

inforum

INFormationsforum des Rechenzentrums der Universität Münster

Jahrgang 14, Nr. 1 – Januar 1990

ISSN 0931-4008

Inhalt

Editorial	2
<u>RUM-Aktuell</u>	
Erfahrungsbericht zu SAS/PC	3
VAMP	4
PCHECK: Neues CMS-Kommando für PL/I-Programmierer	5
JAHR, ein Kalender-EXEC	7
Personalia	7
<u>RUM-Tutorial</u>	
Viren	8
<u>RUM-Lehre</u>	
Lehrveranstaltungen im 1. Halbjahr 1990	13
<u>RUM-Text</u>	
Neues von TEX	22
<u>RUM-Grafik</u>	
Grafik am Universitätsrechenzentrum	24
Explodierende Torten	26
Ein neues POSTER	27
Kreise und Ellipsen mit RUM/GKS	27
<u>RUM-Aroma</u>	
<u>inforum-Quiz</u>	30
<u>Die Statistik-Seite</u>	
Verbrauchsdaten 1989	31
Stichwörter <u>inforum</u> Jahrgang 13	32

Impressum**inforum**

ISSN 0931-4008

Redaktion: W. Bosse (Tel. 83-2461)
H. Pudlatz (Tel. 83-2472)
E. Sturm (Tel. 83-2609)

Satz: A. Ibach
Th. Mosqua

Druck: M. Näther
A. Wolters

Universitätsrechenzentrum
Einsteinstraße 60
4400 Münster

Auflage dieser Ausgabe: 1000

Redaktionsschluß der nächsten
Ausgabe: 15.3.1990

Editorial

von

E. Sturm

Auch für dieses **inforum** haben wir uns etwas Neues ausgedacht: Im Editorial sollen Sie nämlich ab sofort nicht mehr eine lange Liste lesen, in dem jeder der folgenden Artikel einmal erwähnt wird. Stattdessen lassen wir Ihnen das sinnliche Vergnügen des Durchblätterns, wenn Sie erfahren wollen, was es so alles Neues gibt. Damit Sie aber trotzdem schnell einen Überblick gewinnen können, haben wir vor jedem Artikel einen Aufhänger plaziert, der Ihnen verrät, worum es geht.

Eine andere Neuerung würden wir uns wünschen, nämlich daß das **inforum** wie eine ordentliche Zeitschrift geheftet wäre (wir haben ja schließlich eine ISSN-Nummer, s.o.). Aber wahrscheinlich wäre das doch erheblich teurer, da sparen wir lieber für einen neuen Großrechner, wo unser alter inzwischen nur noch als Mini gilt. (Letztes Jahr war er noch ein Super-Mini!)

Damit das Editorial wirklich etwas kürzer wird, höre ich jetzt auf und wünsche nur noch, daß die Sache mit den Aufhängern eine gute Besserung war!

RUM-Aktuell

Erfahrungsbericht zu SAS/PC

von

S. Zörkendörfer

Das Statistical Analysis System SAS gibt es nicht nur als Großrechnerversion, sondern auch auf dem PC.

Zur Erprobung der PC-Version des SAS stand uns im September 1989 ein großzügig ausgestatteter PC zur Verfügung. Ausgehend von der Mächtigkeit des Großrechner-SAS gingen wir mit hohen Erwartungen an dieses PC-Produkt heran, und wir wurden in fast jeder Hinsicht mehr als befriedigt.

Als Rechner stand ein PC mit 386er Processor und 4 MB Erweiterungsspeicher unter MS-DOS (ohne Netzsoftware) zur Verfügung, anschließen konnten wir zeitweise einen HP DeskJet und einen Plotter HP 7475, die Druck- und Plotausgabe für einen HP LaserJet haben wir zunächst auf Disketten abgelegt und dann zu einem am Netzwerk angeschlossenen PC weitergeleitet. Die SAS-Software hatte in etwa den gleichen Umfang wie in unserer CMS-Umgebung, besondere Aufmerksamkeit galt den grafischen Präsentationen (z.B. von PROC GMAP und PROC SURVIVAL), der Matrixsprache und der Aufbereitung vom Menü. Unsere Großrechnerinstallationen des SAS haben das Release 5.18 als aktuelle Version, als nächste Version erwarten wir hier 6.06. Die getestete Version für MS-DOS war 6.03. Daß der Übergang von der Großrechnerversion zur PC-Version recht leicht fiel, möchte ich damit verdeutlichen, daß die Vorbereitung der Lehrveranstaltung mit dieser PC-Version innerhalb eines Werktages erfolgen mußte.

Auffällig in der Programmierumgebung des Display Managers sind die erweiterten Angebote der Fenstertechnik – die SAS-Fans sind erwartungsvoll neugierig, wie dies am Terminal (mit Maus) der Großrechnerversion 6.06 gelingt. Dank der guten Hardwarevoraussetzungen gab es beim Testen und beim Arbeiten mit kleinen Dateien keine Durchsatzprobleme. Bei größerem Daten-

umfang kam es schon zu unangenehm langen Wartezeiten: Suchaufträge im Buchbestand unserer Bibliothek (mit ca. 4000 Büchern) dauerten bis zu einer halben Minute, die Erstellung des kompletten (fast 300 Seiten starken) KWIC-Index erfolgte in etwa einer Stunde. (Mit einer derart umfangreichen Aufgabenstellung hätte ich natürlich gerne den Micro-To-Host-Link erprobt, um ein Programm zunächst am PC zu erstellen und sodann die Produktion am Großrechner zu starten. Leider standen zur Testphase die Hardware- und Software-Voraussetzungen zum Aufbau einer solchen Verbindung nicht zur Verfügung.) Auch bezüglich der Plot-Ausgabe am angeschlossenen DeskJet ist von unangenehm langen Wartezeiten zu berichten – allerdings meldeten Teilnehmer der Lehrveranstaltung wesentlich kürzere Zeiten bei neueren Serien des Tintenspritzers. Zur Bildqualität sowohl des LaserJets wie des DeskJets sei als Beispiel angegeben, daß (jeweils im gewöhnlichen Papierformat A4) eine Landkarte der Bundesrepublik mit eingezeichneten Kreisgrenzen, aber ohne Beschriftungen innerhalb der Kartenfläche alle Qualitätsanforderungen befriedigte. Die Einbindung von SAS-Bildern ins PC-Textverarbeitungssystem WordPerfect 5.0 gelang auf Anhieb.

Vielfach ausgenutzt haben wir – auch in der Lehrveranstaltung – die Möglichkeiten der Screen Control Language. Hiermit sind etwa komplexe Plausibilitätskontrollen bereits während der maskenorientierten Dateneingabe (mit FSEDIT oder nun auch FSPRINT) möglich; dasselbe Werkzeug benutzten wir aber auch, als wir mit dem Produkt AF (Application Facility) Anwendungen in Menüs verpackten, beispielsweise entwickelten wir auf dem Datenbestand unserer Bibliothek ein System mit Anwendungen zum Recherchieren und zur Verwaltung.

Vom Universitätsrechenzentrum wird eine PC-Version des SAS erst angemietet, wenn die Nachfrage entsprechend groß ist; das Rechenzentrum stellt SAS bis auf weiteres daher nur am Großrechner zur Verfügung.

VAMP

von

Hans-Werner Kisker

Neue Wege zur Verteilung von Programmsystemen mit Campus-Lizenz

Die Westfälische Wilhelms-Universität Münster hat in den letzten Jahren eine Reihe von Campus-Lizenzen erworben, die den Angehörigen der Universität zur Verfügung stehen. Im einzelnen handelt es sich um folgende Programme:

- das Satzsystem TEX,
- der Editor KEDIT,
- die Vernetzungssoftware FTP,
- das Statistikpaket STATGRAPHICS,
- die Numerikbibliothek NAG,
- das Datenbanksystem zur Verwaltung bibliographischer Daten ISIS,
- das Textverarbeitungssystem TUSTEP und
- in eingeschränkter Form das Statistikpaket SPSS.

Die Verteilung dieser Programmsysteme wurde bisher von verschiedenen Mitarbeitern des Rechenzentrums abgewickelt. Ab sofort kann jeder Benutzer, der berechtigt ist, eines der obigen Software-Pakete zu betreiben, diese Software selbstständig von jedem in das Universitätsnetz direkt eingebundenen Mikrorechner aus abrufen. Er benötigt hierzu eine normale Verrechnungsnummer für Leistungen des Universitätsrechenzentrums. Zu dieser Verrechnungsnummer muß eine entsprechende Berechtigung für das gewünschte Programmsystem beantragt sein. Anträge für Verrechnungsnummern und für die Programmabrufberechtigungen sind – soweit nicht bereits geschehen – auf den üblichen Wegen ans Rechenzentrum zu richten.

Die Programmsysteme sind auf einem vom Universitätsrechenzentrum bereitgestellten Netzwerk-Server abgelegt. Zu diesem Server hat der Benutzer nur Zugang über ein besonderes im Netz zur Verfügung stehendes Programm mit Namen VAMP (Verteilung Allgemeiner Mikrocomputer-Programme), das von Herrn Coosmann entwickelt wurde. Eine kurze Einführung in VAMP ist dieser Notiz als Anhang beigelegt.

Einführung in VAMP

VAMP ist ein vom Universitätsrechenzentrums der WWU Münster bereitgestelltes Programmpaket. Es wurde geschaffen, um lizenzierte Programme und Programmpakete an berechtigte Benutzer zu verteilen. Es soll helfen, die sichere Abwicklung der Übertragung der lizenzierten Software zu gewährleisten und unsere Verwaltungsarbeit zu entlasten.

Wenn Sie sich für die Zuteilung einer Softwarelizenz interessieren, müssen Sie im Dispatch einen Antrag auf Benutzung von VAMP stellen. Daraufhin wird Ihnen ein Benutzername und ein Paßwort für VAMP zugewiesen.

Nach dem Aufruf von VAMP erscheint ein Logo, das Ihnen zeigt, daß VAMP gestartet wurde. Falls VAMP eine Grafikkarte in Ihrem Rechner entdeckt hat, werden Sie gefragt, ob Sie auch einen Farbbildschirm haben. Sie können nur mit J (Ja) oder N (Nein) antworten. Bei einer Schwarzweißkarte (Monochromkarte) werden Sie nur aufgefordert, eine beliebige Taste zu drücken.

Ab sofort ist das Hilfssystem abruffähig durch F1.

Danach geben Sie auf Anfrage Ihre Benutzerkennung und Ihr Paßwort an. Das Paßwort wird dabei nicht auf dem Bildschirm angezeigt. Es erscheint das Hauptmenü von VAMP.

Sie können nun das Programm verlassen, indem Sie Q eingeben oder Quit ansteuern und RETURN drücken. Es wird vorher gefragt, ob Sie VAMP wirklich beenden wollen.

Sie können weiterhin mittels des Untermenüs Setup Ihr Paßwort für VAMP und Ihr Laufwerk, auf das Sie kopieren wollen, ändern.

Nun kommen wir zum wichtigsten Menüunterpunkt von VAMP. Nach der Wahl von Kopieren können Sie sich aus den Ihnen zugeteilten Kopierlizenzen eine aussuchen, um das entsprechende Softwarepaket zu kopieren. Wenn Sie den Kopiervorgang einleiten, bekommen Sie einen Hinweis darauf, wohin kopiert werden soll (das von Ihnen gewählte Laufwerk), wie das Programmpaket heißt und wieviele Disketten das Paket umfaßt. Bei jedem Aufruf des Kopiervorganges bekommen Sie zuerst eine Datei auf eine formatierte Diskette geschrieben. Diese enthält

kurze Einzelinformationen zu jeder Diskette aus dem Programmpaket.

Wir empfehlen dringend, diese Informationen nachher auf Aufkleber zu schreiben und diese dann auf die kopierten Disketten zu kleben, die Sie beim Kopieren numerieren sollten. Nur so kann man bei größeren Softwarepaketen einem Diskettenchaos entgehen!

Sie können den Kopiervorgang nach jeder Diskette unterbrechen, um später fortzufahren.

Fehler werden in einem Fehlerfenster in der Mitte des Bildschirms angezeigt. Generell fängt VAMP jeden Fehler, der im Rechner verursacht wird, ab.

Die Voraussetzungen für den Betrieb mit VAMP sind ziemlich leicht zu erfüllen, Sie benötigen

- einen IBM-PC-kompatiblen Rechner,
- mindestens 150 KB RAM (freien Hauptspeicher), nachdem die Vernetzung gestartet worden ist,
- eine MSNET- oder PCNET-kompatible Vernetzung,
- MS-DOS oder PC-DOS Version 3.10 oder höher,
- die Berechtigung für den Abruf eines Programmpakets,
- für jedes Programmpaket:
 - eine formatierte Diskette und
 - genügend formatierte oder nicht formatierte Disketten. Die Anzahl hängt vom jeweiligen Softwarepaket ab.

Unterstützung bei der Benutzung von VAMP kann Ihnen die Programmierberatung des Rechenzentrums geben. Normalerweise wird Ihnen dort schnell geholfen werden. Weiterführende Fragen sollten dann an Herrn Dr. Lange (Tel. 2475) weitergeleitet werden.

Die Programmierberatung ist anzutreffen: montags 14–17 Uhr, sowie dienstags bis freitags 10–12 und 14–17 Uhr (Tel. 2486).

Vernetzung

VAMP ist ein Programm, das über das lokale Netz geladen und ausgeführt werden muß. Die Vernetzung darf während der Arbeit nicht ab-

oder unterbrochen werden. VAMP wird dann beendet.

VAMP ist so gebaut, daß mehrere Benutzer gleichzeitig damit arbeiten können. Dies kann in ungünstigen Fällen zu einer kleinen Verzögerung führen, eventuell sogar zu einer Fehlermeldung. Bitte warten Sie ein paar Sekunden und probieren es dann nochmal.

Das Hilffsystem

Das Hilfsmenü ist selbsterklärend. Wenn Sie zum ersten Mal F1 drücken, erscheint die Hilfestellung mit einer Anleitung, wie Sie die Hilfe benutzen können.

PCHECK: Neues CMS-Kommando für PL/I-Programmierer

von

H. Auf der Landwehr

Der PL/I-Compiler überprüft nicht die Übereinstimmung von Argumenten und Parametern beim Aufruf von externen Prozeduren. Ein neues CMS-Kommando hilft bei der Kontrolle.

Wer in PL/I häufig mit externen Unterprogrammen arbeitet, stand sicherlich schon mal vor der folgenden Situation: Durch wiederholtes Ändern der Parameter oder Parameterattribute gab es Unstimmigkeiten zwischen der Deklaration des Unterprogramms im aufrufenden Programm einerseits und den Angaben im externen Unterprogramm andererseits. Solche Unstimmigkeiten können vom PL/I-Compiler nicht erkannt werden und führen oftmals beim Programmlauf zu unerklärlichen Ergebnissen oder gar zum Programmabbruch. Die Fehlerquelle dann zu lokalisieren, bedeutet nicht selten einen erheblichen Zeitaufwand.

Für solche Fälle gibt es jetzt das neue CMS-Kommando PCHECK !!!

PCHECK überprüft alle Deklarationen von externen Unterprogrammen, die in einer Datei bzw. in mit dieser Datei logisch zusammenhängenden

externen Unterprogrammen gefunden werden, auf Übereinstimmung mit den Angaben im externen Unterprogramm selbst. Hierbei werden folgende Punkte berücksichtigt:

- Vorhandensein als externes Unterprogramm oder Entry

Für jede Deklaration eines externen Unterprogramms wird überprüft, ob eine gleichnamige Datei mit Dateityp PLI und beliebigem Filemodus existiert. Wenn keine solche Datei gefunden wird, dann wird nach einem Entry-Punkt gleichen Namens gesucht. Sollte auch kein Entry-Punkt gefunden werden, so wird eine entsprechende Fehlermeldung ausgegeben. Jedes externe Unterprogramm muß also in einer eigenen Datei stehen und darf nicht mit *process angefügt sein, damit es auch gefunden wird. Sollte ein Unterprogrammname sowohl als Datei als auch als Entry-Punkt existieren, dann wird auf diese Fehlerquelle hingewiesen.

- Anzahl und Attribute der Parameter bzw. des Rückgabewertes

Zunächst wird die Anzahl der Parameter verglichen. Bei einer unterschiedlichen Anzahl an Parametern wird eine entsprechende Fehlermeldung ausgegeben. Stimmt die Anzahl der Parameter überein, dann werden für jeden Parameter die Attribute verglichen und ggf. eine Unstimmigkeit angezeigt. Gleiches gilt für den Rückgabewert: Wenn nur einseitig ein Rückgabewert erwartet wird oder wenn unterschiedliche Attribute für den Rückgabewert angegeben sind, dann erscheint eine entsprechende Meldung. Beim Vergleich der Parameterattribute bzw. der Attribute für den Rückgabewert handelt es sich in der jetzigen Version um einen reinen Vergleich auf Übereinstimmung der Zeichenketten (abgesehen von Blanks).

Der Aufruf von PCHECK erfolgt im CMS gemäß folgender Syntax:

```
PCHECK Dateiname [ (ALL []) ]
```

Der Parameter Dateiname bezeichnet den im CMS bekannten Dateinamen der zu untersu-

chenden Datei. Es wird erwartet, daß eine gleichnamige Datei mit Dateityp PLI und beliebigem Dateimodus existiert. Andernfalls wird eine Fehlermeldung ausgegeben und das Programm abgebrochen.

Die Option ALL bewirkt, daß zusätzlich zur Ausgabe der gefundenen Unstimmigkeiten eine Auflistung sämtlicher in einem Programm deklarierten externen Unterprogramme erstellt wird.

PCHECK liefert nur bei PL/I-Programmen mit korrekter Syntax einen fehlerfreien Programmlauf. Einzige Voraussetzung, die an das zu untersuchende Programm gestellt wird, ist, daß jede Variablendeklaration in einer neuen Zeile beginnt, wobei Klammerungen von Variablenamen jedoch erlaubt sind.

Abschließend soll der Gebrauch von PCHECK an einem Beispiel veranschaulicht werden. Vorhanden seien eine Datei HAUPT PLI:

```
Haupt: procedure options (main);
declare
    UP1 entry (char, 1, 2 char,
              2 bit),
    UP2 entry (fixed, ptr)
              returns (bit);
/* PROGRAMMTEXT Haupt */
end Haupt;
```

sowie eine Datei UP1 PLI:

```
UP1: procedure (Zeichen, Struktur);
declare
    1 Struktur (*),
    2 Zeichen char,
    2 Korrekt bit,
    Zeichen char(3);
declare
    UP3 entry (ptr);
/* PROGRAMMTEXT UP1 */
end UP1;
```

und eine Datei UP2 PLI:

```
UP2: procedure (Zahl)
              returns (fixed);
declare
    Zahl fixed;
/* PROGRAMMTEXT UP2 */
UP1: entry (Zahl);
/* PROGRAMMTEXT UP2 */
end UP2;
```

Dann führt der Aufruf „PCHECK Haupt“ zu der in Abbildung 1 gezeigten Ausgabe.

Pro Aufrufmöglichkeit erscheint eine Zeile. Zum Beispiel lese man die erste Zeile als: Programm Haupt ruft Programm UP1 auf, wobei der erste Parameter (P1) im rufenden Programm als char und im gerufenen als char (3) deklariert wurde. Der Buchstabe R betrifft den Funktionswert, also die returns-Angabe. Wie man in der vierten

Zeile lesen kann, besteht eine Diskrepanz beim Aufruf von UP2: Erwartet wird im rufenden Programm ein bit-Wert, zurückgegeben wird dagegen ein fixed-Wert.

Nach unseren Erfahrungen bietet diese knappe Art der Darstellung einen schnellen Überblick über alle Fehlerquellen, die auf falsche Parameter-Deklarationen zurückzuführen sind.

Haupt: P1 (char)	=> UP1: Zeichen (char(3))
Haupt: P2 (1, 2 char, 2 bit)	=> UP1: Struktur (1 (*), 2 char, 2 bit)
Haupt: 2 Parameter	=> UP2: 1 Parameter
Haupt: R (bit)	=> UP2: (fixed)
UP1:	=> UP3: existiert nicht
UP1: existiert doppelt	=> als CMS-Datei
	als Entry von UP2

Abbildung 1.

JAHR, ein Kalender-EXEC

von

H. Pudlatz

Für alle, die schon immer einen Kalender online brauchten, ihn aber nicht zu programmieren wagten!

Wer am Anfang des Jahres einen Kalender als Beilage zum **inforum** erwartet hatte, ihn hier aber vergeblich suchte, soll als kleine Entschädigung auf einen neuen Service im CMS hingewiesen werden.

Hier gibt es jetzt das EXEC JAHR, das den aktuellen Kalender auf dem Bildschirm anzeigt, wobei die Sonntage, die gesetzlichen und kirchlichen Feiertage auf manchen Terminals hervorgehoben dargestellt werden. Die Eingabe von

JAHR

liefert in diesem Jahr also den Kalender von 1990. Für alle, die entweder nicht erraten, warum der 26. Februar 1990 in Münster als Feiertag gilt bzw. deren Terminal die Feiertage nicht hervorhebt, ist unten auf dem Bildschirm im Klartext angegeben, auf welches Datum jeder Feiertag fällt.

Wer wissen möchte, an welchem Tag im Jahr 1968 Ostern war, kann auch

JAHR 1968

eingeben. Bei einem normalen 32-zeiligen Terminal paßt der Kalender auf eine Bildschirmseite. An PCs oder älteren Terminals müssen Sie leider einmal die CLEAR-Taste drücken, um den Rest zu sehen. Und nun viel Spaß mit JAHR!

Personalia

Am 18.10.1989 hat Frau H. Brunstering eine Halbtagsstelle im Bürobereich des Rechenzentrums angetreten. Sie vertritt Frau Knappstein im Schreibzimmer Orleansring 16 (Abteilung 3).

Bei den studentischen Mitarbeitern haben sich folgende Veränderungen ergeben:

Ausgeschieden sind zum 31.10.1989 Herr R. Ratschmeyer und zum 14.12.1989 Frau A. Schulze. Beiden können wir zum bestandenen Examen gratulieren. Nachzutragen ist, daß Herr S. Schnieders bereits zum 31.7.1989 seine Tätigkeit beendet hat.

Neu eingestellt worden sind Frau A. Jankrift und Herr E. Schroer im November sowie die Herren W. Bollweg und M. Schlütz im Dezember 1989.

RUM-Tutorial

Viren

von

Hans-Werner Kisker

Über angemessenes Verhalten gegenüber Programmen mit Virus-Eigenschaft

Um die vermeintliche oder auch tatsächliche Bedrohung von Programmen und Rechnern durch sogenannte Viren rankt sich ein Geflecht von Gerüchten, das von ernstzunehmenden Mahnungen über Halbwahrheiten bis zu barem Unsinn reicht. Dem normalen Mikrorechnernutzer fällt es häufig schwer, hier die sachgerechte Einordnung zu finden und demgemäß sein Verhalten festzulegen. Dieser Artikel soll etwas dazu beitragen, diese Verunsicherung zu beseitigen.

Sicherlich sind Viren ein Ärgernis, die Beschäftigung mit Ihnen lästig und sicherlich stellen sie bei allzu leichtfertiger Vorgehensweise eine Gefährdung für Daten und Programme dar. Bei einer nüchternen Betrachtungsweise stellt sich jedoch schnell heraus, daß das Gefahrenpotential wohl doch überschätzt wird. Angemessene Verhaltensweisen vorausgesetzt, ist die Wahrscheinlichkeit für einem echten Schaden durchaus einzuschränken.

Begriffsbestimmung

Der Jargon, der sich um den Begriff Viren herum gebildet hat, ist meiner Ansicht nach äußerst unglücklich gewählt – da gibt es Würmer, Falltüren, Zeitbomben, trojanische Pferde und da wird von biologischen Verhaltensmustern, Krankheit, Infektion gesprochen. Unangemessen ist vor allem die vorgegaukelte Analogie zur Medizin. Hierdurch wird man leicht dazu verführt, auch an das Umfeld Kriterien anzulegen, wie sie in der Medizin üblich und angebracht sind. Insbesondere wird man dazu verleitet, zum einen die Herkunft der Viren als unabänderlich – sozusagen gottgegeben – hinzunehmen, und zum anderen die Beschäftigung mit ihnen als eine sinnvolle, lohnende Tätigkeit anzusehen. Natürlich ist diese Bewer-

tung völlig unangebracht, ja aus einem gewissen Sichtwinkel heraus, der weiter unten sichtbar wird, sogar gefährlich. Machen wir uns doch klar, was wirklich passiert. Da sitzt irgendwo auf der Welt ein Programmierer, den ich nicht kenne, der auch mich nicht kennt, und der schreibt ein Programm, dessen einziger Zweck es letztlich ist, meine Daten zu zerstören. Vor diesem Hintergrund müssen wir dringend zu einer angemessenen Nüchternheit zurückfinden. Ich möchte deshalb das Thema auch an dieser Stelle nicht dadurch künstlich aufwerten, daß ich an den Anfang abgrenzende, wohlformulierte Definitionen stelle, die die obigen dubiosen Begriffe vertiefend erläutern. Es sei nur kurz aufgezeigt, was ein Programm mit Virus-Eigenschaft anrichten kann.

Die Virus-Eigenschaft eines Programms zeigt sich darin, daß es bei seiner Ausführung neben der normalen und vom Anwender beabsichtigten Wirkung noch weitere unvorhergesehene und unerwünschte Aktivitäten entwickelt. Insbesondere werden andere auf dem Rechner abgelegte Programme so modifiziert, daß sie sich genau so wie das Ausgangsprogramm verhalten. Sie werden also um Code-Teile ergänzt, die diese Manipulation vornehmen. Daneben können noch weitere Schäden angerichtet werden: vom Ausgeben einer unnötigen Meldung auf den Bildschirm bis zum vollständigen Löschen von Platten reicht die Palette.

Solch ein Virus-Programm zu schreiben ist leider kein sonderliches Kunststück. In verschiedenen Zeitschriften und Büchern sind außerdem entsprechende Hinweise vorhanden. Aus naheliegenden Gründen möchte ich von einem solchen Vorgehen absehen.

Die Herkunft der Viren

Es erscheint mir aufschlußreich, die Motive zu betrachten, die jemanden dazu veranlassen, ein Virus-Programm in Umlauf zu setzen. Man kann hier meiner Meinung nach zwei grundsätzlich verschiedene Personengruppen unterscheiden.

Die meisten Virus-Programme werden von Leuten geschrieben, die ihre Tätigkeit als Spiel, Schabernack oder auch Sport bezeichnen würden. Von diesem Personenkreis sind auch die ersten Aktivitäten ausgegangen. Man geht wohl nicht ganz fehl in der Annahme, daß die Wurzeln für ihr Verhalten in einem gestörten Selbstwertgefühl zu finden sind. Sie suchen die Anonymität, da sie sich offenbar nicht trauen, die Ergebnisse Ihrer Arbeit offen vorzulegen. Andererseits möchten sie aber Aufmerksamkeit erregen – und durch nichts erweckt man mehr die Aufmerksamkeit eines anderen, als daß man ihm Schaden zufügt. Daß diese Vermutung über den Personenkreis richtig ist, wird auch dadurch gestützt, daß viele Virus-Programme recht primitiv aufgebaut sind. Offenbar sind sie nicht von Fachleuten, die die Möglichkeiten überschauen, konzipiert worden. Die Aufmerksamkeit, die den Programmierern der Virus-Programme entgegengebracht wird, ist die Quelle für ihre Motivation. Entziehen wir ihnen diese Aufmerksamkeit, so wird auch ihr Interesse abnehmen, uns Schaden zuzufügen.

Während die erste Personengruppe also ziemlich ungerichtet agiert und allenfalls darauf aus ist, so etwas wie eine allgemeine Verunsicherung zu bewirken, handelt der zweite Personenkreis absolut zielgerichtet: eine bestimmte Wirkung soll erreicht oder aber ein bestimmter Schaden soll herbeigeführt werden. Diese zweite Motivationsquelle würde ich als wesentlich schwerwiegender einstufen als die erste. Hier wird sich nämlich – so ist zu befürchten – auch der eine oder andere qualifizierte Fachmann verleiten lassen, aus Verblendung, Habgier oder Rache heraus zu handeln. Ich möchte drei beispielhafte – und erschreckende – Szenarien anführen, die in diesen Zusammenhang gehören.

1. Eine Terroristengruppe lanciert ein Virusprogramm in die Datenverarbeitung staatlicher Stellen.
2. Ein Programm, das z.B. als Reklamebeigabe zu einer Zeitung verteilt wird, verschlüsselt wichtige Daten auf dem Rechner, auf dem es gestartet wird. Den zum Dechiffrieren benötigten Schlüssel erhält man nur nach Zahlung eines Geldbetrages auf ein anonymes Konto in einem fremden Land.

3. Ein unzufriedener Mitarbeiter einer Software-Firma manipuliert ein von dieser Firma vertriebenes Programm. Die Virus-Eigenschaft wird jedoch erst nach einiger Zeit aktiviert, wenn für eine weite Verbreitung gesorgt und alle Spuren zu dem Mitarbeiter verwischt sind.

Die ersten beiden Fälle sind gerüchteweise bereits konkret vorgekommen. Den dritten Fall halte ich persönlich für den beunruhigsten. Hier sind die Software-Firmen ganz eng in die Pflicht zu nehmen. Sie müssen durch strenge und umfassende Kontrollen sicherstellen, daß ihre Programme keine Code-Sequenzen enthalten, die absichtliche Manipulationen vornehmen. Vielleicht wäre es auch hilfreich, wenn der Gesetzgeber durch angemessene Schadenersatzregelungen deutlich machen würde, welche Verantwortung den Firmen hier zukommt. Es wäre damit auch Sorge getragen, daß einer gewissen Leichtfertigkeit bei der Software-Entwicklung von vornherein die wirtschaftliche Rentabilität entzogen ist.

Auswirkungen eines Programms mit Virus-Eigenschaft

Jedes Auftreten der Virus-Eigenschaft ist mit der Veränderung von Programmen oder Daten im Rechner verbunden. Um die Wirksamkeit von Schutzmaßnahmen beurteilen zu können, ist es hilfreich, sich vor Augen zu führen, welche Komponenten wirklich von Manipulationen betroffen sein können und welche nicht. Hier grassieren teilweise wilde Gerüchte – so kann eine schreibgeschützte Diskette, das BIOS-ROM oder ein anderer Festwertspeicher auf keinen Fall abgeändert werden. (Wird bei einem 386er oder einem Rechner mit dem sogenannten NEAT-Chip-Satz der Inhalt eines ROM-Bausteins in den Schreib/Lesespeicher an gleicher Adresse (Shadow-RAM) kopiert, so kann diese Kopie zwar temporär manipuliert werden, die Änderung geht jedoch beim Ausschalten des Rechners verloren.) Manipulierbar ist dagegen dagegen:

- jedes ausführbare Programm (auch command.com),
- ein File des Betriebssystems (ibmbio.com und ibmdos.com),

- das Boot-Record,
- der Partition-Loader,
- die FAT,
- jede Directory,
- jedes Daten-File,
- ein unbelegter Teil des letzten Sektors eines Files,
- ein als defekt markierter Sektor und
- das CMOS-RAM.

Zu beachten ist, daß auf jeden Fall eine der ersten vier Komponenten manipuliert werden muß, da nur diese ausgeführt werden. Die anderen Komponenten können nicht direkt startbare Programme enthalten, sie können allerdings sehr wohl weitere Daten und auch Programmstücke enthalten, die dann von einer der ersten Komponenten geladen oder ausgewertet werden.

Damit ist klar, daß die Wirkung eines Virus-Programms stets mit der Veränderung von Komponenten verbunden ist. Dies gibt uns eine sichere Methode an die Hand, die Auswirkung eines Virus-Programms zu erkennen. Wir müssen den Inhalt aller oben angeführten Komponenten auf unerwartete Änderungen hin überprüfen. Üblicherweise wird zu diesem Zweck zu jeder Komponente die Länge und eine Prüfsumme festgehalten. Diese Angaben werden dann nach Ablauf eines verdächtigen Programms erneut gebildet. In extremen Fällen können die Komponenten auch Byte für Byte mit einer gesondert und geschützt aufbewahrten Referenzversion verglichen werden. Tritt eine Diskrepanz auf, so ist möglicherweise ein Virus-Programm ausgeführt worden.

Schutzmaßnahmen

Wünschenswert wäre es, die Virus-Eigenschaft eines Programms zu erkennen, bevor sie zur Wirkung kommt. Dies erweist sich jedoch als äußerst schwierig.

Code-Analyse

Theoretisch könnte man an eine gründliche Code-Analyse denken, bei der man die zerstörenden Programmteile auffindet. Praktisch ist dies jedoch selbst für den professionellen Anwender unmöglich, vor allem wenn nur das ausführbare Programm vorliegt. Dies muß man also – wie oben gefordert – dem Software-

Hersteller, der ja über den Quell-Code verfügt, überlassen.

Typensammlungen

Als eine andere Möglichkeit wird manchmal eine sogenannte Typensammlung von Virus-programmen angesehen. In ihnen werden typische Code-Sequenzen bekannter Virus-Programme festgehalten. Nach diesen wird dann in einem unsicheren Programm gesucht. Ich persönlich halte von diesen Sammlungen überhaupt nichts. Zwar kann man dadurch manche Programme mit Virus-Eigenschaft erkennen, die Sicherheit, die dies Verfahren jedoch bei negativem Ausgang bietet, ist jedoch gleich Null. Es kann ja durchaus ein Virus-Teil enthalten sein, der nicht in die Sammlung aufgenommen ist. Es wird also eine trügerische Sicherheit geboten, die den unerfahrenen Benutzer zu unangemessener Leichtfertigkeit verleiten kann. Außerdem begibt man sich in die Rolle des Reagierenden, während man dem Programmierer der Viren die Aktion überläßt. Für ihn ist es aber ein leichtes, die ja veröffentlichten Prüfprogramme zu umgehen. Der Nutzen dieser Sammlungen ist also deutlich beschränkt, andererseits bietet er dem anonymen Viren-Entwickler jedoch ein Podium zur Selbstdarstellung. Was wäre größer als die Aufnahme eines selbstgeschriebenen Virus-Programmes in eine „renommierte Viren-Sammlung“. Zum Glück gibt es eine Sammlung mit diesem Etikett noch nicht, und wir sollten Sorge tragen, daß sie auch nicht aufgebaut wird.

Testumgebung

Vorgeschlagen wird manchmal auch der Aufbau einer Testumgebung. Man kann sich hierunter einen Rechner vorstellen, auf dem nur anderweitig gesicherte Programme und Daten abgelegt sind und auf dem das zu untersuchende Programm probeweise installiert und zum Ablauf gebracht wird. Dabei wird – wie oben beschrieben – auf Veränderungen geachtet, die sich außerhalb des angegebenen Wirkungsbereichs des Programms ergeben. Treten keine Veränderungen auf, so geht man davon aus, das keine Virus-Eigenschaft vorliegt. Dieser Vorschlag in seiner einfachen Form birgt zwei Fehler in sich: zum einen kann nicht

jeder Nutzer eines Mikrorechners auch noch einen zweiten Testrechner vorhalten und zum anderen kann es ja sein, daß das Virus-Programm seine Wirkung erst nach mehrmonatigem fehlerfreiem Gebrauch oder erst zu einem bestimmten Datum zeigt. In der Tat sind manche Programme mit Virus-Eigenschaft genau so programmiert. Trotzdem ist dieser Vorschlag – wie wir unten sehen werden – im Ansatz richtig und weist wohl auf die einzige wirklich zuverlässige Absicherung gegen Viren hin.

Verhinderung der Manipulation auf Software-Basis

Wir müssen leider festhalten, daß es keine zuverlässige Methode gibt, ein Virusprogramm als solches vor Auftreten seiner Auswirkung zu erkennen. Die nächste naheliegende Schutzstufe ist dann der Versuch, das Auftreten der Wirkung unmöglich zu machen. Es werden auf dem Markt verschiedene Programme zu diesem Zweck angeboten. Sie setzen fast immer darauf, daß ein Virus-Programm zu seinen Manipulationen DOS- bzw. BIOS-Aufrufe benutzt. Diese Aufrufe werden dann von den Schutzprogrammen überwacht. In der Tat sind die meisten Programme – wahrscheinlich weil sie vom obigen ersten Personenkreis stammen – recht simpel und man kann sich auf diese einfache Art und Weise tatsächlich häufig gegen ihre Zerstörung schützen. Einen sicheren Schutz bieten diese Programme, die ja auf Software-Basis beruhen, jedoch nicht. Gegen Viren-Programme, die die Hardware des Mikrorechners direkt ansprechen, sind sie machtlos. Sie erhöhen jedoch die Know-How-Schwelle, die der Virenschreiber überwinden muß.

Verhinderung der Manipulation auf Hardware-Basis

Sichere Schutzmaßnahmen müssen auf Hardware-Basis beruhen. Insbesondere schreibgeschützte Disketten und Platten sind hier einzusetzen. Wenn ein entsprechender Aufwand zu rechtfertigen ist, kann man daran denken, seinen Rechner mit zwei Platten – einer stets schreibgeschützten Sicherungsplatte und einer beschreibbaren Arbeitsplatte zu versehen. Soll ein Programm ausgeführt werden, so wird es von der Sicherungsplatte auf die Arbeitsplatte

kopiert und dann von dort geladen. Dabei wird – wie oben beschrieben – auf unerwartete Änderungen geachtet. Zeigen sich solche Änderungen, so sind hiervon nur Datenteile der Arbeitsplatte betroffen. Die Daten und Programme der Sicherungsplatte können jedoch nicht angegriffen werden. Der möglicherweise zerstörerischen Wirkung des Virusprogramms ist damit seine Spitze genommen. Allenfalls Daten, die während der aktuellen Ausführung auf der Arbeitsplatte neu erzeugt wurden, sind verloren. Alle anderen Datenbestände stehen ja auf der Sicherungsplatte unverändert zur Verfügung. Es ist klar, daß sich dies Verfahren durch seinen relativ hohen Material- (zweite Platte) und seinen relativ hohen Verfahrensaufwand (ständiges Kopieren der Programme) nur in Bereichen mit besonderen Sicherheitsanforderungen trägt. Hier bietet es jedoch einen sicheren Schutz vor den zerstörerischen Auswirkungen eines Programms mit Virus-Eigenschaft.

Verwendung von Programmen aus vertrauenswürdiger Quelle

Eine weitere wesentliche, aber ganz anders geartete Schutzmöglichkeit ist bisher noch unerwähnt geblieben. Es ist **dringend anzuraten, nur Programme aus vertrauenswürdiger Quelle zu verwenden**. Bisher gibt es meines Wissens nach keinen Hinweis darauf, daß durch die Programmpakete, die von den renommierten Software-Firmen vertrieben werden, Virus-Programme eingeschleppt werden. Für den normalen Benutzer ist ein korrektes Verhalten beim Erwerb von Software wohl als der beste Schutz vor Schaden durch Virus-Programme anzusehen.

Zusammenfassung

Zum Thema Viren gibt es also einige Punkte, die zur Besorgnis Anlaß geben. Vor allem ist es zu bedauern, daß es einem Programm nicht anzusehen ist, ob es eine Virus-Eigenschaft besitzt. Diese zeigt sich erst durch ihre Wirkung, und die kann verheerend sein. Weiterhin kann es leider auch keine Schutzprogramme auf Software-Basis geben, die die Auswirkung der Virus-Eigenschaft zuverlässig verhindern.

Einen sicheren Schutz bieten dagegen Lösungen auf Hardware-Basis. Sie bringen jedoch einen erhöhten Kostenaufwand mit sich und sind aufwendig in der Handhabung. Sie sind wohl nur im sicherheitssensitiven Bereich tragbar.

Der normale Benutzer schützt sich am zuverlässigsten, wenn er Programme aus dubiosen Kanälen vermeidet. Bei Verwendung von korrekt erworbener Software aus zuverlässiger Quelle ist kaum mit Beeinträchtigungen zu rechnen.

Des weiteren sollten wir uns dringend darum bemühen, dem Thema Viren keine überzogene Beachtung zu schenken. Sicher wäre es ein Fehler, die Gefahr ganz zu ignorieren. Einige von uns sollten sich schon daran machen, die Erkennungsmechanismen zu verfeinern. Die „große Bühne“ sollten wir jedoch peinlichst vermeiden. Ich glaube, daß dies ein Weg ist, die verbreitete Erzeugung von Viren auf Dauer zu unterbinden.

RUM-Lehre

Lehrveranstaltungen im 1. Halbjahr 1990

Beratung zum Lehrangebot erfolgt durch Herrn W. Bosse jeweils dienstags und donnerstags von 11 bis 12 Uhr (Tel. 83-2461).

Zu den Ferienkursen

Vor Beginn des SS 1990 werden vom Universitätsrechenzentrum einige ganztägige Intensivkurse durchgeführt, in denen Stoffvermittlung und Übungen integriert sind. Diese Veranstaltungen sollen durch entsprechende Betreuung der Teilnehmer eigene Programmierübungen fördern. Das bedingt eine Begrenzung der Teilnehmerzahl der einzelnen Lehrveranstaltungen. Interessenten werden deshalb gebeten, sich möglichst bald, spätestens jedoch eine Woche vor Beginn der entsprechenden Veranstaltung, im Dispatch des Rechenzentrums (Raum 02) in die Anmeldelisten einzutragen, und sollten unbedingt zu dem angekündigten Beginn anwesend sein.

Die Teilnehmer dieser Kurse werden gebeten, diese im SS 1990 zu belegen. Dies ist neben der bestandenen Abschlußklausur Voraussetzung für die Aushändigung eines Scheines über die erfolgreiche Teilnahme.

Zu den Semesterkursen

Eine Anmeldung ist nur für diejenigen Lehrveranstaltungen erforderlich, die nachfolgend besonders gekennzeichnet sind.

1. Lehrveranstaltungen in den Semesterferien (Februar – März 1990)

- | | | |
|--------|--|------------------------|
| 320012 | Programmieren in FORTRAN
vom 12.2. bis 23.2.1990 ganztägig
Hörsaal: M4, Beginn: 12.2.1990, 13.30 Uhr | <i>Reichel, K.</i> |
| 320027 | Programmieren in Pascal
vom 12.2. bis 23.2.1990 ganztägig
Hörsaal: M3, Beginn: 12.2.1990, 9 Uhr | <i>Nienhaus, R.</i> |
| 320031 | Programmieren in Pascal
vom 19.3. bis 30.3.1990 ganztägig
Hörsaal: M3, Beginn: 19.3.1990, 9 Uhr | <i>Süselbeck, B.</i> |
| 320046 | Programmieren in C
vom 5.3. bis 16.3.1990 ganztägig
Hörsaal: M3, Beginn: 5.3.1990, 9 Uhr | <i>Stockhofe, W.</i> |
| 320050 | Statistische Datenanalyse mit dem Programmsystem SPSS ^X
vom 19.3. bis 30.3.1990 ganztägig
Hörsaal: M4, Beginn: 19.3.1990, 9 Uhr | <i>Steinhausen, D.</i> |

- 320065 Text- und Layout-Strukturen in LaTeX Kaspar, W.
vom 5.3. bis 16.3.1990 ganztägig
Hörsaal: M4, Beginn: 5.3.1990, 9 Uhr
- 320070 Mikrorechner-Praktikum Kisker, H.-W.
vom 5.3. bis 16.3.1990 halbtägig
Pool-Raum RZ, Beginn: 5.3.1990, 9 Uhr

2. Einführende Lehrveranstaltungen im Sommersemester 1990

- 320084 Einführung in die EDV Held, W.
Do 11-13
Hörsaal: M4, Beginn: 5.4.1990
- 320099 Programmieren in FORTRAN Ost, St.
Mo 13-15
Hörsaal: M4, Beginn: 9.4.1990
- 320103 Programmieren in Pascal Hölters, J.
Di 13-15
Hörsaal: M3, Beginn: 10.4.1990
- 320118 Programmieren in Modula-2 Mersch, R.
Mi 13-15
Hörsaal: M4, Beginn: 4.4.1990
- 320122 Programmieren in PL/I Pudlatz, H.
Di, Do 15-17
Hörsaal: M4, M5, Beginn: 10.4.1990 (im M4)
- 320137 *) Textverarbeitung auf Mikrorechnern Kamp, H.
Fr 13-15
Hörsaal: M6, Beginn: 20.4.1990
- 320141 Computerunterstütztes Publizieren mit LaTeX Benduhn-Mertz, A.
Mo 13-15
Hörsaal: M5, Beginn: 9.4.1990
- 320156 Statistische Datenanalyse mit dem Programmpaket SAS Zörkendörfer, S.
Mo 15-17
Hörsaal: M4, Beginn: 23.4.1990

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

3. Weiterführende Lehrveranstaltungen im Sommersemester 1990

320160	Datenstrukturen und Programmierverfahren in Pascal Di 13-15 Hörsaal: M6, Beginn: 3.4.1990	<i>Bosse, W.</i>
320175	Programmieren in PL/I für Fortgeschrittene Mi 15-17 Hörsaal: M6, Beginn: 4.4.1990	<i>Sturm, E.</i>
320180	Programmieren in C Do 15-17 Hörsaal: M4, Beginn: 12.4.1990	<i>Lange, W.</i>
320194	SQL - Vergleich von Implementationen Mi 9-11 Hörsaal: M4, Beginn: 4.4.1990	<i>Achilles, A.</i>
320209	SPSS ^X : Fortgeschrittene Anwendungen Mo 15-17 Hörsaal: M5, Beginn: 9.4.1990	<i>Steinhausen, D.</i>
320213	Rechnernetze Do 13-15 Hörsaal: M4, Beginn: 19.4.1990	<i>Richter, G.</i>
320228	Dialogprogrammierung mit REXX Fr 13-15 Hörsaal: M4, Beginn: 6.4.1990	<i>Mertz, K.-B.</i>
320232	Betriebssystemdienste für interaktive Anwendungen Mi 15-17 Hörsaal: M4, Beginn: 11.4.1990	<i>Neukäter, B.</i>
320247	Kolloquium über Themen der Informatik Fr 15-17 Hörsaal: M4	<i>Held, W./ die wiss. Mitarbeiter des Rechenzentrums</i>
320251	Anleitung zum Einsatz der EDV bei wissenschaftlichen Arbeiten nach Vereinbarung	<i>die wiss. Mitarbeiter des Rechenzentrums</i>

Kommentare zu den Lehrveranstaltungen

Wer als Anfänger eine Programmiersprache sucht und sich auf die Erläuterungen keinen Reim machen kann, sollte sich an den Autos orientieren.

Einführung in die EDV (320084)

Die Vorlesung vermittelt Grundkenntnisse der Datenverarbeitung für Hörer aller Fachbereiche. Behandelt werden folgende Themen:

- Geschichtliche Entwicklung
- Theoretische Grundlagen
- Hardware (Zentraleinheit, Peripheriegeräte)
- Software (Programmiersprachen, Compiler, Betriebssysteme)
- Anwendungen (Textverarbeitung, Datenbanken, Statistikprogramme u.a.)

Literatur:

Dworatschek: Grundlagen der Datenverarbeitung, de Gruyter

Sommer: Informatik – Eine PC-orientierte Einführung

Wolters: Schlüssel zur Computer-Praxis



Programmieren in FORTRAN (320012, 320099)

FORTRAN ist eine Programmiersprache, die vorwiegend für die Formulierung von Problemlösungen aus dem Bereich der Naturwissenschaften (Numerik, Statistik) geeignet ist. Unverwundlichkeit, leichte Erlernbarkeit und Anwendbarkeit auf

Computern fast aller Hersteller haben dieser Programmiersprache eine weite Verbreitung gesichert.

Literatur:

VS FORTRAN Release 4.0 (erhältlich im Rechenzentrum)

Wehnes: FORTRAN 77, Hanser

Kießling/Lowes: Programmierung mit FORTRAN 77, Teubner Studienskripten

Brauer: Programmieren in FORTRAN 77, Hüthig

Brauer: FORTRAN 77 – Ständig im Griff, Hüthig

Metcalf: Effective FORTRAN 77, Oxford University Press



Programmieren in Pascal (320027, 320031, 320103)

Die Programmiersprache Pascal ist aufgrund ihres didaktischen Konzepts für Programmieranfänger besonders zu empfehlen. Andererseits ist Pascal durch die Vielzahl von Datentypen und Strukturierungsmöglichkeiten für Anwendungen numerischer wie nichtnumerischer Art gleichermaßen geeignet.

Literatur:

Marty: Methodik der Programmierung in Pascal, Hanser

Jensen/Wirth: Pascal, User Manual and Report, Springer

Ottmann/Widmeyer: Programmieren mit Pascal, Teubner

Wilson/Addyman: Pascal, Leichtverständliche Einführung in das Programmieren mit Pascal, Hanser

Datenstrukturen und Programmierverfahren in Pascal (320160)

In dieser weiterführenden Lehrveranstaltung werden insbesondere dynamische Datenstrukturen (Stack, Liste, Baum u.a.) sowie Fragen der Programmiermethodik anhand praktischer Beispiele behandelt. Schwerpunkte bilden dabei effiziente Sortierverfahren und Aufgaben der Listenverarbeitung. Im Hinblick auf die Erstellung leistungsfähiger und übersichtlicher Programme sollen bereits vorhandene grundlegende Programmierkenntnisse in Pascal vertieft werden.

Literatur:

Wirth: Algorithmen und Datenstrukturen, Teubner
Lipschutz: Datenstrukturen, McGraw-Hill
Schneider/Bruell: Advanced Programming and Problem Solving with Pascal, J. Wiley & Sons



Programmieren in PL/I (320122)

PL/I ist eine universelle Programmiersprache. Ihre Anwendungsmöglichkeiten reichen von einfach zu handhabender Textverarbeitung über die Programmierung numerischer Probleme bis zur Bearbeitung großer Datenmengen mit Hilfe vielseitiger Ein/Ausgabebefehle. Dynamische Speicherplatzverwaltung und selbst Multitasking, also der gleichzeitige Ablauf mehrerer Unterprogramme, sind in PL/I möglich.

Im Grundkurs können selbstverständlich nicht all diese Sprachelemente vorgestellt werden. Im Vordergrund steht die Vermittlung moderner Programmiermethoden anhand leichtverständlicher Beispiele, deshalb vorwiegend aus dem nichtnumerischen Bereich. Hierzu stellt PL/I alle Werkzeuge strukturierter Programmierung zur Verfügung.

Literatur:

Conway/Gries: An Introduction to Programming, Winthrop
Conway/Gries: Der neue Programmierstil, gezeigt an PL/I, Hanser
Mell/Preuss/Sandner: Einführung in die Programmiersprache PL/I, BI
Smolek/Weissenböck: Grundlagen der Programmiersprache PL/I, Hüthig

Programmieren in PL/I für Fortgeschrittene (320175)

Die Vorlesung wendet sich an Hörer, die bereits an einem PL/I-Grundkurs teilgenommen haben. Behandelt werden Themen wie Ein- und Ausgabe auf Platten und Bändern, Datenorganisation, Unterprogrammtechniken, Listenverarbeitung, Multitasking und die Benutzung des PL/I-Preprocessors. Der Hörer soll einen Einblick in die Möglichkeiten einer universellen Programmiersprache erhalten und in die Lage versetzt werden, effiziente und gut strukturierte Programme zu schreiben.

Literatur:

s. vorangehende Veranstaltung



Programmieren in C (320046, 320180)

C ist eine Programmiersprache, deren Einsatzmöglichkeiten einerseits durch Assembler-ähnliche Sprachelemente und andererseits durch Elemente moderner blockstrukturierter Sprachen sehr vielseitig sind. Durch einen hohen Grad an Portabilität ist C eine Sprache mit stark zunehmender Verbreitung; Compiler-Implementierungen stehen praktisch für alle Betriebssysteme und Rechnertypen zur Verfügung.

Die Lehrveranstaltung ist eine Einführung in die Programmiersprache C. Zum Verständnis sind

aber Kenntnisse einer anderen Programmiersprache kaum verzichtbar.

Literatur:

Harbison/Steele: C Reference Manual, Prentice Hall

Kernighan/Ritchie: Programmieren in C, Hanser



Programmieren in Modula-2 (320118)

Modula-2 ist eine von N. Wirth um 1980 vorgestellte Weiterentwicklung der Sprache Pascal, bei der der Sprachaufbau systematischer und die Ausdrucksmöglichkeiten vielfältiger sind. Durch das „Modul“-Konzept, das die Trennung in einen von außen sichtbaren Definitionsteil und einen von außen unsichtbaren Implementationsteil erlaubt, wird die Programmierung komplexer Systeme im Team unterstützt. Multiprogramming ist möglich („Prozeß“), und Programmkonstrukte auf niedriger Ebene ermöglichen die Umgehung des von Pascal her bekannten „strong typing“. Als weiterer Datentyp ist der Procedure-Typ vorhanden, der die Zuweisung von Prozeduren erlaubt.

Modula-2 steht sowohl in einer Großrechner-Version als auch auf PCs zur Verfügung.

Vorkenntnisse in der Programmierung werden nicht vorausgesetzt.

Literatur:

Wirth: Programming in Modula-2, Springer

Dal Cin/Lutz/Risse: Programmierung in Modula-2, Teubner

Pomberger: Softwaretechnik und Modula-2, Hanser

Pudlatz: Einführung in die Programmiersprache Modula-2, Vieweg

Dialogprogrammierung mit REXX (320228)

REXX („Restructured EXtended eXecutor“) ist eine PL/I-ähnliche Programmiersprache, mit deren Hilfe insbesondere Kommandofolgen für Betriebssysteme wie CMS auf Großrechnern mit VM/SP oder MS-DOS auf Personalcomputern zusammengestellt werden können. Leichte Erlernbarkeit, gut abgestufte Möglichkeiten zur interaktiven Fehleranalyse und -korrektur und eine klare Struktur machen REXX zu einem geeigneten Mittel, auch komplizierte Kommandofolgen schnell zu formulieren und auszutesten. In der Vorlesung soll das Programmieren von Dialogen besprochen werden, beginnend mit einfachen Dialogen bis hin zur Menü-Technik unter Benutzung des Editors XEDIT bzw. KEDIT. Grundkenntnisse in einer Programmiersprache sind erwünscht.

Literatur:

O'Hara/Gomberg: Modern Programming Using REXX, Prentice Hall

Cowliskaw: The REXX Language: A Practical Approach to Programming, Prentice Hall

IBM: Virtual Machine/System Product, System Product Interpreter Reference

Statistische Datenanalyse mit dem Programmsystem SPSS^X (320050)

Das statistische Programmsystem SPSS (Statistical Package for the Social Sciences) wird in der am Rechenzentrum implementierten Version SPSS^X vorgestellt. Mit diesem System stehen bequem aufzurufende Programme zu den gebräuchlichen univariaten und multivariaten statistischen Verfahren sowie zur Datenmanipulation zur Verfügung. SPSS wird z.B. zur statistischen Auswertung von Fragebögen eingesetzt.

In dieser Veranstaltung wird das programmtechnische Rüstzeug zur Durchführung derartiger Auswertungen vermittelt. Solide Grundkenntnisse bezüglich der anzusprechenden statistischen Verfahren sowie Kenntnisse der Anwendungsmöglichkeiten dieser Verfahren im jeweiligen Fachgebiet sind erwünscht und bei den praktischen Übungen von großem Nutzen.

Literatur:

Steinhausen/Zörkendörfer: Statistische Datenanalyse mit dem Programmsystem SPSS^X und SPSS/PC⁺, Oldenbourg Verlag (Hörerscheine bei den Autoren erhältlich)

SPSS^X: Fortgeschrittene Anwendungen
(320209)

Diese Veranstaltung stellt eine Fortsetzung des einführenden SPSS-Kurses „Statistische Datenanalyse mit dem Programmsystem SPSS^X“ dar. Die Vertiefung des Stoffes bezieht sich dabei sowohl auf die Behandlung „höherer“ Verfahren als auch auf kompliziertere „Programmierung“ und Dateienbehandlung.

Literatur:

s. vorangehende Veranstaltung

Statistische Datenanalyse mit dem Programmpaket SAS
(320156)

Vom Statistical Analysis System SAS werden in dieser Lehrveranstaltung vornehmlich jene Komponenten behandelt, die die Aufbereitung der Daten (z.B. einer Fragebogenerhebung) und statistische Auswertungen (einschließlich der Ergebnispräsentation, z.B. als Farbgrafik) in einer anwendungsfreundlichen interaktiven Programmierungsumgebung am Großrechner erlauben.

Die Teilnehmer sollen herangeführt werden, mit diesem mächtigen Werkzeug ihre Dateien aufzubereiten und zu pflegen und sodann vorprogrammierte SAS-Prozeduren aus den jeweiligen Fachgebieten aufzurufen.

Literatur:

SAS User's Guides (Basics / Statistics / FSP / Graph) der aktuellen Version

Textverarbeitung auf Mikrorechnern
(320137)

Im ersten, theoretischen Teil der Veranstaltung sollen Konzeption und Realisierungsmöglichkeiten textverarbeitender Systeme besprochen werden. Im zweiten, mehr praxisorientierten Teil, soll den Teilnehmern die Gelegenheit geboten werden, ausgewählte Systeme zu testen und miteinander zu vergleichen. Die Tests sollen auf Mikrorechnern des Universitätsrechenzentrums durchgeführt werden.

Computerunterstütztes Publizieren mit L^AT_EX
(320141)

L^AT_EX ist eine Erweiterung des Satzsystems T_EX, die das Arbeiten mit T_EX wesentlich erleichtert. Dem Autor werden z.B. fertige Layouts für Bücher, Reports und Artikel zur Verfügung gestellt, die er selbst in gewissen Grenzen seinen eigenen Vorstellungen leicht anpassen kann. L^AT_EX steht sowohl auf IBM-kompatiblen PCs als auch auf unserem Großrechner zur Verfügung.

In dieser Veranstaltung werden die Einsatzmöglichkeiten von L^AT_EX im Publikationsprozeß vorgestellt. Es wird gezeigt, wie Texte mit Editoren wie KEDIT unter DOS und XEDIT unter CMS für L^AT_EX erfaßt, mit T_EX formatiert, zur Kontrolle am Bildschirm angezeigt und auf unterschiedlichen Druckern ausgegeben werden können.

Die Hörer sollten Grundkenntnisse im Umgang mit PCs oder dem Großrechner besitzen.

Literatur:

Kopka: L^AT_EX - Eine Einführung, Addison-Wesley

Lamport: L^AT_EX - A Document Preparation System, Addison-Wesley

Partl/Schlegel/Hyna: L^AT_EX-Kurzbeschreibung

Wonneberger: Kompaktführer L^AT_EX, Addison-Wesley

Text- und Layoutstrukturen in L^AT_EX (320065)

Zusammen mit dem Makropaket L^AT_EX unterstützt das Satzsystem T_EX die Trennung der Logik- und Layout-Struktur eines Dokumentes, d.h. der Autor kann sich ganz auf den Inhalt und die logische Struktur seines Textes, der aus Textelementen wie Überschriften, Absätzen und Anmerkungen besteht, konzentrieren und muß sich nicht gleichzeitig, sondern in einem separaten Arbeitsschritt um das Aussehen dieser Textelemente im gedruckten Dokument kümmern.

In L^AT_EX sind eine Reihe von Textelementen definiert, denen mehrere Layoutstrukturen zugeordnet werden können. Ein Dokument, dessen Textelemente mit L^AT_EX-Markierungen versehen sind, kann so in unterschiedlicher Gestalt formatiert und gedruckt werden.

Die in L^AT_EX standardmäßig vorgegebenen Strukturen decken allerdings nicht den Bedarf aller Publikationsformen an unserer Universität ab, so daß es für viele Autoren unumgänglich ist, neue Textelemente einzuführen und die vorgegebene Layout-Struktur zu verändern.

Diese Veranstaltung richtet sich an Autoren, die schon erste Erfahrungen mit L^AT_EX gemacht haben und die solche Erweiterungen und Veränderungen durchführen wollen.

Literatur:

Lamport: L^AT_EX: A Document Preparation System, Addison-Wesley

Schwarz: Einführung in T_EX, Addison-Wesley

Appelt: T_EX für Fortgeschrittene, Addison-Wesley

Kopka: L^AT_EX - Eine Einführung, Addison-Wesley

Mikrorechner-Praktikum (320070)

Das Praktikum führt in den Umgang mit den technischen Komponenten eines Mikrorechners ein. Es ist eingeteilt in 10 Themenbereiche:

- Festplatte
- Diskette
- Tastatur
- RAM- und ROM-Speicher

RUM-Lehre

- V.24-Kommunikation
- Drucker
- Video
- Leistungsmessungen
- Konfiguration von Programmen
- Systemplatine

Jeweils zwei bis drei Personen beschäftigen sich in einer Stunde mit einem dieser Themenbereiche. Für die Vorlesung ist eine Voranmeldung bis zum 16.2.1990 erforderlich.

SQL - Vergleich von Implementationen (320194)

Nachdem nun auch dBASE IV SQL-fähig geworden ist, wird die Frage nach Gemeinsamkeiten und Unterschieden verschiedener SQL-Systeme immer wichtiger. Nach einer allgemeinen Einführung in den SQL-Standard sollen die Systeme dBASE, DB2, DDB/4, Oracle und SQL/DS verglichen werden. Daran schließen sich allgemeine Überlegungen zum Aufbau einer Datenbank-Anwendung an.

Literatur:

Achilles: SQL - Standardisierte Datenbanksprache vom PC bis zum Großrechner, Oldenbourg

Achilles/Kämmerer: Benutzerhandbuch SQL, Software-Information 18

Rechnernetze (320213)

Die Entwicklung der Mikrocomputer zu leistungsfähigen autonomen Arbeitsplatzrechnern, der verstärkte Einsatz zahlreicher Mikrocomputer innerhalb einer Organisation und die damit einhergehende Dezentralisierung von Rechenleistung erfordert effiziente Möglichkeiten des Informationsaustausches zwischen Rechnern auch unterschiedlicher Typen und Hersteller und der Informationssammlung auf zentralen EDV-Anlagen. Dies hat die Entwicklung und Verbreitung lokaler Rechnernetze sowie deren Anbindung an bestehende, große Entfernungen überbrückende, globale Rechnernetze intensiv vorangetrieben. In der Veranstaltung soll auf logische und technische Grundlagen sowie theoretische Konzeptionen vor

allen lokaler Rechnernetze eingegangen werden. Der Praxisbezug soll durch Anwendungsbeispiele aus dem Bereich der Universität hergestellt werden. Dazu werden z.B. der Betrieb der Ausbildungsrechnernetze („CIP-Pools“), die Ansätze für ein hochschulweites Datenverbundsystem und der Zugang zu Datenbanken über öffentliche Netze behandelt.

Literatur:

Meißner: Arbeitsplatzrechner im Verbund, Hanser

Schnupp: Rechnernetze – Entwurf und Realisierung, de Gruyter

Hopper, Temple, Williamson: Local Area Network Design, Addison-Wesley

Betriebssystemdienste für interaktive Anwendungen (320232)

Um dem Benutzer den Umgang mit dem Rechner zu erleichtern, werden in zunehmendem Maße Dialogschnittstellen angeboten. In dieser Vorlesung wird anhand praktischer Beispiele gezeigt, wo Dialoge benutzt werden können und wie man Dialoge programmiert.

Programmierkenntnisse werden vorausgesetzt.

Kolloquium über Themen der Informatik (320247)

Im Rahmen des Kolloquiums werden Vorträge über neuere Entwicklungen der Informatik gehalten. Vortragstermine werden durch Aushang im Universitätsrechenzentrum bekanntgegeben.

RUM-Text

Neues von TEX

von

W. Kaspar

Über neue Wege der Verteilung, neue Hilfsprogramme und Styles und auf Neues in der Zukunft darf sich der TEX-Anwender freuen.

Seit der Einführung von TEX für den PC vor drei Jahren hat das Universitätsrechenzentrum im Rahmen einer Campuslizenz mehr als 100 Kopien dieses Satzsystems an universitäre Einrichtungen und über eine zusätzliche persönliche Nutzungsoption mehr als 300 Kopien an Universitätsangehörige weitergegeben.

Durch die Verbreitung von TEX auch an vielen anderen Universitäten in Deutschland und im westlichen Ausland wächst auch immer mehr der Kreis derjenigen TEX-Anwender, die eigene Styles, Schriften und Hilfsprogramme für TEX entwickeln.

Auf Grund der weltweiten Vernetzung der EDV-Anlagen verbreiten sich Nachrichten über solche Entwicklungen recht schnell, und auch die Programme können, soweit sie von den Autoren kostenlos zur Verfügung gestellt werden, über solche Netze leicht beschafft werden.

Verfügbarkeit der neuesten TEX-Entwicklungen an der WWU

Im PC-Pool des Universitätsrechenzentrums steht seit kurzem ein Programm für die Verteilung Allgemeiner Mikrocomputer-Programme (VAMP) zur Verfügung, mit dem nun auch das gesamte TEX-System an alle PC-Benutzer regelmäßig verteilt werden kann.

Jede an TEX interessierte universitäre Einrichtung und jeder Universitätsangehörige, der TEX auf seinem privaten PC nutzen möchte, kann eine Benutzerkennung für VAMP beantragen. Die in VAMP gespeicherten TEX-Installationsdisketten können dann an jedem PC, der auf den Server des

Universitätsrechenzentrums Zugriff hat, auf eigene Disketten kopiert werden.

Die folgende Liste gibt einen kurzen Überblick der zur Zeit in VAMP verfügbaren TEX-Software. Ausführlicheres Dokumentationsmaterial ist auf den Installationsdisketten abgelegt. Eine Kurzanleitung zu VAMP finden Sie in diesem *Inform* („Neue Wege zur Verteilung von Programmsystemen mit Campus-Lizenz.“). „TEX-Installationshinweise für den PC“ liegen ebenfalls in gedruckter Form vor.

PCTEX Version 2.93: das zentrale Satzprogramm. Gegenüber der bisherigen Version 2.0 wurden einige Fehler im TEX-Programmcode beseitigt, die sich allerdings nur in sehr seltenen Fällen bemerkbar machten.

Außerdem kann mit der neuen Version die Struktur der Unterverzeichnisse flexibler gestaltet werden, da unter anderem für die TEX-, TFM- und FMT-Dateien mehrere Zugriffspfade angegeben werden können.

LATEX Version 2.09 vom 14.01.90: das Standard-Makropaket, das für nahezu alle Publikationsarten eingesetzt werden kann.

SLITEX: ein Makropaket zur Erstellung von Folien für Overheadprojektoren.

BIBTEX: ein Dienstprogramm, das durch die Verwaltung einer Literaturdatenbank und in Zusammenarbeit mit LATEX das Zusammenstellen von Bibliographien erleichtert.

MAKEINDEX: ein Dienstprogramm, das in Zusammenarbeit mit LATEX die automatische Generierung eines Buchindex unterstützt.

Zusatzstile für LATEX: wie z.B. für den Satz von

- zusätzlichen diakritischen Zeichen,
- Texten, die unformatiert gesetzt werden sollen, wie z.B. Quelltexte von Computerprogrammen,
- Seitenkopf- und Fußzeilen in unterschiedlichen Layouts,

- Baumdiagrammen,
- kommutativen Diagrammen,
- chemischen Formeln und
- mehrseitigen Tabellen.

Ausgabegerätetreiber für

- Bildschirme in Verbindung mit Hercules-, EGA-, VGA- und anderen Grafikkarten,
- 9- und 24-Nadeldrucker,
- HP LaserJet-Drucker,
- HP DeskJet-Drucker und
- hierzu kompatible Drucker.

Schriften aus der „Computer Modern“-Familie

- in Schriftstilen wie z.B. Fett, Kursiv, mit und ohne Serifen, Typewriter und Kapitälchen,
- in Schriftgrößen von 5 bis 25 Punkt, für SLITEX von 14 bis 41 Punkt (nur ohne Serifen) und
- in Auflösungen von 100, 150, 180, 240 und 300 dpi.

Ausblick auf zukünftige TEX-Entwicklungen

Wie schon oben erwähnt, beschäftigen sich weltweit TEX-Anwender mit der Entwicklung neuer Produkte für TEX.

Einige davon werden im folgenden kurz aufgezählt:

- eine nahezu geräte- und treiberunabhängige Einbindung grafischer Objekte in TEX-Texte,
- LATEX-Briefstil mit Universitätsbriefkopf,
- LATEX-Stil für komfortablen Satz mehrerer Spalten,
- Erweiterungen zum LATEX-Tabellensatz,
- Implementierung der AMSTEX-Makros in LATEX,
- Ausgabegerätetreiber für 24-Nadel-Drucker am PC mit 360-dpi-Fonts, und VGA-Grafikkarten mit einer Auflösung von 600x800,
- Einbindung von Bitstream-Schriften in LATEX,
- Schriften und LATEX-Makros für griechische, kyrillische und außereuropäische Texte, auch in Rechts-Links-Schreibweise,

- TEX Version 3.0, mit Erweiterungen zur noch flexibleren Verarbeitung fremdsprachiger Texte,
- Notensatz.

TEX-Anwendervereinigungen

Die TEX Users Group (TUG) ist der älteste weltweite Zusammenschluß von TEX-Benutzern. Seit 1980 unterstützt dieser Verein verschiedene TEX-Entwicklungen und bietet mit seiner Zeitschrift TUGboat ein wichtiges Informations- und Diskussionsmedium. Seit der neuesten Ausgabe von Ende 1989 werden neben den normalerweise englischsprachigen Artikeln auch Kurzfassungen in Deutsch abgedruckt.

Die Mitgliedschaft kostet zur Zeit \$ 45. Interessenten können an folgende Adresse schreiben:

The TEX Users Group
P. O. Box 9506
Providence, Rhode Island 02940
U.S.A

Zusätzlich sind in den letzten beiden Jahren noch national bzw. sprachlich begrenzte TEX-Anwendervereine gegründet worden, so in Frankreich, Holland, England und Skandinavien.

Für den deutschsprachigen Raum wurde im April 1989 der Verein DANTE, Deutschsprachige Anwendervereinigung TEX, gegründet. Zweck dieses Vereins ist die Beratung bei der Anschaffung, Implementation und bei Anwendungsproblemen von TEX, aber auch die Verteilung von Public-Domain-Software an die Vereinsmitglieder. Über eine 1/4-jährlich erscheinende Vereinszeitschrift werden Neuigkeiten aus der TEX-Welt vorgestellt.

Der Einzelbetrag für die Mitgliedschaft einer Privatperson liegt bei DM 60, für Studierende bei DM 40. Die Anschrift des Vereins lautet:

DANTE
Deutschsprachige Anwendervereinigung TEX e.V.
Rechenzentrum der Universität Heidelberg
Im Neuenheimer Feld 293
D-6900 Heidelberg 1

Für weitere Auskünfte zu allen in diesem Artikel erwähnten Themen kann sich jeder Interessent gerne an mich wenden (URZ86 at DMSWWU1A).

RUM-Grafik

Grafik am Universitätsrechenzentrum

von

E. Sturm

Für all die, die gerade überlegen, wie sie am besten an grafische Darstellungen ihrer Daten kommen, wird im folgenden eine Übersicht der Möglichkeiten gegeben.

Wer sich in der Grafikwelt umschaute, wird mit einer Unzahl von Geräten und Programmsystemen konfrontiert, die alle mehr oder weniger inkompatibel sind. Im Bereich der genormten Software spielt GKS (Grafisches Kern-System) als ISO- und DIN-Norm die wichtigste Rolle.

Das Universitätsrechenzentrum stellt zum einen zentral ein Grafik-System zur Verfügung und berät zum anderen auch, wie man am PC grafische Datenverarbeitung betreiben kann. Für den Grafik-Benutzer sollte es zunächst egal sein, ob er PC oder Großrechner verwendet, Hauptsache, die Bilder können leicht programmiert werden und liegen danach in der gewünschten Qualität vor. Ich möchte zunächst das zentrale Grafik-System vorstellen, danach komme ich dann auf die Möglichkeiten am PC zu sprechen.

Das Zentrale Grafik-System

Auf dem Großrechner sind je nach Anwendungsgebiet verschiedene „Schienen“ zu unterscheiden:

I. Die Hauptschiene ist ein Grafiksystem, das vom Programm über Plot-Preview bis zum fertigen Bild eine vollständige Kette bildet und auf unserer GKS-Implementierung RUM/GKS beruht. Diese Kette im einzelnen:

1. Zunächst zur Erzeugung des Bildes:

- FORTRAN- und PL/I-Programmierer tätigen GKS-Aufrufe, z.B. des Unterprogramms POLYLINE und erzeugen damit entweder eine CMS-Datei des Typs PLOT oder eine Spool-Datei im MVS (Näheres kann man über `help rumgks` erfragen).

- Für SAS-Benutzer gibt es ebenfalls eine Möglichkeit, zu einer solchen PLOT-Datei zu gelangen (Näheres bei Herrn Dr. Zörkendörfer).

- Wer nicht programmieren will, der kann im CMS die Plot-Datei auch mit dem CHART-Kommando (früher SHOW) erzeugen (siehe *inforum* Nr. 3/1989).

- SPSS-Benutzer wiederum können eine Ausgabe erzeugen, die als Eingabe vom CHART-Kommando akzeptiert wird (Näheres bei Herrn Dr. Steinhäuser).

- Ohne Programmierkenntnisse kommt auch aus, wer Diagramme mit Hilfe des CMS-Kommandos POSTER erzeugt (siehe den Artikel in diesem *inforum*).

2. Man kann sich seine Bilder vorher anschauen (Plot-Preview):

- Dies geht einmal im CMS mit Hilfe des PPV-Kommandos an einem grafikfähigen Terminal (einschließlich PC als GVT-Terminal), im MVS an einem der zentralen Grafik-Terminals (siehe *inforum* Nr. 4/1988).

- Eine andere Möglichkeit bietet das in *inforum* Nr. 2/1989 vorgestellte PPV-Programm, das auf einem über Ethernet angeschlossenen PC läuft.

3. Möchte man das Bild geplottet haben, so gibt es wieder mehrere Möglichkeiten:

- Im CMS zeigt man dies an mittels des PLOT-Kommandos (oder schon direkt beim CHART-Kommando, siehe auch `help plot`).

- Im MVS macht man das entsprechende Kreuz in der Plot-Preview-Sitzung.

- Wer sich der Korrektheit seiner Bilder sicher ist, der kann im MVS auch statt Preview die JCL-Prozedur PLOTSERV aufrufen.

In jedem Fall hat man die Auswahl zwischen einem Stift-Plotter (Pen-Plotter) mit

einer maximalen Zeichenfläche von 68cm x 100cm und einem Monochrom-Rasterplotter (Agfa P400) im Format DIN A4 (siehe Inform Nr. 4/1986).

4. Das Bild landet dann entweder im üblichen Ausgabefach oder in einem der speziellen Plotterfächer in der „Eingangshalle“ des Rechenzentrums.

Interaktiv kann man GKS nur von PL/I aus betreiben, wiederum an den grafikfähigen Terminals (denselben wie beim PPV-Kommando, da sich hinter diesem ein interaktives PL/I-Programm verbirgt, siehe auch help rumgks oder Inform Nr. 4/1989).

- II. Noch nicht in die eben beschriebene Kette gehört unser Farbrasterplotter. Dies hat seinen Grund u.a. darin, daß unser momentanes Plot-File nur Linien in maximal 3 Farben kennt, also noch sehr stiftplotterorientiert ist. Die Vorteile eines Rasterplotters liegen aber in seiner Möglichkeit, Flächen zu färben. Diese Möglichkeiten kann man auf einer zweiten Schiene ausnutzen. Grundlage ist hier nicht das übliche Plot-File, sondern ein COLPLOT-File. Die Reihenfolge ist aber ähnlich:

1. Man stellt (per Programm) eine Farbmatrix zusammen, z.B. aus Satellitendaten.
2. Für das Anschauen auf einem Farbrasterterminal (IBM 5080 bzw. IBM 3192) gibt es das COLPLOT-Kommando.
3. Die Erzeugung des Bildes erfolgt mit der JCL-Prozedur COLPLOT im MVS.
4. Das fertige Bild kann man dann im Dispatch abholen (siehe Inform Nr. 4/1988).

- III. Die dritte Schiene ist für dreidimensionale Plots vorgesehen:

1. Man erstellt (per Programm) eine Koordinatendatei.
2. An der IBM 5080 kann man sich mit Hilfe einer Rot-Grün-Brille und den Drehknöpfen die Schokoladenseite des Bildes suchen und abspeichern. Dazu gibt es das Kommando VIEW3D. Möglich sind Punktwolken und „Drantgebilde“.

3. In einem MVS-Job mit der JCL-Prozedur PLOT3D wird danach ein Rot-Grün-Bild auf dem Farbrasterplotter erzeugt.

4. Nach dem Abholen im Dispatch kann man mit einer Rot/Grün-, besser noch mit einer Rot/Blau-Brille sein Werk betrachten (siehe Inform Nr. 3/1988).

- IV. Die letzte Schiene am Großrechner führt über das CMS-Kommando PIC400. Dieses Kommando hilft einem, wenn man Bilder in DCF- oder TEX-Dokumente einbinden will. Ausgangspunkt kann das übliche Plot-File, eine Bitmap oder eine Scanner-Datei des Scanners Agfa S400PC sein (siehe Inform Nr. 1/1989 für TEX, Nr. 2/1988 für DCF, sowie unter help pic400).

Auf diese Schiene gelangt man auch, wenn man das IBM-Programmprodukt GDDM (Graphical Data Display Manager) oder das darauf aufsetzende Produkt ICU (Interactive Chart Utility) benutzt (siehe Inform Nr. 4/1989). Der Umweg erfolgt über eine BITMAP-Datei.

Grafiksysteme auf dem PC

In den letzten Jahren ist eine große Zahl von PC-Programmen auf den Markt gekommen, die auf komfortable Art das interaktive Erstellen von Bildern ermöglichen. Man kann vier Arten unterscheiden, wobei durchaus Überschneidungen vorhanden sind:

1. Programme zum Malen von Rasterbildern (z.B. MS-PAINT),
2. Programme, die Präsentationsgrafiken erzeugen (z.B. Harvard Graphics),
3. Programme für technische Zeichnungen (z.B. AutoCAD), und
4. Programme zum Entwerfen von Grafiken (z.B. Corel DRAW).

Hiermit ist aber nicht gemeint, daß diese Programme „nur“ Grafik erzeugen können. Manchmal ist die Grafik nur ein Nebenprodukt z.B. eines Statistik-Pakets.

Will ein Benutzer die Zeichnungen ausgeben, so braucht er sich nur einen Drucker bzw. Plotter anzuschaffen, für den der Software-Lieferant

einen sogenannten Treiber mitgeschickt hat. Oftmals soll ein auf dem PC erzeugtes Bild aber auch in DCF- oder TEX-Dokumente eingebunden oder auf einem hochwertigen Plotter des Rechenzentrums ausgegeben werden. In diesem Fall muß eine Bilddatei an den zentralen Rechner übertragen werden. Das Problem ist jetzt, ein Format zu finden, für das man einen Treiber besitzt und das auch im Großrechnerbereich verstanden wird.

Dieses Problem ist leider nicht hundertprozentig lösbar. All diese Programme sind zwar in der Lage, Bilddateien eines oder mehrerer Formate auszugeben, nur in den seltensten Fällen ist darunter das genormte Format CGM (Computer Graphics Metafile). Also muß man sich zur Zeit mit irgendeinem der sogenannten Industrie-Standards zufriedengeben. „Verstehen“ tut der Großrechner bis jetzt schon das DXB-Format von AutoCAD. Wir arbeiten am DXF- und am HPGL-Format. Im nächsten *Inform* wird darüber zu berichten sein. Auch das CGM-Format wird natürlich hinzukommen. Wer diesbezüglich konkrete Wünsche hat, kann sie gerne bei mir anmelden (URZ31, Tel. 2609)!

In Vorbereitung befindet sich ein anderer Ansatz zum Einbinden von Bildern in TEX-Dokumente. Während man beim PIC400-Kommando im CMS Bilder nur für den Agfa P400 vorbereiten kann, ermöglicht das Programm RUMPIXL das Einbinden von Bildern, die im HP-PCL-Format vorliegen, für beliebige Geräte, also auch für PC-Drucker.

Sollten noch Fragen offen sein, so wenden Sie sich bitte an die Programmierberatung (Tel. 2486), dort können alle zitierten *Inform*s eingesehen werden.

Explodierende Torten

von

B. Samland

CHART-Kommando und RUM/GKS in PL/I sind erweitert worden.

Bei Kreissektordiagrammen ist es jetzt möglich, ein Stück aus der Torte herauszuziehen. Dazu wird bei `Generate_Piechart` als letzter Parameter die Nummer des explodierenden Sektors angegeben. Voreingestellt für die Explosionsweite ist $\text{Kreisradius} / 2$. Möchte man die Weite verändern, so gibt man im letzten Parameter bei `Define_Piechart` an, wie weit das Tortenstück im Verhältnis zum Radius explodieren soll.

Ein Beispiel: Die Aufrufe

```
call Define_Piechart (Startwinkel,
                     Strichlänge, Texthöhe, 0.6);
call Generate_Piechart ('OUTSIDE',
                       X_Mitte, Y_Mitte, Radius, Werte,
                       Texte, Schraffuren, Farben, 5);
```

erzeugen

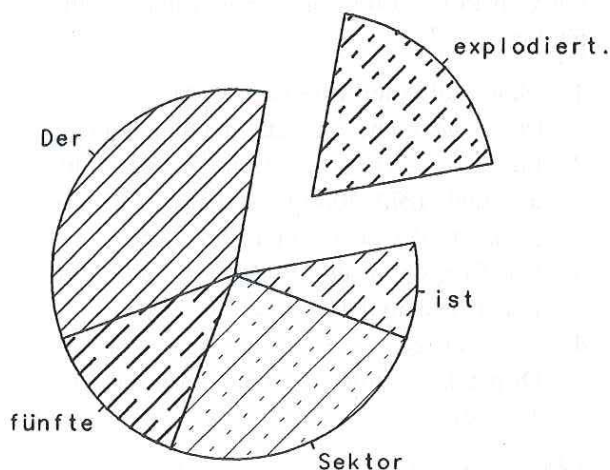


Abbildung 2.

wobei Werte, Texte, Schraffuren und Farben Arrays der Typen float, char (*) var bzw. bin fixed sind.

Auch mit Hilfe des CHART-Kommandos kann man Torten explodieren lassen. Dazu trägt man im Define-Piechart-Formular die gewünschten Werte bei `EXPL_PART` und `EXPL_FACTOR` ein.

Ein neues POSTER

von

H. Brinkmann

Jetzt können POSTER-Diagramme auch mit Platten und Bändern versehen werden.

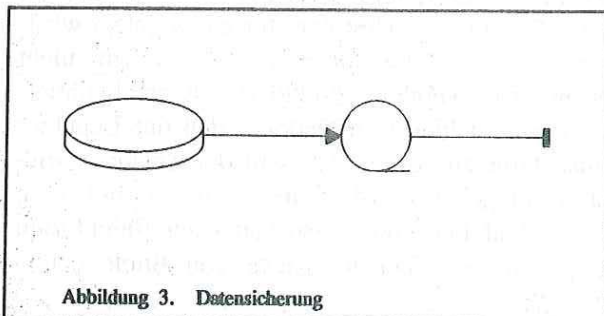
Bei dem in **forum** Nr. 1/88 und 3/89 beschriebenen Programm POSTER zum Zeichnen von Diagrammen u.ä. sind einige Erweiterungen vorgenommen worden.

Es sind zwei neue „Kästchen“-Typen hinzugekommen. Diese stellen Symbole für Magnetplatte bzw. Magnetband dar. Ihre Schlüsselwörter heißen DISK und TAPE. Die möglichen Parameter sind dieselben wie bei allen anderen „Kästchen“, also Name, horizontale und vertikale Position sowie Breite und Höhe.

Außerdem kann man am Ende von Verbindungslinien einen Querbalken, der das Null-Symbol darstellen soll, zeichnen lassen. Dazu muß man beim Schlüsselwort CONNECT innerhalb der Klammern an Stelle des A (Arrow) ein N angeben. Dahinter kann wie bei einer Pfeilspitze noch eine Größenangabe folgen. Gebrauchen kann man das Null-Symbol z.B. bei Pointerlisten.

Mit dem folgenden Beispiel wurde die Abbildung 3 erzeugt.

```
SIZE 14 4
DISK 1 2 2 4 2
TAPE 2 9 2 2 2
POINT 3 14 2
CONNECT 1 2 (A .5)
CONNECT 2 3 (N .5)
```



Kreise und Ellipsen mit RUM/GKS

von

H. Stubbe

Der Kreis der RUM/GKS-Routinen wurde für PL/I-Programmierer um Ellipsen erweitert.

Hier stelle ich zwei Routinen vor, die in der Programmiersprache PL/I geschrieben wurden. Mit ihnen können Kreise, Ellipsen sowie Teile von Kreisen und Ellipsen gezeichnet oder berechnet werden. Zum Beispiel kann man einen Plattenstapel aus Ellipsen und Ellipsenbögen zusammenstellen, wie Abbildung 4 auf Seite 28 zeigt. Bevor der Aufruf durch den GKS-Benutzer vorgestellt wird, soll zum näheren Verständnis dieser Routinen das Prinzip der Berechnung angegeben werden (in kleiner Schrift):

Das Prinzip, das hinter der Berechnung steht, ist die lineare Approximation des Kreises bzw. der Ellipse durch ein Polygon. Dazu müssen einige Randpunkte berechnet werden, bevor RUM/GKS diese zu einem Polygonzug verbinden kann.

Kreise sind bei gleicher Qualität leichter zu realisieren als Ellipsen, da nur einmal die Winkelfunktionen Sinus und Cosinus berechnet werden müssen und danach alle benötigten Punkte aus ihren unmittelbaren Vorgängern direkt berechnet werden können. Bestimmt werden müssen lediglich Sinus und Cosinus des Winkels α , der zwischen den Verbindungen von Mittelpunkt und Randpunkten auftritt. Dieser wird konstant gewählt. Mit den Sinus- und Cosinuswerten von α kann nun über die Additionstheoreme von Sinus und Cosinus aus jedem Randpunkt P_i der Punkt P_{i+1} berechnet werden. Der Winkel α kann über den maximal zugelassenen Approximationsfehler bestimmt werden. Der evtl. angegebene Anfangspunkt wird einfach als P_1 zugewiesen und der Endpunkt wird direkt mit dem vorletzten berechneten Punkt verbunden.

Etwas komplizierter ist die Berechnung von Ellipsen. Das Prinzip ist zwar ähnlich dem bei Kreisen, jedoch besteht die Hauptschwierigkeit darin, daß der Winkel α nicht konstant gewählt werden kann. Dies liegt daran, daß die Ellipse an dem Schnittpunkt der einen Achse eine wesentlich größere Krümmung aufweisen kann als am anderen und bei starker Krümmung mehr Punkte nötig sind, um gleich genau zeichnen zu können. So nutzt man die Eigenschaft der Ellipse, daß sie gewissermaßen als ein langgezogener Kreis angesehen werden kann. Dies führt allerdings zu einem zweiten Hindernis, wenn man die Ellipse mit wenig Rechenaufwand berechnen will. Da der Kreis nicht nur entlang der X-Achse gestreckt sein kann, sondern in jede beliebige Richtung, berechnet man den Winkel der Streckrichtung. Jetzt kann bei der Rechnung von einem Kreis ausgegangen werden, der erst entlang der X-Achse gestreckt und dann um den Winkel der Streckachse zur X-Achse gedreht wird. Der zugrundeliegende Kreis wird genauso wie oben berechnet. Wenn nur ein Teil einer Ellipse berechnet werden soll, so werden Anfangs- und Endpunkt bezüglich des zugrundeliegenden Kreises bestimmt.

Um die Routinen aufrufen zu können, muß der Benutzer in seinem PL/I-Programm folgenden Befehl einbauen:

```
%include GKSCIR;
/* für Kreise bzw. */
%include GKSELL;
/* für Ellipsen */
```

Damit stehen dem GKS-Benutzer jeweils für Kreise und Ellipsen drei Aufrufe zur Verfügung. Dies sind für Kreise

1. Generate_Circle
2. Inquire_Circle
3. Define_Circle

sowie für Ellipsen

1. Generate_Ellipse
2. Inquire_Ellipse
3. Define_Ellipse

Da gleichnumerierte Routinen in beiden Fällen einander entsprechen, werden hier nur die Routinen für Kreise beschrieben.

Generate_Circle berechnet die Randpunkte des Kreises und plottet mit ihnen die approximierende Kurve. Define_Circle definiert den maximalen Fehler, der bei der Approximation erlaubt ist. Inquire_Circle berechnet die Randpunkte des Kreises, die zur Approximation nötig wären, plottet jedoch nicht, sondern gibt sie dem Benutzer zur Weiterverarbeitung zurück.

Für den Aufruf von Generate_Circle gibt es zwei Möglichkeiten:

```
call Generate_Circle (Center_X,
Center_Y, Edge_X, Edge_Y);
```

oder

```
call Generate_Circle (Center_X,
Center_Y,
Start_X, Start_Y, End_X, End_Y )
```

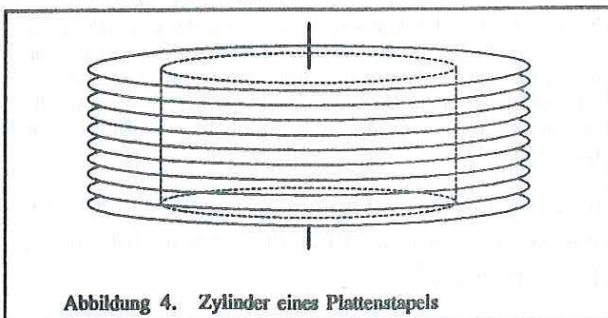


Abbildung 4. Zylinder eines Plattenstapels

Center_X und Center_Y (deklariert als FLOAT) geben den Mittelpunkt des Kreises an. Edge_X und Edge_Y (ebenfalls als FLOAT deklariert) geben einen beliebigen Randpunkt des Kreises an. Start_X und Start_Y geben den Anfangspunkt und End_X und End_Y (alle als FLOAT deklariert) den Endpunkt des Kreisbogens im Gegenuhrzeigersinn an. Alle Parameter werden als Weltkoordinaten im GKS-Sinne angegeben.

Falls der Benutzer die Zeichengenauigkeit verändern will, so kann er dies mit folgendem Aufruf tun:

```
call Define_Circle (Epsilon)
```

Epsilon (deklariert als FLOAT) ist das Maß für die Genauigkeit der zu zeichnenden Kurve. Es gibt den maximalen Abstand zwischen dem Polygonzug, der aus Sekanten des Kreises besteht, und dem Kreis selbst an (in Weltkoordinaten). Der implizite Wert von Epsilon ist so klein gewählt (0.00005), daß ihn der Benutzer im Normalfall nicht verändern muß.

Mit der Routine Inquire_Circle besteht die Möglichkeit, die Kreispunkte zu berechnen und selbst weiterzuverarbeiten. Der Aufruf lautet:

```
call Inquire_Circle(Center_X,
Center_Y, Edge_X, Edge_Y,
Back_X, Back_Y);
```

oder

```
call Inquire_Circle(Center_X,
Center_Y, Start_X, Start_Y,
End_X, End_Y, Back_X, Back_Y);
```

Dabei sind Center_X, Center_Y, Edge_X, Edge_Y, Start_X, Start_Y, End_X und End_Y als FLOAT deklariert. Back_X und Back_Y sind als (*) FLOAT CONTROLLED deklariert, d.h. daß erst in der Routine Speicherplatz für sie angelegt wird. Dies ist notwendig, da vor dem Aufruf nicht bekannt ist, wieviele Punkte der Kreis benötigt. Der Nachteil hierbei ist jedoch, daß der Benutzer selbst dafür zu sorgen hat, daß der Speicher wieder freigegeben wird. Dies kann er mit dem PL/I-Befehl free tun. Also lautet der Befehl zum Freigeben des Speicherplatzes von Back_X und Back_Y:

```
free Back_X, Back_Y;
```

Die Routinen `Generate_Ellipse`, `Define_Ellipse` und `Inquire_Ellipse` werden ähnlich wie die Kreisroutinen aufgerufen. Der Unterschied besteht darin, daß bei Ellipsen zwei Radien und zusätzlich evtl. Start- und Endpunkt angegeben werden müssen. Also lauten die Aufrufe

von `Generate_Ellipse`:

```
call Generate_Ellipse (Center_X,  
    Center_Y,  
    Axis_1_X, Axis_1_Y,  
    Axis_2_X, Axis_2_Y);
```

oder

```
call Generate_Ellipse (Center_X,  
    Center_Y,  
    Axis_1_X, Axis_1_Y,  
    Axis_2_X, Axis_2_Y,  
    Start_X, Start_Y,  
    End_X, End_Y);
```

von `Define_Ellipse`:

```
call Define_Ellipse (Epsilon);
```

und von `Inquire_Ellipse`:

```
call Inquire_Ellipse (Center_X,  
    Center_Y,  
    Axis_1_X, Axis_1_Y,  
    Axis_2_X, Axis_2_Y,  
    Back_X, Back_Y);
```

oder

```
call Inquire_Ellipse (Center_X,  
    Center_Y,  
    Axis_1_X, Axis_1_Y,  
    Axis_2_X, Axis_2_Y,  
    Start_X, Start_Y,  
    End_X, End_Y,  
    Back_X, Back_Y);
```

Auch hier sind alle Parameter außer `Back_X` und `Back_Y` als `FLOAT` deklariert. `Back_X` und `Back_Y` sind deklariert als (*) `FLOAT CONTROLLED`. Die Parameter `Axis_1_X`, `Axis_1_Y`, `Axis_2_X` und `Axis_2_Y` geben jeweils die X- bzw. Y-Koordinaten der Schnittpunkte der beiden Ellipsenachsen mit dem Ellipsenrand an.

Für die Plausibilität der Daten hat der Benutzer selbst zu sorgen. Tut er dies nicht, so versuchen die Routinen trotzdem noch irgendwelche Kreise oder Ellipsen zu produzieren. Ob dies aber im Sinne des Benutzers ist, muß er selbst entscheiden.

Die hier vorgestellten Routinen sind in PL/I geschrieben und momentan nur von PL/I aus zu benutzen. Es ist beabsichtigt, sie auch nach FORTRAN zu übersetzen. Falls der Bedarf dringend ist, wende man sich bitte an Herrn Sturm (URZ31, Tel. 2609).

RUM-Aroma

inforum-Quiz

von

E. Sturm

Das letzte **inforum**-Quiz war ein Beispiel dafür, daß in PL/I allgemeine Prinzipien auch dann gelten, wenn das Ergebnis etwas verblüffend ist. Bei der Anweisung

```
if A = ¬A then put
('A ist wahr und falsch zugleich!');
```

ist völlig klar, daß jedes Bit der Variablen A durch den Nicht-Operator umgedreht wird, der Vergleich von A und ¬A also nicht wahr ergeben kann, wenn die Variable auch nur ein Bit enthält. Damit ist die Lösung klar: A ist ein Nullstring! Möglich ist z.B. die Deklaration

```
declare A bit (0);
```

oder auch

```
declare A bit (10) var init ('b');
```

In beiden Fällen ist kein Bit vorhanden, das umgedreht werden könnte und Nullstrings sind einander gleich: die obige put-Anweisung wird also ausgeführt.

Selbst wer nicht wußte, das es variierende Bitstrings gibt, mit Zeichenfolgen ging es auch, da PL/I bereit ist, Zeichen-Nullstrings in Bit-Nullstrings umzuwandeln.

```
declare A char (10) var init ('');
```

Mit anderen Worten, die Kenntnisse eines PL/I-Grundkurses reichten zum Lösen des Quiz.

Das letzte Quiz war so kurz, damit es nicht so auffällt, wenn ich jetzt sage, daß es (vorerst?) keine weitere Fortsetzung der Reihe gibt. Falls jedoch ein Leser ein interessantes Problem weiß (es muß nicht in PL/I sein), möge er doch einen entsprechenden Leserbrief schreiben. Es gibt sogar einen CMS-Befehl namens inforum, der einen dabei unterstützt.

Die Statistik-Seite

Verbrauchsdaten 1989

von

Alfons Ahrens

In der folgenden Tabelle sind die Verbrauchsdaten der einzelnen Benutzerkreise des Rechenzentrums der Universität Münster im Jahre 1989 in Prozent der jeweiligen Gesamtsumme angegeben. Die Strichlänge bezieht sich auf Plots, die auf dem PEN-Plotter erzeugt wurden.

Benutzerkreis	Dialog			Batch				
	Sitzungs-Zeit	CMS-CPU	UNIX-CPU	MVS-CPU	I/O Services	Jobzahl	Zeilen	Plot-Str.-länge
FB 1	0.13	0.47	.	0.07	0.02	0.10	1.11	.
FB 2	0.06	0.73	.	.	.	.	0.16	.
FB 3	0.01	0.13	.	.	.	.	0.01	.
FB 4	2.42	4.89	.	4.37	4.94	7.30	6.07	0.36
FB 5	3.41	6.65	.	0.96	6.68	9.31	9.76	7.83
FB 6	3.47	2.88	.	0.16	0.82	2.10	2.54	0.01
FB 7	2.81	4.24	.	0.01	0.02	0.12	3.14	.
FB 8	1.10	3.03	.	0.37	1.55	5.37	4.44	0.06
FB 9	1.45	1.21	.	0.50	2.20	2.23	2.47	.
FB 10	0.07	0.47	.	0.02	0.22	0.38	0.58	2.51
FB 11	0.38	0.95	0.14	0.01	0.05	0.10	0.35	12.19
FB 12	1.87	0.25	.	0.05	0.30	0.38	0.23	.
FB 13	0.01	0.02	.	.	.	.	0.02	.
FB 14	0.14	0.47	.	.	.	.	0.13	.
FB 15	9.86	10.49	0.20	2.25	5.14	2.47	0.96	0.53
FB 16	21.97	16.71	98.23	43.54	9.92	26.23	3.50	6.42
FB 17	9.68	4.64	0.01	43.51	31.07	9.28	3.76	2.09
FB 18	0.01	0.03	.	.	.	.	.	.
FB 19	7.02	8.66	.	1.59	16.20	11.30	4.27	50.25
FB 20	0.45	1.04	.	0.02	0.12	0.29	0.66	.
FB 21	0.01	0.09	.	.	.	0.01	.	.
URZ und UB	19.09	19.98	0.21	0.78	3.21	5.31	5.32	6.32
ZUV und VMK	6.73	5.74	0.01	1.30	14.97	11.39	44.40	11.18
SF-Bereiche	0.56	0.96	.	0.40	1.82	1.70	3.04	.
Kurse	7.12	4.85	1.20	0.08	0.65	4.29	2.52	0.24
Sonstige	0.14	0.41	.	0.02	0.09	0.24	0.57	.

Stichwörter **Inform** Jahrgang 13**AGFA-P400-Drucker**

13,2-15	H. Kamp/ M. Sandkötter	Ausdruck von WordPerfekt-4.1-Dateien auf dem Agfa-P400-Drucker
13,3-20	W. Kasper	Laserbelichter

Beratung

13,2- 3	-	Änderung der Bibliothekszeiten im Sommersemester 1989
13,2- 3	-	Organisationsplan des Universitätsrechenzentrums

Betriebssystem

13,1- 4	A. Achilles	Mehr Megabyte im CMS
13,1- 7	A. Achilles	Neue Versionen des Betriebssystems CMS
13,2- 3	W. Kasper	TeX-Preview unter CMS
13,2- 6	W. Stockhofe	Rechenzeit und Plattenspeicher im UTS
13,2- 7	W. Lange	OS/2 aus der Sicht des MS-DOS-Anwenders
13,3- 3	A. Achilles	Umgebungen im CMS
13,3- 8	B. Neukäter	Mehr Plattenplatz im MVS
13,4- 5	A. Achilles	CMS Release 5

Campuslizenzen

13,1- 4	H.W. Kisker	Erweiterung der Lizenzliste um SPSS/PC+
13,1- 5	S. Zörkendörfer	PC-Lizenz zu SPSS-Produkten

CHART

13,3-25	B. Samland	Neue SHOW mit CHART
---------	------------	---------------------

CMS

13,1- 4	A. Achilles	Mehr Megabyte im CMS
13,1- 7	A. Achilles	Neue Versionen des Betriebssystems CMS
13,2- 3	W. Kasper	TeX-Preview unter CMS
13,3- 3	A. Achilles	Probleme mit dem PEEK-Kommando?
13,3- 3	A. Achilles	Umgebungen im CMS
13,4- 5	A. Achilles	CMS Release 5

Compiler

13,1- 8	E. Sturm	PL/I Version 2
13,3- 9	St. Ost	FORTRAN Interactive Debug

Datenbanken

13,1- 3	A. Achilles	Abschluß des Studienprojektes
13,3- 4	A. Achilles	Relationale Datenbanksysteme
13,3- 7	A. Achilles	Text-Retrieval mit ISIS
13,4-12	A. Achilles	Volltext-Retrieval mit ISIS

Datex-P

13,1- 7	G. Richter	Datex-P20 im LAN der Universität
---------	------------	----------------------------------

DCF

13,3-19	B. Neukäter	DCF: Vorläufer wird Standard
---------	-------------	------------------------------

ENVIRON

- | | | |
|---------|-------------|-----------------------------------|
| 13,3- 3 | A. Achilles | Umgebungen im CMS |
| 13,4- 4 | A. Achilles | Erweiterung des ENVIRON-Kommandos |

Filetransfer

- | | | |
|---------|----------|----|
| 13,4- 5 | E. Sturm | UM |
|---------|----------|----|

FORTRAN

- | | | |
|---------|---------|---------------------------|
| 13,3- 9 | St. Ost | FORTRAN Interactive Debug |
|---------|---------|---------------------------|

GDDM

- | | | |
|---------|-----------|----------------------------------|
| 13,4-31 | R. Mersch | Erstellung von Grafiken mit GDDM |
|---------|-----------|----------------------------------|

Grafik

- | | | |
|---------|--------------------------------|---|
| 13,1- 3 | E. Sturm | Umfrage „Zukünftige Grafiknorm“ |
| 13,1-23 | R. Schopohl | Scantool |
| 13,1-24 | B. Schlagheck/
Th. Ocón Cao | Interpolierende und approximierende Spline-Kurven
in RUM/GKS |
| 13,1-27 | M. Awe | Integration von Grafik in TeX-Texte |
| 13,2-21 | H. Pudlatz | Plot-Dateien des Großrechners auf dem PC |
| 13,3-21 | H. Brinkmann | Noch mehr POSTER |
| 13,3-25 | B. Samland | Neue SHOW mit CHART |
| 13,4-29 | E. Sturm | RUM/GKS mit PL/I Version 2 |
| 13,4-31 | R. Mersch | Erstellung von Grafiken mit GDDM |

ISIS

- | | | |
|---------|-------------|-----------------------------|
| 13,3- 7 | A. Achilles | Text-Retrieval mit ISIS |
| 13,4-12 | A. Achilles | Volltext-Retrieval mit ISIS |

Jobverteilung

- | | | |
|---------|-----------|------------------------------------|
| 13,4-38 | A. Ahrens | Jobverteilung Juni bis August 1989 |
|---------|-----------|------------------------------------|

Lehrveranstaltungen

- | | | |
|---------|---|---|
| 13,1-15 | - | Lehrveranstaltungen im 1. Halbjahr 1989 |
| 13,2-13 | - | Lehrveranstaltungen im 1. Halbjahr 1989 |
| 13,3-11 | - | Lehrveranstaltungen im 2. Halbjahr 1989 |
| 13,4-27 | - | Lehrveranstaltungen im Wintersemester 1989/90 |

Magnetplatten

- | | | |
|---------|-------------|-----------------------------------|
| 13,2- 6 | K.B. Mertz | Installation von IBM-3380-Platten |
| 13,3- 8 | B. Neukäter | Mehr Plattenplatz im MVS |

Mikrorechner

- | | | |
|---------|-------------------------|---|
| 13,1- 5 | S. Zörkendörfer | PC-Lizenz zu SPSS-Produkten |
| 13,1-11 | H. Kamp/
H.W. Kisker | Generierung eigener Zeichensätze auf dem PC |
| 13,2-21 | H. Pudlatz | Plot-Dateien des Großrechners auf dem PC |

MS-DOS

- | | | |
|---------|----------|---|
| 13,2- 7 | W. Lange | OS/2 aus der Sicht des MS-DOS-Anwenders |
|---------|----------|---|

MVS

- | | | |
|---------|-------------|--------------------------|
| 13,3- 8 | B. Neukäter | Mehr Plattenplatz im MVS |
|---------|-------------|--------------------------|

Netze

13,1- 7	G. Richter	Datex-P20 im LAN der Universität
13,4- 3	W. Held	Datennetz für die Wissenschaft
13,4-18	B. Neukäter	Weltweiter Rechnerverbund im Dienste der Wissenschaft

Organisationsplan

13,2- 3	-	Organisationsplan des Universitätsrechenzentrums
---------	---	--

OS/2

13,2- 7	W. Lange	OS/2 aus der Sicht des MS-DOS-Anwenders
---------	----------	---

PEEK

13,3- 3	A. Achilles	Probleme mit dem PEEK-Kommando?
---------	-------------	---------------------------------

PL/I

13,1- 8	E. Sturm	PL/I Version 2
13,4-29	E. Sturm	RUM/GKS mit PL/I Version 2

Plot-Preview

13,2-21	H. Pudlatz	Plot-Dateien des Großrechners auf dem PC
---------	------------	--

POSTER

13,3-21	H. Brinkmann	Noch mehr POSTER
---------	--------------	------------------

Programmquerschnitt

13,2-23	A. Ahrens	Programmquerschnitt Januar bis März 1989
---------	-----------	--

Quiz

13,1-29	E. Sturm	Inform-Quiz
13,2-22	E. Sturm	Inform-Quiz
13,3-27	E. Sturm	Inform-Quiz
13,4-37	E. Sturm	Inform-Quiz

REXX

13,4-10	A. Achilles	RXSQL - die Schnittstelle zwischen REXX und SQL/DS
---------	-------------	--

RUM/GKS

13,1-24	B. Schlagheck/ Th. Ocón Cao	Interpolierende und approximierende Spline-Kurven in RUM/GKS
13,4-29	E. Sturm	RUM/GKS mit PL/I Version 2

RXSQL

13,4-10	A. Achilles	RXSQL - die Schnittstelle zwischen REXX und SQL/DS
---------	-------------	--

SAS/PC

13,3- 6	S. Zörkendörfer	SAS/PC-Probeinstallation
---------	-----------------	--------------------------

Scanner

13,1-23	R. Schopohl	Scantool
13,2-19	H. Pudlatz	Texte „scannen“ - Komfort bei der Eingabe

SHOW

13,3-25 B. Samland Neue SHOW mit CHART

Splines

 13,1-24 B. Schlagheck/
Th. Ocón Cao Interpolierende und approximierende Spline-Kurven
in RUM/GKS

SPSS

 13,1- 4 H.W. Kisker Erweiterung der Lizenzliste um SPSS/PC+
13,1- 5 S. Zörkendörfer PC-Lizenz zu SPSS-Produkten
13,3- 4 S. Zörkendörfer Neues vom SPSS

SQL

 13,4- 7 A. Achilles Der Tabellen-Editor VMSQL/EDIT
13,4-10 A. Achilles RXSQL - die Schnittstelle zwischen REXX und SQL/DS

Statistik

 13,3- 4 S. Zörkendörfer Neues vom SPSS
13,3- 6 S. Zörkendörfer SAS/PC-Probeinstallation

Studienprojekt

13,1- 3 A. Achilles Abschluß des Studienprojektes

Textverarbeitung

 13,1-27 M. Awe Integration von Grafik in TeX-Texte
13,2- 3 W. Kasper TeX-Preview unter CMS
13,2-15 M. Sandkötter/
H. Kamp Ausdruck von WordPerfekt-4.1-Dateien auf dem
AGFA-P400-Drucker
13,2-19 H. Pudlatz Texte „scannen“ - Komfort bei der Eingabe
13,3-19 B. Neukäter DCF: Vorläufer wird Standard
13,3-20 W. Kasper Laserbelichter

UM

13,4- 5 E. Sturm UM

UTS

13,2- 6 W. Stockhofe Rechenzeit und Plattenspeicher im UTS

Verbrauchsdaten

13,1-34 A. Ahrens Verbrauchsdaten 1988

WordPerfekt

 13,2-15 H. Kamp/
M. Sandkötter Ausdruck von WordPerfekt-4.1-Dateien auf dem
AGFA-P400-Drucker

Zeichensatz

 13,1-11 H. Kamp/
H.W. Kisker Generierung eigener Zeichensätze auf dem PC