur einmal benötigte Formate auch als Literal-Konstante im READ/WRITE-Statement angegeben werden können:

```
WRITE(6,'(E12.5,2I4.2,E15.5E3,Q20.12)') A,I,J,D,E
```

Innerhalb der begrenzenden Apostrophs sind öffnende und schließende Klammern - wie bisher vom einzulesenden "variablen" Format bekannt - mit anzugeben. Am Beispiel fällt neben dem bereits früher genannten Q-Formatcode für erweiterte REAL-Größen zweierlei auf:

- Neben dem bisher gebräuchlichen aIw ist der Code aIw.d erlaubt, wobei a der Wiederholungsfaktor und w die Feldweite zur Übertragung von INTEGER-Zahlen ist. In der Ausgabe bedeutet "d", daß genau d Ziffern (also u.U. mit führenden Nullen) auszugeben sind. Die Angabe "d" wird bei der Eingabe ignoriert und bei der Ausgabe dann, wenn die auszugebende Zahl mehr als d Stellen hat und ≠ 0 ist. Mit Iw.0 erscheint eine Null in der Form von w Blanks.
- aEw.dEe bedeutet, daß für die Übertragung des Exponenten von REAL-Zahlen e Stellen vorgesehen sind. Der E-Format-Code in der vorgestellten erweiterten Form kann auch zur Ein/Ausgabe von REAL*8- und REAL*16- (bzw. COMPLEX*16- und COMPLEX*32-) Größen benutzt werden. Die Angabe der Exponentenlänge ist nämlich beim D- und Q-Format-Code nicht vorgesehen, er ist aber in Verbindung mit dem G-Format möglich (aGw.dEe).

Bei der Schachtelung von Format-Gruppen ist die maximale Klammerungstiefe 2 aufgehoben worden. Schachtelungen sind daher unbegrenzt möglich. Für die Wiederholung von Formatgruppen bei nicht erschöpfter Datenliste gilt die bisherige Regel. Als weiterer Formatcode-Separator (neben "," und "/") ist ":" einge-

führt worden mit der Wirkung, daß die Formatsteuerung beim Doppelpunkt abbricht, wenn die Datenliste erschöpft ist. Dies hat zur Konsequenz, daß z.B. ein nachfolgendes Literal (das oft zur Beschreibung des nächsten Ausgabedatums dient) nicht mehr gedruckt wird.

Neben dem existierenden Tabulator-Steuerformat TC gibt es jetzt auch die Codes TLC und TRC, die besagen, daß die nächste Datenübertragung c Zeichen links bzw. rechts von der aktuellen Position erfolgen soll. Ferner existieren zwei Gruppen von Steuerformatcodes zur Ausgabe von +-Vorzeichen und zur Behandlung von Blanks in der Eingabe:

- Der Steuerformatcode SP sorgt dafür, daß in allen nachfolgenden Zahlenausgaben mit den Codes D,E,F,G,I und Q ein +-Zeichen vor positiven Werten erscheint. Ein Format SS macht diese Setzung für alle nachfolgenden Zahlenausgaben wieder rückgängig.
- Die Steuerformatcodes BZ und BN steuern die Interpretation von "nicht führenden" (d.h. eingestreuten oder anhängenden) Blanks aller nachfolgenden numerischen Eingabefelder. BZ interpretiert solche Blanks als Nullen, BN ignoriert diese, d.h. linksbündige Eingabe formatierter Eingabefelder produziert dann nicht mehr unbeabsichtigte Zehnerpotenzen. Die Blankinterpretation kann im OPEN-Statement für ein Eingabe-File voreingestellt werden (s.u.). Geschicht dies nicht, so ist die Wirkung von BZ voreingestellt, d.h. implizit werden Blanks als Nullen interpretiert.

VS FORTRAN kennt drei Arten von Files: sequentiell, direkt und intern. Zur Charakterisierung der "herkömmlichen" sequentiellen Files ist ein OPEN-Statement vor dem ersten Zugriff auf ein solches File erlaubt, aber nicht erforderlich. Files, auf die direkt (d.h. wahlfrei mit Satznummern) zugegriffen werden soll, erfordern zwingend die vorherige Ausführung eines OPEN-Statements (früher DEFINE FILE), während für sogenannte interne Files (=Datenbereiche im Hauptspeicher) die Angabe eines OPEN-Statements verboten ist. Bevor näher auf das OPEN-Statement eingegangen wird, einige grundsätzliche Bemerkungen:

FORTRAN-Konstrukte, die die Behandlung von Files betreffen, haben folgende allgemeine Gestalt: Schlüsselwort (z.B. OPEN, BACKSPACE, READ), gefolgt von einer eingeklammerten Liste von Positions- und/oder Keyword-Parametern (z.B. 6,100,END=200 im READ), ggfs. gefolgt von einer Liste der zu übertragenden Daten. Grundsätzlich können alle Parameter in VS FORTRAN als Keyword-Parameter geschrieben werden, was den Vorteil hat, daß es dann auf ihre Reihenfolge nicht mehr ankommt. Um den Übergang von FORTRAN IV-Programmen ohne Umkodierung zu ermöglichen, ist für den Einheiten- und Format-Parameter auch die Schreibung als Positionsparameter erlaubt, dann müssen diese aber an erster bzw. zweiter Stelle stehen:

```
FORTRAN IV: READ(5,100,END=200) X
VS FORTRAN: READ(UNIT=5,FMT=100,END=200) X , aber auch
              READ(FMT=100,END=200,UNIT=5) X , oder
              READ(5,END=200,FMT=100) X
```

In allen File-bezogenen Statements (außer denen zum asynchronen Datentransfer, s.u.) sind zwei weitere Keyword-Parameter möglich, die die Behandlung von Fehlern bzw. Ausnahmen kontrollieren. Es sind dies:

```
IOSTAT=ios
ERR=stn
```

Wenn weder IOSTAT noch ERR angegeben sind, bricht das Programm bei einem E/A-Fehler nach diesem Statement ab. Ist ERR=stn spezifiziert, so verzweigt das Programm zur Statementnummer stn. Ist END=stn angegeben, so wird beim Eintreffen der End-of-File-Bedingung bei der Ausführung dieses Statements nach stn verzweigt. Ist weder ERR noch END, aber IOSTAT angegeben, so wird beim Auftreten eines Fehlers (einer Ausnahme) die Programmausführung fortgesetzt. Die INTEGER-Variable ios enthält dann den folgenden I/O-Status:

```
ios=0 : kein Fehler
ios>0 : Übertragungsfehler aufgetreten
ios<0 : End-of-File aufgetreten
```

Als UNIT kann i.a. ein Ausdruck von Typ INTEGER*4 angegeben werden (Ausnahme: Ein/Ausgabe bei internen Files, s.u.), während der FMT-Parameter folgende Angaben enthalten kann:

- eine Statementnummer, die ein FORMAT-Statement bezeichnet,
- eine INTEGER-Variable, der über ein ASSIGN-Statement die Nummer eines FORMAT-Statements zugewiesen wurde,
- eine CHARACTER-Konstante, -Variable oder ein CHARACTER-Array-Element, das ein gültiges Format enthält (s.o.),
- ein (CHARACTER-) Array, bei dem ein gültiges Format sich über mehrere Arrayelemente erstrecken kann (Arrays vom Typ INTEGER, REAL etc. sind als IBM-Erweiterung zulässig)

Das OPEN-Statement dient der Verbindung eines (externen) Files mit einer Einheitennummer (UNIT). Es hat die Form:

```
OPEN ([UNIT=un][,ERR=stn][,IOSTAT=ios][,ACCESS=acc][,RECL=rcl]
      [,FILE=fil][,STATUS=sta][,BLANK=blk][,FORM=frm])
```

Dabei bedeuten:

- acc ein CHARACTER-Ausdruck mit dem Wert 'SEQUENTIAL' oder 'DIRECT'. Letzteres muß für Files, auf die wahlfrei zugegriffen werden soll, angegeben werden. SEQUENTIAL wird implizit angenommen.
- rcl ein INTEGER-Ausdruck, der die Satzlänge eines DIRECT-Files bestimmt. Diese Angabe wird bei SEQUENTIAL-Files ignoriert.

fil ein CHARACTER-Ausdruck, der den DD-Namen für ein File (1 - 8 Zeichen) enthält. Falls nicht angegeben, wird dieser aus dem UNIT-Parameter abgeleitet (FunF001).

sta ein CHARACTER-Ausdruck mit dem Wert 'NEW', 'OLD', 'SCRATCH' oder 'UNKNOWN'. Bei NEW bzw. OLD muß auch FILE=fil angegeben sein. Im ersten Fall darf das File noch nicht existieren, im letzteren muß es.

blk ein CHARACTER-Ausdruck mit dem Wert 'NULL' oder 'ZERO'. Er gibt die Voreinstellung bzgl. der Interpretation von Blanks an, wenn von diesem File Zahlenwerte mit Format gelesen werden sollen. NULL ignoriert alle nichtführenden Blanks, ZERO behandelt diese wie Nullen (vgl. BN, BZ oben).

frm ein CHARACTER-Ausdruck mit dem Wert 'FORMATTED' bzw. 'UNFORMATTED', der besagt, ob Ein/Ausgabe von/zu einem (externen) File formatiert oder unformatiert geschehen soll. Falls dieser Parameter fehlt, wird für sequentiellen Zugriff formatierte und für direkten Zugriff unformatierte Übertragung angenommen.

Nach dieser ausführlichen Darstellung des OPEN-Statements kann eine Übersicht über alle Ein/Ausgabe-Statements in VS FORTRAN in der unten angegebenen Tabelle gegeben werden. Bei den Statements READ und WRITE muß dabei zwischen

- formatiert (FMT=fmt ist angegeben),
- unformatiert (FMT fehlt, Ein/Ausgabe von/auf Band/Platte in der internen Darstellung)

und

- listengesteuert ([FMT=]*, nur für sequentielle Ein/Ausgabe)

unterschieden werden. Bei der listengesteuerten Ausgabe werden die Daten in einem festen Format ausgegeben. Bei der listengesteuerten Eingabe können die Daten im freien Format, getrennt durch Komma, Blank oder Schrägstrich (/) vorliegen. Dabei ist zu beachten, daß ein Schrägstrich die Eingabeaktion beendet, unabhängig davon, ob alle Elemente der Datenliste zugewiesen werden konnten.

Bei direktem Zugriff muß mit REC=rec die relative Satznummer als INTEGER-Ausdruck mit einem Wert >0 angegeben werden.

"Interne Files" sind als CHARACTER-Variable, -Arrays, -Array-Elemente oder Teilstrings bezeichnete Bereiche im Hauptspeicher, deren Werte z.B. durch ein READ-Statement von einem externen File erhalten wurden. Eingabe von einem internen File ist z.B. dann sinnvoll, wenn in Abhängigkeit von einem bestimmten Eingabeelement verschiedene weitere Daten (mit verschiedenen Formaten) vom selben Eingabesatz gelesen werden sollen. In diesem Fall ist ein Eingabesatz zunächst auf eine CHARACTER-Variable zu lesen, von der danach "intern" gelesen werden kann (vgl. früheres REREAD-Statement). Beispiel für Lesen vom internen File C:

```
CHARACTER*80 C
READ (5, '(A)') C
IF (C(:1).EQ.'A') READ (UNIT=C, FMT='(1X,2I5)') M,N
```

Statement	[UNIT=] un	[FMT=] fmt	IOSTAT =ios	ERR= stn	END= stn	REC= rec	ID= id	STATUS =sta
READ/WRITE sequentiell	+	x	x	x	bei READ	-	-	-
READ/WRITE direkt	+	x	x	x	-	+	-	-
READ/WRITE intern	+	x	x	x	bei READ	-	-	-
READ/WRITE sequentiell mit NAMEDLIST	+	-	x	x	bei READ	-	-	-
READ/WRITE asynchron mit WAIT	+	+	-	-	-	-	+	-
OPEN/CLOSE	+	-	x	x	-	-	-	x
BACKSPACE ENDFILE REWIND	+	-	x	x	-	-	-	-
INQUIRE	x	-	x	x	-	-	-	-

Mit '+' bezeichnete Einträge dieser Tabelle sind obligatorisch, mit 'x' bezeichnete wahlfrei bzw. im jeweiligen Zusammenhang erforderlich (z.B. formatierte Ein/Ausgabe) und ein '-' deutet ein Verbot dieses Parameters für dieses Statement an.

READ/WRITE mit NAMEDLIST und READ/WRITE asynchron mit WAIT sind die vom FORTRAN IV-H-Extended-Compiler her bekannten IBM-Erweiterungen, die auch in VS FORTRAN als solche anzusehen, also nicht im 77-Standard enthalten sind. Bei NAMEDLIST sind die ersten beiden Parameter ("UNIT=" darf nicht geschrieben werden) die Einheitennummer und der NAMEDLIST-Name. Das (verständliche) Fehlen von IOSTAT und ERR bei der asynchronen (d.h. mit der weiteren Programmausführung konkurrierenden) Datenübertragung wird durch zwei Parameter im zugehörigen (d.h. über den ID-Parameter mit einem eindeutigen INTEGER-Wert an das zugehörige READ/WRITE gekoppelten) WAIT-Statement ersetzt: COND gibt Auskunft über Erfolg oder Mißerfolg und NUN über die Anzahl der übertragenen Bytes der READ/WRITE-Operation.

Das CLOSE-Statement hängt ein externes File logisch von der mit ihm verbundenen Einheit ab, so daß diese mit einem anderen File über ein nachfolgendes OPEN verbunden werden kann. Für STATUS sind hier die CHARACTER-Werte 'KEEP' und 'DELETE' erlaubt. Fehlt STATUS, so wird für SCRATCH-Files DELETE und für

alle anderen KEEP angenommen.

Das BACKSPACE-Statement setzt ein zum Lesen eröffnetes sequentielles File (nicht bei listengesteuertem READ oder READ mit NAMELIST) um einen Satz zurück.

Ist die UNIT-Angabe der einzige Parameter für BACKSPACE, END-FILE und REWIND, so kann er (ohne vorgestelltes UNIT=) weiterhin ohne Klammern hinter das entsprechende Schlüsselwort gestellt werden, z.B. REWIND 1.

Das INQUIRE-Statement soll nicht näher besprochen werden, da hier annähernd 20 Parameter im einzelnen zu erläutern wären. Hier nur so viel: es gestattet die Abfrage einer Fülle von Informationen über die Existenz, den Zustand und den Zugriff zu einem externen File.

Abschließend sei bemerkt, daß auch auf VSAM-Files mit den genannten Statements zugegriffen werden kann, in diesem Fall haben BACKSPACE, ENDFILE, REWIND und IOSTAT eine modifizierte Bedeutung.

inforum-Quiz

E. Sturm

Die Lösung des letzten inforum-Quiz war "PL/I IS THE BIGGEST", nicht "PL/I IS THE GREATEST". Dies soll auch das heutige Quiz untermauern. Zunächst aber einige Bemerkungen zum letzten Programm. Wie manche vielleicht mit Erstaunen festgestellt haben, können in PL/I BUILTIN-Funktionen auch mit Array-Argumenten aufgerufen werden. Dies in völliger Analogie zu den Operatoren: so wie man von zwei Vektoren A und B die Summe A+B bilden kann, ist es auch möglich, z.B. SQRT(A) zu benutzen. Das Ergebnis ist in jedem Fall ein Vektor der gleichen Dimension wie A (bzw. B). Eine BUILTIN-Funktion kann also im Gegensatz zu den selbstgeschriebenen Funktionen als Funktionswert auch einen Array zurückgeben.

Auch die Zeichenketten verarbeitenden BUITLIN-Funktionen können mit Array-Argumenten aufgerufen werden, wie das letzte Quiz-Programm zeigen sollte. INDEX (D,'O') liefert einen Array der Positionen des jeweils ersten 'O' innerhalb der Elemente von D. TRANSLATE (A,B,C) übersetzt die Elemente von A, indem es als Vorlage die Elemente von B bzw. C mit demselben Array-Index benutzt. SUBSTR wiederum liefert einen Array der Teilketten des übersetzten Arrays A, deren Anfänge durch die INDEX-Funktion und deren Längen durch den Vektor L gegeben sind. Dieser Array wird auf den Vektor F zugewiesen, dessen VARYING-Elemente dann durch die Funktion STRING zu dem oben verkündeten Spruch verkettet werden. STRING, ADDR, ALLOCATION und alle "array handling built-in-functions" sind die einzigen Ausnahmen von der Regel, daß Array-Argumente auch zu Array-Funktionswerten führen.

Die vielen sprachlichen Möglichkeiten von PL/I bringen eine Vielzahl von Regeln mit sich. Nicht umsonst ist das PL/I-Handbuch so dick. An Hand des neuen Problems können Sie testen, ob Sie bei Variablennamen immer wissen, um welche Variablen es sich handelt. Das folgende Programm ist natürlich kein allzu sinnvolles, aber es zeigt doch, daß man in PL/I eine ganze Menge (aus Versehen) hinschreiben kann, ohne daß der Compiler sich beschwert. Damit Sie nicht zu früh glauben, die Lösung gefunden zu haben, sei hier verraten, daß der gesuchte Wert von B, der mit dem PUT-Statement ausgedruckt wird, nicht unter 20 liegt.

```
WHAT_IS_WHAT:
PROCEDURE OPTIONS (MAIN);
DCL 1 A,
    2 A,
    3 A BIN FIXED INIT (1),
    3 B BIN FIXED INIT (2);
DCL B BIN FIXED INIT (3);
CALL WHAT_IS_WHAT (B, B, (B));
PUT LIST (B);
WHAT_IS_WHAT:
PROCEDURE (X, Y, Z);
DCL (X, Y, Z) BIN FIXED;
DCL 1 A,
    2 B BIN FIXED INIT (4),
    2 A BIN FIXED INIT (5);
A = A + B;
Z = X + A.A;
X = Z + A.B;
Y = Y + A.A.A;
END WHAT_IS_WHAT;
END WHAT_IS_WHAT;
```

Programmquerschnitt Januar - April

H. Schlattmann

Die folgende Übersicht zeigt die Benutzung ausgewählter Programme und VSPC-Prozessoren und ihren Verbrauch an CPU-Zeit auf. Die Prozentangaben beziehen sich dabei auf die Gesamtzahl der Programm- oder Prozessoraufrufe bzw. die den beiden Tabellentteilen entsprechenden Anteile an der Gesamt-CPU-Zeit im angegebenen Zeitraum.

Programm-Benutzung

Programm	Aufrufe	in %	CPU-Zeit	in %
PL/1-OPTIMIZING-COMPILE	21084	9.18	22:52:52	2.79
PL/1-OPTIMIZING-LINKAGE	6492	2.82	2:41:40	0.32
PL/1-OPTIMIZING-EXECUTE	29703	12.94	140:18:12	17.13
PL/1-CHECKOUT-COMPILE	2366	1.03	1:15:16	0.15
PL/1-CHECKOUT-LINKAGE	35	0.01	0:00:24	0.00
PL/1-CHECKOUT-EXECUTE	2092	0.91	2:14:07	0.27
FORTRAN IV-G1-COMPILE	8408	3.66	7:23:58	0.90
FORTRAN IV-H EXTENDED-COMPILE	14760	6.43	13:34:05	1.65
FORTRAN IV-LINKAGE	5895	2.56	2:40:28	0.32
FORTRAN IV-EXECUTE	38940	16.97	440:32:16	53.80
FORTRAN 77-COMPILE	11811	5.14	4:47:28	0.58
FORTRAN 77-LINKAGE	49	0.02	0:00:52	0.00
FORTRAN 77-EXECUTE	2601	1.13	3:49:14	0.46
ALGOLW-COMPILE	22	0.00	0:00:12	0.00
ALGOLW-EXECUTE	22	0.00	0:00:16	0.00
PLC-MONITOR	1137	0.49	2:01:16	0.24
FORTRAN-MONITOR	1828	0.79	12:45:09	1.55
ALGOLW-MONITOR	89	0.03	0:01:34	0.00
PASCAL-COMPILE	1755	0.76	1:11:10	0.14
PASCAL-LINKAGE	25	0.01	0:00:07	0.00
PASCAL-EXECUTE	1720	0.74	3:02:12	0.37
SPSS	18876	8.22	26:49:23	3.27
SERVICE	10403	4.53	7:23:48	0.90
MPSX-COMPILE	511	0.22	0:04:59	0.01
MPSX-EXECUTE	510	0.22	0:41:20	0.08
GPSS	94	0.04	0:06:42	0.01
BASIC-COMPILE	13	0.00	0:00:05	0.00

Anzahl der Programm-Aufrufe gesamt : 229458
mit einer CPU-Zeit von : 818:42:16

VSPC-Prozessoren

PROZESSOR	Aufrufe	in %	CPU-Zeit	in %
APL	7481	4.12	3:29:15	46.92
BASIC	17804	9.82	1:58:23	26.55
CLIST	92114	50.83	0:08:44	1.96
EDIT	4053	2.23	0:04:21	0.97
LOGON	43857	24.20	0:55:47	12.51
PLOT	2565	1.41	0:35:34	7.97
RUN	12670	6.99	0:11:50	2.65

Anzahl der Prozessor-Aufrufe gesamt: 181211
mit einer CPU-Zeit von : 7:25:53