

inforum

INFORMATIONsforum des Rechenzentrums der Universität Münster

Jahrgang 10, Nr. 4

Oktober 1986

Inhalt

Editorial	2
<u>RUM-Aktuell</u>	
Gedenkkolloquium zu Ehren von Helmut Werner	3
Kein BASIC mehr auf dem Großrechner	3
Termine zur Benutzerinformation	3
Datenschutz und Personal-Computer	4
Notizen zum Thema Datenbanken	4
Umgebungen im CMS	4
Mathematische Formeln mit DCF	6
Personalia	8
RUM/GKS-Dokumentation online	8
Neues vom SAS	8
Undokumentierte OpCodes und Macros endlich beschrieben	9
Sprechzeiten	11
<u>RUM-Lehre</u>	
Lehrveranstaltungen im 2. Halbjahr 1986	12
Einführung in die Benutzung der Rechenanlage	13
<u>RUM-Graphik</u>	
Farbige Rasterbilder II	14
Plotausgabe auf dem Agfa P400	15
<u>RUM-Tutorial</u>	
Bildschirmsteuerung im Dialog mit ISPF	16
<u>Die Statistik-Seite</u>	
Jobverteilung im August/September 1986	21
Stichwörter <u>inforum</u> Jahrgang 10	22

Impressum

Redaktion *infor um*:

A. Achilles (Tel. 83-2607)
 W. Bosse (Tel. 83-2461)
 H. Pudlatz (Tel. 83-2472)
 E. Sturm (Tel. 83-2609)

Satz: P. Bigdon

E. Krause

Druck: H. Mecke

Universitätsrechenzentrum
 Einsteinstraße 60
 4400 Münster

Auflage dieser Ausgabe: 1000

Redaktionsschluß der nächsten
 Ausgabe: 30.12.1986

Editorial

von

A. Achilles

Nun ist es endlich so weit: 10 Jahre *infor um* liegen hinter uns – und die Hefte, ordentlich gebündelt, hoffentlich vor Ihnen. Diese 10 Jahre haben einige Veränderungen mit sich gebracht, vergleichen Sie allein die Veränderungen im Layout unserer Informationsschrift. Eine der Änderungen führt zur erneuten Veröffentlichung der Sprechzeiten: seit kurzem ist das Rechenzentrum über drei Gebäude verteilt, und Sie sowohl wie wir Mitarbeiter erfahren, daß auch heute, im Computer-Zeitalter, die praktische Seite der Kommunikation nur unvollständig gelöst ist.

Sie finden in dieser Ausgabe nicht nur eine (unkommentierte) Übersicht über das Lehrangebot und eine Aufstellung der Veranstaltungsreihe "Einführung in die Benutzung der Rechenanlage" (siehe RUM-Lehre), sondern auch weitere Termine, die Sie sich vormerken sollten: unter der Überschrift "Termine zur Benutzerinformation" wird auf das nächste Benutzertreffen hingewiesen und noch einmal an den geplanten Erfahrungsaustausch über Graphikanwendungen erinnert.

Graphiken können nicht nur erzeugt, sondern auch auf immer besseren Geräten ausgegeben werden. Als Jubiläumsbeigabe finden Sie einen Kalender aus dem Jahr 1829, Blatt für Blatt original Agfa P400. Haben Sie's gemerkt? Es handelt sich nicht um Makulatur, Sie können diesen Kalender auch im Jahre 1987 benutzen. Näheres zu den neuen Ausgabemöglichkeiten finden Sie in den Beiträgen "Plotausgabe auf dem Agfa P400" und "Farbige Rasterbilder II".

Im Beitrag "Umgebungen im CMS", einer Fortsetzung der Diskussion um Konventionen, wird beschrieben, wie Sie Ihre Umgebung mit einfachen Mitteln nach Ihren Bedürfnissen gestalten können. Wenn Sie weitergehende Wünsche haben, so geben Sie doch Ihrerseits Anregungen.

Datenschutz bei personenbezogenen Daten ist nicht nur ein Problem des Großrechners, wie im gleichnamigen Artikel zu lesen ist. Leider konnte aus gleichem Grunde auch ein Beitrag nicht gebracht werden, der gut in diese Ausgabe gepaßt hätte: ein Mitschnitt einer Sitzung der *infor um*-Redaktion.

Der Beitrag "Mathematische Formeln mit DCF" beschreibt lang erwartete und überfällige Erweiterungen von SCRIPT. Nun können auch mit SCRIPT-Mitteln Formeln gesetzt werden – in vielen Fällen mag das ein Ausweichen auf TEX erübrigen.

In dem Beitrag über die "OpCodes" wurde versucht, die Hintergründe der Schwierigkeiten, die Sie sowohl wie wir mit der Datenverarbeitung haben, zu analysieren. Daß es in diesem Stile nicht weitergeht, beweisen die wenigen Zeilen über die Abschaffung von BASIC. Die Lage ist ernst, aber hoffnungslos!?

Nach den Beiträgen über PANELIO beginnt hier – anstelle eines Handbuchs – eine Folge von Beiträgen über ISPF, einem Programm zur Bildschirmsteuerung. Der erste Beitrag führt in die Benutzung von Bildschirmmasken ein.

Und nun wünsche ich Ihnen nach den vergangenen 10 Jahren weitere 10 informative Jahre mit unserer Informationszeitschrift *infor um* nach dem Motto

*Am Rechner, am Schreibtisch,
 in den Bergen, am Strand,
 das neuste infor um
 ist immer zur Hand.*

RUM-Aktuell**Gedenkkolloquium
zu Ehren von Helmut Werner**

Am 21. und 22. 11. 1986 findet ein Kolloquium zu Ehren des im letzten Jahr verstorbenen ehemaligen Leiters des Universitätsrechenzentrums, Prof. Dr. Helmut Werner, statt. Eine Reihe namhafter Wissenschaftler der Universität Münster als auch anderen deutscher und ausländischer Hochschulen werden Vorträge aus ihren Fachgebieten halten. Soweit es sich hierbei um außermathematische Disziplinen handelt, wurden die Forschungen z.T. durch den allzu früh Verstorbenen angeregt oder waren Ergebnis interdisziplinärer Zusammenarbeit des Mathematikers Werner mit den betreffenden Wissenschaftlern.

Das Kolloquium beginnt am Freitag, dem 21. 11. 1986 um 10.30 Uhr im Hörsaal M4 des Fachbereichs Mathematik, Einsteinstr. 60, und wird eröffnet durch den Rektor der Universität, den Dekan des Fachbereichs Mathematik und den Leiter des Universitätsrechenzentrums und endet gegen 18 Uhr. Es wird fortgesetzt am Samstagvormittag um 9.15 Uhr mit einer Reihe von mathematischen Fachvorträgen.

Kein BASIC mehr auf dem Großrechner

von

B. Neukäter

Wegen der angespannten Haushaltsslage wird ab 1. November 1986 der IBM-BASIC-Compiler nicht mehr zur Verfügung stehen. Die betroffenen Benutzer werden gebeten, sich mit ihrem Problemlöser im Rechenzentrum in Verbindung zu setzen, um gemeinsam nach einem Ausweg zu suchen.

Termine zur Benutzerinformation

von

*W. Bosse***Erfahrungsaustausch**

Wie bereits in der letzten *inforum*-Ausgabe mitgeteilt, ist für

Freitag, den 17. Oktober 1986, 13.15 Uhr

im Hörsaal M4, Einsteinstraße 64, ein Erfahrungsaustausch über Graphikanwendungen geplant, bei dem problem- und sachbezogene Fragen zur graphischen Datenverarbeitung erörtert werden sollen. Interessierte Benutzer haben auch die Gelegenheit, ihre Erfahrungen mit anderen auszutauschen.

Bei Interesse besteht die Möglichkeit, sich regelmäßig über entsprechende Themen zu informieren.

Benutzertreffen

Das nächste Benutzertreffen ist für

Freitag, den 9. Januar 1987, 13.15 Uhr

geplant und wird im Hörsaal M4, Einsteinstraße 64, stattfinden.

Vorgesehene Tagesordnung:

1. Bericht des Rechenzentrums
2. Beantwortung von Fragen und Diskussion
3. Verschiedenes

Fragen sollten nach Möglichkeit bis Dienstag, den 6.1.1987, im Rechenzentrum eingereicht werden, damit wir fundiert dazu Stellung nehmen können.

Alle Benutzer sind herzlich zu diesen Veranstaltungen eingeladen.

Datenschutz und Personal-Computer

von

W. Held

Aus gegebenem Anlaß möchten wir darauf hinweisen, daß Benutzer, die personenbezogene Daten verarbeiten, grundsätzlich die entsprechenden Datenschutzgesetze zu beachten haben. Ob die Verarbeitung auf einem Großrechner oder einem Personal-Computer durchgeführt wird, macht keinen Unterschied. Bei der Benutzung eines Personal-Computers liegt in vielen Fällen allerdings eine noch größere Sorgfaltspflicht beim Benutzer selbst, da der organisatorische Rahmen eines Rechenzentrumsbetriebs fehlt.

Das Rechenzentrum ist gerne bereit, hinsichtlich bestimmter Arbeitsabläufe eine Beratung zu geben. Setzen Sie sich deshalb gegebenenfalls mit Herrn Dr. Kamp (Tel. 2474) in Verbindung, wenn ein Informationsbedürfnis besteht.

Notizen zum Thema Datenbanken

von

A. Achilles

Mit einiger Verspätung ist jetzt das bereits in der letzten *inforum*-Ausgabe angekündigte STAIRS-Handbuch erhältlich. Wir bedauern die Verzögerung.

Unter dem Namen URZL steht jetzt eine STAIRS-Datenbank zur Verfügung, die eine Übersicht über die Bücher der Bibliothek des Rechenzentrums enthält. Diese Datenbank löst die Version BIBL ab.

Die Benutzer der relationalen Datenbank "RDS" (alter Name "VDN") sollten sich jetzt endgültig auf den neuen Datenbanknamen "REFLEX" umstellen. REFLEX liegt jetzt in der Version 2.1 vor, der Betrieb wirkt stabil. Leider ist noch nicht das volle SQL implementiert. Der Precompiler (Pascal) ist noch mit gewissen Problemen behaftet, im wesentlichen:

- keine Überprüfung der SQL-Anweisungen,
- Beschränkung im Umfang der Prozeduren,
- keine Zugriffe auf das Data-Dictionary.

Umgebungen im CMS

von

A. Achilles

Wie bereits aus den NEWS ersichtlich, wurde vor kurzem das ENVIRON-Kommando eingeführt. Dieses Kommando soll nicht nur die Ausführungszeit von Compiler-Aufrufen verkürzen, sondern soll allgemein dazu dienen, dem Benutzer beim Bereitstellen seiner CMS-Umgebung zu helfen. Zur Ausführung vieler Kommandos müssen erst die benötigten Systemplatten für den Benutzer bereitgestellt werden, auf denen die Komponenten gespeichert sind. Wird die Umgebung mit dem ENVIRON-Kommando voreingestellt, so wird das Bereitstellen der Platten bei der Ausführung des ENVIRON-Kommandos vorgenommen und nicht jedesmal bei der Ausführung des gewünschten Kommandos. Zusätzlich werden (z. Zt. nur für Compiler-Aufrufe) Kommandos zum Bereitstellen von MACLIBs, LOADLIBs und TXTLIBs ausgeführt.

Kann z. B. abgesehen werden, daß die meisten Arbeiten der derzeitigen Sitzung SAS-Anwendungen sind, so empfiehlt es sich, durch das Kommando

```
environ sas
```

die Umgebung entsprechend einzustellen. Soll andererseits vorwiegend Textrecherche mit der STAIRS-Datenbank gemacht werden, wobei die gefundenen Dokumente mit DCF (SCRIPT) aufbereitet werden sollen, so stellt der Aufruf

```
environ dcf stairs
```

die gewünschte Umgebung her. Hat der Benutzer vorwiegend mit einer bestimmten Anwendung zu tun, so sollte er erwägen, den Aufruf des ENVIRON-Kommandos in seinem PROFILE EXEC ausführen zu lassen.

Werden beim Einstellen der Umgebung auch ein oder mehrere Compiler angegeben, so werden die Compiler-Optionen, die benötigten FILEDEFs, die mitzuladenden TEXT-Dateien etc. überprüft und entsprechende Voreinstellungen vorgenommen. Hierdurch kann die Ausführungszeit der Compiler-EXECs – nicht jedoch der Übersetzungs- bzw. Ausführungsphase selbst – merkbar reduziert werden.

Das ENVIRON-Kommando kann nicht nur eine Umgebung bereitstellen, es kann auch die eingestellte Umgebung erweitern oder verkleinern und Auskünfte geben. Die Syntax lautet:

ENVIRON [komponenten] [(optionen)]

Wird das ENVIRON-Kommando mit einer oder mehreren Komponenten ohne weitere Optionen aufgerufen, so wird die durch die angegebenen Komponenten beschriebene Umgebung aufgebaut. Die zulässigen Optionen und ihre Wirkungen sind in der folgenden Liste beschrieben:

EXPAND fügt eine oder mehrere Komponenten zur bestehenden Umgebung hinzu. Die gleiche Wirkung kann erzielt werden, indem das ENVIRON-Kommando aufgerufen wird, wobei alle gewünschten Komponenten zugleich angegeben werden. Wird ENVIRON mit Komponenten aufgerufen, ohne die Option **EXPAND** und die folgende Option **RESET** anzugeben, so wird die alte Umgebung abgebaut und eine neue Umgebung, die nur aus den angegebenen Komponenten besteht, aufgebaut.

RESET entfernt die angegebenen Komponenten aus der derzeitigen Umgebung. Wurden keine Komponenten angegeben, so wird diejenige Umgebung wiederhergestellt, die zu Beginn des ersten Aufrufs des ENVIRON-Kommandos nach dem letzten "IPL CMS" bestand.

DISPLAY zeigt die gerade eingestellte Umgebung an. Es werden nicht nur die Platten der bereitgestellten Komponenten, sondern auch die eingestellten MACLIBs, TXTLIBs, LOADLIBs und FILEDEFs angezeigt und zusätzlich – sofern auch eine Compiler-Umgebung eingestellt wurde – genauere Auskünfte über den zugrundeliegenden Compiler gegeben.

LIST gibt an, welche Komponenten dem Benutzer überhaupt zur Verfügung stehen. Die Liste ist aufgeteilt in eine alphabetische Liste der Compiler und eine alphabetische Liste der übrigen Komponenten.

Beim Aufbau der Umgebung können mehrere Compiler-Namen angegeben werden. Das besagt aber nicht, daß die verkürzte Ausführungszeit für alle Compiler-Aufrufe gilt. Der erste angegebene

Compiler-Name entscheidet, für welchen Compiler die Umgebung eingestellt wird, die folgenden Namen dienen nur der interlanguagecommunication, d. h. der Möglichkeit, von einer Sprache aus bei der Ausführung Prozeduren anderer Sprachen aufzurufen. Die Reihenfolge entscheidet dabei zusätzlich über die Suchreihenfolge der Einträge in den TXTLIBs und LOADLIBs. Diese Möglichkeit erlaubt es dem Benutzer, komplexe Compiler-Umgebungen einzustellen, ohne vorher mit Hilfe des DEFAULTS-Kommandos diese Festlegung zu treffen.

Welche Rolle spielt das DEFAULTS-Kommando beim Aufbau von Compiler-Umgebungen? Das DEFAULTS-Kommando speichert dauerhafte Voreinstellungen (auch für Compiler), die mit Aufruf des jeweiligen Kommandos aktiviert werden. Wird mit dem ENVIRON-Kommando eine Umgebung für einen Compiler eingestellt, so wird die mit dem DEFAULTS-Kommando definierte Voreinstellung aktiviert. Generell gilt: das DEFAULTS-Kommando soll nur dann aufgerufen werden, wenn für diese und die folgenden Sitzungen eine andere Voreinstellung gelten soll. Die Aktivierung erfolgt erst durch den Aufruf der Kommandos – bei Compilern z. B. durch den Aufruf des ENVIRON-Kommandos.

In naher Zukunft ist eine weitere Überarbeitung des ENVIRON-Kommandos sowie der Compiler-EXECs geplant. Folgende Erweiterungen sind zu erwarten:

- ⊙ Einbindung der Projekte,
- ⊙ Möglichkeit, für die Übersetzungsphase der Compiler-EXECs FILEDEFs für LISTING und TEXT-Dateien anzugeben.

Im Rahmen dieser Änderungen kann es sich als notwendig erweisen, aus wartungstechnischen Gründen die internen EXEC-Prozeduren RUM\$ACC, RUM\$COMP, RUM\$COM0, RUM\$COM1 und RUM\$COM2 umzubenennen. Den Benutzern wird empfohlen, diese internen EXECs nicht zu verwenden bzw. durch den Aufruf des ENVIRON-Kommandos und/oder durch dokumentierte Compiler-Aufrufe zu ersetzen.

Sollten Sie noch Vorschläge zur Erweiterung der angegebenen Kommandos haben, so wenden Sie sich bitte an den Autor (URZ06). Vorschläge, die bis Mitte November eingegangen sind, werden – soweit möglich – vor der nächsten Überarbeitung berücksichtigt.

Mathematische Formeln mit DCF

von

B. Neukäter

Das Setzen komplizierter mathematischer Formeln mit DCF war bislang eine mühsame Angelegenheit und erforderte die Kenntnisse eines Spezialisten. Mit Hilfe des SCRIPT Mathematical Formula Formatter, einer Erweiterung von DCF, die versuchsweise zur Verfügung gestellt wird, kann der Aufwand erheblich reduziert werden.

Griechische Buchstaben und mathematische Symbole können verwendet werden, soweit sie auf dem Drucker P400 verfügbar sind. Es gibt verschiedene Arten von Klammern. Bruchstriche mit komplexen Zählern und Nennern machen keine Schwierigkeit. Weitere Einzelheiten erfahren Sie in der Programmierberatung, wo auch die Dokumentation eingesehen werden kann.

Es ist zu erwarten, daß nicht alle Möglichkeiten des Mathematical Formula Formatter ausgeschöpft werden können, weil das Programm für den Drucker IBM 4250 geschrieben wurde. Im Rechenzentrum wird jedoch der Drucker AGFA P400 eingesetzt, der zur Zeit nicht alle im Mathematical Formula Formatter erwarteten Zeichen anbietet.

Bekannte Einschränkungen sind:

- Die mathematischen Symbole gibt es nur in der Größe 10 Punkt. Das gilt insbesondere für Integral-, Summen- und Produktzeichen.
- Das Wurzelzeichen steht nicht zur Verfügung.
- Verschiedene seltenere mathematische Symbole werden nicht angeboten. Informieren Sie sich bitte in der Programmierberatung über den angebotenen Satz von Symbolen (GML-Befehl `:symbols`).

Beispiele zum SCRIPT Mathematical Formula Formatter

In diesen Beispielen soll gezeigt werden, wie man mit dem SCRIPT Mathematical Formula Formatter auf einfache Art und Weise Formeln setzen kann.

Den Mathematical Formula Formatter aktiviert man mit den GML-Befehlen

```
:mff on.
:symbols math.
```

Damit die Formel $a_k \sin \beta + H_2^3$ in der laufenden Zeile erscheint, muß man sie zwischen den beiden GML-Befehlen `:f.` und `:ef.` in einer speziell für den Mathematical Formula Formatter vorgesehenen Darstellung eingeben. Formatierungsbefehle wie Hoch- und Tiefstellung werden in dieser Darstellung durch Operatoren beschrieben (*sup* und *sub*).

```
:f. a sub k sin beta
    + H sub 2 sup 3 :ef.
```

Mit

```
:df align=center.
eqno (1)
a sub k sin beta + H sub 2 sup 3
:edf.
```

kann man erreichen, daß die Formel in der Mitte einer Zeile erscheint:

$$a_k \sin \beta + H_2^3 \quad (1)$$

Hier wurde noch eine Numerierung hinzugefügt.

Voraussetzung dafür, daß die Variable *beta* als β ausgegeben wird ist, daß diese Variable als Symbol definiert ist. Dieses wurde mit `:symbols math.` erreicht.

Gruppierung, verschiedene Klammern

```
:df frame=box align=center.
eqno (2)
( % Gamma { x + pi lbracket u over v
- w rbracket } - Omega % )%%
(% a above b % ) %%
{ x over <x sup 2 - 1> }
:edf
```

Der Parameter `frame=box` erzeugt einen Rahmen für die Formel. Klammern dienen zur Gruppierung zusammengehöriger Wörter. Spitze Klammern werden nicht ausgegeben. Das Zeichen `%` bewirkt, daß ein zusätzlicher Zwischenraum von etwa einem Viertel der Dichte eines m gelassen wird. Leerzeichen (blanks) haben keine Bedeutung für die Formatierung einer Formel. Das gesetzte Ergebnis zu obigem Beispiel ist in Formel (2) am Ende dieses Artikels wiedergegeben.

Große gezeichnete eckige Klammern, Integral, Summe und Produkt

```
.se ask = 'alpha sub k'
:df frame=rule align=center.
leqno(3)
s(x) = "u" sub 0 %%
integral
  from adjust(down 6 left 6) 0
  to adjust(up 4 left 5) x
of adjust(left 5) <sigma(xi)d xi>
  - q over 1 %%
left lb
l + delta b+2
sum from <k=1> to infinity
of <product from <j=1> to k
  of <a sub j>
  + b tanh &ask delta>
  over
<&ask %(&ask sup
  adjust(right 3.5) 2 +
  b &ask tanh &ask delta + 1 ) >%%
cos &ask x
right rb
:edf.
```

(Das Ergebnis dieser Eingabe ist Formel (3) am Ende dieses Artikels.)

Da der Text *alpha sub k* an mehreren Stellen vorkommt, wurde er vorher einem Symbol zugewiesen. Die Klammerung *left lb* und *right rb* bewirkt, daß die eckigen Klammern in ausreichender Größe gezeichnet werden. Mit *adjust* erreicht man, daß das folgende Wort oder die folgende in Klammern eingeschlossene Wortgruppe in die entspre-

chenden Richtungen um die angegebenen Einheiten verschoben werden. Ein Zehntel der Größe des Fonts dient dabei als Einheit.

Ein Kettenbruch

```
:df frame=rule align=15mm.
eqno lbracket 4 rbracket
a sub 0 plus
b sub 1 over <a sub 1 plus
  <b sub 2 over <a sub 2 plus
    <b sub 3 over <a sub 3 plus
      <b sub 4 over <a sub 4 plus
        ellipsis
      >>
    >>
  >>
  >
:edf
```

Das Schlüsselwort *plus* hat die gleiche Wirkung wie das Zeichen *+*. Der Parameter *align=15mm* läßt die Formel 15 mm vom linken Rand erscheinen. Die Punkte von *ellipsis* sind nach einem Operator in der Mitte der Zeile, sonst auf der Basislinie zu finden.

$$a_0 + \frac{b_1}{a_1 + \frac{b_2}{a_2 + \frac{b_3}{a_3 + \frac{b_4}{a_4 + \dots}}}} \quad [4]$$

$$\left(\Gamma \left\{ x + \pi \left[\frac{u}{v} - w \right] \right\} - \Omega \right) \left(\frac{a}{b} \right) \left\{ \frac{x}{x^2 - 1} \right\} \quad (2)$$

$$(3) \quad s(x) = u_0 \int_0^x \sigma(\xi) d\xi - \frac{q}{l} \left[1 + \delta b + 2 \sum_{k=1}^{\infty} \frac{\prod_{j=1}^k a_j + b \tanh a_k \delta}{a_k (a_k^2 + b a_k \tanh a_k \delta + 1)} \cos a_k x \right]$$

Personalia

Zum 11.7.1986 ist Frau I. Kranz als Mitarbeiterin im Geschäftszimmer des Rechenzentrums eingestellt worden, nachdem sie ihre Ausbildung an der WWU erfolgreich abgeschlossen hat und bereits die letzten Wochen davor bei uns tätig gewesen ist.

Als studentische Mitarbeiter sind die Herren R. Everaers, H. Groeneveld und F. Lechtenberg zum 1.10.1986 eingestellt worden.

Ausgeschieden als studentische Mitarbeiter sind zum 30.6.1986 Herr G. Scheunemann, zum 31.8.1986 Frau I. Bartodziej und Herr M. Steimke sowie zum 30.9.1986 die Herren G. Hillebrand und R. Timmermann. Herrn Scheunemann können wir zum erfolgreichen Abschluß seines Physik-Studiums beglückwünschen.

RUM/GKS-Dokumentation online

von

E. Sturm

Wie bereits im letzten *infor um* im Artikel "Neue Möglichkeiten im CMS" angedeutet, sind jetzt auch zu RUM/GKS durch das Help-Kommando Erklärungen zu bekommen. Das Kommando

```
help rumgks
```

führt in ein Auswahlménü, wo alle RUM/GKS-Unterprogramme, die bisher dokumentiert sind, aufgelistet sind. Im diesem Auswahlménü sind der Kurzname, wie er in FORTRAN geschrieben werden muß, als auch der Name aus der ISO-Norm, wie er im allgemeinen in PL/I zu schreiben ist, die Kategorie, z.B. Transformation oder Output, sowie das Datum der Eintragung in das Help-System zu finden. Um schneller die gewünschte GKS-Funktion zu finden, kann man man durch wiederholtes Drücken von PF9 ein Sortierkommando in die Kommandozeile holen, welches dann beim Drücken der ENTER-Taste ausgeführt wird, dh. man kann die gesamte Liste nach einem der oben angegebenen vier Kriterien sortieren.

Bringt man dann den Cursor in die Zeile der in Frage kommenden Funktion, so erscheint nach Drücken der ENTER- oder PF1-Taste die entsprechende Dokumentation. Hier erfolgte eine Einteilung in die Punkte:

- Parameter in FORTRAN,
- Parameter in PL/I,
- Beschreibung des Unterprogramms,
- Besonderheiten in FORTRAN,
- Besonderheiten in PL/I (z.B. %INCLUDEs),
- Referenzen (z.B. infor um).

Wir hoffen, demnächst die bisher bescheidene Zahl von dokumentierten Unterprogrammen wesentlich erhöhen zu können, ja irgendwann auch Vollständigkeit zu erreichen.

Neues vom SAS

von

S. Zörkendörfer

Sowohl im Stapelbetrieb (MVS) wie im Dialogbetrieb (CMS) ist momentan die Version 5.08 des "Statistical Analysis System" SAS aktuell. Für beide Betriebssysteme wird in unserem Haus die Probeinstallation 5.15P getestet, die endgültigen Arbeitsversionen 5.16 werden voraussichtlich noch im laufenden Jahr eintreffen und bereitgestellt. Eine Beschreibung der Neuerungen dieser Version 5.15 ist als "Technical Report P-146" (433 Seiten) erschienen und kann bei mir eingesehen werden. Besonders hinweisen möchte ich diesbezüglich auf die neue SPARSEDATA-Option bei der linearen Programmierung (SAS/OR) und auf die direkte Unterstützung (also nicht ausschließlich über GDDM) des Farbgraphikschirmes "IBM 3179 G" (SAS/GRAPH) sowie den Font XSWIS4 mit deutschen Umlauten. Mit der bereits angesprochenen Version 5.15P stehen uns zur kurzzeitigen Erprobung außer den von uns angemieteten Produkten

- SAS/Basics (System, Programmier- und Macrosprache, Statistik),
- SAS/FSP (Insbesondere FSEDIT zur Bildschirmfassung),
- SAS/GRAPH (Graphik, derzeit beschränkt auf Farbgraphikschirme)
- SAS/OR (Lineare Programme, Netzwerke),
- SAS/ETS (Zeitreihen, Modelle)

auch die Produkte

- SAS/IML ("Interactive Matrix Language"),
- SAS/AF ("Application Facility"),

● SAS/QC ("Quality Control")

zur Verfügung.

Die Objekte der Matrixsprache IML sind zweidimensionale Matrizen. Zum Sprachumfang gehören arithmetische und logische Operatoren, zahlreiche Funktionen (z.B. INV, SOLVE, GINV, EIGEN, FFT, IFFT, RANK, SPLINE), skalare SAS-Funktionen, Zugriffsmöglichkeiten auf SAS-Dateien und Routinen zur grafischen Ausgabe. SAS/IML kann vom "Display Manager" des Basisproduktes als eine Prozedur aktiviert werden, die Interaktion geschieht dabei befehlsweise (immediate mode) oder in Moduln (collected mode, etwa bei Schleifen). Programmsegmente können innerhalb des IML selbst als Text-Matrizen verwaltet werden (store, load, parse). In einer Demonstrationsveranstaltung (voraussichtlich am 7.11.1986 15 Uhr im Hörsaal M4, Einsteinstr. 64) wird SAS/IML am graphischen Farbbildschirm IBM 3279 vorgeführt. Ich nehme gerne Anregungen zu Problemen entgegen, deren Lösungen bei dieser Gelegenheit einbezogen werden sollen. Das Rechenzentrum erwägt die Anmietung des SAS/IML am Großrechner unter CMS.

In einer weiteren Vorführung (am 14.11.1986 15 Uhr im Hörsaal M4, Einsteinstr. 64) soll mit SAS/AF die Erstellung, Wartung und Ausführung einer Menü-gesteuerten Anwendung demonstriert werden, und zwar an einem vorgegebenem Lernprogramm zur Anwendung der Regressionsanalyse.

Undokumentierte OpCodes und Macros endlich beschrieben

von

A. Achilles

Geht es Ihnen auch so, lieber Leser, daß Sie manchmal dem Gefühl, "man sollte doch die ganze EDV ...", kaum widerstehen können. Endlich habe ich die Quelle dieser Gefühle entdeckt: ganz offensichtlich ist doch nur ein geringer Teil der möglichen Assembler-Codes dokumentiert, der weitaus größere Teil - und zwar derjenige, der für so manch unliebsame Überraschung gut ist - wird von der großen blauen Firma dezent verschwiegen.

Als ich über mögliche Artikel zum 10-jährigen Jubiläum unserer Informationsschrift *inforu m* nachdachte, stieß ich durch Zufall auf ein heißes Eisen. Beim Blättern in der Zeitschrift "VM/COM, CsNews

Network Newsletter" entdeckte ich sie plötzlich - schwarz auf weiß abgedruckt - diese OpCodes und Macros, deren Existenz wir alle vermuten, die aber dennoch nirgendwo offiziell dokumentiert sind.

Im Stile jedes schlechten Handbuchs waren die Anweisungen - wirre Buchstabenkombinationen, die sich ja doch keiner merken kann - in schlechtem Englisch dokumentiert. Eine Auswahl davon möchte ich Ihnen zum Besten geben; ich hoffe, dabei diejenigen gefunden zu haben, die Sie in letzter Zeit am meisten beunruhigten. Natürlich werde ich Ihnen auch den englischen Kommentar, so schlecht er sein mag, nicht vorenthalten.

Sie kennen sicher das Problem, daß Programme, auch nach mehrmaliger Überprüfung des Quelltextes, nicht die Ergebnisse produzieren, die Sie erwarten. Das Problem ist ganz einfach, irgendwo baut der benutzte Compiler Anweisungen wie

JNL Jump when programmer is Not Looking
BOMB Branch Out and Make Bug

ein. Trösten Sie sich damit: einem kommerziellen Software-Hersteller ist es erst nach mehr als einem Jahr gelungen, die beiden Anweisungen

JRAN Jump RANdom (not to be confused with IRAN - Idiote RANdom)

CRASH Continue Running after Stop or Halt

durch eine Anweisung, nämlich

JT Jump if Tuesday

zu ersetzen, wodurch das Programm wesentlich zuverlässiger wurde. Ein anderer Software-Hersteller benutzt offensichtlich an versteckter Stelle Codes wie

CCB Consult Crystal Ball
SHIT Stop Here If Thursday

Sie wollen wissen, wen ich meine? Na, haben Sie noch nie Schwierigkeiten mit der angebotenen Software gehabt?

Gehören Sie auch zu den Geschädigten, denen das Mißgeschick widerfuhr, keine Datei auf Ihrer A-Platte mehr lesen zu können? Obwohl die zugrundeliegende Anweisung nicht eindeutig lokalisiert ist, dürfen wir vermuten, daß Sie in nächster Zeit nicht wieder auf solche Probleme stoßen werden. Vermutlich handelte es sich dabei um eine der folgenden Anweisungen:

CFS	Corrupt File Structure
DD	Destroy Disk
FDR	Fill Disk Randomly - unwahrscheinlich
KUD	Kill User's Data
RMF	Ruin My Files

Auch an anderer Stelle waren kleine Überraschungen im Betriebssystem eingebaut. Eindeutig konnten folgende OpCodes und Macros identifiziert werden:

CDLD	Crash Disk, Lose Data
EBRS	Emit Burnt Resistor Smell
ICB	Interrupt, Crash and Burn
SPHW	Spill Plenty of Hot Water

Dafür wurde andererseits

SMT	Stretch Magnetic Tape
------------	-----------------------

anscheinend aus dem Betriebssystem verbannt.

Wenn Sie zum Kreis der Benutzer gehören, die den Protokollkonverter benutzen, haben Sie vielleicht schon die Auswirkung von

TDB	Transfer and Drop Bits
TALC	Transfer And Lose Connection

kennengelernt. Nahezu jeder Benutzer wird jedoch indirekt mit folgenden OpCodes konfrontiert worden sein:

ASD	Alter Some Data
PAS	Print And Smear
PUD	Print Upside Down
SP	Scatter Print

Nun wissen Sie endlich, warum bei der Benutzung der verschiedenen Drucker und Plotter das Resultat nicht immer Ihren Qualitätsanforderungen entspricht.

Anweisungen, die sich jeder Analyse weitgehend entziehen, wie

CVG	ConVert to Garbage
UAI	Use Alternate Instruction set

sind glücklicherweise nur an wenigen Stellen des Betriebssystems zu finden, und werden offensichtlich nur über das Macro

UOY	Use at most Once a Year
------------	-------------------------

erreicht. Für Sie vielleicht ärgerlicher ist das vermehrte Auftreten von

TTCWN	Take Time and Conversate With Neighbour
--------------	---

aber denken Sie dabei auch an die positiven Seiten: dadurch wird im Terminalraum für vermehrte soziale Kontakte gesorgt.

Während auch noch im kleinsten Programm die Anweisung

OML	Obey Murphy's Law
------------	-------------------

nachzuweisen ist, habe ich die folgenden Macros, die ebenfalls in der Zeitschrift angeführt wurden, bislang nirgends finden können (und zweifle auch an ihrer Existenz):

GENT	GENERate Thesis
SOAWP	SOLVE All the World's Problems

An der Existenz einer anderen Anweisung zweifle ich jedoch keinen Moment, obwohl sie noch nicht ausfindig gemacht werden konnte. Sie erleben es selbst in jedem Handbuch:

TRA	Te Rds Arvs (Type Ridiculous Abbreviations)
------------	---

Offensichtlich steckt diese Anweisung in allen Systemen, die zur Textverarbeitung benutzt werden, und wird zumindest bei der Produktion jedes Handbuchs aktiviert.

Nach der Lektüre, die ich Ihnen nur verkürzt wiedergegeben habe, fühle ich mich viel besser: endlich gelingt es mir, bislang unverstandene Probleme richtig einzuordnen, meine vage Ahnung ist bestätigt worden und ich versuche, meine eigenen Programme durch ein paar

DBVSI	Do it Better in Very Small Increments
--------------	---------------------------------------

gegen die schlimmsten Fallen zu schützen.

Sprechzeiten

Die Mitarbeiter des Rechenzentrums haben über ihre Beratungstätigkeit hinaus auch verschiedene andere Aufgaben wahrzunehmen, die es insbesondere wegen der Verteilung des Rechenzentrums auf drei verschiedene Gebäude mit sich bringen, daß sie nicht immer in ihrem Büro unter der angegebenen Telefonnummer erreichbar sind. Darum

seien im folgenden Zeiten für eine erste Kontaktaufnahme genannt. Außer diesen gesicherten Zeiten für eine erste Kontaktaufnahme besteht selbstverständlich die Möglichkeit, weitere Gesprächstermine mit den einzelnen Mitarbeitern zu vereinbaren.

Name	Zimmer	Tel.-Nr.	Zeit
Dr. Achilles	E 109	2607	di 10-11.30
Ahrens	E 109	2607	mo-fr 9-11; mo-mi 14-15
Frau Benduhn-Mertz	H 13	2682	mo 8-10
Bosse	E 102	2461	nach Vereinbarung
Bothe	O 11	2475	mo 9-11
Goorkotte	O 03	2672	do 9-11
Dr. Held	E 105	3791	nach Vereinbarung
Kämmerer	H 13	2682	
Dr. Kamp	H 11	2474	mo,do 11-12
Kaspar	H 12	2473	mi 15-16
Kisker	O 01	2467	mo 9-11
Mecke	O 03	2466	mo 9-11
Nabrotzki	H 14	2468	mi 8.30-11.00
Frau Dr. Nienhaus	H 02	2483	mo 14-15
Pietrucha	E 02	2482	mo-fr 9-11, 13-15
Dr. Pudlatz	H 15	2472	di,fr 10-11
Reichel	O 02	2481	mo 9-11
Dr. Richter	H 22	2476	mo 10-12
Dr. Steinhausen	H 36	2464	mo,mi 10-11
Dr. Stöckelmann	H 26	2682	mo 10-12
Stockhofe	O 10	2685	mo 9-11
Sturm	H 16	2609	do 10-11
Dr. Süselbeck	H 34	2686	do 15-16
Dr. Zörkendörfer	H 01	2471	do 10-11
Abteilung <i>Systemsoftware</i>	E	2689	di-fr 9-12; mo-mi,fr 14-17
<i>Problembberatung</i>	H		mo,mi,fr 10-11
<i>Programmierberatung</i>	E	2486	di-fr 10-12; mo-fr 14-17

Gebäude: H = Hittorfstraße 27
 E = Einsteinstraße 60
 O = Orléansring 16

Die Abteilung "Mikrorechner, Rechnernetze und Prozeßdatenverarbeitung" befindet sich jetzt im Orléansring 16, 2. Etage, (Telefonzentrale), Eingang Ostseite neben der Waschanlage.

RUM-Lehre**Lehrveranstaltungen im 2. Halbjahr 1986****1. Einführende Lehrveranstaltungen im Wintersemester 1986/87**

320086	Programmieren in FORTRAN mi, fr 13-15; Hörsaal: M4, M3, Beginn: 22.10.1986 im M4	Mertz, K.-B.
320090	Programmieren in FORTRAN mo, do 15-17; Hörsaal: M4, Beginn: 16.10.1986	Zörkendörfer, S.
320105	Programmieren in Pascal mo 13-15; Hörsaal: M2, Beginn: 20.10.1986	Mersch, R.
320110	Programmieren in Pascal fr 11-13; Hörsaal: M1, Beginn: 17.10.1986	N. N.
320124	Programmieren in PL/I mi 15-17; Hörsaal: M5, Beginn: 15.10.1986	Sturm, E.
320139	Textverarbeitung auf Mikrorechnern di 13-15; Hörsaal: M6, Beginn: 21.10.1986	Kamp, H.
(122723)	*) English for Computer Science do 13-15; Hörsaal: M5, Beginn: 16.10.1986	Gress, E.

*) Wegen der Begrenzung der Teilnehmerzahl ist für diese Lehrveranstaltung eine frühzeitige Anmeldung im Dispatch des Rechenzentrums erforderlich.

2. Weiterführende Lehrveranstaltungen im Wintersemester 1986/87

320143	Programmieren in PL/I für Fortgeschrittene mo 13-15; Hörsaal: M4, Beginn: 20.10.1986	Ost, St.
320158	Datenstrukturen und Programmierverfahren in Pascal di 15-17; Hörsaal: M2, Beginn: 21.10.1986	Bosse, W.
320162	Programmieren in C mi 11-13; Hörsaal: M4, Beginn: 22.10.1986	Richter, G.
320177	Höhere Programmiersprachen: Modula-2 mi 15-17; Hörsaal: M4, Beginn: 22.10.1986	Pudlatz, H.
320181	Werkzeuge der Dialogprogrammierung di 15-17; Hörsaal: M4, Beginn: 21.10.1986	Neukäter, B.
320196	Mikrorechner mo 13-15; Hörsaal: M3, Beginn: 20.10.1986	Kisker, H.-W.
320200	Datenbanken: Theorie und Anwendungen di 13-15; Hörsaal: M4, Beginn: 14.10.1986	Achilles, A.
320215	Rechnernetze mi 9-11; Hörsaal: M4, Beginn: 15.10.1986	Held, W./ Schulze, D.

- | | | |
|--------|--|---|
| 320220 | Anwendungen in SAS: SAS/OR und SAS/ETS
mo 15-17; Hörsaal: M3, Beginn: 20.10.1986 | <i>Steinhausen, D.</i> |
| 320234 | Forschungskolloquium "Computermusik"
mi 8-10; Hörsaal: Bibliothek des Rechenzentrums,
Beginn: 22.10.1986 | <i>Achilles, A./
Brockhoff, M. E.</i> |
| 320249 | Kolloquium über Themen der Informatik
fr 15-17; Hörsaal: M4 | <i>Held, W./
die wiss. Mitarbeiter
des Rechenzentrums</i> |
| 320253 | Anleitung zum Einsatz der EDV bei
wissenschaftlichen Arbeiten | <i>die wiss. Mitarbeiter
des Rechenzentrums</i> |

Einführung in die Benutzung der Rechenanlage

Durch die im folgenden genannten Einzelveranstaltungen soll allen Benutzern des Rechenzentrums die Gelegenheit gegeben werden, sich über die unterschiedlichen Möglichkeiten des Rechnereinsatzes zu informieren. Die Teilnahme an jeder dieser Veranstaltungen ist möglich, ohne daß vorangegangene Veranstaltungen besucht wurden. Eine Anmeldung zu den einzelnen Terminen ist nicht erforderlich.

Die Veranstaltungen finden jeweils freitags von 13 - 15 Uhr (c.t.) im Hörsaal M4 der Mathematischen Institute, Einsteinstraße 64, statt.

- | | | |
|----------|---|-----------------|
| 7.11.86 | Ein Überblick über das Dialogsystem VM/CMS | B. Neukäter |
| 14.11.86 | XEDIT — Arbeiten mit dem System Product Editor | H. Pudlatz |
| 28.11.86 | REXX — Eine Sprache zur Erzeugung von Kommandofolgen
im Dialogsystem CMS | A. Achilles |
| 5.12.86 | Benutzung des Batch-Systems MVS vom Dialogsystem aus:
Aufbau eines Jobs, Kommandos zur Kontrolle des Batch-Jobs,
Bearbeitung der Ausgabedateien | St. Ost |
| 12.12.85 | Einführung in die Benutzung des Dienstprogramms RUMSERV | K.B. Mersch |
| 19.12.86 | Benutzung von Magnetplatten und Magnetbändern im
Batch-System MVS | H. Schalthöfer |
| 16. 1.87 | Einsatz von Datenbanken | A. Achilles |
| 23. 1.87 | Graphische Datenverarbeitung — Einsatzmöglichkeiten
und Hinweise zur Benutzung | E. Sturm |
| 30. 1.87 | SCRIPT — die Document Composition Facility (DCF) für die
Formatierung von Textdaten | W. Kaspar |
| 6. 2.87 | Ein Überblick über das Dialogsystem UNIX | H.W. Kisker |
| 13. 2.87 | Das Statistical Analysis System (SAS) im Dialogsystem CMS | S. Zörkendörfer |

RUM-Graphik

Farbige Rasterbilder II

von

E. Sturm

Wie schon in *inforum* Nr. 2/1986 berichtet wurde, können farbige Rasterbilder, die in Form einer Matrix vorliegen, auf dem Farbbildschirmgerät IBM 5080 angeschaut werden. Die damals ausgesprochene Hoffnung, diese Bilder auch auf Papier ausgeben zu können, scheint sich jetzt zu erfüllen. Der zu diesem Zweck angeschaffte Farbrasterplotter Benson Colorscan 800 arbeitet nun schon seit zwei Monaten ohne wesentliche Fehler (einmal fehlte die Farbe "cyan", kam aber von selbst wieder).

Für die, die die Vorankündigung in den "NEWS" noch nicht gelesen haben, hier noch einmal schwarz auf weiß die Regeln der Benutzung. Bedingt durch die hohen Tusche- und Papierkosten sollte man erst dann ein Bild auf dem Plotter zeichnen lassen, wenn man mit der gewählten Farbdefinition am Terminal zufrieden ist. Erfolgt das Anschauen auf dem Bildschirm auf der C-Maschine unter CMS, so geschieht das Erzeugen des Papierbildes mit Hilfe eines MVS-Jobs auf der B-Maschine. Die das Bild beschreibenden zwei Dateien - eine für die Matrix, eine zur Definition der Farben - sind für beide Systeme identisch und müssen, je nachdem, wo die Daten entstanden sind, mit RUMSERV, SENDFILE oder SENDMVS an das jeweils andere System übertragen werden.

Ein Job im MVS kann ungefähr das folgende Aussehen haben:

```
//ABC99COL JOB (GRAF,B99),NAME,
//  PASSWORD=(<PASSWORD>)
/*JOBPARM T=20
// EXEC COLPLOT,PARME='L=30,C=40,S=5'
//MAT DD DSN=ABC99.BILD.MAT,DISP=SHR
//COL DD DSN=ABC99.BILD.COL,DISP=SHR
```

Der Jobtyp GRAF ist zur Zeit notwendig, um zu gewährleisten, daß nur ein Job auf den Plotter

zugreift, später wird möglicherweise der Jobtyp TEST Verwendung finden. 20 Sekunden Rechenzeit sollten mindestens angegeben werden. Das Parameterfeld ist dasselbe wie beim CMS-Kommando COLPLOT (siehe dazu die oben erwähnte *inforum*-Ausgabe!). Erforderlich sind die DD-Namen MAT sowie COL (hinter den beiden Schrägstrichen), als DSN ist natürlich der gewählte Dateiname im MVS zu spezifizieren. Die Version mit "DD *" ist möglich, wenn man die Farbdefinition direkt hinterher folgen läßt.

Für diesen Job muß im Dispatch ein Job-Begleitzettel abgegeben werden, der das Operating veranlaßt, ihn zu rechnen. Das Bild kann dann auch im Dispatch abgeholt werden. Es empfiehlt sich, dort eine Zeichenmappe oder Plastikhülle zu hinterlegen, da die Farben wasserlöslich sind. Mit handelsüblichem Foto-Spray kann das Bild fixiert werden, falls man die Absicht hat, mit dem Finger auf ihm herumzufahren. Fehlermeldungen suche man im JES2 JOB LOG, also ganz am Anfang des Job-Outputs.

Sowohl beim COLPLOT-Kommando als auch bei der MVS-Prozedur COLPLOT ist eine kompatible Erweiterung vorgenommen worden. Bisher mußte jede Farbe, die verwendet wurde, auch in der COL-Datei definiert sein. In der neuen Version füllt das Programm die Lücken zwischen Farben, die man angegeben hat, durch Interpolation. Dies erspart dem Benutzer die mühsame eigene Definition eines Farbkeils. Alle anderen Farben gelten als weiß (sofern sie also weder explizit gesetzt noch durch Interpolation gewonnen wurden). Als Beispiel diene die folgende COL-Datei:

```
10 1 0 0 "rot"
20 0 1 0 "gruen"
```

Allein die Definition der beiden Farben 10 und 20 bewirkt dann folgende Farbtabelle:

```

1    1  1  1
...
9    1  1  1
10   1  0  0
11   .9 .1  0
12   .8 .2  0
13   .7 .3  0
14   .6 .4  0
15   .5 .5  0
16   .4 .6  0
17   .3 .7  0
18   .2 .8  0
19   .1 .9  0
20   0  1  0
21   1  1  1
...
255  1  1  1
    
```

Bedenken sollte man aber, daß "schöne" Farbkeile vor allem dann entstehen, wenn man sich parallel zu den Kanten des sogenannten RGB-Würfels bewegt. Was es damit auf sich hat, versteht man wahrscheinlich besser, wenn man sich das Bild im Schaukasten im ersten Stock des Rechenzentrums anschaut. Dort sind einzelne Ebenen des RGB-Würfels ausgestellt, gezeichnet vom Farbraster-Plotter.

Plotausgabe auf dem Agfa P400

von
E. Sturm

Wer bisher schon vom Agfa P400 als Drucker beeindruckt war, kann nun auch dessen Qualitäten als Plotter bewundern. Obwohl bisher nur vom CMS aus zu beschicken, sind in den ersten Wochen nach der Ankündigung in den "NEWS" schon hunderte von Bildern auf dem zweiten Agfa-Drucker erzeugt worden. Auch Sie sollten sich überlegen, ob es nicht ausreicht, ein DIN-A4-Bild in einer Farbe statt eines größeren in drei Farben zeichnen zu lassen. Von der einfacheren Bedienung durch die Operateure, die nun nicht mehr jeden Plot von der Rolle abschneiden müssen, über die sehr viel höhere Zeichengeschwindigkeit (die praktisch nur von der Übertragungsrate abhängt) bis zur höheren Strichqualität (Auflösung 16 Pixel/mm) spricht an sich alles für den "Monochrom-Rasterplotter". Nur wer unbedingt mehrere Farben oder ein größeres Format benötigt, sollte noch Bilder auf den Pen-Plotter schicken.

Das Bild muß nicht unbedingt das Format DIN A4 besitzen. Wenn die Koordinaten in der einen Richtung im Bereich von 0 bis 0,255 m und in der anderen Richtung im Bereich von 0 bis 0,175 m liegen, erscheint das Bild unverändert, entweder im Hoch- oder im Querformat. Ist es möglich, das Bild durch Verschiebung in diesen Bereich zu bringen, wird es ebenfalls zentimetergenau gezeichnet. In allen anderen Fällen wird das Bild so verschoben, gedreht und verkleinert, daß es gerade in den oben angegebenen Bereich paßt.

Die in unserem veralteten Plotfile spezifizierbaren Farben schwarz, rot und blau werden durch normale, dicke bzw. dünne Linien angedeutet. Hardwaretext (durch PCHARS bei der alten Plot-Software oder durch Text-Qualität 'STRING' oder 'CHAR' evtl. bei Benutzung von RUM/GKS entstanden) wird nicht abgebildet.

Vorläufig kann man nur vom CMS aus Bilder mit Hilfe des PLOT-Kommandos an den P400 schicken:

```
PLOT fn ft fm (MONochrome)
```

Als ft wird PLOT und als fm * angenommen, falls Angaben fehlen. Im Gegensatz zur bisherigen Konvention ist eine der Optionen MON (dh. an Agfa P400), PPV (d.h. an Plot Preview) oder PEN (dh. an einen der üblichen Pen-Plotter) erforderlich.

Will man im MVS erzeugte Bilder auf den Monochrom-Plotter bringen, muß man PLOT=Q in der EXEC-Anweisung verwenden und das dadurch zum eigenen Reader geschickte File erst mit RECEIVE auf eine Platte übertragen, bevor man das PLOT-Kommando anwenden kann. Auch hier empfiehlt sich zunächst die Option PPV, um sicher zu sein, daß das Bild den Vorstellungen entspricht.

Demnächst wird auch der fürs MVS übliche Weg über Plot Preview und das PLOTSERV-Programm möglich sein. Bei Plot Preview wird man dann außer Hardcopy und Pen-Plot auch Monochrom-Plot ankreuzen können - bei PLOTSERV gibt es dann noch die zusätzliche Angabe MONochrome. Der Job

```

//ABC99XYZ JOB (TEST,B99),Name,
//  PASSWORD=(<PASSWORD>)
// EXEC PLOTSERV
PLOT ABC99XYZ 4042 MON
    
```

schickt dann alle Plots des Jobs mit Namen ABC99XYZ und der Nummer 4042 auf den Agfa-P400-Plotter.

RUM-Tutorial

Bildschirmsteuerung im Dialog mit ISPF

von

A. Achilles

Wenn Sie bei der Benutzung eines Personal-Computers die Möglichkeiten der übersichtlichen Gestaltung der Ein- und Ausgabe auf dem Bildschirm kennengelernt haben, fragen Sie vielleicht, warum dieser Komfort häufig beim Großrechner fehlt. Einer der Gründe, warum beim Großrechner hierfür andere Wege als beim PC beschritten werden, liegt in der Vielzahl verschiedener Bildschirmtypen, die – bei zum Teil unterschiedlichen Steuerbefehlen – sinnvoll bedient werden müssen. Trotzdem braucht niemand auf Bildschirmmasken (Panels) zur Ein- und Ausgabe zu verzichten. Die folgende Artikelserie soll in die Benutzung des Systems ISPF (Interactive System Productivity Facility) einführen, das zur Bildschirmgestaltung benutzt werden kann. Es sei darauf hingewiesen, daß ISPF nicht die einzige Möglichkeit ist, Bildschirmmasken zu benutzen. So können Sie z. B. Bildschirmmasken aufrufen mit

- APL oder SAS in den jeweiligen Umgebungen,
- GDDM über FORTRAN- und PL/I-Programme,
- der RUM/GKS-Komponente PANELIO (vgl. *inforum* Nr. 3/1984) aus PL/I-Programmen,
- XEDIT aus EXECs heraus.

ISPF ist aber das zur Zeit einzige Werkzeug, das Masken sowohl von EXECs (inzwischen auch REXX-EXECs) als auch von den Programmiersprachen APL2, FORTRAN, Pascal und PL/I aus unterstützt.

Dieser Beitrag beschäftigt sich mit dem Problem, wie eine einfache Ein-/Ausgabemaske aufgebaut und wie sie benutzt wird. Folgende Artikel werden sich mit den "Programmieranweisungen" in ISPF, mit der Benutzung von Funktionstasten, mit dem Aufbau dynamischer Masken, mit Fragen des Designs usw. beschäftigen.

Der Aufbau einer Bildschirmmaske

Das Initialisieren von ISPF, der Aufruf der Maske und die Übergabe der Variablen wird im nächsten Kapitel behandelt.

Eine Bildschirmmaske dient dazu, den gesamten Bildschirm zu formatieren. Dabei müssen Textteile, die festgelegt wurden, ausgegeben werden sowie in formatierten Feldern Variablen – aus dem die Maske benutzenden Programm – ausgegeben bzw. eingelesen werden. Die Felder tragen Attribute, die beschreiben, ob es sich um Ein- oder Ausgabefelder handelt, ob der Inhalt hell erleuchtet, normal oder gar nicht zu sehen sein soll, etc.

Die Definition einer Maske kann bis zu sieben Abschnitte enthalten, zwei der Abschnitte müssen Sie bei jeder Maske angeben. Die Reihenfolge der Abschnitte ist vorgeschrieben:

1. Der Attributabschnitt kann benutzt werden, um spezielle Zeichen zu definieren, die den Bildschirmaufbau der Maske steuern. Voreingestellt sind die Zeichen % + _ .
 - % erzeugt ein helles Textfeld,
 - + erzeugt ein Textfeld mit normaler Helligkeit,
 - _ erzeugt ein hellerleuchtetes Eingabefeld.
2. Der eigentliche Körper (Body) der Maske ist erforderlich. In diesem Teil wird der Bildschirmaufbau definiert. Die Maskenbeschreibung unterscheidet sich nur geringfügig von dem Bild, das bei der Ausgabe der Maske zu sehen ist (siehe unten).
3. Der Modell-Abschnitt ist nur bei einer bestimmten Sorte von Masken zulässig und wird im Rahmen dieser Serie zurückgestellt.
4. Der Initialisierungsabschnitt wird dazu benutzt, vor der Ausgabe der Maske Variablen zu initialisieren.
5. Der Reinitialisierungsabschnitt kann benutzt werden, um eine Verarbeitung vorzunehmen, wenn die Maske erneut dargeboten werden soll (etwa nach einem erkannten Eingabefehler).

6. Der Verarbeitungsabschnitt (Processing) dient dazu, festzulegen, wie nach der Darstellung der Maske die Eingaben auszuwerten, insbesondere zu überprüfen sind.
7. Das Ende muß – wie der Körper – in jedem Fall angegeben werden. Es besteht nur aus der Anweisung ")END".

Jeder der Abschnitte wird, sofern er vorhanden ist, durch ein Startsymbol eingeleitet. Die Startsymbole müssen, in der ersten Spalte beginnend, folgendermaßen eingetragen werden:

```
)ATTR
)BODY
)MODEL
)INIT
)REINIT
)PROC
)END
```

Der Körper einer Bildschirmmaske besteht aus Steuerzeichen, Textteilen und Ein-/Ausgabefeldern, die durch Namen (maximal 8 Zeichen, Großbuchstaben) gekennzeichnet sind. Diese Namen stimmen, mit Ausnahme der verwendeten ISPF-internen Variablen, mit den Variablennamen der entsprechenden im EXEC oder Programm zur Ein- bzw. Ausgabe benutzen Variablen überein. Bei der Ausgabe werden die Werte dieser Variablen in den Feldern eingesetzt und die Steuerzeichen ausgewertet. Die Felder und Textteile erscheinen in der Anordnung auf dem Bildschirm, die sie auch im Körper der Maske haben.

Bevor Sie Ihre erste Maske entwerfen, müssen Sie sich noch mit einigen Eigenschaften von ISPF vertraut machen:

- wollen Sie z. B. bei fehlerhaften Eingaben Nachrichten auf der Maske erscheinen lassen, so benutzt ISPF dazu bei einer Kurzmeldung den rechten Teil der ersten Zeile, bei einer langen Meldung die dritte Zeile (diese Voreinstellung kann geändert werden);
- wollen Sie Kommandos (ISPF, aber auch CMS) eingeben können, so sollten Sie dazu in der zweiten Zeile ein Eingabefeld mit dem Namen ZCMD anlegen; ZCMD ist eine spezielle ISPF-Variable (diese Voreinstellung kann geändert werden);
- ISPF verträgt es nicht, wenn in dem Körper mehr Zeilen angegeben werden, als der Bild-

schirm enthält – entwerfen Sie einen Dialog mit Maskenbenutzung, der sowohl von 24-zeiligen wie auch von 32-zeiligen Bildschirmen aus aufgerufen werden soll, so müssen Sie entweder Mittel einsetzen, die erst später besprochen werden sollen, oder aber die Maske von vornherein auf 24 Zeilen auslegen.

- für jede Variable darf die Länge des Inhalts nicht die Feldlänge überschreiten.

Wollen Sie nun mit den bereits bekannten Mitteln eine Maske entwerfen, die in der ersten Zeile auf dem Bildschirm (hell erleuchtet) den Text "==== Testmaske =====", auf der letzten Zeile (normal) "----- Ende -----" ausgibt, in der vierten Zeile ein Ausgabefeld und in der fünften Zeile ein Eingabefeld enthält, so sieht die Lösung folgendermaßen aus:

```
)BODY
%==== Testmaske =====
+ _ZCMD +
+
+Ausgabefeld ==>_AUSGABE +
+Eingabefeld ==>_EINGABE+
+
+----- Ende -----
)END
```

Die ISPF-Variable ZCMD, die in der zweiten Zeile aufgeführt wird, ist nur nötig, wenn Sie von dieser Maske aus auch Kommandos eingeben können wollen. In der vierten Zeile erscheint zusätzlich der Text "Ausgabefeld ==>", in der fünften Zeile "Eingabefeld ==>". In beiden Zeilen dient das Zeichen + dazu, die Felder zu begrenzen: ein längerer Text in der vierten Zeile wird bei der Ausgabe dort abgeschnitten, in der fünften Zeile können nicht mehr Zeichen eingegeben werden. Wenn diese Maske mit den im folgenden besprochenen Mitteln ausgegeben wird, so sieht man, daß die Steuerzeichen %+_ nicht auf dem Bildschirm erscheinen, dafür aber an ihrer Stelle jeweils ein Leerzeichen zu sehen ist. Weiter fällt auf, daß die Zeichenkette, die dem Feld AUSGABE zugewiesen wurde, in Großschreibung umgesetzt wird. Das AUSGABE-Feld ist außerdem noch beschreibbar. Bei der Besprechung des Attributabschnittes werden Möglichkeiten aufgezeigt, diese unerwünschten Effekte zu vermeiden.

Die Initialisierung von ISPF

Bevor die eben definierte Maske von einem EXEC oder Programm aus aufgerufen werden kann, muß zunächst ISPF initialisiert werden. Wollen Sie mehrere ISPF-Anwendungen nacheinander starten, so empfiehlt es sich, die Systemplatte, die die Komponente ISPF enthält, mit dem Befehl

```
ENVIRON ISPF
```

in den Zugriff zu nehmen. Wollen Sie nur eine ISPF-Anwendung starten, so ist das Vorgehen nicht nötig, da beim Aufruf von ISPF die benötigte Platte bereitgestellt (anschließend aber auch wieder entfernt) wird, wenn sie nicht vorhanden ist.

Als nächstes müssen Sie mit Hilfe von FILEDEF-Kommandos Ihr ISPF-Profil und die benötigten Masken dem ISPF bekannt machen. Besitzen Sie kein ISPF-Profil, so wird es während der ISPF-Anwendung unter dem angegebenen Dateinamen angelegt. Die Dateien, die Sie für die Arbeit mit ISPF bereitstellen müssen, sind MACLIBs, TXTLIBs bzw. LOADLIBs. (An dieser Stelle soll schon auf die Erweiterung durch das ISPF-LIBDEF-Kommando verwiesen werden, die in einem späteren Beitrag beschrieben wird). Die Masken, die Sie verwenden wollen, müssen mit dem XEDIT-Kommando zuvor in eine geeignete MACLIB geschrieben werden, z. B.:

```
XEDIT MASKEN MACLIB ( MEMBER TEST
```

Folgende FILEDEF-Setzungen müssen Sie vereinbaren:

- ISPPROF für das ISPF-Benutzer-Profil,
- ISPLIB für die zu verwendenden Masken.

Weiter können folgende Vereinbarungen mit FILEDEF-Kommandos getroffen werden:

- ISPLIB für Nachrichten,
- ISPTLIB für Tabellen (wird zurückgestellt),
- ISPSLIB für die Aufbereitung von Tabellen (wird zurückgestellt),
- ISPLIB für die Deklaration von TXTLIBs,
- ISPLIB für die Deklaration von LOADLIBs.

Die im zitierten Beispiel benötigten FILEDEF-Kommandos lauten:

```
FILEDEF ISPPROF DISK
      MEINISPF PROFILE A (PERM
FILEDEF ISPLIB DISK
      MASKEN MACLIB A (PERM CONCAT
```

Die Option PERM wird benötigt, damit der ISPF-Aufruf das abgegebene FILEDEF-Kommando nicht überschreibt, die Option CONCAT wird benötigt, da die Initialisierung automatisch noch einige weitere MACLIBs unter dem Namen ISPLIB verkettet.

ISPF-Initialisierung eines EXECs

Wollen Sie ein EXEC starten, das die Maske TEST aufruft, so muß erst noch das EXEC, z. B. mit dem Namen TESTEXEC, geschrieben werden:

```
/* TESTEXEC zum Aufruf der
   Bildschirmmaske TEST */
ausgabe = 'Guten Tag'
Address ISPEXEC,
      'DISPLAY PANEL(TEST)'
Say 'Im Feld EINGABE steht',
   'der Wert:' eingabe
Exit 4
```

In der ersten Anweisung wird die Variable "ausgabe" initialisiert, die zweite Anweisung dient dem Aufruf der Bildschirmmaske TEST, die Variable "ausgabe" wird dabei übergeben und in der Maske angezeigt. Die dritte Anweisung zeigt Ihnen, daß Ihre Eingabe auch richtig im EXEC angekommen ist. Wenn Sie das EXEC "TESTEXEC" nicht mit dem Return-Code 4 verlassen, durchlaufen Sie zunächst noch ein zusätzliches Menü, an dem Sie in der Regel nicht interessiert sind. Der Aufruf des EXECs "TESTEXEC" unter ISPF lautet:

```
ISPSTART CMD(TESTEXEC)
```

ISPF-Initialisierung eines PL/I-Programms

Das Benutzen eines Programms, das Masken benutzt, bringt etwas mehr Aufwand mit sich. Wenn Sie ein PL/I-Programm mit dem Namen TESTE geschrieben haben, so müssen Sie es mit dem Aufruf

```
PLIC TESTE
```

übersetzen. Außer den oben angegebenen FILEDEF-Kommandos müssen Sie noch zusätzlich die Kommandos

```
ENVIRON PLI (EXPAND
FILEDEF ISPTLIB DISK
      PLILIB TXTLIB * (PERM CONCAT
FILEDEF ISPTLIB DISK
      CMSLIB TXTLIB * (PERM CONCAT
```

(Zur Zeit muß leider die PL/I-Version 4 - VERSION(OLD) - benutzt werden). Der Aufruf erfolgt dann mit

```
ISPSTART PGM(TESTE)
```

Nun wird das PL/I-Programm TESTE mit Kommentaren vorgestellt:

```
TESTE: PROC OPTIONS(MAIN);

/* Hier beginnen die fuer die
   Verbindung zu ISPF
   notwendigen Deklarationen
   und Voreinstellungen. */
DCL ISPLINK EXTERNAL ENTRY
   OPTIONS(ASM INTER RETCODE);
DCL EINGABE CHAR( 7) INIT((7)' ');
DCL AUSGABE CHAR(22)
   INIT('Guten Tag');
DCL LEINGABE FIXED BIN(31,0);
DCL LAUSGABE FIXED BIN(31,0);
LEINGABE = LENGTH(EINGABE);
LAUSGABE = LENGTH(AUSGABE);

/* Hier werden die Variablen
   EINGABE und AUSGABE
   dem ISPF bekannt gemacht. */
CALL ISPLINK('VDEFINE ',
             '(EINGABE)', EINGABE,
             'CHAR', LEINGABE);
CALL ISPLINK('VDEFINE ',
             '(AUSGABE)', AUSGABE,
             'CHAR', LAUSGABE);

/* Hier wird die Maske TEST
   aufgerufen und die in der Maske
   im Feld EINGABE eingegebene
   Zeichenkette in der Anweisung
   PUT SKIP ... ausgegeben. */
CALL ISPLINK('DISPLAY ', 'TEST ');
PUT SKIP LIST('Die Eingabe lautet',
              EINGABE);

/* Hier wird die
   ISPF-Anwendung beendet. */
CALL ISPLINK('VRESET ');
CALL PLIRETC(4);

END TESTE;
```

Aufruf eines Programms aus einem EXEC

Sie können auch von einem EXEC aus das Programm TESTE ausführen lassen. Das EXEC PROCEXEC leistet dies:

```
/* PROCEXEC zum Aufruf der
   Bildschirmmaske TEST */
EXEC PLIE TESTE'
Exit 4
```

Die FILEDEFS, die gesetzt werden müssen, entsprechen denen des vorigen Kapitels. Gestartet werden muß das EXEC durch

```
ISPSTART CMD(PROCEXEC)
```

(in diesem Falle können Sie auch die PL/I-Version 5.1 benutzen.)

ISPF-Initialisierung eines Auswahl-Menüs

Als letzte Möglichkeit wird gezeigt, wie die Maske als Auswahl-Menü aufgerufen werden kann:

```
ISPSTART PANEL(TEST)
```

Das funktioniert zwar, aber... Das Ausgabefeld bleibt leer, und was nützt es, etwas in das Eingabefeld zu schreiben? Wenn Sie diese ISPF-Anwendung durch einen Druck auf die ENTER- oder die Funktionstaste PF3 beenden, so erscheint eine Fehlermeldung, die besagt, daß es sich hier nicht um eine Auswahlmaske handelt. Trotzdem sollten Sie sich diese Methode merken. Mit erweiterten ISPF-Kenntnissen läßt sich diese Form manchmal gut verwenden.

Attribute zur Definition der Feldsteuerung

Im Attributabschnitt können Sie die Voreinstellungen %+_ verändern oder weitere Zeichen zur Charakterisierung der Felder vereinbaren. Sie können angeben,

- ob Felder Texte darstellen, nur zur Ausgabe oder auch zur Eingabe dienen sollen,
- ob der Inhalt hell erleuchtet, normal oder unsichtbar dargestellt werden soll,
- welche Farbe ein Feld haben soll (nur bei wenigen Bildschirmgeräten ausnutzbar),
- ob bei Feldern für die Ein- oder Ausgabe der Inhalt von Klein- in Großbuchstaben geändert werden soll,
- ob der Inhalt ggf. rechts- oder linksbündig dargestellt werden soll,
- ob der Inhalt mit speziellen Zeichen aufgefüllt werden soll,

- ob der Cursor zum nächsten Eingabefeld vorrücken soll, wenn das Feld vollständig beschrieben wurde,
- ob das Textfeld durch "CursorSelect" ausgewählt werden kann,
- ob das Feld durch Unterstreichung, Negativdarstellung oder Blinken hervorgehoben werden soll,
- ob auch DBCS-Zeichen in der Eingabe erscheinen dürfen,

um nur die wichtigsten Einstellungen zu nennen. Die Voreinstellungen werden durch die Anweisung **DEFAULT** geändert; dabei werden drei Zeichen in Klammern eingeschlossen, die – in der angegebenen Reihenfolge – die Funktion von **%+__** übernehmen. Die Zeile

```
)ATTR DEFAULT($-/)
```

bewirkt, daß die Voreinstellung folgendermaßen geändert wird:

- \$ erzeugt ein helles Textfeld anstelle von %,
- erzeugt ein Textfeld mit normaler Helligkeit anstelle von +,
- / erzeugt ein hellerleuchtetes Eingabefeld anstelle von __.

In den folgenden Zeilen des Attributabschnittes können Zeichen oder zweistellige Hexadezimalzahlen verwendet werden, um weitere Bezeichner zu definieren. Die Anweisungen beginnen mit dem Zeichen, gefolgt von Schlüsselwörtern, denen jeweils ein Attribut begefügt ist. Die Schlüsselwörter für die oben angeführten Möglichkeiten lauten:

TYPE(TEXT|INPUT|OUTPUT) spezifiziert, ob es sich um Text, Eingabe- oder Ausgabefelder handelt. Die Attribute lauten dementsprechend **TEXT**, **INPUT** bzw. **OUTPUT**.

INTENS(HIGH|LOW|NONE) gibt die Helligkeit an, in der das Feld – oder der Text dargestellt wird: **HIGH**, **LOW** bzw. **NONE**. **NONE** kann nur bei Eingabefeldern verwendet werden und unterdrückt dann die Ausgabe.

CAPS(ON|OFF) steuert, ob der Inhalt des Feldes von Klein- in Großbuchstaben umgewandelt wird. **OFF** beläßt die Darstellung, **ON** wandelt die Darstellung in Großbuchstaben um.

JUST(ASIS|LEFT|RIGHT) bewirkt eine Ausrichtung des Feldinhalts nach links bei **LEFT**, nach rechts, wenn **RIGHT** gesetzt wird, oder unterdrückt die Ausrichtung bei **ASIS**.

PAD('x') füllt den nicht ausgefüllten Teil des Feldes mit dem angegebenen Zeichen – in diesem Falle **x** – auf.

Diese Methode soll an einem Beispiel verdeutlicht werden. Mit den Zeichen **.** und **-** sollen weitere Feldmerkmale definiert werden:

- soll Ausgabefelder charakterisieren, bei denen keine Umwandlung in Großbuchstaben vorgenommen wird und die Ausgabe in normaler Helligkeit geschieht,
- soll Eingabefelder definieren, bei denen die Eingabe in Großbuchstaben umgewandelt und die Darstellung unterdrückt wird.

```
)ATTR
```

```
. TYPE(OUTPUT) CAPS(OFF) INTENS(LOW)
- TYPE(INPUT) CAPS(ON) INTENS(NONE)
```

Mit den zur Verfügung gestellten Mitteln ist es Ihnen bereits möglich, einfache Maskendialoge zu entwerfen und aufzurufen. Die nächste Folge wird sich mit dem (Re-)Initialisierungs- und dem Verarbeitungsabschnitt der Maskendefinition und den daraus folgenden weiteren Möglichkeiten befassen.

Die Statistik-Seite

Jobverteilung im August/September 1986

von

A. Ahrens

Angegeben ist eine Übersicht des Durchsatzes in den einzelnen Jobklassen. Die angegebenen Zeiten sind dabei der JOBPARM-Karte entnommen worden. Aus diesem Grunde gibt es auch ca. 140 Testläufe, bei denen mehr als 40 Sek. CPU-Zeit angefordert wurden. Diese Jobs sind jedoch sofort vom System zurückgewiesen worden.

	1-10 sec	11-20 sec	21-40 sec	41-100 sec	101-500 sec	> 500 sec	total
TEST ohne Band	27768 39.29%	10791 15.27%	16569 23.44%	62 0.09%	62 0.09%	13 0.02%	55265 78.19%
TEST mit Band	2803 3.97%	1207 1.71%	589 0.83%	0 0.00%	0 0.00%	5 0.01%	4604 6.51%
PROD ohne Band	116 0.16%	266 0.38%	299 0.42%	1701 2.41%	3598 5.09%	1937 2.74%	7917 11.20%
PROD mit Band	94 0.13%	108 0.15%	466 0.66%	520 0.74%	521 0.74%	294 0.42%	2003 2.83%
HUGE ohne Band	17 0.02%	5 0.01%	3 0.00%	18 0.03%	40 0.06%	125 0.18%	208 0.29%
HUGE mit Band	0 0.00%	2 0.00%	3 0.00%	5 0.01%	0 0.00%	0 0.00%	10 0.01%
LONG ohne Band	0 0.00%	1 0.00%	1 0.00%	0 0.00%	1 0.00%	541 0.77%	544 0.77%
LONG mit Band	0 0.00%	0 0.00%	1 0.00%	0 0.00%	0 0.00%	57 0.08%	58 0.08%
SORT ohne Band	2 0.00%	0 0.00%	2 0.00%	5 0.01%	1 0.00%	32 0.05%	42 0.06%
SORT mit Band	0 0.00%	0 0.00%	0 0.00%	2 0.00%	17 0.02%	10 0.01%	29 0.04%
total	30800 43.58%	12380 17.52%	17933 25.37%	2313 3.27%	4240 6.00%	3014 4.26%	70680 100.00%

Stichwörter *inforum* Jahrgang 10**Ausbildung**

10,3-11 D.Stöckelmann Ausbildung Mathematisch-technischer Assistent(inn)en an der Universität Münster

BASIC

10,4- 3 B.Neukäter Kein BASIC mehr auf dem Großrechner

Benutzerkennung

10,1- 8 W.Bosse Neue Fachbereiche - neue Benutzerkennungen
10,2- 3 A.Achilles Änderung der Benutzerkennungen

Beschaffung

10,1- 7 W.Held/
H.W.Kisker Beschaffungen im Rahmen des CIP 1985

Betriebssystem

10,2-31 M.Brox UNIX - Eine Einführung

CIP

10,1- 7 W.Held Computer-Investitions-Programm 1986
10,1- 7 W.Held/
H.W.Kisker Beschaffungen im Rahmen des CIP 1985

CMS

10,1- 4 A.Achilles Projekte unter CMS
10,1-20 A.Achilles Benutzung fremder CMS-Maschinen
10,1-21 A.Achilles CMS nach Ladenschluß
10,2- 3 A.Achilles Wozu Konventionen?
10,2- 6 W.Kaspar Neue Bezeichnungen für P400-Fonts im FPRINT-Kommando
10,3-13 B.Neukäter Neue Möglichkeiten im CMS
10,3-14 A.Achilles Initialisierung durch SPROFILE
10,3-15 E.Sturm CMS-Kommandos in Programmen
10,3-16 A.Achilles Bildschirmsteuerung mit ISPF
10,4- 4 A.Achilles Umgebungen im CMS
10,4-16 A.Achilles Bildschirmsteuerung im Dialog mit ISPF

Compiler

10,3- 6 St.Ost Neue Version des PL/I-Compilers
10,4- 3 B.Neukäter Kein BASIC mehr auf dem Großrechner

Datenbank

10,3- 3 W.Held Erprobung verschiedener Datenbankanwendungen
10,3-26 H.Pudlatz Datenbankprogramme auf Mikrorechnern
10,4- 4 A.Achilles Notizen zum Thema Datenbanken

Datenschutz

10,4- 4 W.Held Datenschutz und Personal-Computer

Datex-P

10,1- 3 V.Bake Benutzung der CYBER 205 in Bochum über Datex-P

dBASE III

10,3-26 H.Pudlatz Datenbankprogramme auf Mikrorechnern

DCF/GML

10,2- 7 W.Kaspar Hinweise zur Arbeit mit DCF
10,2- 8 W.Kaspar Neue Sonderzeichen in DCF
10,2- 8 W.Kaspar P400-Fonts in DCF
10,3-16 B.Neukäter Berichtigung zu "P400-Fonts in DCF"
10,4- 6 B.Neukäter Mathematische Formeln mit DCF

Dialogsystem		
10,3-13	B.Neukäter	Neue Möglichkeiten im CMS
10,3-14	A.Achilles	Initialisierung durch SPROFILE
10,3-15	E.Sturm	CMS-Kommandos in Programmen
10,3-16	A.Achilles	Bildschirmsteuerung mit ISPF
10,4-16	A.Achilles	Bildschirmsteuerung im Dialog mit ISPF
Disketten		
10,3- 5	H.W.Kisker	Verarbeitung von Fremddisketten
Druckausgabe		
10,3- 4	W.Bosse	Computerdruck und Kostendruck
Expertensysteme		
10,3- 7	B.Süselbeck	Interessengruppe Expertensysteme
GKS		
10,1- 5	E.Sturm	Neue RUM/GKS-Terminals
10,3-17	D.Lammers	Der PC als RUM/GKS-Terminal
10,4- 8	E.Sturm	RUM/GKS-Dokumentation online
Größtrechner		
10,1- 3	V.Bake	Benutzung der CYBER 205 in Bochum über Datex-P
Graphik		
10,1- 5	E.Sturm	Neue RUM/GKS-Terminals
10,2-13	D.Eckey	VIEW3D - räumliche Darstellung von 3D-Objekten
10,2-18	E.Sturm	Farbige Rasterbilder
10,3-10	W.Bosse	Erfahrungsaustausch über Graphikanwendungen
10,3-17	D.Lammers	Der PC als RUM/GKS-Terminal
10,4- 8	E.Sturm	RUM/GKS-Dokumentation online
10,4-14	E.Sturm	Farbige Rasterbilder II
10,4-15	E.Sturm	Plotausgabe auf dem Agfa P400
ISPF		
10,3-16	A.Achilles	Bildschirmsteuerung mit ISPF
10,4-16	A.Achilles	Bildschirmsteuerung im Dialog mit ISPF
Jobverteilung		
10,4-21	A.Ahrens	Jobverteilung im August/September 1986
Künstliche Intelligenz		
10,1-22	P.Bigdon	Künstliche Intelligenz: Ersatz für den Menschen oder großer Bluff?
KEDIT		
10,3- 7	H.Pudlatz	Ein neuer Editor für den PC
Kolloquium		
10,4- 3	-	Gedenkkolloquium zu Ehren von Helmut Werner
Lochkarte		
10,2- 5	B.Neukäter	Sichern Sie den Inhalt Ihrer Lochkarten
Mikrorechner		
10,1-26	H.W.Kisker	Die Struktur eines MS-DOS-Rechners
10,2-20	H.Kamp	Textverarbeitung mit Mikrorechnern
10,3- 7	H.Pudlatz	Ein neuer Editor für den PC
10,3- 9	W.Bosse	Sammelbestellung von PC-Software
10,3- 9	S.Zörkendörfer	SAS auf PCs im Universitätsbereich?
10,3-26	H.Pudlatz	Datenbankprogramme auf Mikrorechnern
10,4- 4	W.Held	Datenschutz und Personal-Computer

MS-DOS		
10,1-26	H.W.Kisker	Die Struktur eines MS-DOS-Rechners
Numerik		
10,1-30	S.Zörkendörfer	Neues vom SAS
10,1-30	S.Zörkendörfer	Mitteilungen zum SPSS
10,4- 8	S.Zörkendörfer	Neues vom SAS
PL/I		
10,3- 6	St.Ost	Neue Version des PL/I-Compilers
Programmquerschnitt		
10,2-35	A.Ahrens	Programmquerschnitt Januar bis März 1986
Projekte		
10,1- 4	A.Achilles	Projekte unter CMS
P400		
10,2- 6	W.Kaspar	Neue Bezeichnungen für P400-Fonts im FPRINT-Kommando
10,2- 8	W.Kaspar	P400-Fonts in DCF
Rechenanlage		
10,4-13	-	Einführung in die Benutzung der Rechenanlage
REDUCE		
10,2-22	B.Süselbeck	REDUCE - Ein System zur algebraischen Formelmanipulation
RUMSERV		
10,1- 8	St.Ost	Neue RUMSERV-Anweisungen
10,3-12	P.Bigdon	HELP-Dateien für das Dienstprogramm RUMSERV
SAS		
10,1-30	S.Zörkendörfer	Neues vom SAS
10,3- 9	S.Zörkendörfer	SAS auf PCs im Universitätsbereich?
10,4- 8	S.Zörkendörfer	Neues vom SAS
SCRIPT		
10,2- 7	W.Kaspar	Hinweise zur Arbeit mit DCF
10,2- 8	W.Kaspar	Neue Sonderzeichen in DCF
10,2- 8	W.Kaspar	P400-Fonts in DCF
10,3-16	B.Neukäter	Berichtigung zu "P400-Fonts in DCF"
10,4- 6	B.Neukäter	Mathematische Formeln mit DCF
Software-Information		
10,3-10	W.Bosse	Software-Informationen
Sprechzeiten		
10,1-12	-	Sprechzeiten
10,4-11	-	Sprechzeiten
SPSS		
10,1-30	S.Zörkendörfer	Mitteilungen zum SPSS
Statistik		
10,1-30	S.Zörkendörfer	Mitteilungen zum SPSS
10,1-30	S.Zörkendörfer	Neues vom SAS
10,3- 9	S.Zörkendörfer	SAS auf PCs im Universitätsbereich?
10,4- 8	S.Zörkendörfer	Neues vom SAS
Textverarbeitung		
10,2-20	H.Kamp	Textverarbeitung mit Mikrorechnern

UNIX			
10,2-31	M.Brox	UNIX - Eine Einführung	
Vektorrechner			
10,1- 3	V.Bake	Benutzung der CYBER 205 in Bochum über Datex-P	
Verbrauchsdaten			
10,1-31	A.Ahrens	Verbrauchsdaten 1985	
Zufallszahlen			
10,1-30	-	Zufallszahlen	