

# inforum

---

INFORMATIONSFORUM des Rechenzentrums der Universität Münster

Jahrgang 10, Nr. 2

Mai 1986

---

## Inhalt

Editorial .....	2
<b><u>RUM-Aktuell</u></b>	
Wozu Konventionen? .....	3
Änderung der Benutzerkennungen .....	3
Leserforum .....	4
Sichern Sie den Inhalt Ihrer Lochkarten! .....	5
Personalia .....	5
Neue Bezeichnungen für P400-Fonts im FPRINT-Kommando .....	6
Hinweise zur Arbeit mit DCF .....	7
Neue Sonderzeichen in DCF .....	8
P400-Fonts in DCF .....	8
<b><u>RUM-Lehre</u></b>	
Übersicht über die Lehrveranstaltungen im SS 1986 .....	11
<b><u>RUM-Graphik</u></b>	
VIEW3D - räumliche Darstellung von 3D-Objekten .....	13
Projizierte Flächen mit verdeckten Linien .....	14
Farbige Rasterbilder .....	18
<b><u>RUM-Tutorial</u></b>	
Textverarbeitung mit Mikrorechnern .....	20
REDUCE - Ein System zur algebraischen Formelmanipulation .....	22
UNIX - Eine Einführung .....	31
<b><u>Die Statistik-Seite</u></b>	
Programmquerschnitt Januar bis März 1986 .....	35

**Impressum**Redaktion *inforum*:

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: 20.6.1986

**Editorial**

von

*E. Sturm*

Wir hoffen, in dieser Ausgabe des *inforum* zum letzten Mal etwas über das Thema Lochkarten zu bringen. Wenn Sie jetzt zum ersten Mal hören, daß der Lochkartenleser demnächst abgebaut wird und mit Schrecken an Ihre Schränke voller Lochkarten-

kisten denken, sollten Sie zuerst prüfen, ob Sie sie nicht wegwerfen können (nicht nur die Kisten, auch die Schränke, wer braucht schon Lochkartenschränke!). Falls Sie doch noch einige der Karten dringend benötigen und es Ihnen ein Rätsel ist, wie man deren Löcher auf Magnetband überträgt, möchte ich Ihnen den entsprechenden Artikel in der Rubrik RUM-Aktuell empfehlen.

Unter der Überschrift RUM-Graphik werden drei Programme vorgestellt, von denen zwei die neuen Hardware-Möglichkeiten des Rechenzentrums zugänglich machen: Zum einen können jetzt dreidimensionale Gebilde wie z.B. Moleküle am Farbrasterbildschirm IBM 5080 betrachtet werden. Mit der Rot/Grün-Brille auf der Nase, die Hand an den Drehknöpfen, kann man seine "Draht"-Modelle oder Punktwolken nach Belieben drehen und wenden! Wer Verbindungen zu Landsat oder zur ESA hat, kann mit dem zweiten Programm Satellitenbilder oder solche vom Halleyschen Kometen auch bei uns betrachten, demnächst hoffentlich auch auf Papier mitnehmen. Mit dem dritten Programm, dem RUM/GKS-Unterprogramm GENERATE\_SURFACE, kann man gekrümmte Flächen mit verdeckten Linien darstellen, wie üblich zunächst in PL/I, später in FORTRAN. Die Abbildungen dieses Artikels wurden übrigens mit demselben Drucker erzeugt wie der Rest dieser *inforum*-Ausgabe. Das Rechenzentrum plant, auch den Benutzern ähnliche Möglichkeiten zur Verfügung zu stellen.

Neben der Fortsetzung der Artikelreihe über UNIX finden Sie zwei weitere Artikel im RUM-Tutorial: Sollten Sie die vielen Formeln im Artikel über REDUCE schrecken, so werden Sie bestimmt an der "Textverarbeitung mit Mikrorechnern" Gefallen finden.

*RUM-Aktuell***Wozu Konventionen?**

von

*A. Achilles*

Wartung an Betriebssystemen ist von Zeit zu Zeit nötig. Der Grund für Änderungen liegt in Maßnahmen zur Fehlerbehebung oder in dem Anbieten verbesserter Leistungen. Wartungsarbeiten am System werden in der Regel so durchgeführt, daß die Benutzer der Rechenanlage davon nicht betroffen werden. Um dies unter dem Betriebssystem CMS garantieren zu können, wurde das Kommando SPROFILE zur Verfügung gestellt, in dem einige Voreinstellungen gesetzt werden, und das als einer der ersten Aufrufe im PROFILE EXEC jedes Benutzers stehen sollte.

Vor kurzem erfolgte eine Änderung am CMS-System. Auch diese Änderung war so geplant und getestet worden, daß Benutzer, die sich an die vorgegebenen Konventionen hielten, dadurch nicht betroffen wurden. Um so erstaunter waren wir über die Vielzahl der Anfragen, warum plötzlich eine Reihe von Funktionen versagten. Nachfragen bei den Betroffenen brachten eine Klärung des Problems: entweder hatten sie ihr PROFILE EXEC gelöscht oder aber bei einer Modifizierung des PROFILE EXECs den Aufruf EXEC SPROFILE herausgenommen.

Diese Beobachtung veranlaßt mich, hier über den Sinn von Konventionen im CMS nachzudenken. CMS ist ein Betriebssystem, das den Benutzern weitestgehende Freiheiten läßt. Diese Bemerkung macht deutlich, daß der Sinn von Konventionen nicht darin gesucht werden kann, die Möglichkeiten der Benutzer einzuschränken. Auch wenn Sie sich als Benutzer an die vorgeschlagenen Konventionen halten, werden Ihre Möglichkeiten nicht beschnitten.

Nun kann man die Meinung vertreten, daß die Konventionen ja doch nur für den Durchschnitt der Benutzer gelten. Leider zeigen sich die Folgen dieser Ansicht leicht schon bei kleinen Änderungen des zur Verfügung gestellten Systems.

Was kann man hieraus lernen? Konventionen sollen sicherstellen, daß die Benutzer möglichst wenig durch Systemänderungen betroffen werden. Kon-

ventionen für die hier angesprochene Situation gilt: sofern Ihr PROFILE EXEC nicht die Zeile

'EXEC SPROFILE'

bzw.

EXEC SPROFILE

enthält, sollten Sie diese Zeile dringend als eine der ersten Anweisungen einfügen. (Die erste Zeile sollten Sie verwenden, wenn Ihr PROFILE in REXX geschrieben ist, d.h. mit einem Kommentar /\* ... \*/ beginnt, die untere Zeile, wenn die erste Zeile Ihres PROFILE EXEC mit &TRACE beginnt.) Sie haben dann immer noch genug Möglichkeiten, Ihre CMS-Umgebung so zu gestalten, wie Sie sie benötigen.

Weitere Konventionen und Anregungen finden Sie sowohl in dieser Zeitschrift als auch aktuell in den NEWS. Sie ersparen sich Ärger (und uns Zeit), wenn Sie diese Informationsquellen regelmäßig beachten. Dafür können Sie dann auch Ihrem Unmut zu Recht Luft machen, wenn's bei Beachtung aller Konventionen wirklich mal nicht klappt.

**Änderung der Benutzerkennungen**

von

*A. Achilles*

Mit der Neuordnung der Fachbereiche wurde auch eine Umstellung einiger Benutzer-Nummern nötig. Diese Umstellung wurde am 1. April 1986 vorgenommen. Wenn Sie zu dem betroffenen Benutzerkreis gehören, haben Sie im voraus einen Brief vom Universitätsrechenzentrum bekommen, dem Sie die Angaben über Ihre neue Benutzerkennung und Ihr neues MVS-Paßwort entnehmen konnten.

Für Sie als Benutzer können Probleme auftreten, unabhängig davon, ob Ihre Nummer geändert wurde oder nicht. Wenn Sie zum Kreis derer gehören, deren Nummer geändert wurde, so werden Sie beim ersten LOGON nach dem 1. April 1986 bei der Umstellung im CMS durch ein EXEC-Kommando unterstützt, das von SPROFILE EXEC aufgerufen wird. Die nicht automatisierbaren Änderungen müssen Sie, wie alle anderen Benutzer, von Hand durchführen. Die wichtigsten seien im folgenden genannt:

Alle Benutzer sollten Ihre NAMES-Datei und die CP LINK-Kommandos in Ihren EXECs daraufhin überprüfen, ob die vorkommenden Benutzerkennungen zu den geänderten zählen. Sie müssen dann durch die neuen Kennungen ersetzt werden. Diese Information finden Sie in der Datei USERID CHANGED Y. In der linken Spalte finden Sie die alten, rechts daneben die zugehörigen neuen Benutzerkennungen.

Bei Problemen mit der Umstellung im CMS können Sie sich an A. Achilles (URZ06, Tel. 2607) wenden.

Für die Umstellung im MVS wurden von Seiten des Rechenzentrums folgende Änderungen vorgenommen:

- die privaten Magnetplattendateien der betroffenen Benutzer wurden umbenannt,
- die Eintragungen (member) in den Benutzerbibliotheken SYSTEM.xxxxxx wurden umbenannt,
- die Zugriffsberechtigungen zu den Magnetbändern oder Magnetplattendateien wurden entsprechend geändert.

Sie müssen noch Ihre JCL-Anweisungen daraufhin überprüfen, ob auf die alten Benutzerkennungen Bezug genommen wird, um sie ggf. zu ändern. Bei Problemen wenden Sie sich bitte an St. Ost (URZ49, Tel. 2681).

### Leserforum

*In der letzten Zeit stelle ich fest, daß der Terminalraum des Rechenzentrums immer voller und voller wird. Dies allerdings nicht aufgrund der steigenden Programmieraktivität der Benutzer, sondern mehr durch ein verstärktes Erstellen von Texten mittels SCRIPT. Für einen Benutzer wie mich, der die angebotenen Leistungen des Rechenzentrums "nur" für Programmierzwecke im Rahmen seiner wissenschaftlichen Arbeit benutzt, bleibt angesichts der vielen "Scripter" praktisch keine Möglichkeit mehr, vernünftig zu arbeiten (z.B. mit GKS). Ich bin gezwungen, auf Terminals außerhalb des Hauses auszuweichen und jeweils für jeden Plot (bzw. Plotversuch, wenn es um die Erstellung eines neuen Programmes geht) hierhin zu kommen. Wenn ich Glück habe, wird das "schnelle" Textronix-Terminal nicht für den*

*VM-Betrieb benutzt. Eigentlich ist es ja eine Verschwendung von Mitteln, daß diese sicherlich teuren Display-Terminals nur für die Erstellung von Texten benutzt werden, und damit die Ausbildung, z.B. in den Programmierkursen, und die Forschung dafür zurückbleiben muß.*

*Clement, QPC*

Die Redaktion hat diesen Leserbrief gekürzt wiedergegeben. Wir danken für den Denkanstoß, der hier gegeben wurde. Erste Lösungen, mit dem Problem der Dateneingabe umzugehen, werden im Artikel "Hinweise zur Arbeit mit DCF" von W. Kaspar in dieser inforum-Ausgabe aufgezeigt.

*Seit einiger Zeit nun gibt es das Programmpaket GKS am RZ und es läuft ja wirklich sehr zufriedenstellend. Doch leider ist die Dokumentation zum GKS ausgesprochen schlecht organisiert. Das einzig Verfügbare ist eine Loseblatt-Sammlung von inforum-Artikeln, die irgendwann mal zum Thema GKS geschrieben wurden. Außerdem ist diese Sammlung nur in der Beratung einzusehen und in der beratungslosen Zeit leider nicht zugänglich.*

*Ergäbe sich also die Situation, daß nach 17.00 Uhr eine Achse gezeichnet werden soll und die inforums X,Y,Z gerade nicht greifbar sind, so muß man eben ohne Achse auskommen. Ich finde, im Zeitalter von CMS ein unhaltbarer Zustand.*

*Warum wird nicht eine Datei eingerichtet, in der alle GKS-Befehle kurz erläutert sind und aus der man sich mit PPUT die benötigten Aufrufe direkt in sein Programm kopieren kann, wo man dann nur noch die Variablennamen ändern muß?*

*Meiner Meinung nach ein vertretbarer Aufwand mit großer Hilfe für den Benutzer....*

*M. Hannemann, SGP*

Nachdem jetzt endlich die GKS-Norm auch als DIN-Norm veröffentlicht worden ist, werden wir nach und nach alle im RUM/GKS möglichen Unterprogramme dokumentieren, und zwar im CMS online zugreifbar. Als Soforthilfe empfehlen wir PL/I-Benutzern, in den %INCLUDE-Dateien GKSAXIS, GKSBAR usw. nachzuschlagen, die auch eine Minimaldokumentation enthalten. Zudem ist es durchaus möglich, die GKS-Unterlagen der Programmierberatung für die eigene Arbeit zu kopieren.

## Sichern Sie den Inhalt Ihrer Lochkarten!

von

B. Neukäter

Sollten Sie zu den Liebhabern alter Lochkarten gehören, die noch immer nicht deren Inhalt auf Magnetband gesichert haben, obwohl der Kartenleser laut Ankündigung in *inforum* Nr. 4/1985 bereits am 31.3.1986 außer Betrieb gesetzt werden sollte, so bitten wir Sie inständig, das Versäumnis nachzuholen, bevor es zu spät ist. Der Kartenleser steht zwar noch bereit, wird aber nicht mehr gewartet und im Laufe dieses Sommersemesters endgültig abgebaut!

Sollte Ihr Versäumnis darin begründet sein, daß Sie nicht genau wissen, wie man den Inhalt von Lochkarten auf ein Magnetband überträgt, so lesen Sie bitte folgende Hinweise.

Haben Sie weniger als ca. 1000 Lochkarten zu übertragen, so stellen Sie folgenden Auftrag zusammen:

```
//... JOB ...
/*JOBPARM TAPES=1
// EXEC IEBCGENR
//SYSUT2 DD UNIT=TAPE,VOL=SER=valid,
//          DSN=dsname,DISP=(NEW,KEEP),
//          LABEL=labnr,
//          DCB=(RECFM=FB,LRECL=80,
//          BLKSIZE=8000)
//SYSUT1 DD *
```

Ihre Lochkarten, die kein // oder /\* in den beiden ersten Spalten enthalten dürfen.

/\*

Es bedeuten:

**valid** Bezeichnung Ihres vom Rechenzentrum zugeteilten Magnetbandes

**dsname** Bezeichnung der Magnetbanddatei

**labnr** Folgenummer der Magnetbanddatei

Sollten unter Ihren zu übertragenden Lochkarten solche sein, die in den ersten beiden Spalten // oder /\* enthalten aber nicht ::, so schreiben Sie statt

```
//SYSUT1 DD *
```

die Anweisung

```
//SYSUT1 DD DATA,DLM='::'
```

und beenden den Kartenstapel mit :: statt mit /\*. Anstelle von :: können Sie natürlich auch eine andere Kombination verwenden.

Sollten Sie noch ganze Kisten mit wertvollen Lochkarten gehortet haben, so füllen Sie einen Jobbegleitzettel aus und schreiben folgenden Job:

```
//... JOB ...
/*JOBPARM TAPES=1
// EXEC SERVICE
//SYSIN DD *
CARDIN DSN(dsname) VOL(valid) -
LABEL(labnr) ALL LIST
```

Sollen nur erfolgreich gelesene Karten in die Magnetbanddatei übertragen werden, so lassen Sie das Schlüsselwort ALL weg. Andernfalls kann Ihre Magnetbanddatei auch fehlerhafte Daten enthalten.

Mit Jobbegleitzettel, Job und Lochkartenkisten wenden Sie sich an Herrn Pietrucha (Maschinen-saal) und bitten ihn, Ihnen beim Einlesen behilflich zu sein, da große Mengen von Lochkarten erfahrungsgemäß nicht einfach zu beherrschen sind. Außerdem sind bei diesem Verfahren noch weitere Vorbereitungen des Operateurs notwendig.

## Personalia

Seit dem 28.1.1986 ist Frau G. Koopmann im Rechenzentrum vertretungsweise im Sekretariatsbereich tätig.

Am 1.2.1986 ist Herr Dr. D. Stöckelmann vom Institut für Mineralogie zum Rechenzentrum gekommen, um hier als Ausbildungsleiter des neu eingerichteten Ausbildungsganges für Mathematisch-technische Assistenten tätig zu sein.

Herr D. Schulze konnte zum 1.4.1986 sein 25-jähriges Dienstjubiläum feiern, zu dem auch die *inforum*-Redaktion gratuliert. Nach zwei Jahrzehnten Tätigkeit im Institut für Kernphysik ist er nun auch bereits 5 Jahre bei uns im Bereich Datenfernübertragung zum Großrechner aktiv.

Frau C. Schütz ist bis zum 31.5.1988 von ihrer Tätigkeit im Rechenzentrum beurlaubt.

Bei den Studentischen Mitarbeitern hat es folgende Veränderungen gegeben:

Ausgeschieden sind zum 28.2.1986 Herr S. Heuer und zum 31.3.1986 Herr H. Aschmann.

Am 1.4.1986 haben Frau E. Krause sowie die Herren F. Budde, G. Hillebrand und D. Lammers ihre Tätigkeit im Rechenzentrum aufgenommen.

## Neue Bezeichnungen für P400-Fonts im FPRINT-Kommando

von

W. Kaspar

Am 26. Mai 1986 werden nahezu alle P400-Fonts gegen neuere Versionen ausgetauscht. Dabei werden sich die Namen der Fonts ändern, da Agfa-Gevaert die Systematik der Namensgebung geändert hat. Dieser Umtausch ist erforderlich, da die meisten unserer bisherigen P400-Fonts nicht alle von uns benötigten Zeichen enthielten.

Abbildung 2 auf Seite 7 zeigt eine Gegenüberstellung der alten und neuen nicht-proportionalen Fonts.

Um die Benutzung der Fonts im FPRINT-Kommando zu vereinfachen, werden ab dem 26. Mai 1986 folgende Optionen neu eingeführt:

### TYPEFace s

Auswahl des Schrifttyps *s*. Die zur Verfügung stehenden Schrifttypen sind in Abbildung 1 aufgeführt. Der Schrifttyp "Modern" ist voreingestellt.

### POINTsize n

Festlegen der Schriftgröße *n*. Diese Angabe ist ein Maß für die Höhe der Buchstaben eines Fonts und wird in Pica (1 Pica  $\triangleq$  0,35 Millimeter) gemessen.

Da nicht jeder Schrifttyp in allen Schriftgrößen vorhanden ist (pro Schriftgröße wird ein separater P400-Font benötigt), können nur bestimmte Kombinationen von Schrifttypen und -größen benutzt werden (siehe Abbildung 1). Die Schriftgröße "10" ist voreingestellt.

Über diese beiden Optionen wird immer einer der Fonts aus Abbildung 2 auf Seite 7 ausgewählt – wie z.B. mit den Angaben TYPEFACE MODERN POINTSIZE 8 der Font "Modern15:E.N.8.P".

Schrifttyp	Schriftgröße		
	6	8	10
Modern		X	X
Courier			X
Bulletin	X		

Abbildung 1. Zulässige Kombinationen von Schrifttyp und Schriftgröße.

Wird zusätzlich zu diesen beiden Optionen noch die Font-Option angegeben, so werden die Optionen TYPEFACE und POINTSIZE ignoriert und es wird der in der Font-Option angegebene Font benutzt. Fehlen alle diese drei Optionen, wird der Font "Modern12:E.N.10.P" benutzt.

### Beispiele

Die Datei "P400 TEST A1" soll aus folgender Zeile bestehen:

Dies ist ein Fonttest.

Mit dem Kommando

```
fprint p400 test a
(dev p400 typeface modern pointsize 8)
```

könnte dann folgendes Druckbild erreicht werden:

Dies ist ein Fonttest.

Wird das Kommando

```
fprint p400 test a
(dev p400 typeface courier)
```

abgesetzt, so liefert der P400 folgenden Ausdruck (Pointsize 10 ist voreingestellt):

Dies ist ein Fonttest.

Bei der Eingabe von

```
fprint p400 test a
(dev p400 pointsize 8
font bulletin22:e.n.6.p)
```

wird die Pointsize-Option nicht berücksichtigt, da zusätzlich die Font-Option angegeben wurde. Das Druckergebnis sieht dann so aus:

Dies ist ein Fonttest.

## Hinweise zur Arbeit mit DCF

von

W. Kaspar

Immer mehr Benutzer des Rechenzentrums wenden das Formatierungssystem DCF an, um ihre wissenschaftlichen Arbeiten in ansprechender Form zu Papier zu bringen.

Da die Verweilzeit von DCF-Benutzern, die an CMS-Terminals Texte erfassen, i.a. groß ist, müssen viele Benutzer inzwischen immer längere Wartezeiten in Kauf nehmen, bevor ein CMS-Terminal für sie frei wird.

Neben den Maßnahmen des Rechenzentrums, die Wartezeiten auf ein freies Terminal zu verkürzen, z.B. durch Anschaffen weiterer Terminals, könnten auch einige Benutzer zur Entlastung der CMS-Bildschirme beitragen, indem sie die erste Erfassung der Texte an PCs durchführten.

Für diese Arbeit ist jeder PC-Editor geeignet, der außer der Markierung von Zeilenenden keine weitere Steuerinformationen in den Text einstreut (z.B. Professional Editor, RED, WordStar im Non-Document-Modus, WordPerfect). Für XEDIT-gewohnte Benutzer ist ein vergleichbarer Editor unter MS-DOS angekündigt worden.

Die so erfaßten Daten können dann über einen IBM-PC im Seminarraum des Rechenzentrums (Raum 107) ins CMS übertragen werden. Sind die Daten nicht im IBM-PC-Format auf Diskette gespeichert, so können sie an einem anderen PC im gleichen Raum auf das zur Übertragung ins CMS benötigte IBM-PC-Format umgesetzt werden. Dokumentationen zur Umsetzung der Diskettenfor-

mate (z.B. welche Formate unterstützt werden) und zur Übertragung ins CMS befinden sich bei den entsprechenden PCs. Durch diese Vorgehensweise können unter Umständen die Wartezeiten vor besetzten Terminals verkürzt werden.

Da einerseits immer mehr Institute eigene PCs beschaffen, andererseits über das Computer-Investitions-Programm (CIP) ganze PC-Pools an mehreren Orten der Universität eingerichtet werden, müßte demnächst sehr viel mehr Benutzern ein PC zugänglich sein. Dadurch wird in der Regel der Weg zum nächsten PC kürzer sein als zu einem CMS-Terminal, so daß auch umfangreichere Arbeitsunterlagen zur Textaufnahme leicht mitgenommen werden können. Die weitere Bearbeitung und endgültige Produktion kann dann wie bisher von CMS-Terminals aus erfolgen.

Ein weiteres Problem im Zusammenhang mit DCF sind die Betriebskosten des Agfa-Druckers P400, die innerhalb eines Jahres sprunghaft gestiegen sind (Jan. 1985: 500 Seiten pro Tag; Jan. 1986: 2000 Seiten pro Tag). Zur Entlastung des P400 soll deshalb daran erinnert werden, daß zur inhaltlichen Kontrolle der Texte ein Schnelldruckerprotokoll in vielen Fällen ausreicht.

Erfolgt die Textaufnahme an einem PC, so kann in der Regel ein angeschlossener Drucker die ersten Korrekturfahnen liefern, so daß die komplette inhaltliche Fertigstellung des Textes am PC - d.h. in der Nähe des Autors - erfolgen kann.

Erst wenn die inhaltlichen Arbeiten am Text abgeschlossen sind, sollte die Kontrolle des Zeilen- und Seitenumbruchs am P400 durchgeführt werden.

Font	Zeich. Breite	Zeichen/Zeile		Pitch	max. Zeilen pro Seite	
	in Dots	Rand 34mm	Rand 2mm		FCB in Dots	Zeilen
Bulletin22:E.N.6.P (Bulletin22.N.6.P)	20 (20)	140 (140)	165 (165)	22 (22)	38 (38)	123 (123)
Modern15:E.N.8.P (Antic15.N.8.P)	27 (27)	103 (103)	122 (122)	15 (15)	49 (49)	95 (95)
Modern12:E.N.10.P (Modern10.N.12.P)	33 (33)	84 (84)	100 (100)	12 (12)	66 (66)	71 (71)
Courier10:E.N.10.P (Courier10.N.10.P)	41 (41)	68 (68)	80 (80)	10 (10)	67 (65)	70 (72)

Abbildung 2. Neue und alte (in Klammern) nicht-proportionale Fonts.

Wenn dabei nach dem dritten P400-Ausdruck auf den Seiten 5 und 97 immer noch Trennungen zu korrigieren sind, genügt es vielfach, nur diese Seiten<sup>1</sup> oder, wenn dies nicht ausreicht, noch einige der folgenden Seiten, z.B. bis zum nächsten Kapitelanfang<sup>2</sup>, erneut zu drucken.

Wir appellieren also an unsere DCF-Benutzer, Texterfassung und erste Korrekturen nicht am Großrechner vorzunehmen, sondern dazu PCs zu nutzen.

## Neue Sonderzeichen in DCF

von  
W. Kaspar

Ab sofort stehen unter DCF eine Reihe von mathematischen, griechischen und diakritischen Zeichen über Symbole zur Verfügung.

Da in den meisten Texten nicht alle Zeichen benötigt werden, ist standardmäßig nur ein Teil der Symbole definiert. Über das neue GML-Tag

`:symbols` *Symbolgruppe*.

können bei Bedarf weitere Symbole aktiviert werden. Folgende *Symbolgruppen* werden zur Zeit angeboten: *general*, *math*, *text* und *all*.

Eine Liste aller Symbole liegt in Kürze als Sonderdruck bei der Programmierberatung vor.

## P400-Fonts in DCF

von  
W. Kaspar

Die meisten Texte, die von DCF formatiert werden sollen, enthalten "Starter Set Tags", die die einzelnen Textabschnitte markieren. Bei der Formatierung wird über diese Markierungen (Tags) die Bearbeitung von Script-Steuerwörtern angestoßen, die dann ein bestimmtes Druckbild erzeugen.

Unter anderem werden für bestimmte Textteile verschiedene Fonts benutzt, die über das Script-Steuerwort `.BF fontname` ein und mit `.PF` wieder ausgeschaltet werden.

<sup>1</sup> mit der Option "page (5 only 97 only)" des Script-Kommandos

<sup>2</sup> "page (5 to 10 97 to 98)"

Am Anfang von Beispielen, die bekanntlich mit `:XMP.` und `:EXMP.` markiert werden, wird z.B. ein Font Namens XMPFONT mit

```
.bf xmpfont
```

eingeschaltet und am Ende mit

```
.pf
```

wieder ausgeschaltet. Damit der DCF-Bezeichner XMPFONT als ein Fontname erkannt wird, muß er vor seinem Auftreten in einem `.BF`-Steuerwort mit dem `.DF`-Steuerwort vereinbart werden.

Im System-Profile von Script steht deshalb die folgende Zeile:

```
.df xmpfont ('Modern' 10
              type
              medium normal)
```

Hiermit wird festgelegt, daß mit dem Fontnamen XMPFONT ein Font mit Schrifttyp (typeface) MODERN, Schriftgröße (point-size) 10, mittlerer Stärke (medium weight) und normaler Breite (normal width) gemeint ist. Vereinbart man nun im eigenen Profile

```
.df xmpfont ('Courier' 10
              type
              medium normal)
```

so würden alle Beispiele im Schrifttyp COURIER gedruckt.

Auf diese Weise lassen sich auch noch eine Reihe weiterer Fontnamen, die sich auf andere Textgruppen des "Starter Sets" beziehen, undefinieren:

<code>xmpfont</code>	In <code>:xmp.</code> und <code>:exmp.</code> eingeschlossene Beispiele.
<code>figfont</code>	Text innerhalb von Abbildungen ( <code>:fig.</code> , <code>:efig.</code> )
<code>figcap</code>	Text in Abbildungsbezeichnungen ( <code>:figcap</code> )
<code>figdesc</code>	Text in Abbildungsbeschreibungen ( <code>:figdesc</code> )
<code>hi0</code>	Nicht hervorgehobener Text ( <code>:hp0.</code> )
<code>hi1</code>	Hervorgehobener Text ( <code>:hp1.</code> )
<code>hi2</code>	Hervorgehobener Text ( <code>:hp2.</code> )
<code>hi3</code>	Hervorgehobener Text ( <code>:hp3.</code> )
<code>fn</code>	Fußnoten
<code>lqfont</code>	Zitate ( <code>:lq.</code> , <code>:elq.</code> )
<code>@rh</code>	Laufende Kopfzeile
<code>@rf</code>	Laufende Fußzeile

Typeface	Point size	Width	Weight	Attributes
Triumvirate	6	normal	medium	
Triumvirate	6	normal	semibold	
<i>Triumvirate</i>	6	<i>normal</i>	<i>medium</i>	<i>italic</i>
<i>Triumvirate</i>	6	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Triumvirate	8	normal	medium	
Triumvirate	8	normal	semibold	
<i>Triumvirate</i>	8	<i>normal</i>	<i>medium</i>	<i>italic</i>
<i>Triumvirate</i>	8	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Triumvirate	10	normal	medium	
Triumvirate	10	normal	semibold	
<i>Triumvirate</i>	10	<i>normal</i>	<i>medium</i>	<i>italic</i>
<i>Triumvirate</i>	10	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Triumvirate	12	normal	medium	
Triumvirate	12	normal	semibold	
<i>Triumvirate</i>	12	<i>normal</i>	<i>medium</i>	<i>italic</i>
<i>Triumvirate</i>	12	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Triumvirate	14	normal	medium	
Triumvirate	14	normal	semibold	
<i>Triumvirate</i>	14	<i>normal</i>	<i>medium</i>	<i>italic</i>
<i>Triumvirate</i>	14	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Triumvirate	18	condensed	bold	
<i>Triumvirate</i>	18	<i>condensed</i>	<i>bold</i>	<i>italic</i>

Abbildung 1. Proportionale Schriften

**title** Titelseite: Titel  
**docnum** Titelseite: Dokumentnummer  
**date** Titelseite: Datum  
**author** Titelseite: Autor  
**address** Titelseite: Adresse

In Abbildung 1 und 2 sind alle Fonts aufgeführt, die ab dem 26. Mai 1986 zur Verfügung stehen werden.

Alle Benutzer, die zur Zeit schon das .DF-Steuerwort benutzen, werden gebeten, die Schrifttypenbezeichnungen der nicht-proportionalen Schriften den neuen Konventionen anzupassen. Diese Umstellung wurde durch die Namensänderung einiger P400-Fonts notwendig (siehe hierzu auch den Artikel "Neue Bezeichnungen für P400-Fonts im FPRINT-Kommando"). Die alten Bezeichnungen werden ab dem 26. Mai 1986 noch ca. 8 Wochen gültig bleiben und dann ohne eine weitere Ankündigung gelöscht.

Typeface	Point size	Width	Weight	Attributes
Bulletin	6	normal	medium	
<i>Bulletin</i>	6	<i>normal</i>	<i>medium</i>	<i>italic</i>
Bulletin	6	normal	semibold	
<i>Bulletin</i>	6	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Modern	8	normal	medium	
<i>Modern</i>	8	<i>normal</i>	<i>medium</i>	<i>italic</i>
Modern	8	normal	semibold	
<i>Modern</i>	8	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Modern	10	normal	medium	
<i>Modern</i>	10	<i>normal</i>	<i>medium</i>	<i>italic</i>
Modern	10	normal	semibold	
<i>Modern</i>	10	<i>normal</i>	<i>semibold</i>	<i>italic</i>
Modern	18	condensed	medium	
<i>Modern</i>	18	<i>condensed</i>	<i>medium</i>	<i>italic</i>
Modern	18	condensed	semibold	
<i>Modern</i>	18	<i>condensed</i>	<i>semibold</i>	<i>italic</i>
Courier	10	normal	medium	
<i>Courier</i>	10	<i>normal</i>	<i>medium</i>	<i>italic</i>
Courier	10	normal	semibold	
<i>Courier</i>	10	<i>normal</i>	<i>semibold</i>	<i>italic</i>

Abbildung 2. Nicht-proportionale Schriften

**RUM-Lehre**

**Übersicht über die Lehrveranstaltungen im SS 1986**

**1. Einführende Lehrveranstaltungen**

320078	Einführung in die EDV di 13-15; Hörsaal: M3, Beginn: 29.4.1986	Schalzhöfer, H.
320082	Programmieren in FORTRAN fr 9-11; Hörsaal: M1, Beginn: 25.4.1986	Achilles, A.
320097	Programmieren in FORTRAN mo 13-15; Hörsaal: M2, Beginn: 28.4.1986	Stöckelmann, D.
320101	Programmieren in Pascal mi 11-13; Hörsaal: M3, Beginn: 23.4.1986	Benduhn-Mertz, A.
320116	Programmieren in Pascal mi 13-15; Hörsaal: M3, Beginn: 23.4.1986 und 2 Std. Übungen n. V.	Mertz, K.-B.
320120	Programmieren in PL/I di 15-17; Hörsaal: M5, Beginn: 22.4.1986	Neukäter, B.
320130	Programmieren in SAS do 15-17; Hörsaal: M4, Beginn: 24.4.1986	Zörkendörfer, S.

**2. Weiterführende Lehrveranstaltungen**

320140	Programmieren in FORTRAN für Fortgeschrittene mo 13-15; Hörsaal: M3, Beginn: 28.4.1986	Ost, St.
320154	Programmieren in Pascal für Fortgeschrittene di 15-17; Hörsaal: M4, Beginn: 29.4.1986	Pudlatz, H.
320169	UNIX fr 13-15; Hörsaal: M4, Beginn: 25.4.1986	Kisker, H.-W.
320173	Programmieren in C mi 11-13; Hörsaal: M5, Beginn: 30.4.1986	Richter, G.
320188	Höhere Programmiersprachen: SNOBOL4 und Icon di 15-17; Hörsaal: M6, Beginn: 29.4.1986	Bosse, W.
320192	Graphische Datenverarbeitung mi 15-17; Hörsaal: M5, Beginn: 23.4.1986	Sturm, E.
320207	Entwurf von Layout-Strukturen für den Einsatz in DCF mi 13-15; Hörsaal: M5, Beginn: 23.4.1986	Kaspar, W.
320211	Datenbanken: Theorie und Anwendungen di 13-15; Hörsaal: M4, Beginn: 29.4.1986	Achilles, A.
320226	Betriebssysteme mo 15-17, Hörsaal: M4, do 15-17, Hörsaal: M6, Beginn: 24.4.1986	Held, W.

- |          |   |   |
|----------|---|---|
| 320230   | Mikrocomputer-Anwendungen<br>mi 13-15; Hörsaal: M4, Beginn: 30.4.1986   | Kamp, H./<br>Kisker, H.-W./<br>Pudlatz, H./<br>Richter, G.  |
| 320245   | Clusteranalyse-Verfahren und Software zur<br>automatischen Klassifikation<br>mo 15-17; Hörsaal: M6, Beginn: 28.4.1986 | Steinhausen, D.   |
| 320250   | Forschungskolloquium "Computermusik"<br>mi 8.30-10; Hörsaal: Bibliothek des Rechenzentrums,<br>Beginn: 30.4.1986      | Achilles, A./<br>Brockhoff, M.E.                            |
| 320264   | Kolloquium über Themen der Informatik<br>fr 15-17; Hörsaal: M6  | Held, W. und die<br>wiss. Mitarbeiter<br>des Rechenzentrums |
| 320279   | Anleitung zum Einsatz der EDV bei<br>wissenschaftlichen Arbeiten  | die wiss. Mitarbeiter<br>des Rechenzentrums                 |
| (056915) | Einführung in INTERLISP<br>mi 13-15; Hörsaal: M6, Beginn: 30.6.1986<br>und 2 Std. Übungen n.V.                        | Göttsche, H.  |

## RUM-Graphik

### VIEW3D - räumliche Darstellung von 3D-Objekten

von

D. Eckey

Mit dem Programm VIEW3D ist es möglich, 3-dimensionale Objekte (Gittermodelle oder Punktwolken) räumlich darzustellen und von allen Seiten zu betrachten.

Für die Arbeit mit VIEW3D benötigt man:

- eine Rechennummer für die C-Maschine,
- eine Rot-Grün-Brille und
- eine oder zwei CMS-Dateien mit Dateityp LINES oder POINTS.

Die Nummer für die C-Maschine ist notwendig, weil die Bilder auf dem Graphischen Bildschirmsystem IBM 5080 erzeugt werden und dieses Gerät an die C-Maschine angeschlossen ist.

Die Rot-Grün-Brille wird benötigt, um den räumlichen Effekt zu erreichen: Es werden zwei perspektivische Projektionen durchgeführt, deren Projektionspunkte dicht nebeneinander liegen. Dabei wird eine Projektion in rot, die andere in grün auf dem Bildschirm dargestellt. Betrachtet man dieses Bild durch eine Rot-Grün-Brille, so sieht jedes Auge nur die Projektion, die es auch sehen würde, wenn man einen realen Körper vor sich hätte.

Die CMS-Dateien mit Dateityp LINES (bzw. POINTS) enthalten die Koordinaten, die das darzustellende Objekt beschreiben. Diese Koordinaten beziehen sich auf ein rechtshändiges kartesisches Koordinatensystem, wobei die x-Achse von links nach rechts, die y-Achse von unten nach oben und die z-Achse von hinten nach vorne verläuft. Ein Record dieser Dateien hat folgende Form:

$$X_1 Y_1 Z_1 X_2 Y_2 Z_2 \dots X_n Y_n Z_n$$

Dabei bezeichnet  $X_i Y_i Z_i$  den i-ten Punkt in diesem Record.  $X_i$ ,  $Y_i$  und  $Z_i$  sind 16-Bit-Zahlen, die in dem Intervall (-2048, 2048) liegen sollten. Wenn die Datei den Dateityp LINES hat, werden jeweils alle Punkte, die in einem Record stehen, durch einen

Polygonzug miteinander verbunden. Hat die Datei den Typ POINTS, werden nur die Punkte selbst dargestellt.

Beispiel zur Erstellung eines solchen Records in PL/I:

```

DECLARE
  N          BIN FIXED,
  OUT        FILE RECORD OUTPUT
             ENVIRONMENT
             (V BLKSIZE (32000)),
  1 RECORD (*) CONTROLLED,
  2 X        BIN FIXED,
  2 Y        BIN FIXED,
  2 Z        BIN FIXED;
. . .
/* Es seien N Punkte: */
ALLOCATE RECORD (N);
/* An dieser Stelle sind die
Koordinaten einzutragen: */
. . .
WRITE FILE(OUT) FROM (RECORD);
FREE RECORD;
    
```

Vor dem Programmstart ist das CMS-Kommando FILEDEF OUT DISK fn LINES A (PERM einzugeben.

In FORTRAN wird die gleiche Datei folgendermaßen erzeugt:

```

* Ein Linienzug habe z.B. maximal
* 100 Punkte:
      INTEGER*2 X (100), Y (100),
              Z (100)
      INTEGER  I, N
. . .
    
```

```

* Es seien N Punkte.
* An dieser Stelle sind die Arrays
* X, Y und Z zu füllen:
. . .
    
```

```

WRITE (8) (X(I),Y(I),Z(I),
           I=1,N)
    
```

Vor dem Programmstart ist das CMS-Kommando FILEDEF 8 DISK fn LINES A (PERM RECFM V BLKSIZE 32000 einzugeben.

Bevor man VIEW3D aufrufen kann, muß man die IBM 5080 an die eigene virtuelle Maschine anschließen. Dazu trägt man im Monitorbild der IBM 5080 seine Benutzerkennung ein. Sollte kein Bild zu sehen sein, ENTER drücken.

Der Aufruf von VIEW3D geschieht folgendermaßen:

```
VIEW3D fn1 [ fn2 ]
```

Dabei sind *fn1* und *fn2* die Namen von Dateien mit Typ LINES oder POINTS, die so aufgebaut sind wie oben beschrieben. Werden zwei Dateien angegeben, so werden die Objekte, die dadurch beschrieben sind, gleichzeitig dargestellt. Man kann dann mit Hilfe der Einstellräder 4 und 8 (s.u.) die Helligkeit der beiden Objekte getrennt einstellen. Beispielsweise kann man mit *fn2* die Koordinatenachsen definieren, damit man beim Drehen des Objekts in *fn1* nicht die Orientierung verliert. Falls die Koordinatenachsen den Eindruck stören, kann man sie einfach ausblenden.

Das Programm sucht jeweils zuerst nach einer Datei 'fn LINES \*'. Wird diese nicht gefunden, sucht es nach 'fn POINTS \*'. Will man also eine Punktwolke darstellen ('fn POINTS \*'), so darf keine Datei existieren, die den gleichen Namen und den Dateityp LINES hat.

Mit Hilfe der Einstellräder, die neben dem 5080-Bildschirm stehen, kann man die Darstellung verändern. Man hat dabei folgende Möglichkeiten:

**Rad 1** Drehwinkel um die z-Achse des Bildschirms ("von hinten nach vorne")

**Rad 5** Drehwinkel um die x-Achse des Bildschirms ("von links nach rechts")

Durch die Größe dieser Winkel wird die ursprüngliche y-Achse ("von unten nach oben") in eine bestimmte Achse (im folgenden "Rotationsachse" genannt) transformiert. Um diese Achse kann man das Objekt mit den Rädern 2 und 6 drehen:

**Rad 2** Drehwinkel um die Rotationsachse

**Rad 6** Rotationsgeschwindigkeit um die Rotationsachse. Wenn man hier einen Wert  $\Delta\omega$  ungleich 0 einstellt, dreht sich das Objekt ständig mit der angegebenen Geschwindigkeit (d.h. bei jedem neuen Bild wird der Drehwinkel um die Rotationsachse um  $\Delta\omega$  verändert). Der Eindruck einer kontinuierlichen Bewegung entsteht allerdings nur, wenn die Anzahl der zu transformierenden Punkte klein genug ist. Man

kann die Rotationsgeschwindigkeit auf Null zurücksetzen, indem man die Funktionstaste 3 am Cursor des Grafik-Tablets drückt.

Die aktuellen Werte der Drehwinkel werden ständig am unteren Rand des Bildschirms eingeblendet.

**Rad 3** Verschieben der vorderen und hinteren Clip-Ebenen. Drehung des Rades im Uhrzeigersinn verschiebt die Ebenen nach hinten, Drehung im Gegenuhrzeigersinn nach vorne. In der linken unteren Ecke des Bildschirms wird eingeblendet, welche Clip-Ebene man gerade verschieben kann. Mit den Funktionstasten 1 und 2 am Cursor kann man zwischen beiden Ebenen umschalten.

**Rad 7** Zoom. Mit diesem Rad kann man das Objekt innerhalb bestimmter Grenzen vergrößern oder verkleinern.

**Rad 4** Intensität des Objekts, das durch die erste Datei im Aufruf definiert ist.

**Rad 8** Intensität des Objekts, das durch die zweite Datei im Aufruf definiert ist. Wurde nur eine Datei angegeben, hat dieses Rad keine Funktion.

Die Bedeutung der einzelnen Räder wird auch auf dem 3270-Terminal ausgegeben, von dem aus das Programm gestartet worden ist (i.a. das Terminal, das neben der 5080 steht).

## Projizierte Flächen mit verdeckten Linien

von

*R. Schulte*

Bei der Analyse von Daten ist es sicherlich immer von Vorteil, auch über graphisches Anschauungsmaterial zu verfügen. Mit der PL/I-Prozedur GENERATE\_SURFACE ist nun mit Mitteln des GKS die Möglichkeit geschaffen worden, numerische Daten, die in Form einer Matrix vorliegen, als 3-dimensionale Gitterfläche darzustellen.

Dazu werden zunächst die Daten der Matrix als Funktionswerte einer Abbildung  $z = f(x,y)$  auf einem rechteckigen äquidistanten Gitter in der Ebene interpretiert. Die Gitterfläche kann dann als Interpolation des Funktionsgraphen aufgefaßt werden. Um diese Gitterfläche auf ein 2-dimensionales Medium (Bildschirm, Plotter) abbilden zu können,

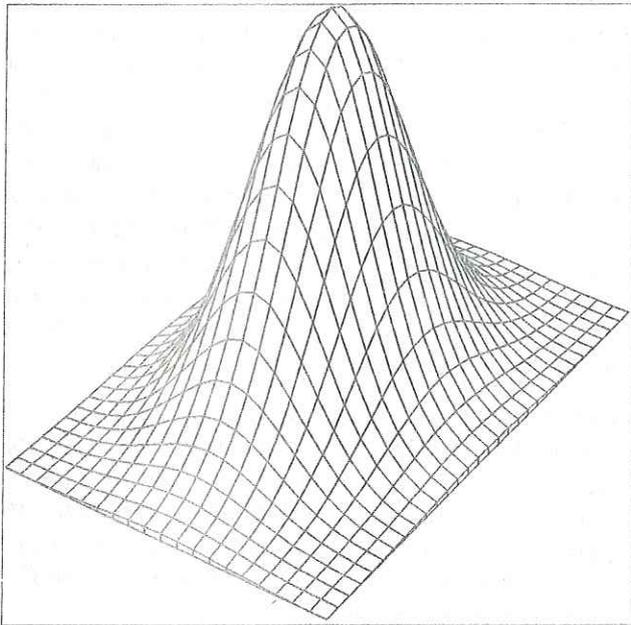


Abbildung 1.

wird nun in der Prozedur eine Zentralprojektion durchgeführt. Daran anschließend findet die Elimination der verdeckten Linien statt, und das Ergebnis ist schließlich ein Bild, das sich mit einer Fotografie des Gebildes vergleichen läßt.

Wie auch bei GENERATE\_BARCHART und den anderen RUM/GKS-Routinen der Präsentationsgra-

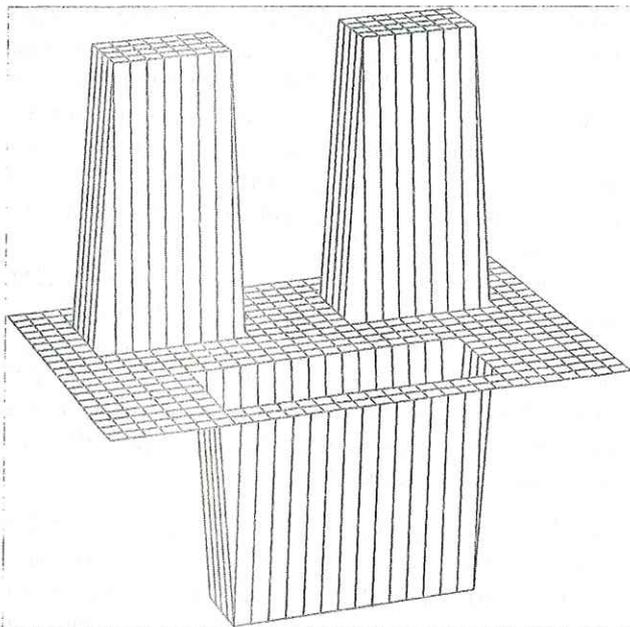


Abbildung 2.

phik gibt man mit SET\_VIEWPORT den Bereich auf dem Bildschirm oder Papier an, in dem das Bild gezeichnet werden soll. Der Aufruf der Prozedur geschieht dann durch:

```
CALL GENERATE_SURFACE
      (TREATMENT, MATRIX);
```

Der zweite Parameter (vom Typ (\*,\*) FLOAT) dient zur Aufnahme der bereits oben erwähnten Datenmatrix. Mit Hilfe des ersten Parameters (vom Typ CHAR (\*) VAR) dagegen kann, falls dieser die Zeichen 'D' oder 'O' enthält, Einfluß auf bestimmte implizite Setzungen genommen werden.

So wird erstens normalerweise die untere Seite der Gitterfläche durch einen Sockel verdeckt (siehe Abb. 1, deutlicher in Abb. 3). Enthält TREATMENT aber ein 'D' (für double-sided), so werden beide Seiten gezeichnet (siehe Abb. 2). Die Laufzeit des Programms verdoppelt sich dabei allerdings auch.

Zweitens verläuft implizit die Blickrichtung des Beobachters horizontal, so daß senkrechte Linien auch nach der Projektion senkrecht verlaufen. Bei einer natürlichen Betrachtungsweise dagegen ist der Blick auf den Mittelpunkt des Gebildes gerichtet. Dies entspricht einer Neigung des Blickwinkels und wird durch die Angabe von 'O' (für oblique view) erreicht. Der Unterschied wird bei Betrachtung von Abb. 3 klar, in der die eigentlich parallelen senkrechten Linien deutlich auseinanderlaufen.

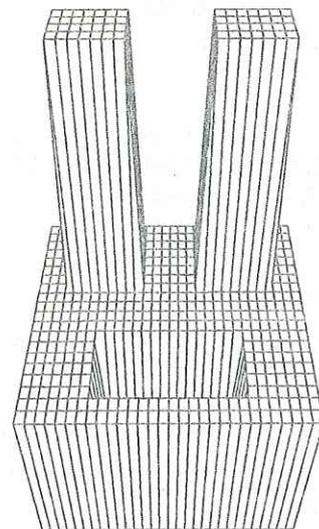


Abbildung 3.

Schließlich wird innerhalb der Prozedur ein Window gesetzt, das normalerweise aus den Extremwerten der projizierten Grundfläche in x-Richtung und den Extremwerten aller Projektionspunkte in y-Richtung gebildet wird. Außerdem wird die Ausdehnung der Grundfläche allein durch die Dimensionen der Datenmatrix bestimmt. Ist über die Größe der Grundfläche weiter nichts bekannt (z.B. bei empirischen Daten), so ist diese Vorgehensweise auch sinnvoll. Hat der Benutzer jedoch konkrete Vorstellungen über die Ausmaße der Grundfläche (z.B. bei Funktionsgraphen), und sollen die richtigen Proportionen (Breite, Tiefe und Höhe) des Urbildes erhalten bleiben, so kann er dies dem Programm mitteilen, indem er beim Aufruf von GENERATE\_SURFACE zwei weitere Argumente anfügt:

```
CALL GENERATE_SURFACE
(TREATMENT, MATRIX, X_SIZE, Y_SIZE);
```

In diesem Fall werden für die Ausdehnung der Grundfläche in x- und y-Richtung die Werte X\_SIZE und Y\_SIZE benutzt. So stimmen auch erst in Abb. 3 die Verhältnisse zwischen Höhe, Breite und Tiefe mit denen des wirklichen Funktionsgraphen überein.

Richtig interessant wird GENERATE\_SURFACE erst durch die Möglichkeit, auch den Projektionspunkt selbst zu bestimmen. Hierdurch kann die Gitterfläche praktisch von allen Seiten aus betrachtet werden. Vorgesehen hierfür ist der Befehl:

```
CALL DEFINE_SURFACE
(DISTANCE,
ALPHA,
BETA,
INTERPOLATIONSTEPS_BETWEEN_ROWS,
INTERPOLATIONSTEPS_BETWEEN_COLS,
COLOUR_INDEX_ABOVE,
COLOUR_INDEX_BELOW,
LINETYPE_ABOVE,
LINETYPE_BELOW);
```

Die lange Parameterliste wirkt sicherlich auf den ersten Blick ein wenig abschreckend, zur Änderung des Projektionspunktes benötigt man jedoch nur die ersten drei: DISTANCE, ALPHA und BETA (alle vom Typ FLOAT). Sie geben die räumlichen Polarkoordinaten des Projektionspunktes an. Der Ursprung des Koordinatensystems befindet sich dabei im Mittelpunkt des Quaders, der die gesamte Gitterfläche gerade noch einhüllt. Die Polarkoordinaten wurden gewählt, weil sich mit ihrer Hilfe Drehungen sehr leicht nachvollziehen lassen.

Die Größe DISTANCE legt den Abstand des Beobachters vom Quader-Mittelpunkt fest. Und zwar wird implizit als Abstand die Länge der Raumdiagonalen des Quaders benutzt. Eine angegebene DISTANCE bezieht sich auf eine halbe Raumdiagonalenlänge, so daß man sich bei DISTANCE > 1 stets außerhalb des Gebildes befindet. Enthält der Parameter TREATMENT aber ein 'W', das heißt, der Benutzer macht selbst Angaben über die Größe der Grundfläche, so wird der Wert von DISTANCE unmittelbar als Abstand angenommen.

Die Variable ALPHA entspricht dem Winkel zwischen dem Vektor zum Projektionspunkt und der (x,y)-Ebene. Gilt  $0 < \text{ALPHA} < 180$ , so befindet man sich oberhalb, andernfalls unterhalb der Ebene. Liegt ALPHA zwischen 180 und 360 Grad, so entsteht ein Bild, welches das Gebilde auf den Kopf gestellt wiedergibt.

Jetzt muß nur noch der Lotpunkt des Projektionspunktes in der (x,y)-Ebene festgelegt werden. Dies geschieht durch den Winkel BETA, der die Abweichung des Lotpunktes von der positiven x-Achse angibt, und zwar in Richtung auf die positive y-Achse. Eine Vergrößerung von BETA führt demnach zu einer Drehung des Gebildes nach links.

Die Drehungen lassen sich beim Vergleich der drei Abbildungen sicherlich klar erkennen. Anzumerken ist hier noch, daß bei einer Zentralprojektion zu den Punkten der Ebene, die parallel zur Projektionsebene durch den Projektionspunkt verläuft, keine Bildpunkte existieren. Schneidet also diese sogenannte Verschwindungsebene die Gitterfläche, so läßt sich die Projektion nicht sinnvoll durchführen. Bei horizontaler Blickrichtung tritt dies immer dann auf, wenn sich der Projektionspunkt genau ober- bzw. unterhalb der Grundfläche befindet, bei geneigter Projektion dagegen nur, falls der Abstand vom Mittelpunkt zu klein gewählt wurde. Zwar erzeugt die geneigte Projektion im ersten Fall sinnvolle Bilder, aber der verwendete Eliminationsalgorithmus ist nicht mehr anwendbar. Sollte eine dieser Ausnahmen eintreten, so erhält der Benutzer über die Standardausgabedatei SYSPRINT eine entsprechende Meldung.

Die beiden folgenden Parameter (vom Typ BIN FIXED) sollen es auch den Benutzern, die nur über wenig Ausgangsdaten verfügen, ermöglichen, ein möglichst glattes Bild zu erhalten. Dazu wird, um die großen Lücken im Gitter aufzufüllen, eine Polynominterpolation 3. Grades durchgeführt.

Anzugeben sind die Anzahl der Interpolationsschritte pro Stützstelle in x- und in y-Richtung. Soll nicht interpoliert werden, so sind beide Parameter auf 1 zu setzen. Notwendig für die Interpolation sind mindestens 4 Werte pro Richtung. Es sei darauf hingewiesen, daß zuviele Interpolationsschritte auch zu einer Verfälschung des Bildes führen können.

Die letzten vier Parameter (ebenfalls vom Typ BIN FIXED) entsprechen den vom GKS bekannten Angaben für Farbe und Linientyp der Polygonzüge, jeweils getrennt für Ober- und Unterseite.

Mit Hilfe des folgenden Beispielprogramms, das die drei hier gezeigten Abbildungen erzeugt, soll zum Schluß die Benutzung der verschiedenen Parameter noch einmal verdeutlicht werden. Man beachte die %INCLUDE-Anweisung.

```

FLAECHEH:
PROCEDURE OPTIONS (MAIN);

CALL START_PLOT;
CALL SET_PLOT_LIMITS (200, 100, 3); /* 3 BILDER */
CALL SET_VIEWPORT (1, 0, 0.3, 0, 0.3);

/* ERZEUGUNG DER DATENMATRIX DES ERSTEN BILDES: */
X = XMIN;
DO I = 0 TO 30;
  Y = YMIN;
  DO J = 0 TO 20;
    MATRIX(I,J) = EXP (-X**2 - Y**2);
    Y = Y + (YMAX - YMIN) / 20;
  END;
  X = X + (XMAX - XMIN) / 30;
END;

/* ERSTES BILD: GAUSS-GLOCKE */
CALL GENERATE_SURFACE ('', MATRIX);

/* ERZEUGUNG DER DATENMATRIX FUER DIE WEITEREN BILDER: */
X = XMIN;
DO I = 0 TO 30;
  Y = YMIN;
  DO J = 0 TO 20;
    SELECT;
      WHEN (((-1.5 <= X) & (X <= -0.5) | ( 0.5 <= X) & ( X <= 1.5))
        & ((-1.5 <= Y) & (Y <= -0.5)))
        MATRIX(I,J) = 4;
      WHEN ((-1 <= X) & (X <= 1) & (0.5 <= Y) & (Y <= 1.5))
        MATRIX(I,J) = -3;
      OTHERWISE
        MATRIX(I,J) = 0;
    END;
    Y = Y + (YMAX - YMIN) / 20;
  END;
  X = X + (XMAX - XMIN) / 30;
END;

```

```

/* ZWEITES BILD: */
CALL CLEAR_PLOT;
CALL DEFINE_SURFACE (10, 20, 70, 1, 1, 1, 1, 1, 1);
CALL GENERATE_SURFACE ('D', MATRIX);

/* DRITTES BILD: */
CALL CLEAR_PLOT;
CALL DEFINE_SURFACE (10, 45, 90, 1, 1, 1, 1, 1, 1);
CALL GENERATE_SURFACE ('O', MATRIX, XMAX-XMIN, YMAX-YMIN);

CALL END_PLOT;

DECLARE
  MATRIX (0:30,0:20) FLOAT,
  XMIN   FLOAT INIT (-2),
  XMAX   FLOAT INIT (2),
  YMIN   FLOAT INIT (-2),
  YMAX   FLOAT INIT (2),
  (X,Y)  FLOAT,
  (I,J)  BIN FIXED,
  EXP    BUILTIN;

%INCLUDE GKS, GKSPLLOT, GKSSURF;

END FLAECHEH;

```

## Farbige Rasterbilder

von

*E. Sturm*

In vielen wissenschaftlichen Bereichen dienen mehr oder weniger grob gerasterte Bilder - oft auch mit willkürlich festgelegten "falschen" Farben - der Veranschaulichung von gemessenen Werten. Solche Matrixbilder können jetzt auch auf dem Farbraster-Bildschirmgerät IBM 5080 dargestellt werden, ohne daß der Benutzer Kenntnisse in der graphischen Programmierung benötigt. Die Farben können dann interaktiv mit Hilfe von XEDIT geändert werden.

Die Bildmatrix kann maximal 800 Zeilen und maximal 528 Spalten besitzen. Für jedes Feld der Matrix muß man eine von maximal 256 Farben festlegen.

In der Matrix-Datei ist pro Matrixzeile ein Record gespeichert. Im Record gibt es für jedes Feld einen Farbindex. Diese Zahl ist, um Speicherplatz zu sparen, in einem Byte verschlüsselt (CHAR (1) in PL/I, CHARACTER\*1 in FORTRAN), so daß

Farbindizes zwischen 0 und 255 möglich sind. Die Matrix-Datei ist also binär codiert und kann nicht mit XEDIT in der üblichen Weise geändert werden. Zur Erzeugung dieser Datei verwendet man die im folgenden angegebenen Funktionen:

```

/* Zuweisung einer Zahl auf
   ein Zeichen in PL/I: */
DCL C CHAR (1), N BIN FIXED;
/* Wertzuweisung auf N */
UNSPEC (C) = BINARY (N, 8);

```

```

* Zuweisung einer Zahl auf
* ein Zeichen in FORTRAN:
CHARACTER*1 C
INTEGER N
* Wertzuweisung auf N
C = CHAR (N)

```

In der Farb-Datei ist für jeden vorkommenden Farbindex gespeichert, welche Farbe gemeint ist. In jedem Record müssen vier Zahlen stehen: der Farbindex (zwischen 0 und 255) und die drei Intensitäten für Rot, Grün und Blau (zwischen 0 und 1) - dahinter evtl. als Kommentar der umgangssprachliche Name der Farbe:

0	1	1	1	weiß
1	0	0	0	schwarz
2	1	1	0	gelb
3	0.2	0.1	0	dunkelbraun

Diese Zahlen können auf beliebigen Positionen im Record stehen (wie für GET LIST in PL/I bzw. READ(5,\*) in FORTRAN). Die Farb-Datei kann mit XEDIT nach Belieben verändert werden.

Im CMS gibt man für das Bildschirmgerät IBM 5080 das COLPLOT-Kommando ein. Z.B. benötigt man bei der Eingabe von

```
COLPLOT BILD (L=30,C=40,S=5
```

die Dateien 'BILD MAT \*\*' und 'BILD COL \*\*'. Sie müssen der obigen Beschreibung entsprechen. Hinter der Klammer muß man die Abmessungen des Bildes und einer Bildzelle angeben: L steht für 'linecount' (Anzahl der Zeilen), C für 'columncount' (Anzahl der Spalten) und S für 'size' (Anzahl der

Bildpunkte pro Matrixzellenbreite). 'S=5' hieße also, daß eine Matrixzelle aus 5 mal 5 Bildpunkten bestehen soll. Das Produkt aus L und S darf nicht größer als 800, das Produkt aus C und S nicht größer als 528 sein.

Man beachte, daß das COLPLOT-Kommando nur auf der C-Maschine implementiert ist und daß vor seinem Aufruf die IBM 5080 an die eigene virtuelle Maschine angeschlossen werden muß (Eingabe der eigenen Benutzernummer auf dem Monitorbild der IBM 5080). Außerdem ist entweder mit LOGOFF die Sitzung zu beenden oder vor DISCONN der Befehl 'DET 505' zu geben, damit der nächste Benutzer ohne Operator-Maßnahmen Zugriff auf die IBM 5080 hat.

Wir bemühen uns, den Benutzern in Zukunft die Ausgabe farbiger Rasterbilder auf Papier anzubieten.

## RUM-Tutorial

### Textverarbeitung mit Mikrorechnern

von

H. Kamp

In den vergangenen Jahren ist die Zahl der an unserer Universität vorhandenen Mikrorechner ständig gestiegen; besonders das CIP-Programm (vgl. *inforum* Nr. 4/1985 und Nr. 1/1986) hat mittel- und unmittelbar einen kräftigen Beschaffungsschub ausgelöst. In vielen Instituten war der Wunsch, über eine effiziente Textverarbeitung im eigenen Hause verfügen zu können, die Ursache für die Beschaffung eines Mikrorechners nebst entsprechender Peripherie und Software. Die Vorteile einer rechnergestützten Textverarbeitung haben sich inzwischen herumgesprochen; sie sind im wesentlichen im Artikel "Textverarbeitung mit WordStar" (*inforum* Nr. 1/1984) zusammengefaßt und brauchen daher hier nicht wiederholt zu werden. Auch für eine Erläuterung der Grundlagen und Begriffe sei auf den genannten Beitrag verwiesen.

Schwierig ist dagegen die Frage, welche Textverarbeitungssoftware für welchen Zweck eingesetzt werden soll. Neben dem alteingeführten WordStar bietet eine Reihe neuerer Produkte — Word, Papyrus, WordPerfect u.v.a. — ihre Dienste an. Im Rahmen des CIP-Programms wurde, quasi als Grundausstattung der Pools, WordStar beschafft; zusätzlich wurde jedoch eine größere Anzahl von Lizenzen für WordPerfect sowie eine geringere Zahl von Lizenzen für Word erworben.

Als Vorteil von WordStar gilt seine leichte Erlern- und Benutzbarkeit; vorteilhaft ist sicher auch der z. Zt. recht geringe Anschaffungspreis. Als Manko empfanden vor allem Benutzer aus den geisteswissenschaftlichen Fachbereichen, daß eine automatische Verwaltung von Fußnoten nicht unterstützt ist. Für diesen Zweck ist zwar Footnote als (gemessen am Preis/Leistungsverhältnis von WordStar relativ teures) Ergänzungsprogramm erhältlich, das jedoch nicht in WordStar integriert und daher umständlich in der Handhabung ist. Die Umstellung eines Textstücks etwa, das auf eine Fußnote Bezug nimmt, gestaltet sich zu einer in mehreren Schritten durchzuführenden Aktion, die mühsam und zeitaufwendig ist. Auch ist es in der Regel nötig, nach der Bearbeitung des Textes

durch Footnote nochmals WordStar aufzurufen, um eine Endredaktion des Textes vorzunehmen.

In WordPerfect oder Word ist eine Textumstellung dagegen völlig problemlos; der Autor plaziert den Text an die gewünschte Stelle, eventuell dazu gehörige Fußnoten werden automatisch mit umgestellt, die Neunumerierung der Fußnoten besorgt das System. Ein anderer Kritikpunkt an WordStar — dieses Mal eher aus der Sicht der naturwissenschaftlichen Fachbereiche — ist, daß das System dem Benutzer keine Sonderzeichen zur Formelschreibung anbietet. WordPerfect oder Word stellen dagegen, je nach Rechner typ, einen Grundstock zur Verfügung, der Zeichen für 'Integral', 'Wurzel', 'Unendlich' etc. sowie eine Reihe griechischer Zeichen umfaßt.

Vergleicht man die einzelnen Systemfunktionen, so darf man sagen, daß die von WordStar aufgestellten Standards von WordPerfect oder Word in den meisten Fällen erreicht und in vielen auch übertroffen werden (eine Zitierung dieses Satzes ist nur in seiner Ganzheit zulässig). Die Basisfunktionen zum Eingeben, Ändern, Löschen oder Kopieren von Text sind in den drei genannten Systemen in Handhabung und Auswirkung ähnlich; alle bieten inzwischen auch eine brauchbare deutsche Dokumentation sowie am Bildschirm abrufbare Hilfen an. Der Benutzer mag daher entscheiden, ob ihm für seine Textverarbeitung (Fließ- oder Korrespondenztext) die Mittel von WordStar ausreichen oder ob er für komplexere Texte (mit Fußnoten und evtl. Formeln) auf WordPerfect, Word etc. zurückgreifen möchte.

WordPerfect bietet zudem eine Reihe zusätzlicher Funktionen an, die dem Benutzer weitere Bearbeitungsmöglichkeiten einer gespeicherten Datei eröffnen; genannt seien hier:

- Zum Lieferumfang des Systems gehört ein Lexikon, in dem etwa 80.000 Wortformen des Deutschen gespeichert sind; der Benutzer kann noch etwa 10.000 Wortformen eigener Wahl hinzufügen. Eine Programmfunktion erlaubt eine Rechtschreibkontrolle gespeicherter Dokumente mit Hilfe des Lexikons. Ein Praxistest im Fachbereich Jura ergab, daß diese Funktion gute Ergebnisse erbrachte, nachdem das fachspezifische Vokabular zusätzlich gespeichert war.

- WordPerfect erlaubt dem Benutzer die Vereinbarung eigener Makros, in denen Text oder Befehlsfolgen abgelegt und beliebig oft aufgerufen werden können. Sollen etwa die Fußnoten eines Textes als Endnoten am Schluß des Dokuments erscheinen, kann die erforderliche Befehlsfolge zum Transfer einer Fußnote als Makro abgelegt und für den automatischen Transfer aller Fußnoten aufgerufen werden. Ein Makro für den umgekehrten Weg ist ebenfalls leicht formulierbar.
- Eine Sortier-/Selektierkomponente ermöglicht die Sortierung einer Datei nach bis zu neun Kriterien; zusätzlich können Merkmale angegeben werden, die ein Herausfiltern aller Datensätze gestatten, die die gesuchten Merkmale aufweisen. Bei der Sortierung kann der Benutzer u.a. angeben, ob seine Daten eine Zeilen- oder Absatzstruktur haben; die letztgenannte Möglichkeit kann z.B. genutzt werden, um die Titel einer Bibliographie, die in beliebiger Reihenfolge eingegeben wurden, alphabetisch nach Verfassernamen umzusortieren.
- Der Benutzer verfügt über einen Modus, in dem ihm die Grundrechenarten zur Verfügung stehen. Ein Textverarbeitungsprogramm sollte zwar nicht als Instrument zur Tabellenkalkulation mißverstanden werden, es gibt jedoch sicherlich Beispiele, bei denen es ärgerlich ist, wenn im Rechner (!) gespeicherte Daten zur Summierung, Prozentermittlung o.ä. nochmals in einen Taschenrechner getippt werden müssen.

WordPerfect und Word verfügen zwar über eine Grundausstattung mathematischer Zeichen und griechischer Buchstaben, die jedoch auf der einen Seite für sehr komplexe Formeln nicht ausreichen und auf der anderen Seite auch dem Graecisten nicht genügen werden. Für den erstgenannten Fall sei auf PC-TEX verwiesen, das speziell zum Satz mathematischer Formeln entworfen wurde, darüber hinaus aber auch über einen großen Zeichenvorrat, einschließlich vieler Sonderzeichen und Diacritica

verfügt. Es darf jedoch nicht verschwiegen werden, daß PC-TEX zwar ein sehr leistungsfähiges, aber auch ein sehr aufwendiges Instrument ist; es benötigt an Hauptspeicher mindestens 512 KB sowie mindestens eine 10 MB Festplatte. Der mit PC-TEX gesetzte Text ist äußerst akkurat, die für Formatierung und Druck benötigte Zeit — selbstverständlich abhängig vom Schwierigkeitsgrad des jeweiligen Textes — jedoch auch sehr hoch. Es sei auch darauf hingewiesen, daß sich PC-TEX als ein Satzsystem, jedoch nicht als Textverarbeitungssystem versteht. Komponenten zum Edieren eines erfaßten Textes, wie etwa Suchen und Tauschen von Wörtern, Manipulationen mit vom Benutzer gekennzeichneten Textblöcken etc., gehören nicht zum Bestandteil des Systems.

Ist die Verfügbarkeit fremdsprachiger Alphabete, z.B. Griechisch oder Kyrillisch, eine Voraussetzung für den Einsatz eines Mikrorechners, könnten Produkte wie etwa Scientex, Academic Font oder T<sup>3</sup> eingesetzt werden. Da ausführlichere Tests unsererseits für diese Systeme noch ausstehen, eine Bitte an die *inforum*-Leser: Sofern Sie bereits Erfahrungen mit einem der genannten oder im Leistungsumfang ähnlichen Systeme haben, wären wir für einen Bericht sowie einige Referenzseiten dankbar. Mit der gleichen Bitte sei der Wunsch eines Musikwissenschaftlers weitergegeben, der im selben Dokument Text und Notenschrift mischen möchte — hat einer unserer Leser einen Tip?

Die vorangegangenen Überlegungen sollten auch zeigen, daß aufgrund des breiten Anwendungsspektrums an einer Universität eine pauschale Empfehlung für ein Textsystem, das alle Bereiche gleich gut und zu tragbarem Preis abdecken kann, wohl kaum gegeben werden kann. Das Rechenzentrum berät seine Benutzer, um im jeweiligen Fall eine den Anforderungen entsprechende Lösung zu finden. Für WordStar und WordPerfect steht eine größere Zahl von Mustertexten zur Verfügung, die den Benutzer anhand von Beispielen in die einzelnen Systemfunktionen einführen. Interessenten wenden sich bitte an den Verfasser.

## REDUCE - Ein System zur algebraischen Formelmanipulation

von

B. Süsselbeck

Wenn man heute davon spricht, eine Formel oder ein Integral mit Hilfe eines Computers auszurechnen, so ist in der Regel folgendes gemeint: Für bestimmte Variablen (z.B. Integrationsgrenzen oder Argumente von Funktionen) werden Zahlen eingesetzt; aus diesen Zahlen ermittelt der Rechner dann anhand bestimmter Algorithmen als Ergebnis wieder eine (oder mehrere) Zahlen, z.B. ergibt der arithmetische Ausdruck (in FORTRAN)

$$Y = A**2 + 2*A*B + B**2$$

mit  $A = 1.0$  und  $B = 2.0$  für  $Y$  den Wert  $9.0$ .

Eine andere Art des "Rechnens" mit Formeln, besteht darin, die Formeln selbst zu manipulieren, also z.B. aus  $A**2 + 2*A*B + B**2$  den formal gleichwertigen Ausdruck  $(A + B)**2$  zu erhalten. Hierbei sind die zu manipulierenden Objekte also die Formeln selbst und nicht die Zahlenwerte.

Ein anderes Beispiel wäre die Berechnung der Stammfunktion einer vorgegebenen Funktion (im Gegensatz zur numerischen Berechnung eines Integrals).

Während die Möglichkeit, mit Hilfe eines Computers numerische Rechnungen durchzuführen, als selbstverständlich gilt, ist die Tatsache, daß es auch Programme zur sogenannten algebraischen Formelmanipulation gibt, weit weniger bekannt, zumal derartige Konzepte in den herkömmlichen Programmiersprachen, wie FORTRAN, Pascal, PL/I etc. nicht enthalten sind.

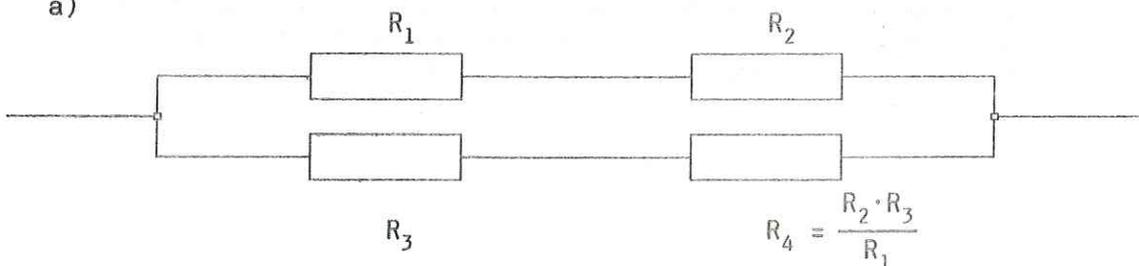
Deshalb soll in diesem Artikel kurz das System REDUCE vorgestellt werden, das von Anthony C. Hearn (Rand Corporation) in den USA basierend auf der Sprache LISP entwickelt wurde.

Die Möglichkeiten von REDUCE sollen zunächst an zwei Beispielen diskutiert werden, die nicht zur Demonstration konstruiert wurden, sondern bei denen REDUCE wirklich zur Lösung eines Problems eingesetzt wurde.

Als Beispiel soll die sogenannte Wheatstone'sche Brückenschaltung von vier Widerständen  $R_1$ ,  $R_2$ ,  $R_3$ ,  $R_4$  betrachtet werden mit

$$R_4 = \frac{R_2 \cdot R_3}{R_1}$$

a)



Der Gesamtwiderstand  $R_{\text{ges}}$  wird nach folgender Formel berechnet:

$$R_{\text{ges}} = \frac{(R_1 + R_2) \cdot (R_3 + \frac{R_2 \cdot R_3}{R_1})}{R_1 + R_2 + R_3 + \frac{R_2 \cdot R_3}{R_1}}$$

Mit einer solchen Formel wird man sich im allgemeinen aber nicht zufrieden geben, sondern möglichst versuchen, sie zu vereinfachen. (Denn bevor man eine Formel zur *numerischen* Auswertung programmiert, sollte sie *algebraisch* einfach sein; dies spart Speicherplatz und Rechenzeit, trägt aber auch zur numerischen Stabilität bei.)

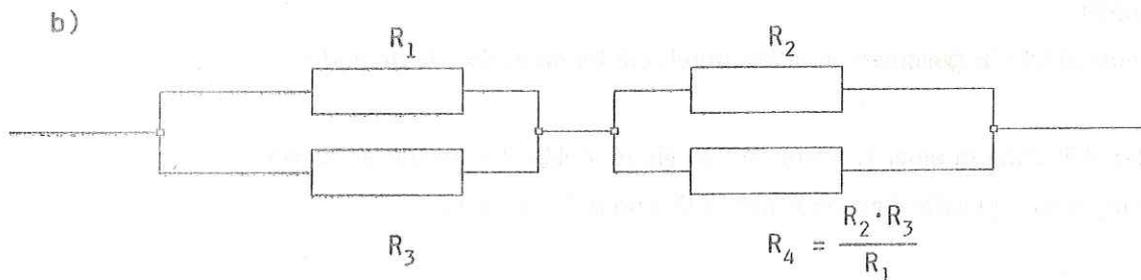
Für  $R_{\text{ges}}$  ergibt sich nach kurzer "manueller" Rechnung, bei der die üblichen Standardtechniken wie

- Bestimmung des Hauptnenners,
- Ausklammern,
- Kürzen

verwendet werden, folgender Ausdruck:

$$R_{\text{ges}} = R_3 \cdot \frac{R_1 + R_2}{R_1 + R_3}$$

Modifiziert man die Schaltung a) zu



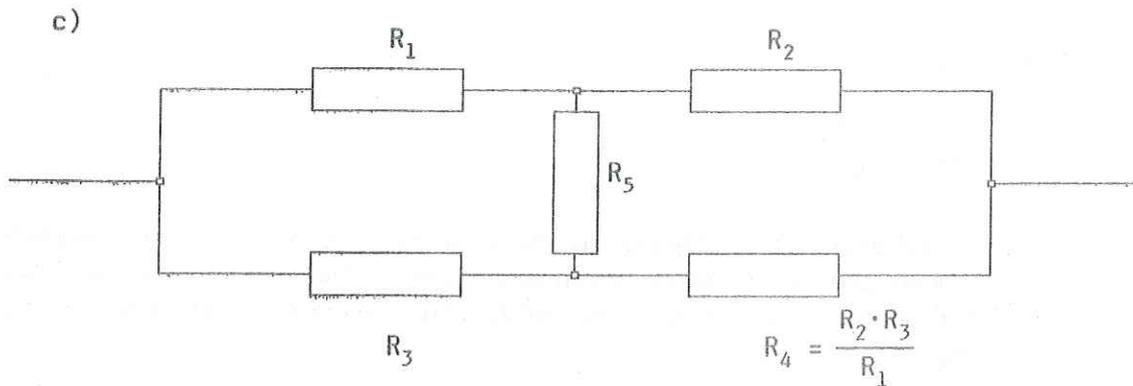
so erhält man:

$$R_{\text{ges}} = \frac{R_1 \cdot R_3}{R_1 + R_3} + \frac{R_2 \cdot \frac{R_2 \cdot R_3}{R_1}}{R_2 + \frac{R_2 \cdot R_3}{R_1}}$$

Hier ergibt sich, ebenfalls nach kurzer Rechnung, das gleiche Ergebnis wie bei a), was auf den ersten Blick nicht zu erkennen ist.

Neben der Vereinfachung von Formeln spielt also auch der Vergleich von Ausdrücken in verschiedener Gestalt eine wichtige Rolle.

Die Äquivalenz der Ergebnisse von Schaltung a) und b) läßt vermuten, daß ein Widerstand  $R_5$  keinen Einfluß auf  $R_{\text{ges}}$  hat, wenn die Schaltung folgendermaßen aufgebaut ist:



Allerdings ergibt sich hierbei die relativ komplizierte Formel

$$R_{\text{ges}} = \frac{R_1 \cdot R_3}{R_1 + R_3 + R_5} + \frac{\left( \frac{R_1 \cdot R_5}{R_1 + R_3 + R_5} + R_2 \right) \cdot \left( \frac{R_3 \cdot R_5}{R_1 + R_3 + R_5} + R_4 \right)}{\frac{R_1 \cdot R_5}{R_1 + R_3 + R_5} + R_2 + \frac{R_3 \cdot R_5}{R_1 + R_3 + R_5} + R_4}$$

deren Vereinfachung "von Hand" schon recht mühselig ist. Das Ergebnis ist aber dasselbe wie in a) und b), insbesondere also unabhängig von  $R_5$ .

Im folgenden wird das Protokoll eines Dialogs in REDUCE wiedergegeben, der dazu dient, die Formel c) zu vereinfachen.

Der Aufruf von REDUCE geschieht im CMS durch die Eingabe des Kommandos

```
REDUCE
```

Man befindet sich dann in einer Umgebung, die ein interaktives Arbeiten ermöglicht.

Die Beendigung einer REDUCE-Sitzung erfolgt z.B. durch das Kommando

```
BYE;
```

das wie alle anderen Kommandos durch ein Semikolon abgeschlossen werden muß. Kommentarzeilen beginnen mit %.

In diesem Programm werden verschiedene Hilfsgrößen benutzt, um die Eingabe der Formel für  $R_{\text{ges}}$  zu vereinfachen:

```
%
%   Beginn des Programms
%
OFF GCD;

OFF FACTOR;

%
%   Berechnung der Hilfsvariablen RHILF.
%
RHILF:=R1+R3+R5;
```

```
RHILF := R5 + R3 + R1
```

```
%
%   Berechnung der Widerstaende R10,R20,R30.
%
```

```
R10:=R1*R3/RHILF;
```

```
R10 := (R3*R1)/(R5 + R3 + R1)
```

```
%
%   R20:=R1*R5/RHILF;
```

```
R20 := (R5*R1)/(R5 + R3 + R1)
```

```
%
%   R30:=R3*R5/RHILF;
```

```
R30 := (R5*R3)/(R5 + R3 + R1)
```

```
%
%   Berechnung des Gesamtwiderstandes aus R10,R20,R30 sowie R2,R4.
%
```

```
R:=R10+(R20+R2)*(R30+R4)/(R20+R2+R30+R4);
```

```
R := (R4*R2*R5 + R4*R2*R3 + R4*R2*R1 + R4*R5*R1 + R4*R3*R1 + R2*R5*
      R3 + R2*R3*R1 + R5*R3*R1)/(R4*R5 + R4*R3 + R4*R1 + R2*R5 + R2
      *R3 + R2*R1 + R5*R3 + R5*R1)
```

```
%
%   Einsetzen der Formel R4 = R2*R3/R1 in R,
%   Ergebnis ist noch nicht reduziert.
%
```

```
RGES:=SUB(R4=R2*R3/R1,R);
```

```
RGES := (R3*(R22*R5 + R22*R3 + R22*R1 + 2*R2*R5*R1 + R2*R3*R1 + R2*
        R1 + R5*R12 ))/(R2*R5*R3 + R2*R5*R1 + R2*R32 + 2*R2*R3
        *R1 + R2*R12 + R5*R3*R1 + R5*R12 )
```

```
%
%   Anforderung zur Kuerzung gemeinsamer Teiler.
%
```

```
ON GCD;
```

```
%  
%   Endergebnis ausgeben.  
%  
ERG:=RGES;  
  
ERG := (R3*(R2 + R1))/(R3 + R1)  
  
%  
%   Zur Ueberpruefung des Ergebnisses wird der Wert in  
%   faktorisierte Form dargestellt.  
%  
OFF GCD;  
  
ON FACTOR;  
  
RGES;  
  
((R2*R5 + R2*R3 + R2*R1 + R5*R1)*(R2 + R1)*R3)/(((R5 + R3 + R1)*R2  
    + R5*R1)*(R3 + R1))  
  
OFF FACTOR;  
  
%  
%   Welchen Wert muss man fuer R4 in R einsetzen,  
%   damit man ERG erhaelt?  
%  
SOLVE(R-ERG,R4);  
  
SOLN(1,1) := (R2*R3)/R1  
  
1  
  
%  
%   Ende des Programms  
%  
;  
  
END;
```

Mit Hilfe sogenannter 'switches' können Optionen zur Auswertung von Formeln gesetzt werden. z.B.

- GCD** kürzen
- FACTOR** Ergebnis in faktorisierter Form

Als Operatoren werden verwendet

- :=** Definition von Formeln
- SUB** Einsetzen von Formeln in andere
- SOLVE** (formales) Lösen von Gleichungen.

Die Möglichkeiten von **REDUCE** gehen aber über das in obigem Beispiel gezeigte Reduzieren von Brüchen hinaus. Hier wären vor allem das formale Differenzieren und das eingangs erwähnte formale Integrieren zu nennen.

Das folgende Beispiel beschäftigt sich mit dem formalen Differenzieren:

Will man an eine Zeitreihe  $z_1, \dots, z_T$  von Meßwerten in einem nichtlinearen Regressionsansatz mit der Methode der kleinsten Quadrate z.B. die Funktion

$$e^{-\frac{x \cdot t^y}{x + t^y}}$$

mit den Parametern  $x, y$  anpassen, so erfordert der NAG-Algorithmus E04HEF zur Lösung von

$$\sum_{t=1}^T (z_t - e^{-\frac{x \cdot t^y}{x + t^y}})^2 \rightarrow \min_{x,y}$$

die Angabe der ersten und zweiten partiellen Ableitungen in FORTRAN-Unterprogrammen.

Die formale Berechnung dieser Ableitungen kann in **REDUCE** folgendermaßen geschehen:

```
ON FACTOR;

%
% DEFINITION DER FUNKTION F
%
F:=EXP(-X*T**Y/(X+T**Y));

      Y      Y
      ((T *X)/(T  + X))
F := 1/E

%
% BERECHNUNG DER ERSTEN UND ZWEITEN
% PARTIELLEN ABLEITUNGEN
%
FX :=DF(F,X);
```

$$FX := (-T^{(2*Y)} \cdot ((T*X)/(T+X))^Y \cdot Y \cdot 2) / (E^{((T*X)/(T+X))} \cdot (T+X)^Y)$$

FY :=DF(F,Y);

$$FY := (-T^Y \cdot \text{LOG}(T) \cdot X^2) / ($$

$$E^{((T*X)/(T+X))} \cdot (T+X)^Y \cdot Y \cdot 2)$$

FXX:=DF(F,X,2);

$$FXX := (T^{(2*Y)} \cdot Y \cdot (2*(T+X) + T^{(2*Y)})) / ($$

$$E^{((T*X)/(T+X))} \cdot (T+X)^Y \cdot Y \cdot 4)$$

FXY:=DF(F,X,Y);

$$FXY := (T^{(2*Y)} \cdot Y \cdot (T*X - 2*T^Y - 2*X) \cdot X \cdot \text{LOG}(T)) / ($$

$$E^{((T*X)/(T+X))} \cdot (T+X)^Y \cdot Y \cdot 4)$$

FYX:=DF(F,Y,X);

$$FYX := (T^Y \cdot (T*X - 2*T^Y - 2*X) \cdot T^Y \cdot \text{LOG}(T) \cdot X) / ($$

$$E^{((T*X)/(T+X))} \cdot (T+X)^Y \cdot Y \cdot 4)$$

```
FYY:=DF(F,Y,2);
```

$$FYY := \frac{Y^{(2*Y)} + T * X^2 - X^2 * LOG(T) * X}{E^{((T * X)/(T + X)) * Y} * (T + X)^4}$$

```
%
% FXY = FYX ?
%
FXY-FYX;
```

```
0
```

```
;
```

```
END;
```

Man kann nun aus diesen Formeln FORTRAN-Programme entwickeln. Der REDUCE-Switch FORT ermöglicht aber auch eine direkte Ausgabe der Ausdrücke als FORTRAN-Statements. Dies sieht für das oben angegebene Problem folgendermaßen aus:

```
F=1./E**((T**Y*X)/(T**Y+X))
FX=(-T**(2.*Y))/(E**((T**Y*X)/(T**Y+X))*(T**Y+X)**2)
FY=(-T**Y*LOG(T)*X**2)/(E**((T**Y*X)/(T**Y+X))*(T**Y+X)**2)
)
FXX=(T**(2.*Y)*(2.*(T**Y+X)+T**(2.*Y)))/(E**((T**Y*X)/(T**
Y+X))*(T**Y+X)**4)
FXY=(T**(2.*Y)*(T**Y*X-2.*T**Y-2.*X)*X*LOG(T))/(E**((T**Y*
X)/(T**Y+X))*(T**Y+X)**4)
FYX=(T**Y*(T**Y*X-2.*T**Y-2.*X)*T**Y*LOG(T)*X)/(E**((T**Y*
X)/(T**Y+X))*(T**Y+X)**4)
FYY=(T**Y*(T**(2.*Y)+T**Y*X**2-X**2)*LOG(T)**2*X**2)/(E**
(T**Y*X)/(T**Y+X))*(T**Y+X)**4)
```

Allerdings kann man einen solchen Programmausschnitt verbessern, indem man gemeinsame Faktoren der Ableitungen getrennt berechnen läßt. Dieses "Herausziehen" von Faktoren kann selbstverständlich wieder mit Hilfe von REDUCE geschehen.

Eine vollständige Beschreibung von REDUCE findet sich im

```
REDUCE User's Manual
Version 3.2
Edited by Anthony C. Hearn
The Rand Corporation
Santa Monica, CA 90406, April 1985
Rand Publication CP78 (Rev. 4/85)
```

Für REDUCE-Anfänger stehen aber auch sieben Lektionen im Dialog zur Verfügung, die mit dem Kommando

```
IN XMPLIB(LESSn); n = 1,...,7
```

aufgerufen werden können.

Abschließend noch einige weitere Möglichkeiten von REDUCE:

```
%
% FORMALE INTEGRATION
%
INT(SIN(X)**10,X);
```

$$\left( - 128 \sin^9(X) \cos(X) - 144 \sin^7(X) \cos(X) - 168 \sin^5(X) \cos(X) - \right.$$

$$\left. 210 \sin^3(X) \cos(X) - 315 \sin(X) \cos(X) + 315 X \right) / 1280$$

```
%
% RECHNEN MIT EINER BELIEBEN ZAHL VON STELLEN.
% ES WERDEN DIE WERTE FUER 50! (50 FAKULTAET) UND
% 2**2000 BERECHNET.
%
FOR I:=1:50 PRODUCT I;
```

```
30414093201713378043612608166064768844377641568960512000000000000
```

```
2**2000;
```

```
114813069527425452423283320117768198402231770208869520047764273682576626
139237031385665948631650626991844596463898746277344711896086305533142593
135616665318539129989145312280000688779148240044871428926990063486244781
615463646388363947317026040466353970904996558162398808944629605623311649
536164221970332681344168908984458505602379484807914058900934776500429002
716706625830522008132236281291761267883317206598995396418127021779858404
042159853183251540889433902091920554957783589672039160081957216630582755
380425583726015528348786419432054508915275783882625175435528800822842770
817965453762184851149029376
```

```
;
```

```
END;
```

Es soll nicht verschwiegen werden, daß die algebraische Formelmanipulation sehr rechenzeitintensiv sein kann. So benötigt obige Integration ca. 20 sec. CPU-Zeit im CMS.

## UNIX - Eine Einführung

von  
M. Brox

Seit ungefähr zwei Monaten steht am Rechenzentrum die neue Version UTS/V des Betriebssystems UNIX zur Verfügung und wird verstärkt zur Durchführung von Lehrveranstaltungen eingesetzt. Dieser Artikel soll dem Anfänger Hilfestellung bei der Benutzung dieses Betriebssystems geben.

### Zugang zum System

Wenn Sie mit UNIX arbeiten wollen, so benötigen Sie zunächst eine auch für dieses Betriebssystem gültige Benutzerkennung des Rechenzentrums. Die Eröffnung einer UNIX-Sitzung ist in folgenden Schritten durchzuführen:

1. Geben Sie das Kommando

UNIX userid

ein (userid steht hierbei für die Ihnen mitgeteilte Benutzerkennung). Wie alle Eingaben ist auch dieses Kommando mit der ENTER-Taste abzuschließen.

2. Warten Sie, bis das VM-Symbol für die C-Maschine auf Ihrem Bildschirm erscheint.
3. Drücken Sie danach die ENTER-Taste.
4. Warten Sie, bis der Bildschirm gelöscht wird.
5. Geben Sie das Kommando ein:

DIAL RUMUNIX

6. Warten Sie, bis auf dem Bildschirm die Aufforderung

login:

erscheint. Sie haben jetzt eine erste Verbindung zum UNIX.

7. Geben Sie Ihre UNIX-Kennung ein. Diese ist identisch mit der unter 1. angegebenen **userid**.
8. Warten Sie, bis die Aufforderung  
Password:  
erscheint, und geben Sie Ihr Paßwort ein.
9. Warten Sie, bis das Zeichen \$ ("Standard-Prompt") erscheint. Sie haben jetzt eine UNIX-Sitzung eröffnet und können unter UNIX arbeiten.

### Anmerkungen:

- Der umständliche Zugang zum UNIX-System ist durch die Einbettung in die VM-Umgebung und die Hardware-Gegebenheiten am Rechenzentrum bedingt.
- Die oben beschriebene Eröffnungsprozedur ist recht anfällig gegen Fehlbedienungen. Situationen, die sich dann ergeben, sind für den unerfahrenen Benutzer nur schwer zu bereinigen. Sollte Ihr Terminal in einen für Sie unerklärlichen Zustand geraten, so können Sie folgendermaßen zum Anfangszustand zurückkehren:
  - Drücken Sie die SYS-REQ-Taste.
  - Geben Sie das Kommando 'logoff' ein.
- Eine UNIX-Sitzung wird beendet durch Drücken der Taste PF3.

### Die Dateistruktur von UNIX

Die Dateien unter UNIX sind hierarchisch strukturiert. Das heißt: alle Dateien befinden sich

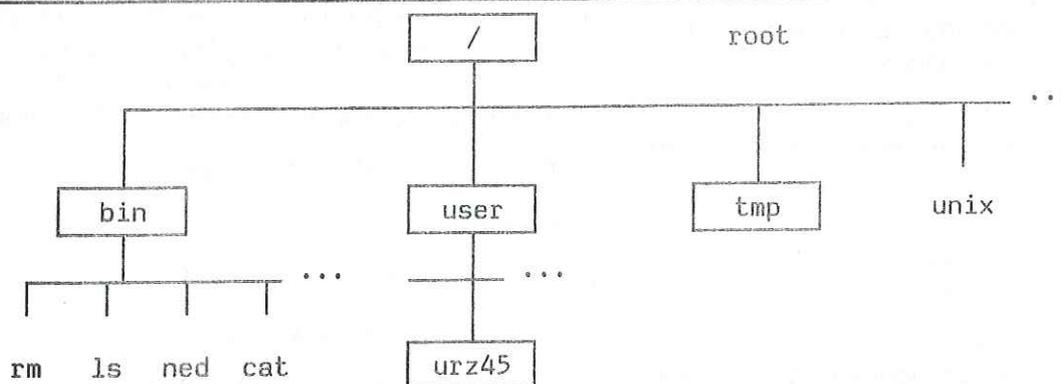


Abbildung 1.

in Verzeichnissen (unter UNIX *directory* genannt). Eine *directory* kann wiederum beliebig viele Unterverzeichnisse (*subdirectories*) enthalten, so daß sich insgesamt das Bild eines auf dem Kopf stehenden Baumes ergibt. Das Verzeichnis, welches den gesamten UNIX-Dateibaum enthält, wird *root-directory* genannt und kurz mit / bezeichnet. Abbildung 1 auf Seite 31 stellt einen kleinen Ausschnitt aus dem Dateibaum dar (eingerahmte Namen sollen hier *directories* symbolisieren). In der *directory /bin* befinden sich zum Beispiel die wichtigsten UNIX-Dienstprogramme und in der *directory /user* die *home-directories* der Benutzer. Dieses Wort der UNIX-Terminologie besagt, daß jedem eingetragenen Benutzer ein Ast des Dateibaumes zur eigenen Verwendung freisteht. Da man in der *home-directory* auch wieder *subdirectories* anlegen kann, bietet sich damit die Möglichkeit, seine eigenen Dateien einfach und effektiv selbst zu verwalten. Da es jedoch unpraktisch wäre, immer den kompletten Pfadnamen zur Bezeichnung einer Datei anzugeben (so bezeichnet zum Beispiel */user/urz45/src/program.c* die Datei *program.c* in der *subdirectory* *src* der *home-directory* des Benutzers *urz45*), bietet UNIX das Konzept der *current directory* an. Wählt man als *current directory /user/urz45/src*, so bezeichnet *program.c* die gleiche Datei wie der obige Ausdruck. Das heißt: allen Dateinamen, welche nicht mit einem / beginnen, wird die *current directory* vorangestellt.

Es folgt eine Liste der wichtigsten UNIX-Kommandos, die zur Verwaltung von Dateien und *directories* benötigt werden (spitze Klammern sind hier durch vom Benutzer gewählte Namen zu ersetzen).

- cd** set current directory  
Setzen der *current directory*. Wird kein Argument angegeben, so wird die *home-directory* zur *current directory* (dieses ist auch die Setzung nach dem *Logon*).  
Syntax: `cd <directory>`
- pwd** print working directory  
Ausgabe des Namens der *current directory*.
- rm** remove  
Löschen einer Datei.  
Syntax: `rm <Datei>`
- rmdir** remove directory  
Entfernen einer *directory*. Die *directory* muß leer sein, d. h. sie darf keine Dateien und keine *subdirectories* enthalten.

Syntax: `rmdir <directory>`

- mkdir** make directory  
Anlegen einer *directory*.  
Syntax: `mkdir <directory>`
- ls** list  
Auflisten der Dateien und *subdirectories* einer *directory*.  
Syntax: `ls <directory>`
- print** print  
Ausgabe einer Datei auf dem Drucker.  
Syntax: `print <file>`
- mv** move  
Wechsel des Dateinamens.  
Syntax: `mv <alter Dateiname> <neuer Dateiname>`
- cp** copy  
Kopieren einer Datei.  
Syntax: `cp <von Datei> <nach Datei>`
- chmod** change modus  
Änderung der Zugriffsrechte.  
Syntax: `chmod <neuer Modus> <Datei>`

Der letzte Befehl ist relativ wichtig, da UNIX, wie aus den obigen Ausführungen deutlich geworden sein sollte, ein Mehrbenutzersystem ist. Jeder Benutzer besitzt zwar seinen eigenen Zweig in der UNIX-Dateistruktur (*home-directory*), jedoch kann zunächst auch jeder andere Zweig erreicht werden (`cd <directory>`). Somit sind Schutzmechanismen vorzusehen, die verhindern, daß eigene Dateien von fremden Benutzern verändert werden. Gibt man das UNIX-Kommando

```
ls -l /
```

ein (UNIX interpretiert Argumente, die mit einem "-" beginnen, als Optionen), so erscheint auf dem Bildschirm eine ausführliche (Option `l`=long) Liste der *root-directory*. Unter anderem befindet sich darin der Eintrag, der die *directory user* beschreibt:

```
drwxrwxr-x 334 root other ... user
```

Bedeutung:

- Pos. 1 Art (hier `d` = *directory*)  
Pos. 2-4 Zugriffsrechte des Besitzers  
`r` = read (Lesezugriff gestattet)  
`w` = write (Schreibzugriff gestattet)

**x =** execute (bei ausführbaren Programmen ist die Ausführung, bei *directories* das Durchsuchen gestattet)

**Pos. 5-7** Zugriffsrechte der Gruppe (alle eingetragenen Benutzer gehören zur Zeit der Gruppe *other* an)

**Pos. 8-10** Zugriffsrechte aller übrigen Benutzer

Danach findet sich der Name des Besitzers der *directory* (hier: *root*) und der der Gruppe. In diesem Fall ist die Gruppe *other*, da sich in dieser *directory* ja alle *home-directories* befinden und somit jeder Benutzer schreibberechtigt sein muß.

Für weitere Informationen möchte ich hier auf das UNIX-Manual verweisen, das auch als Online-Hilfe zur Verfügung steht. Die Syntax des entsprechenden Kommandos ist:

Man <Kommandoname>

Zum Beispiel also: 'Man chmod', um explizit zu erfahren, wie bestimmte Zugriffsrechte gesetzt werden können.

### Der Editor

Der mit UNIX gelieferte Editor heißt *ned*. Der Aufruf geschieht mit:

ned <Dateiname>

Er bietet eine CMS-ähnliche Umgebung. Ich möchte hier nur die Belegung der wichtigsten PF-Tasten angeben:

**PF-3** Verlassen des Editors, wobei die geänderte Datei zurückgeschrieben wird

**PF-7** Blättern rückwärts

**PF-8** Blättern vorwärts

Zur Erklärung der weitergehenden Edierkommandos sei auch hier ein Blick in das Manual empfohlen ('Man ned'). Nach dem Aufruf von 'Man' befinden Sie sich übrigens auch in einer Editorsitzung, so daß die PF-Tasten obige Funktionen besitzen.

### Compiler

UNIX stellt Compiler für die Sprachen C, FORTRAN und Pascal zur Verfügung. Das Rechenzentrum hat dafür mit Hilfe des UNIX-Kommandos 'make' eine

gemeinsame Benutzeroberfläche geschaffen. Die Syntax für den Aufruf eines dieser Compiler ist:

make program

Hier bezeichnet *program* den Namen des zu übersetzenden Programmes. Es ist nur die Einhaltung der UNIX-Konvention nötig:

**C** Programme enden auf .c

**Pascal** Programme enden auf .p

**FORTRAN** Programme enden auf .f

Diese Endung (den Punkt eingeschlossen) ist bei dem Aufruf von 'make' wegzulassen. Das übersetzte Programm wird gestartet durch Eingabe seines Namens (ebenfalls ohne Endung). Sollte es wider Erwarten fehlerhaft arbeiten, so kann es durch Drücken der Taste PA1 abgebrochen werden.

### Umlenkung von Aus- und Eingabe von Programmen

Unter UNIX arbeitende Programme, die einfache Ein-/Ausgabe-Mechanismen benutzen, also nicht explizit auf Dateien zugreifen, lesen von der Tastatur ("Standardeingabe") und schreiben auf den Bildschirm ("Standardausgabe"). Durch einfache Modifikationen können Ein- und Ausgabe jedoch umgelenkt werden.

#### 1. Umlenkung der Ausgabe:

Das UNIX-Kommando 'who' gibt eine Liste aller gerade aktiven Benutzer aus. Soll diese Liste in die Datei mit dem Namen Benutzer geschrieben werden, so lautet die Syntax:

who > Benutzer

Somit bewirkt das Zeichen '>' die Umlenkung der Ausgabe in eine Datei.

#### 2. Umlenkung der Eingabe:

Mit dem UNIX-Kommando 'sort' können Eingaben zeilenweise sortiert werden. Das Sortieren einer Datei mit dem Namen Input geschieht einfach folgendermaßen:

sort < Input

Soll die sortierte Datei noch in eine Datei namens Output geschrieben werden, so lautet das Kommando:

sort < Input > output

3. Umlenkung der Ausgabe eines Programms in die Eingabe eines zweiten:

Dieser, in UNIX als *pipe* bezeichnete Vorgang besitzt vielfältige Anwendungsmöglichkeiten. Eine mit 'sort' sortierte Auflistung der gerade aktiven Benutzer erhalten Sie z. B. mit dem Kommando:

```
who | sort
```

Die Angabe des Zeichens | bewirkt dann, daß die Ausgabe des Programmes 'who' nicht auf den Bildschirm oder in eine Datei gelangt, sondern direkt an die Eingabe von 'sort' weitergeleitet wird.

**Die Statistik-Seite**

**Programmquerschnitt Januar bis März 1986**

von  
A. Ahrens

Im ersten Quartal 1986 ergab sich im Batch-Betrieb auf Jobstep-Basis folgende Verteilung:

Programm	Anzahl	(%)	CPU-Zeit	(%)
PL/I-Optimizing-Compile	10937	4.97	12:55:44	0.98
PL/I-Optimizing-Linkage	2767	1.26	0:42:33	0.05
PL/I-Optimizing-Execute	17343	7.87	82:51:54	6.29
FORTRAN-66-Compile	9777	4.44	3:52:47	0.30
FORTRAN-66-Linkage	15944	7.24	15:42:36	1.19
FORTRAN-66-Execute	31266	14.19	350:40:14	26.62
FORTRAN-77-Compile	40558	18.41	26:26:51	2.01
FORTRAN-77-Linkage	1445	0.66	0:38:52	0.05
FORTRAN-77-Execute	32807	14.89	751:16:01	57.03
Pascal-Compile	402	0.18	0:12:35	0.02
Pascal-Execute	407	0.18	12:23:16	0.94
SPSS	21750	9.87	25:39:45	1.95
SAS	5976	2.71	17:20:58	1.32
IEB / IEH - Programme	10433	4.74	1:50:17	0.14
Sort	4149	1.88	2:40:04	0.20
Service	12940	5.87	8:53:49	0.71
andere Dienstprogramme	1420	0.64	2:41:30	0.20
Summe der Jobsteps:	220321			
Summe CPU-Zeit:			1316:49:46	