

inforum

INfOrMationsforum des Rechenzentrums der Universität Münster

Jahrgang 17, Nr. 1 – Februar 1993

ISSN 0931-4008

Inhaltsverzeichnis

Editorial	2
RUM-Aktuell	
Unix im Rechenzentrum	3
Tips zur effizienteren Netznutzung	4
Unix-Batchverarbeitung	5
VAST-2 im MVS/ESA	7
Abteilungsübergreifende Einrichtung Unix	8
Neue Modems für den Zugang über das Telefonnetz	8
Übertragung binärer Daten zwischen MVS (IBM) und VMS (DEC)	9
Neue Software-Lizenzen für Workstations	12
Benutzung der Schnelldrucker des Rechenzentrums von AIX-Rechnern aus	13
Abschicken von MVS-Jobs mittels FTP	13
Neues vom SPSS	15
Zentrale Sicherung von Workstation-Daten	17
Das neue TeX-System unter AIX und SunOS	19
Anwendungssoftware in Unix	21
Neues vom SAS	22
RUM-Tutorial	
E-Mail und NetNews mit Unix und VM/CMS	23
Ein neuer Schriftauswahlmechanismus für L ^A T _E X – letzter Teil (3)	26
RUM-Grafik	
„Besser als das BKA erlaubt“	31
RUM-Lehre	
Lehrveranstaltungen im 1. Halbjahr 1993	32
Stichwörter inforum Jahrgang 16	38

Impressum**inforum**

ISSN 0931-4008

Redaktion: W. Bosse (Tel. 83-2461)
 St. Ost (Tel. 83-2681)
 H. Pudlatz (Tel. 83-2472)
 E. Sturm (Tel. 83-2609)

Satz: S. Arnold
 R. Wilmes

Satzsystem: $\text{\LaTeX}$

Druck: Drucktechnische
 Zentralstelle der WWU

Universitätsrechenzentrum
 Einsteinstraße 60
 4400 Münster

Auflage dieser Ausgabe: 1000

Redaktionsschluß der nächsten
 Ausgabe: 8.4.1993

langsam und haben schlechte Antwortzeiten. Aber das sind „natürliche“ Phänomene, die genauso alt wie die Datenverarbeitung selbst sind. Sie verschwinden nach einer Neu-Installation vorübergehend, tauchen aber bald wieder auf.

Ein kleines Nachbeben des Sommers, diesmal ein erfreuliches, gibt es aber doch. Im Zuge einer Schadensersatzregulierung für die Ausfälle der ES/9000 wurden zwei Unix-Workstations RS6000-580 beschafft. Sinnigerweise heißen sie „schadnix“ und „machtnix“. Zusammen mit dem seit einem Jahr benutzten „obelix“ sollen sie die Grundlage der Unix-basierenden Aktivitäten des Rechenzentrums sein. Alle drei sind sehr schnell miteinander verbunden (220 MBit/s), was insbesondere Möglichkeiten der Parallelisierung eröffnet. Wie wir uns die Aufgabenverteilung zwischen diesen drei Rechnern vorstellen, erfahren Sie in einer Reihe von Artikeln in diesem Heft.

Überhaupt ist diese **inforum**-Ausgabe dem Thema „Unix“ gewidmet. Dies unterstreicht die vor einem Jahr vorgenommene grundsätzliche Orientierung des Rechenzentrums im Workstation- und Server-Bereich. Damit sind nicht nur IBM-Unix-Systeme gemeint. So wurden eine Reihe von SUN-Workstations beschafft, die die Unterstützung gerade der auf diesen Systemen populären Anwendungs-Software durch das Rechenzentrum ermöglichen. Um andere Unix-Derivate können und wollen wir uns vorerst nicht kümmern. Dies steht im Einklang mit den grundsätzlichen Entscheidungen der Universitätskommissionen und dem Gutachten der DFG zur DV-Gesamtversorgung der Universität, die eine Konzentration auf eine, maximal zwei Unix-Varianten fordern.

Editorial

von

St. Ost

Nach dem großen Sommertheater scheint – ungerufen – an der Hardware-Front Ruhe eingekehrt zu sein. Die Rechner tun, was man von ihnen erwartet, oder doch jedenfalls meistens. Natürlich sind sie überlastet, zu

RUM-Aktuell

Unix im Rechenzentrum

von

St. Ost

Der Artikel beschreibt das Angebot des Rechenzentrums im Unix-Bereich, sowohl die zur Verfügung stehende Hardware als auch einen Teil der benutzbaren Anwendungs-Software.

Das Rechenzentrum hat sich im Workstation- und Serverbereich grundsätzlich für Unix entschieden. Nicht für jedes Unix, wohlbemerkt. Was keine Qualitätsaussage sein soll, sondern allein der Versuch ist, die Anzahl der zu unterstützenden Unix-Varianten klein zu halten. Die Wahl fiel auf Sun, da Suns in der Universität bereits seit Jahren genutzt werden und gut eingeführt sind, und auf IBM, da die RS/6000 zum Zeitpunkt der Beschaffung die schnellsten Workstations (bezogen auf die zur Verfügung stehenden Geldmittel) waren. Sun und IBM sind zusätzlich zwei prominente Vertreter der beiden um Macht- und Marktanteile ringenden „Zusammenschlüsse“ von Computerherstellern, nämlich Unix International (Sun) und Open System Foundation (IBM).

RS/6000

Augenblicklich sind in der Universität ca. 60 Maschinen dieses Typs im Einsatz, rund die Hälfte davon im Fachbereich Chemie und ein Viertel im Rechenzentrum. Die aktuelle Version des Betriebssystems ist AIX 3.2.3. Folgende RS/6000 des Rechenzentrums stehen der Universitätsöffentlichkeit zur Verfügung:

obelix

ist der zentrale File- und NIS-Server (*Network Information Services*) und etwa 28 MFlops schnell. Augenblicklich sind 11 GByte Plattenperipherie installiert. Gedacht ist obelix für interaktive Anwendungen. Mit Ausnahme der Filesysteme der Chemie sind alle User-Filesysteme auf obelix.

schadnix und machtnix

sind beide etwa 42 MFlops schnell und ausschließlich für numerisch intensive Anwendungen gedacht. Sie werden mittels DQS (siehe den Artikel „Unix-Batchverarbeitung“ in diesem *inforum*) genutzt, ein interaktives Arbeiten an diesen Rechnern ist nicht möglich.

comix

ist etwa 22 MFlops schnell. Er soll alle gängigen Kommunikationsprotokolle unterstützen und der erste „Ansprechpartner“ beim Kontakt von außen sein und ggf. zum gewünschten Zielrechner weitervermitteln.

urix2 bis urix6

sind etwa 15 MFlops schnell, befinden sich im Übergang zur Mathematik und stehen jedermann zur Verfügung, der die Mühe auf sich nimmt und persönlich zum Rechenzentrum kommt. In der Regel werden diese Rechner von Studenten genutzt.

obelix, schadnix und machtnix sind sehr schnell (220 MBit/s) über einen „serial optical link“ (SOL) miteinander verbunden. Spezielle Software erlaubt die Parallelisierung von Anwendungen. So könnte ein entsprechend geschriebenes Programm auf allen drei Rechnern parallel arbeiten und somit die Verweilzeit des Programms im Idealfall dritteln.

Da die Funktionsfähigkeit des RS/6000-Clusters von der Funktionsfähigkeit von obelix abhängt, planen wir mit Hilfe von schadnix ein ausfallsicheres System zu installieren, indem schadnix Zugriff auf die obelix-Platten bekommt. Für den Fall, daß obelix ausfällt, sorgen entsprechende System-Routinen in schadnix dafür, daß sich der Rechner (schadnix) gegenüber dem Netz als obelix verhält und die File- und NIS-Server-Dienste übernimmt.

Sun

Im letzten Jahr wurden 9 Rechner des Typs Sun Sparc Station 10 Model 20 (9.5 MFlops) beschafft, hauptsächlich um die Sun-Welt sowohl systemtechnisch als auch auf der Anwendungsseite unterstützen zu können. Als Betriebssystem wird SunOS 4.1.3 verwendet. Zwei Sun-Stations werden allgemein als Server unter DQS und zum interaktiven Arbeiten zur Verfügung gestellt, sobald betriebliche Fragen (Diebstahlsicherung etc.) geklärt sind. Die Suns des Rechenzentrums benutzen obelix als NIS- und File-Server, d. h. Benutzerkennungen und Paßwörter, HOME-Verzeichnisse und Mail-Queue sind die gleichen wie auf den RS/6000-Systemen.

Software

Es steht ein Reihe von Softwareprodukten sowohl für Suns als auch RS/6000 zur Verfügung. Für die letzteren bietet IBM spezielle Software-Lizenzen („Basis- und Ergänzungskörbe“) an. Vergleichen Sie hierzu den Artikel „Neue Software-Lizenzen für Workstations“ in dieser Ausgabe. Für Suns und RS/6000 gibt es darüber hinaus noch Standardanwendungspakete, die zentral auf obelix vorgehalten werden. Sehen Sie hierzu den Artikel „Anwendungssoftware in Unix“ in dieser Ausgabe.

Tips zur effizienteren Netznutzung

von

R. Perske

Dieser Artikel soll einige Tips und Hinweise geben, wie die Netzbelastung durch Inanspruchnahme zusätzlicher Dienste gesenkt werden kann, ohne daß wesentliche Einschränkungen in Kauf genommen werden müssen.

FTP und Archie

Immer wieder kommt es vor, daß Dateien von Servern in Übersee mit FTP geholt werden, obwohl diese bereits lokal vorhanden sind oder auf näherliegenden FTP-Servern verfügbar sind. Um dieses zu vermeiden, kann man den Archie-Dienst im Internet verwenden. Mit Hilfe dieses Dienstes, der von verschiedenen, weltweit verteilten Servern angeboten wird, kann man erfragen, auf welchen FTP-Servern eine Datei vorgehalten wird, und sich anschließend den nächstgelegenen FTP-Server aussuchen.

Da die Archie-Server reihum alle ihnen bekannten FTP-Server nach ihren Inhalten fragen und jeder Umlauf auf etwa einen Monat verteilt wird, befinden sich die Informationen immer auf einem Stand, der nicht älter als ein Monat ist.

Zum Suchen nach den Dateien muß man nur den Dateinamen oder einen Teil davon kennen. Diesen Teil kann man, je nach den gerade eingestellten Optionen, als Teilzeichenkette oder Suchmuster angeben. Der Server liefert dann alle Vorkommen dieser Datei auf den ca. 300 ihm zugänglichen FTP-Servern.

Eine Möglichkeit, Archie zu benutzen, ist, eine Telnet-Verbindung zu einem Archie-Server aufzubauen. Für uns am günstigsten gelegen ist der Server `archie.th-darmstadt.de`; als Benutzerkennung ist `archie` einzugeben. Auf dem dann erscheinenden Bildschirm wird darüber informiert, wie man sich die Hilfstexte anzeigen lassen kann.

Der wichtigste Archie-Befehl ist `prog wort`, mit dem man die Archie-Datenbank nach allen bekannten Vorkommen aller Programme durchsucht, die `wort` heißen oder deren Namen `wort` enthält. Mit dem Befehl `set search typ` kann man vorher einstellen, wie `wort` zu verstehen ist. Für `typ` muß eines der Wörter `exact` (`wort` ist der exakte Programmname), `sub` oder `subcase` (`wort` ist im Programmnamen enthalten, bei `subcase` ist Groß-/Kleinschreibung zu beachten) oder `regex` (`wort` ist ein Suchmuster) eingesetzt werden.

Eine andere Möglichkeit besteht darin, einen lokal installierten Archie-Client zu benutzen. Auf den AIX-Systemen steht hierzu der Befehl `archie` zur Verfügung. Eine kurze Erläuterung dieses Shellscripts erhalten Sie, wenn Sie `archie` ohne Argumente aufrufen.

Der Archie-Dienst wird auch vom Infosystem aus erreicht, wenn man unter der Rubrik *Auswärtige Infosysteme* den entsprechenden Punkt auswählt.

Mehr zum Archie-Dienst und zu vielen anderen Internet-Diensten finden Sie in der Publikation *Zen and the Art of the Internet – A Beginner's Guide to the Internet* von Brendan P. Kehoe, die in der DV-Beratung zur Einsicht und zum Kopieren ausliegt.

Gopher und Veronica

Gopher ist ein weltweites, verteiltes Informationssystem, welches Zugang auch zu vielen anderen Informationssystemen bietet. AIX-Benutzern steht der Befehl `gopher` zur Verfügung, der Zugriff auf dieses Informationssystem bietet. Außerdem kann er auch vom Infosystem aus durch Auswahl des entsprechenden Punktes unter der Rubrik *Auswärtige Infosysteme* aufgerufen werden.

Zur Funktionsweise von Gopher: Auf den Gopher-Servern sind Objekte verschiedener Typen gespeichert. Diese Typen können unter anderem sein: Verzeichnisse von Objekten, Dokumente, Telnet-Verbindungen, Schlüsselwortsuchen in Datenbanken u. a. In den Verzeichnissen wird wiederum auf die Objekte verwiesen, wobei zu jedem Objekt angegeben sind:

- der Typ des Objekts,
- der anzuzeigende Titel,
- eine Kennung (vergleichbar einem Dateinamen),

- die Adresse des Rechners, auf dem sich das Objekt befindet, und
- die Unteradresse des Gopher-Servers.

Der Client erfragt als erstes von einem Gopher-Server ein Verzeichnis, in der Regel das durch Angabe der leeren Kennung zu erreichende Anfangsverzeichnis. Der Gopher-Server sendet als Antwort dieses Verzeichnis, wonach die Verbindung beendet wird.

Dieses Verzeichnis wird vom Gopher-Client angezeigt. Der Benutzer wählt daraus ein Objekt aus. Falls das ausgewählte Objekt ein Verzeichnis ist, wird es vom entsprechenden Gopher-Server wie oben beschrieben erfragt. Dieses kann sich beliebig wiederholen.

Falls ein Dokument ausgewählt wurde, holt es der Gopher-Client nach der gleichen Methode wie bei Verzeichnissen und zeigt es dann an. Bei einer Telnet-Verbindung ruft der Gopher-Client das Telnet-Programm mit der angegebenen Server-Adresse auf. Bei einer Schlüsselwortsuche erfragt der Gopher-Client das zu suchende Wort, sendet es an den entsprechenden Gopher-Server, holt die Antwort, welche wieder die Form eines Verzeichnisses hat, und zeigt sie an.

Es existiert also immer nur eine kurzzeitige Verbindung zu einem Gopher-Server. Jede neue Abfrage kann an einen anderen Gopher-Server gehen, der Benutzer bekommt davon nichts mit.

Leider erzeugt die Gopher-Benutzung für unseren WIN-Anschluß ein gewisses Problem: Die kurzzeitigen Verbindungen werden zwar TCP/IP-mäßig sofort abgebaut, aber systembedingt nur mit deutlicher Verzögerung im darunterliegenden X.25-Protokoll. Ein schneller Wechsel zwischen verschiedenen Gopher-Servern kann also relativ schnell unseren WIN-Anschluß verstopfen. Deshalb bitten wir um Beachtung der jungen Dame Veronica:

Der gerade erst entstandene Dienst Veronica stellt für Gopher das gleiche dar wie der Archie-Dienst für FTP. Die Veronica-Server, selbst Gopher-Server, erfragen von den Gopher-Servern die Titel und speichern sie. Mittels der Gopher-Schlüsselwortsuche kann man die so entstandene Datenbank durchsuchen und in dem daraufhin angezeigten Verzeichnis direkt das gesuchte Objekt auswählen.

Veronica steht unter AIX unter dem Befehl `veronica` (welcher nur `gopher` mit entsprechenden Parametern aufruft) zur Verfügung und kann ebenfalls aus der genannten Rubrik des Infosystems heraus aufgerufen werden.

NFS

Eine stärkere Belastung des lokalen Netzes ergibt sich, wenn von Arbeitsplatzrechnern aus Befehle abgesetzt werden, die riesige Dateien (Megabytes und mehr) in den Benutzerdateisystemen kopieren, da diese Dateien dann zweimal zwischen dem jeweiligen NFS-(Network-File-System-)Server (z. B. obelix oder cuba) und der Workstation kopiert werden. In diesem Fall sollte man überlegen, ob es nicht günstiger ist, eben ein `rlogin` zum jeweiligen Server zu machen und den Kopierbefehl dort abzusetzen. Eine andere Frage ist, ob wirklich zwei getrennte Kopien der Datei benötigt werden oder ob nicht eine Kopie, die ggf. über zwei Verzeichniseinträge (sogenannte *links*) zu erreichen ist, ausreicht. Dann kann man nämlich statt des `cp`-Befehls den `mv`- oder den `ln`-Befehl verwenden.

Bei Fragen wenden Sie sich bitte an den Autor dieses Artikels unter perske@uni-muenster.de.

Unix-Batchverarbeitung

von

St. Ost

Das Distributed Queuing System (DQS) implementiert eine gemeinsame Batch-Queue für Unix-Workstations. Es ist Grundlage der unix-basierenden Batch-Verarbeitung auf den AIX-Rechnern des Rechenzentrums, steht aber auch für andere Plattformen wie Suns zur Verfügung.

Entwickelt wurde DQS an der Florida State University im Institut für Supercomputer. Es ist ein gemeinsames Batchsystem für einen ganzen Zoo von Unix-Workstations auch unterschiedlicher Hardware-Architektur. Der Begriff „Zoo“ meint hier eine Gruppe von Unix-Workstations, die miteinander über ein relativ schnelles LAN verbunden sind und idealerweise ein gemeinsames Filesystem (realisiert etwa über NFS) besitzen. Diese Voraussetzungen sind in Münster gegeben.

Aus Endbenutzersicht gesehen realisiert DQS eine Reihe von Batch-Queues mit unterschiedlichen Eigenschaften wie etwa der maximal möglichen Rechenzeit, der maximal möglichen Memory-Größe und der Hardware-Architektur, auf der das Programm ablaufen kann. Die Benutzung von Batch-Queues kann bei Bedarf auf einzelne Nutzer eingeschränkt werden.

Queues werden auf Workstations realisiert, jede Workstation kann mehr als eine Queue bearbeiten. Workstation-Queues mit gleichen Charakteristiken sind zu Gruppen zusammengefaßt, die nicht einzelnen Workstations zugeordnet sind. Rechenaufträge (Jobs) werden als Shell-Scripts formuliert und dem DQS zur Bearbeitung in einer bestimmten Workstation-Queue oder einer bestimmten Gruppe übergeben. Normalerweise wird man die Jobs an Gruppen zur Verarbeitung übergeben, da damit sichergestellt ist, daß die nächst freie Workstation-Queue den Job abarbeitet. Eine bestimmte Workstation-Queue wird man nur dann auswählen, wenn es spezielle Ressourcen (Software oder Hardware) nur an dieser einen Workstation gibt. Dies ist aber augenblicklich bei uns nicht der Fall. Wir empfehlen dringend, Jobs an Gruppen und nicht an einzelne Workstation-Queues zu übergeben.

Das DQS reiht den Auftrag in die gewünschte Gruppe ein und bearbeitet die so entstandene Warteschlange der Reihe nach (FIFO). Der Systemadministrator hat jeder Gruppe eine Reihe von Workstation-Queues zugeordnet, die die Rechenaufträge, die sich in der Warteschlange befinden, abarbeiten. DQS informiert den Benutzer auf Wunsch über jede Status-Änderung seines Jobs via Mail.

Die Benutzerschnittstelle zum DQS wird durch eine Reihe von Kommandos realisiert; für den Endbenutzer interessant sind die nachstehend aufgeführten Kommandos, deren genaue Syntax den entsprechenden *man pages* entnommen werden kann:

qdel
löscht DQS-Jobs eines Benutzers,

qstat
zeigt den Zustand von Queues und Gruppen an,

qsub
übergibt Rechenaufträge ans DQS.

Bislang wurden vier DQS-Gruppen eingerichtet, von denen eine nicht allgemein zur Verfügung steht:

testaix
Hier werden Shell-Scripts getestet. Maximale Rechendauer ist hier 1 Minute.

prodaix
Rechenaufträge mittlerer Länge bis zu 1 Stunde.

longaix
Ausgesprochene Langläufer mit maximal 3 Tagen Rechenzeit.

q0qc
Eine Gruppe nur für die Mitglieder des Projektes „Quantenchemie“.

Realisiert werden die Gruppen auf den Workstations *obelix*, *schadnix*, *machtnix*, *xfbhsrv1*, *xfbhsrv2* und *tobago*. *obelix* und *tobago* sind etwa 27.9 Mflops (Linpack), die anderen jeweils 42.1 Mflops schnell. Dies gibt einen Eindruck von der Leistungsfähigkeit dieses Workstation-Zoos. Sobald wir entsprechende Server in der Sun-Welt zur Verfügung haben, werden wir ähnliche Queues auch für Sun-Nutzer einrichten.

Wie sieht nun ein DQS-Job aus? Keine Angst, Sie müssen nicht JCL erlernen, es reichen einige wenige Anweisungen. Wie man Shell-Scripts programmiert, ist dagegen nicht so schnell zu erlernen. Wenn man sich zu Anfang ein Shell-Script als einfache Folge von Unix-Kommandos vorstellt, macht man nichts falsch. Ich werde im nächsten Wintersemester eine Vorlesung zur Programmierung von Shell-Scripts anbieten.

Vom Standpunkt der Unix-Shell sind DQS-Anweisungen Kommentare, beginnen also mit einem #. Ist das zweite Zeichen ein \$, so handelt es sich um eine DQS-Anweisung. Alle DQS-Anweisungen eines Shell-Scripts werden zu Anfang, unabhängig von ihrer Stellung in der Datei, abgearbeitet.

Als Beispiel folgt nun ein Job, der ein C-Programm übersetzt und einmal ausführt. Gespeichert sei der Job in der Datei *job1*.

```
#
## -cwd
## -mu ost@obelix.uni-muenster.de
## -mb
## -me
## -eo job1.ausgabe
#
echo hello_world.c wird kompiliert.
cc hello_world.c -o hello_world
#
echo Gruppe: $GROUP
echo Queue: $QUEUE
echo JobId: $JOB_ID
echo Typ: $ARC
echo hello_world wird ausgeführt.
#
hello_world
```

In Zeile 2 bis 6 stehen DQS-Anweisungen. Die erste setzt das Arbeitsverzeichnis auf das Verzeichnis, in dem *job1* steht. Danach wird die Mail-Adresse desjenigen angegeben, der über Status-Änderungen des Jobs informiert wird, und zwar bei Job-Beginn (*-mb*) und bei Job-Ende (*-me*). Zuletzt (*-eo*) wird die Datei genannt, in die *stdout* und *stderr* umgeleitet werden. Fehlt diese letzte Angabe, so wird *stdout* in eine Datei *nnnn.job1.OUT* und *stderr* in die Datei *nnnn.job1.ERR* umgeleitet, wobei *nnnn* die Nummer

des Rechenauftrags (Shell-Variable `JOB_ID`) ist. Der Sinn der anderen Shell-Variablen ist leicht ersichtlich. Das Kommando

```
qsub -G testaix 1 job1
```

übergibt den Rechenauftrag `job1` (s. o.) an die DQS-Gruppe `testaix`. Dabei wird ein Rechner (die „1“ zwischen `testaix` und `job1`) benötigt. DQS unterstützt nämlich auch die Parallelisierung von Rechenaufträgen und an dieser Stelle würde in einem solchen Falle die Anzahl der benötigten Workstation-Queues anzugeben sein.

VAST-2 im MVS/ESA

von

R. Perske

Eine große Hilfe bei der Vektorisierung und Optimierung von FORTRAN-Programmen stellt der Precompiler *VAST-2* dar.

Der FORTRAN-Precompiler *VAST-2* steht jetzt auch im MVS/ESA zur Verfügung. (Unter AIX wird er mit der Compiler-Option `-Pv` aufgerufen.) *VAST-2* verarbeitet den Quelltext derart, daß DO- und IF-Schleifen bezüglich Vektorisierung und Parallelisierung optimiert werden sowie andere Optimierungen vorgenommen werden können. Weiterhin versteht er bereits einige Sprachelemente des neuen Standards Fortran 90 und übersetzt sie in FORTRAN 77. Sein Bericht über erfolgte und nicht erfolgte Optimierungen mit Angabe der Gründe hilft bei der weiteren manuellen Optimierung. Der *User's Guide* für den Precompiler liegt in der DV-Beratung im Rechenzentrum aus.

Aufgerufen wird *VAST-2* über Sprachprozeduren, die die entsprechenden FORTRAN-Sprachprozeduren ersetzen und sowohl den Precompile- als auch den Compile-Schritt beinhalten:

VASTC statt FORTC

VASTO statt FORTO

VASTVC statt FORTVC

VASTVO statt FORTVO

Für die nächsten Verarbeitungsschritte sind die üblichen FORTRAN-Prozeduren zu verwenden, beispielsweise in der Form:

```
// EXEC VASTVC
```

```
//C.SYSIN DD *
```

```
... FORTRAN-Text ...
```

```
// EXEC FORTVLE
```

```
//E.SYSIN DD *
```

```
... Eingabedaten ...
```

Die Voreinstellungen der symbolischen Parameter `OPT` und `PARMC` der Prozeduren *VASTxxx* bzw. *VASTVxxx* haben sich gegenüber denen der Prozeduren *FORTxxx* bzw. *FORTVxxx* geändert; weitere symbolische Parameter sind hinzugekommen:

symbolische Parameter	Bedeutung (Voreinstellung)
OPT	Optimierungsstufe (3 statt 0)
PARMC	Compilerparameter ('VECTOR FLAG(I)' statt 'FLAG(I)')
PARMV	VAST-2-Parameter ('OPTON=V')
REGIONV	VAST-2-Hauptspeicher (32M)
VERSIONV	VAST-2-Version (121)
VPRINT	VAST-2-Ausgabeklasse ('*')

Als Beispiel sei der folgende Job gegeben. Er bereitet das (zugegebenerweise nicht sehr sinnvolle) FORTRAN-77-Programm mit *VAST-2* auf, übersetzt es, spielt es in eine Objektdatenbibliothek und führt es anschließend testweise einmal aus.

```
//PERSKET JOB (TEST,A07),PERSKE
// EXEC VASTVO,PARMV='OPTON=78V',
//   OBJECT='PERSKE.A.TXTLIB',
//   MODULE=TESTVAST
//C.SYSIN DD *
  DIMENSION N(99,99,99)
  DO 1 I=1,99
    DO 1 J=1,99
      DO 1 K=1,99
1        N(I,J,K)=1
      DO 2 I=1,99
        DO 2 J=2,98
          DO 2 K=1,99
2            N(I,J,K) = N(I,J-1,K)
          & + N(I,J+1,K)
          WRITE(*,*) N(50,50,50)
        END
      // EXEC FORTVE,COND=(4,LT),
      //   OBJECT='PERSKE.A.TXTLIB',
      //   MODULE=TESTVAST
```

Der *VAST*-Präprozessor schreibt dann in die Ausgabedatei, welche Schleifen er durch Vektorisierung (Vectorized, V) oder Ersetzen von verschachtelten Schleifen durch eine einzige (Collapsed, C) optimiert und welche Schleifen aus welchen Gründen nicht optimiert werden. Bei den oben gesetzten Optionen

LINE TYP MSG (T=TRANSLATION, D=DEPENDENCY, W=WARNING, S=SYNTAX)

9 D (EPD037) FEEDBACK OF ARRAY ELEMENTS (N)
CONFLICT ON LINE 9. THE DO INDEX IS J, THE DO LABEL IS 2.

Abbildung 1: VAST-2-Meldungen

(OPTON=78V) würde er sogar versuchen, Code aus Unter-routinen in aufrufende Routinen zu integrieren, um die Zeit für die Unterprogrammaufrufe zu sparen. Darauf wird bei der Voreinstellung (OPTON=V) verzichtet. Hier ist ein Ausschnitt aus der Ausgabedatei:

```

1.      DIMENSION N(99,99,99)
2. V----- DO 1 I=1,99
3. V C----- DO 1 J=1,99
4. V C C-- DO 1 K=1,99
5. V--C--C--1 N(I,J,K)=1
6. V----- DO 2 I=1,99
7. V D----- DO 2 J=2,98
8. V D N-- DO 2 K=1,99
9. V--D--N--2 N(I,J,K) = N(I,J-1,K)
10.      &      + N(I,J+1,K)
11.      WRITE(*,*) N(50,50,50)
12.      END

```

Man sieht an den Buchstaben V und C, daß die oberen drei Schleifen zu einer zusammengefaßt und diese vektorisiert wurde. Von den unteren drei Schleifen kann nur die äußere vektorisiert werden. In der mittleren, markiert durch D (Data dependent), wird bezüglich des Laufindex J auf Feldelemente zugegriffen, die in vorherigen Durchläufen der gleichen Schleife verändert worden sind. Deshalb kann auch die innere Schleife nicht bearbeitet werden (N, Not chosen). Zur Erläuterung dieses Sachverhalts taucht in der Ausgabedatei die in Abb. 1 gezeigte Meldung auf.

Abteilungsübergreifende Einrichtung Unix

von

St. Ost

Dem besonderen Stellenwert, den das Betriebssystem Unix in der Universität genießt, wurde durch Einrichtung einer einzigen Anlaufstelle für Fragen und Probleme im Unix-Bereich Rechnung getragen.

Die Unterstützung von Unix-Systemen und -Anwendungen ist nach der Organisationsstruktur des

Rechenzentrums auf drei Abteilungen verteilt. Die systemtechnische Beratung und Betreuung erfolgt durch die Abteilung 2, bei Fragen im Kommunikationsbereich hilft die Abteilung 3, während die Anwendungs-Software durch die Abteilung 4 betreut wird. Das hat für den Nutzer den schwerwiegenden Nachteil, daß er i. a. nicht weiß, an wen er sich bei Fragen und Problemen im Unix-Bereich wenden soll. Ideal wäre eine einzige Anlaufstelle. Aus diesem Grunde wurde eine neue abteilungsübergreifende Einrichtung Unix im Rechenzentrum gegründet, denen die Herren Ost, Hölter und Mersch aus der Abteilung 2, Herr Speer aus der Abteilung 3 sowie Dr. Süselbeck und Herr Grote aus der Abteilung 4 angehören. Koordinator der Einrichtung bin ich (Tel. 26 81, E-Mail ost@uni-muenster.de).

Neue Modems für den Zugang über das Telefonnetz

von

D. Schulze

Vor einigen Wochen wurden im Rechenzentrum leistungsfähigere Wählmodems als Multimodems an das Telefonnetz der Telekom angeschlossen. Damit wurde der Zugang vom häuslichen PC an das Rechenzentrum verbessert.

Die neuen Modems unterstützen neben den bisherigen Übertragungsverfahren nach CCITT V.22 und V.22bis (1200 und 2400 bit/s) auch alle Varianten nach CCITT V.32 und V.32bis. Das bedeutet, daß Geschwindigkeiten bis 14400 bit/s bei normaler Übertragung erreicht werden. Aus Kostengründen wurden drei solcher Modems installiert, ein viertes, das später ersetzt werden soll, unterstützt wie bisher nur die Übertragungsverfahren nach V.22/V.22bis.

Die Amtsnummer 897611 der vier Anschlüsse mit Sammelruf bleibt unverändert. In diesem Jahr werden zusätzlich Anschlüsse im Telefonnetz der Universität für Modems bereitgestellt werden.

Ihre alten Modems können Sie allerdings weiterverwenden. Durch eine automatische Prozedur paßt sich das angewählte Modem an das schnellste und beste Übertragungsverfahren an, das Ihr Modem unterstützen kann. Erst dann erhalten Sie die Meldung über einen erfolgreichen Verbindungsaufbau (CONNECT ...).

Die Modems unterstützen die Networking-Protokolle der Fa. Microcom (MNP) bis Klasse 5 und die entsprechenden Funktionen nach CCITT V.42 und V.42bis. Damit wird zwischen den beiden Modems ein synchrones Übertragungsverfahren mit Fehlererkennung und Fehlerkorrektur angewandt. Zusätzlich können die Daten zum Übertragen komprimiert bzw. dekomprimiert werden. Ist die Verbindung störungsfrei, ergibt sich dadurch eine Verdopplung der Übertragungsrate, z. B. bei einer Geschwindigkeit von 2400 bit/s werden die Daten mit ca. 4800 bit/s übertragen.

Sie müssen folgendes bei der Einstellung Ihres Modems beachten:

- Die Übertragungsgeschwindigkeit Ihrer Computerschnittstelle sollte mindestens doppelt so hoch sein wie die Geschwindigkeit auf der Telefonleitung.
- Sie sollten die XON/XOFF-Flußkontrolle einstellen. Das Modem kann mit XOFF den Empfang weiterer Daten vom angeschlossenen Rechner vorübergehend unterbrechen und mit XON die Übertragung wieder freigeben.

Im übrigen sind die Einstellvorschriften und die entsprechenden Kommandos bei den Modems recht unterschiedlich – trotz des Quasistandards nach der Fa. Hayes. Es lohnt sich in jedem Fall, das Handbuch gut durchzulesen.

Noch ein wichtiger Hinweis:

Lassen Sie die Finger von Endgeräten (Modems, Anrufbeantworter, FAX-Geräte usw.), die von der Telekom nicht zugelassen worden sind. Die Zulassung, erkennbar an einer Zulassungsnummer, die am Gerät angebracht sein muß, garantiert dem Anwender, daß das Endgerät den Betrieb des Telefonnetzes nicht stört oder sonst nachhaltig beeinflusst. Sie garantiert nicht immer die einwandfreie Funktion des Gerätes. Durch den Anschluß nicht zugelassener Endgeräte gefährden Sie unter Umständen auch das Wartungspersonal der für die Telefonübertragung verantwortlichen Institutionen. Sie machen sich also strafbar, wenn Sie nicht zugelassene Modems anschließen!

Übertragung binärer Daten zwischen MVS (IBM) und VMS (DEC)

von

D. Schaube

Für die Übertragung binärer Daten zwischen DEC- und IBM-Rechnern stehen unter MVS Umwandlungshilfen zur Verfügung.

Im MVS sind jetzt Routinen vorhanden, die es erlauben, unter MVS auf der IBM erzeugte binäre Daten in wiederum binäre Daten umzuwandeln, die auf der VAX unter VMS weiterverarbeitet werden können. Ebenso einfach ist es in umgekehrter Richtung, unter VMS erzeugte binäre Daten zu konvertieren, so daß sie unter MVS wiedereingelesen werden können. Die hierfür benötigten Routinen stehen im MVS bereit:

Datentyp	IBM-Format in VAX-Format	Vax-Format in IBM-Format
INTEGER*2	SIBMVAX	SVAXIBM
INTEGER*4	IIBMVAX	IVAXIBM
REAL*4	RIBMVAX	RVAXIBM
REAL*8	DIBMVAX	DVAXIBM
CHARACTER	CIBMVAX	CVAXIBM

Eine genaue Beschreibung der einzelnen Routinen und ihrer Übergabeparameter erhält man im VM/CMS mit dem Kommando `HELP VAX MENU`.

Das eigentliche Problem besteht darin, die Daten mittels FTP zu übertragen, ohne dabei die Satzstruktur zu zerstören, da sonst das Einlesen fehlschlägt und man böse Überraschungen erleben kann.

Erzeugung binärer Daten, Konvertierung derselben und was dabei zu beachten ist

Sollen Daten binär abgelegt werden, so wird meistens unformatiert in eine mit `FORM='UNFORMATTED'` geöffnete Datei geschrieben. Dies allein reicht nicht, falls die Daten vom VMS ins MVS bzw. umgekehrt übertragen werden sollen. In diesem Falle muß folgendes beachtet werden:

Die (neu anzulegende) Datei, in die geschrieben werden soll, sollte feste Satzlänge haben. Dies kann im VMS direkt im `OPEN`-Befehl festgelegt werden, während dies im MVS über `DD`-Anweisungen gesteuert wird. Unformatiertes Wegschreiben der Daten erfolgt mit einem `WRITE`-Befehl ohne Formatangabe.

Eine andere Vorgehensweise besteht darin, die Daten im A-Format, also REAL*4 mit A4, INTEGER*2 mit A2 wegzuschreiben, was intern eine binäre Abspeicherung bewirkt. Variables Datenformat und einfaches unformatiertes Wegschreiben funktionieren nicht, da dann zusätzliche Steuerzeichen in die Datei eingefügt werden, die im MVS zu Problemen bei der Konvertierung führen. Das Folgende zeigt ein VMS-FORTRAN-Programm zur Erzeugung von Binär-Dateien:

```

program conver
integer      i,j
integer*2    sdat(80)
real*4       rdat(80)
character*18 word

open(2,file='vax1.dat',status='new',
&    recordtype='fixed',
&    recl=20,form='unformatted')
open(4,file='vax3.dat',status='new',
&    recordtype='fixed',
&    recl=80,form='formatted')
do 10 i=1,80
    sdat(i)=i
    rdat(i)=sdat(i)/2.
10 continue
    word = 'Beam me up, Scotty'
do 20 i=0,1
    write (2) (sdat(j+i*40),j=1,40)
20 continue
do 30 i=0,3
    write (2) (rdat(j+i*20),j=1,20)
30 continue
write(2) word
write(4,120) (sdat(i),i=1,80)
write(4,130) (rdat(i),i=1,80)
write(4,140) word
120 format (40(a2))
130 format (20(a4))
140 format (a18)
close(2)
close(4)
end

```

In die Datei VAX1.DAT wird unformatiert ausgegeben, während in die Datei VAX3.DAT im A-Format, also unkonvertiert, geschrieben wird. Zu beachten ist der Unterschied in der OPEN-Anweisung der beiden Dateien. Während im Fall des Öffnens der Datei mit FORM='UNFORMATTED' der Parameter RECL die Anzahl der Longwords (1 Longword $\equiv$ 4 Byte) enthält, wird beim Öffnen mit FORM='FORMATTED' in RECL die Zahl der Bytes pro Satz (hier 80 Bytes) festgelegt. Des weiteren sollte man den Qualifizierer /G.FLOATING beim

Aufruf des Compilers vermeiden, da sonst Werte vom Typ REAL*8, COMPLEX*16, DOUBLE PRECISION und DOUBLE COMPLEX nicht mehr richtig konvertiert werden. Normalerweise wird bei der Kompilierung der D_FLOATING-Typ gewählt, mit dem die anschließende Konvertierung keine Probleme bereitet. Wurden die in diesem Abschnitt erläuterten Bedingungen beachtet, so gelingt die Konvertierung sowohl für die Datei VAX1.DAT als auch für die Datei VAX3.DAT.

Die Erzeugung binärer Daten unter MVS verläuft analog dem obigen Beispiel, allerdings mit dem Unterschied, daß die OPEN-Anweisung dann nur die Unit-Nummer und die FORM-Angabe enthält. Die Satzstruktur wird hier durch DD-Anweisungen festgelegt.

Die Konvertierung der Daten erfolgt immer im MVS, da nur hier die Konvertierungsroutinen zur Verfügung stehen. Bevor diese Routinen aufgerufen werden, sollten die Daten mit den gleichen Befehlen und Formatangaben eingelesen werden, mit denen sie weggeschrieben wurden. Nachfolgend ein Beispielprogramm, das die im ersten Beispiel unter VMS erzeugten Daten im MVS in das entsprechende IBM-Format umwandelt:

```

program rdvax
integer      i,j
integer*2    svax,sibm
real*4       rvax,ribm
character*80 wrdvax,out1,out2
parameter    (num=80)
dimension    rvax(num),ribm(num),
&            svax(num),sibm(num)
open (2,form='unformatted')
do 20 i=0,1
    read(2) (svax(j+i*40),j=1,40)
20 continue
do 30 i=0,3
    read(2) (rvax(j+i*20),j=1,20)
30 continue
read(2) wrdvax
close(2)
call svaxibm(svax,1,sibm,1,num)
call rvaxibm(rvax,1,ribm,1,num)
call cvaxibm(wrdvax,out1,80)
call cvaxibm(wrdvax(1:10),out2)
end

```

Nach dem unformatierten Einlesen der Daten werden die Konvertierungsroutinen aufgerufen, um sie vom VAX-Format ins IBM-Format zu transformieren.

Dabei wandelt die Routine SVAXIBM INTEGER*2-Daten um, und die Routine RVAXIBM konvertiert REAL*4-Daten. In diesem Beispiel enthalten die Arrays SVAX und RVAX die unter VMS erzeugten Ein-

gabedaten und die Arrays SIBM und RIBM die konvertierten Ausgabedaten. Der Parameter NUM enthält die Anzahl der zu konvertierenden Daten und der zweite und vierte Übergabeparameter jeweils die Schrittweite der Indizes der beiden Arrays. Die Routine CVAXIBM wandelt CHARACTER vom VAX-Format in das entsprechende IBM-Format um. Hier enthält die Variable WRDVAX den Input-String, während in OUT1 bzw. OUT2 jeweils der konvertierte String abgespeichert wird. NUM enthält wieder die Anzahl der zu konvertierenden Zeichen. Eine andere Möglichkeit zeigt der zweite Aufruf von CVAXIBM. Durch den Aufruf mit WRDVAX(1:10) werden nur die ersten zehn Zeichen des Input-Strings konvertiert und abgespeichert.

Die Konvertierung ist also sehr einfach, die Übertragung zwischen den beiden Systemen geschieht mittels FTP, wobei allerdings einiges zu beachten ist:

Übertragung der Binärdaten zwischen den beiden Systemen

Betrachten wir zuerst den (einfacheren) Fall der Übertragung der Datei vom VMS ins MVS.

Übertragen wird die Datei mittels FTP im BINARY-Mode, wobei vor der Übertragung mit `quote site` die entsprechende Dateistruktur im MVS festgelegt wird, hier also z. B. mit

```
quote site lrecl=80 blksize=6160 recfm=fb
```

Anschließend wird mittels `binary` in den Binär-Modus umgeschaltet und die Datei übertragen. Auf keinen Fall darf der Qualifier `/FDL` benutzt werden, da sonst eine gänzlich andere Satzstruktur erzeugt wird und das Einlesen im MVS schiefgeht.

Nun zu dem etwas umständlicheren Fall der Übertragung der Daten vom MVS ins VMS.

Nachdem die binären Daten ins VMS-Format konvertiert und in eine Datei mit fester Satzlänge geschrieben worden sind, ergibt sich anschließend bei der Übertragung der Datei mittels FTP ins VMS ein Problem. Um die Datei binär zu übertragen, muß im FTP in den Binär-Modus geschaltet werden, was leider außerdem bewirkt, daß die Datei immer mit einer festen Satzlänge von 512 Bytes übertragen wird. Dies ist meistens aber nicht die richtige Satzstruktur. In diesem Falle muß man auf der VMS-Seite eine sogenannte FDL-Datei erzeugen, welche die richtigen Attribute der Datei wie Satzlänge usw. enthält.

Erzeugen kann man diese FDL-Datei auf zwei Arten. Einerseits kann man den FDL-Editor mit dem Kommando `EDIT /FDL` aufrufen, mit dem dann menügesteuert eine FDL-Datei erzeugen werden kann. Eine

andere Möglichkeit besteht darin, eine schon vorhandene Datei einfach zu analysieren und die erhaltene FDL-Datei dann mit irgendeinem Editor so zu modifizieren, daß es die gewünschten Informationen enthält. Der VMS-Befehl lautet:

```
$ ANALYZE fn.ft /RMS_FILE /FDL /OUTPUT=fn.ft
```

Nun können aus dieser FDL-Datei fast alle Zeilen gelöscht werden. Um der einzulesenden Datei eine passende Struktur zu geben, werden nur sehr wenige der Zeilen benötigt, die entsprechend geändert werden. Eine so bearbeitete FDL-Datei könnte wie folgt aussehen:

```
FILE
  ORGANIZATION      sequential
  OWNER              [group,userid]
  PROTECTION         (system:RWED, owner:RWED,
                     group:RE, world:)
RECORD
  CARRIAGE_CONTROL  none
  FORMAT            fixed
  SIZE              80
```

Dabei enthält `[group,userid]` die eigene Benutzerkennung und zugehörige Projektgruppe. Diese Datei wird nun mittels FTP im ASCII-Modus ins MVS übertragen und dort unter dem Namen `fn.ftFDL` abgelegt, wobei `fn.ft` der Dateiname der einzulesenden Datei ist. Dabei ist vorher bei der Vergabe des Dateinamens zu beachten, daß `ft` nicht mehr als 5 Zeichen umfaßt, denn die einzelnen Teile der Dateinamen im MVS dürfen aus jeweils nicht mehr als 8 Zeichen bestehen. Lautet der Name der zu übertragenden Datei `TEST.DAT`, so muß die zugehörige FDL-Datei den Namen `TEST.DATFDL` erhalten.

Anschließend wird die Datei mit dem Qualifizierer `/FDL` übertragen. Damit wird nicht nur die Datei binär übertragen, sondern auch im VMS die notwendige Satzstruktur verwendet. Bei Angabe des Qualifizierers `/FDL` erfolgt automatisch eine Umschaltung in den Binär-Modus. Diejenigen, die nicht mit der VMS-Oberfläche des FTP arbeiten, welche vor eingestellt ist, sondern gewöhnlich mit dem Ultrix-Interface, müssen mit dem Kommando `VMS` auf das VMS-Interface umschalten. Ansonsten erfolgt eine Fehlermeldung, da der `/FDL`-Qualifizierer dort nicht bekannt ist.

Nach der Übertragung können die Daten dann wieder so eingelesen werden, wie sie im MVS abgespeichert wurden. Aber auch hier ist darauf zu achten, daß das Programm ohne den `/G_FLOATING`-Qualifizierer kompiliert wird.

Neue Software-Lizenzen für Workstations

von

W. Bosse

Für Benutzer von Workstations der Firmen IBM (RS/6000) und DEC sind Campus-Lizenzen zur Nutzung der lizenzierten Software dieser Anbieter verfügbar. Die Ende 1992 abgeschlossenen (und finanzierten) Verträge bieten auch für weitere Systeme einen Einstieg.

Software-Körbe für IBM-AIX-Lizenzprogramme

Für ihre Workstation-Familie RS/6000 bietet die Firma IBM AIX-Software zu Campus-Lizenz-Bedingungen in zwei sogenannten Körben an:

- einen Basiskorb mit allgemein benötigter Grundsoftware (Betriebssystem, Compiler, Grafik-Pakete, Unterprogrammibibliothek, PC-Simulator u. a.),
- einen Ergänzungskorb mit spezieller Software (Netz-Software, Grafik-Tools u. a.).

Die Konditionen sind unabhängig von der Leistungsklasse des Rechners und von der Anzahl der gleichzeitigen Benutzer auf der Maschine, allerdings erfolgt eine funktionale Aufteilung nach Software-Server und Workstations (Client).

Das Universitätsrechenzentrum hat als Betreiber eines AIX-Software-Servers entsprechende Server-Lizenzen für den Basis- und den Ergänzungskorb erworben. Unter Bezug darauf können entsprechende Client-Lizenzen für AIX-Workstations zu den Campus-Bedingungen bestellt werden. Dabei wird für die meisten Anwendungen der Basiskorb ausreichen.

Die durch unsere Beschaffung im vergangenen Jahr auf dem Server verfügbaren AIX-Lizenzprogramme sind – entsprechend der IBM-Definition der Körbe – der folgenden Übersicht zu entnehmen.

Basiskorb:

- AIX XL Pascal Compiler
- AIX Windows Environment
- AIX Windows Environment, 3D-Feature
- AIX PC Simulator
- AIX Computer Graphics Interface Toolkit
- AIX X-Station Manager
- AIX SDE Workbench¹
- AIX SDE Integrator¹
- AIX Base
- AIX HANFS
- AIX FDDI
- AIX BMPX
- AIX Info Explorer
- AIX XL Fortran Compiler/Runtime
- AIX XL C++ Compiler
- AIX ESSL (Engineering and Scientific Subroutine Library)
- AIX Visualisation Data Explorer

Ergänzungskorb

- AIX Network Management
- AIX 3278/3279 Emulation
- AIX VS Cobol Compiler¹
- AIX 3270 Host Connection
- AIX SNA Services
- AIX OSL (Optimization Subroutine Library)²
- AIX OSL/GUI (OSL/Graphic User Interface)²
- AIX Netview Service Point
- AIX Windows Interface Composer
- AIX OSI-MF
- AIX Graphics Plotting System
- AIX Graphics File Translator
- AIX APL2
- AIX Netview³
- AIX HA/6000 (High Availability/6000)

Bei Bestellungen im Jahre 1993 sind einige Änderungen in der Zusammenstellung der Körbe zu berücksichtigen (siehe Fußnoten).

DECcampus-Vertrag für DEC-Software

Mit der Firma DEC ist ein Campus-Lizenzvertrag für Workstation-Software abgeschlossen worden, der für den Universitätsbereich die Nutzung fast aller DEC-Software-Produkte für VMS und Ultrix regelt.

¹Diese Produkte sind 1993 nicht mehr Bestandteil der Körbe.

²Diese Software gehört jetzt zum Basiskorb.

³Dies wird ersetzt durch AIX Netview SNMP und AIX Netview EUI.

Die Kosten für einen Rechner ergeben sich aus der Preis-Kategorie, der dieser entsprechend einer Leistungseinstufung durch DEC zugeordnet ist (A-E), und sind für den Erwerb der Lizenz (Einstieg) sowie jährlich für die weitere Pflege und Aktualisierung zu zahlen. Unter den 87 Systemen, die bereits in den Vertrag einbezogen worden sind, befinden sich auch die Bereichs-Server VAX 9000 und VAX 4000.

Weitere Informationen zu den Vertragsbedingungen für entsprechende Systeme in der Universität sind beim Leiter des Universitätsrechenzentrums zu erhalten.

Benutzung der Schnelldrucker des Rechenzentrums von AIX-Rechnern aus

von

J. Hölters

Für die Ausgabe von Textdaten steht seit einiger Zeit das Kommando **lprint** auf den UNIX-Rechnern des Rechenzentrums zur Verfügung. Der Druck erfolgt auf den an der IBM-ES/9000 angeschlossenen Schnelldruckern. Diese Ausgabemöglichkeit ist für Programmlisten und Korrekturausdrucke geeignet.

Die Grundlage für die Nutzung der zentralen Drucker bildet eine Implementierung des Berkley-Netzwerk-Druckprotokolls (auch unter dem Namen *lpr/lpd* bekannt) im MVS. Die von uns genutzte Software ist an einer Hochschule in Kanada entwickelt worden und von uns auf die lokalen Verhältnisse angepaßt worden. Diese Software gestattet es, über das Netz-Druckprotokoll von anderen Rechnern aus Daten, mit den für das MVS spezifischen Attributen wie **FORMS**, **UCS**, **CLASS** und anderen, in der Spool des MVS zur weiteren Verarbeitung abzulegen. Außerdem wird je nach Setzung der Parameter eine Konversion der Daten von ASCII (genauer ISO-Zeichensatz 8859-1) nach EBCDIC (Codepage 037/1) und eine Interpretation von Steuerzeichen, wie *Backspace* zum Übereinanderdrucken von Zeichen und *Newline* zum Kennzeichnen des Zeilenendes, vorgenommen.

Das Kommando **lprint**, das auf den AIX-Rechnern zur Verfügung steht, macht von der oben beschriebenen Übertragungsmöglichkeit Gebrauch. Es erzeugt die für das MVS notwendigen Informationen und sorgt für die Übertragung der Steuerinformationen und der zu druckenden Daten in die Spool des MVS.

Die Syntax für den Aufruf des Kommandos **lprint** lautet:

```
lprint [files]
```

Dabei ist **files** eine Liste von einem oder mehreren Namen von Dateien, die gedruckt werden. Falls **files** nicht angegeben wird, liest das Kommando **lprint** von der Standard-Eingabe, und es werden die von dort gelesenen Daten gedruckt.

Abschicken von MVS-Jobs mittels FTP

von

M. Grote, R. Mersch, E. Sturm

Das Abschicken von MVS-Jobs und der Empfang der Job-Ausgabe mittels FTP ist etwas umständlich. Die in diesem Artikel beschriebenen Werkzeuge schaffen diesbezüglich Erleichterung.

Will man via FTP Jobs zum MVS schicken, so sind dazu immer etliche FTP-Kommandos abzusetzen. Dies wird besonders lästig beim Empfangen der Job-Ausgabe, die aus mehreren Dateien besteht, die einzeln transferiert werden müssen. Es lag daher nahe, diesen Vorgang mit Hilfe von Werkzeugen weitgehend zu automatisieren. Einige Mitarbeiter des Rechenzentrums haben derartige Werkzeuge für die Betriebssysteme DOS, OS/2, VMS und Unix erstellt und den jeweiligen Benutzern dieser Systeme verfügbar gemacht.

DOS-Kommandos zur Job-Steuerung

Über den Server des Rechenzentrums steht DOS-Benutzern das Kommando **MVS** zur Verfügung. Der erste Parameter gibt die gewünschte Funktion an. Bei jedem Aufruf wird FTP aufgerufen, das zunächst die Benutzerkennung und das Paßwort erfragt.

MVS PUT datei

schickt die Datei namens **datei** als Job ans MVS.

MVS DIR

zeigt die dem aufrufenden Benutzer gehörenden Objekte in der MVS-Spool, also auf die Ausführung wartende Jobs, laufende Jobs und Job-Ausgaben, an.

MVS GET jobnr [anzahl] [C]
überträgt die ersten *anzahl*, bzw. 10, wenn *anzahl* nicht angegeben wird, Ausgabedateien des Jobs mit der vierstelligen Spool-Kennung *jobnr* zum PC und faßt sie zu einer Datei zusammen. Bei Angabe von *C* werden alle Ausgabedateien des Jobs in der MVS-Spool gelöscht.

MVS DEL jobnr
löscht den Job mit der Spool-Kennung *jobnr* im MVS.

Die Spool-Kennung und die Anzahl der Ausgabedateien eines Jobs kann man der mittels **MVS DIR** erzeugten Liste entnehmen. Sie sieht typischerweise folgendermaßen aus:

```
MERSCHA JOB08015 OUTPUT 4 Spool Files
MERSCHB JOB08016 INPUT          -HELD-
MERSCHC JOB08561 ACTIVE
```

Hier sieht man, daß der Job mit der Spool-Kennung 8561 gerade ausgeführt wird, daß der gehaltene Job 8016 auf die Ausführung wartet und daß der Job 8015 fertig ist und 4 Ausgabe-Dateien vorliegen. Diese können nun mittels **MVS GET 8015 4 C** zum PC übertragen und im MVS gelöscht werden. Auf dem PC werden sie zu einer Datei namens **JOB08015.OUT** zusammengefaßt.

Anmerkung: Das Funktionieren dieser Kommando-Prozedur hängt von der auf dem PC verwendeten TCP/IP-Version ab. Sie sind erstellt worden für die derzeit im Pool des Rechenzentrums verwendete Version 2.05 von PC/TCP. Wenn Sie ein anderes TCP/IP verwenden, kann es nötig sein, die Kommando-Prozedur anzupassen.

OS/2-Objekt zur Jobsteuerung

In OS/2 möchte man natürlich nicht erst das Befehlszeilenfenster hervorkramen, sondern möglichst alles mit der Maus machen. Auf der Arbeitsoberfläche der PCs im CIP-Pool des Rechenzentrums finden Sie einen Caesarkopf (mit Lorbeerkranz), der zu folgenden Taten bereit ist:

- Wenn man eine Job-Datei über ihm abwirft, wird sie zum MVS geschickt. (Die Datei durch Drücken der rechten Maustaste ergreifen, zu Caesar bewegen, die Taste loslassen.)
- Nach Doppelklicken (mit der linken Maustaste, wenn der Mauszeiger über Caesar schwebt) erscheint ein Menü, in dem man (mit der linken Maustaste) einen Wunsch äußern kann:

- Jobliste anschauen,
- Job abschicken (Alternative zu Abwerfen),
- Job holen,
- Jobs löschen.

Danach wird man in jedem Fall nach Name und Paßwort gefragt, im Falle „Job holen“ auch noch nach dem Namen des Zielordners. Normalerweise klicke man (mit der linken Maustaste) das Feld OK an; wenn man etwas nicht möchte, das Feld CANCEL.

Im Prinzip ist alles problemlos. Wenn jemand noch keine Erfahrung mit einer *filebox* hat, möge er einen Job nur durch Abwerfen (also nicht per Menü) zum MVS schicken. Da die zugrundeliegende Software „Visual REXX“ nicht mehr kann, ist die Auswahl beim Punkt „Jobs löschen“ auf 10 Jobs beschränkt.

VMS-Kommandos zur Job-Steuerung

VMS-Benutzern steht nach Absetzen des Befehls **\$ RUN LIBRARY** das Kommando **MVS** zur Verfügung. Es verhält sich wie das **MVS**-Kommando im DOS (s. o.), verfügt jedoch über einige Erweiterungen. Die wichtigsten Erweiterungen sind:

- Benutzerkennung und Paßwort werden nur beim ersten Aufruf von **MVS** in einer Sitzung erfragt. **MVS** merkt sich diese Informationen und verwendet sie bei allen folgenden Aufrufen während derselben Sitzung, ohne erneut danach zu fragen. Sollte es jedoch einmal nötig sein, die gemerkte Benutzerkennung und/oder das gemerkte Paßwort zu ändern, so kann dies geschehen mittels **\$ MVS USER userid**. Die Angabe *userid* wird als Benutzerkennung für nachfolgende Aufrufe von **MVS** gesetzt. Das zugehörige Paßwort wird erfragt und ebenfalls vermerkt.
- Das **MVS**-Paßwort kann geändert werden: **\$ MVS PASS userid** erfragt das bisherige und das neue **MVS**-Paßwort und veranlaßt diese Paßwort-Änderung im **MVS**.

Unix-Kommandos zur Job-Steuerung

Auf *obelix* und auf allen Unix-Rechnern, die das Dateisystem **/usr/u0** von *obelix* montiert haben, steht das Kommando **mvs** zur Verfügung. Es verhält sich wie das **MVS**-Kommando im DOS (s. o.), mit dem Unterschied, daß die Benutzerkennung nicht erfragt wird (hier wird die Kennung des aufrufenden

Benutzers verwendet), und daß `mvs get` immer alle Ausgabe-Dateien eines Jobs überträgt.

Zusätzlich gibt es einige Kommandos, die die vom CMS gewohnten Kommandos zur Job-Bearbeitung nachbilden. Sie lauten `sendjob`, `jstatus`, `receive`, und `jcancel` und gehen in ihren Möglichkeiten über das von `mvs` gebotene hinaus. Ihr Gebrauch soll im folgenden an Hand von Beispielen erläutert werden. Bei jedem Aufruf wird das MVS-Paßwort des Benutzers erfragt.

```
$ sendjob demo
```

schickt die Datei `demo` als Job ins MVS.

```
$ jstatus
```

fragt den Status der eigenen Jobs in der MVS-Spool ab.

```
$ receive JOB07655
```

kopiert die Ausgabe-Dateien des Jobs mit der Spool-Kennung 7655 in das Unix-Arbeitsverzeichnis. Sie erhalten die Namen `JOB07655.1`, `JOB07655.2` usw. Existieren bereits Dateien mit diesen Namen, so wird die Endung `.1`, wenn es die so entstehenden Namen schon gibt, `.2` usw., angehängt (`JOB07655.1.1`, `JOB07655.2.1`). Die Option `-r` verhindert diese Namenserverweiterung; bereits vorhandene Dateien werden dann überschrieben.

Nicht mehr benötigte Spool-Dateien sollten mit `jcancel` gelöscht werden:

```
$ jcancel JOB07655
```

`sendjob` erlaubt es, mehrere Dateien abzusenden. Jede Datei wird als eigener Job gestartet.

```
$ sendjob listds demo
```

verschickt die Dateien `listds` und `demo` als Jobs ins MVS.

Bei `receive` genügt es, einen Teil des Jobnamens anzugeben:

```
$ receive 59
```

kopiert die Ausgabe-Dateien aller Jobs, deren Spool-Kennung „59“ enthält, ins Arbeitsverzeichnis.

Manchmal ist es wünschenswert, die einzelnen Teile der Job-Ausgabe in einer Datei zu haben. Dies kann durch die Option `-j` verlangt werden:

```
$ receive -j 7659
```

empfängt die Job-Ausgabe in der Datei `JOB07659.0`.

Die Option `-d` beim `receive` bewirkt, daß die Ausgabe-Dateien im MVS gelöscht werden. Dies geschieht auch dann, wenn die Ausgabe nicht empfangen werden konnte. (Normalerweise unterbleibt das Löschen und muß dann mittels `jcancel` nachgeholt werden.)

Die verschiedenen Optionen von `receive` können zusammengefaßt werden:

```
$ receive -dr 59
```

Die Ausgabe-Dateien im MVS werden gelöscht und u. U. existierende Dateien gleichen Namens im Arbeitsverzeichnis werden überschrieben.

Die Eingabe eines Paßworts bei jedem Befehl kann durch Angabe der Option `-n` bei den einzelnen Befehlen vermieden werden. Diese Option setzt aber die Existenz einer Datei `.netrc` im HOME-Verzeichnis des Benutzers voraus, welche das MVS-Paßwort im Klartext enthält. Wenn Sie eine solche Datei anlegen, sollten Sie auf jeden Fall darauf achten, daß kein anderer Benutzer diese lesen kann. `.netrc` hat das Format

```
machine caesar login grote password geheim1
```

Änderungen der Voreinstellungen der Befehle sind mit der Shell-Variablen `JOBFLAGS` möglich, z. B. bewirkt

```
JOBFLAGS= -nd
```

daß die Datei `.netrc` verwendet wird und Output bei `receive` gelöscht wird.

Neues vom SPSS

von

S. Zörkendörfer

SPSS für Windows

Die „Landeslizenz“ für SPSS für IBM-kompatible PCs wurde um SPSS for Windows erweitert. Es sei auch an dieser Stelle nochmal darauf hingewiesen, daß im Rahmen dieser Lizenz Exemplare zur Installation auf Universitätsrechnern bereitgestellt werden, nicht jedoch für häusliche Arbeitsplätze von Studenten. Die Kosten konnten bisher aus zentralen Mitteln bestritten werden, es wird nach Stückzahl abgerechnet. Aus diesem Grund muß jedes der Produkte SPSS/PC+

¹Mein Paßwort ist geheim, also sicherlich nicht geheim!

(derzeit Version 4.01) und SPSS for Windows (Version 5.0) separat beantragt werden; aus Sparsamkeitserwägungen ist mitzuteilen, wenn die Benutzung eines Exemplars eingestellt wird. Beide Produkte werden über das Software-Verteilungssystem VAMP ausgeliefert. SPSS GmbH gewährt zu Handbüchern für die Landeslizenz einen Rabatt.

Mit SPSS for Windows werden im Rahmen dieser Lizenz neben dem „Base system“ die „Professional Statistics“ und die „Advanced Statistics“ bereitgestellt. Mit folgender Kommandoliste, die ich im wesentlichen unverändert aus der Online-Hilfe kopiert habe, gebe ich alten Freunden des SPSS sicherlich eine gute Einschätzung zum Umfang dieses Systems: Add Files, Add Value Labels, Aggregate, Alscal, Anova, Apply Dictionary, Autorecode, Begin Data, Break, Cancorr als Makro, Clear Transformations, Cluster, Cnlnr, Comment, Compute, Correlations, Count, Coxreg, Crosstabs, Data List, Derivatives, Descriptives, Disclass als Makro, Discriminant, Display, Document, Do If, Do Repeat, Drop Documents, End Case, End Data, End File, Examine, Execute, Export, Factor, File Handle, File Label, File Type, Filter, Finish, Flip, Formats, Frequencies, Get, Get Capture, Get SAS, Get Translate, Graph (Bar, Line, Area, Pie, Scatter, Histogram, Boxplot), Hiloglinear, If, Import, Include, Input Program, Kaplan-Meier KM, Leave, List, Logistic, Loglinear, Loop, Makro Facility, Manova, Match Files, Matrix Language, Matrix Data, MConvert, Means, Missing Values, Model Program, Mult Response, N Of Cases, Nlr, Nonpar Corr, Npar Tests, Numbered, Numeric, Oneway, Partial Corr, Plot, Preserve, Print, Print Eject, Print Formats, Print Space, Probit, Procedure Output, Proximities, Quick Cluster, Rank, Recode, Record Type, Regression, Reliability, Rename Vars, Repeating Data, Report, Reread, Restore, Ridge-reg als Makro, Sample, Save, Save Translate, Select If, Set, Show, Sort Cases, Split File, String, Subtitle, Survival, Sysfile Info, Tables, Temporary, Time Program, Title, t-Test, Unnumbered, Update, Value Labels, Variable Labels, Vector, Weight, Write, Write Formats, Xsave.

Erste Erfahrungen gewann ich auf einem 386er mit Koprozessor, VGA, 8 MB, Maus, Deskjet (Grafik!); die vollständige Installation belegt knapp 20 MB der Festplatte. Damit ist wohl auch zur Systemvoraussetzung eine Mindestausstattung genannt. (Mit dieser Ausstattung gelang nach gut einer Stunde Rechenzeit immerhin eine Faktorenanalyse mit 300 Variablen und 1000 Beobachtungen.)

Am PC wird der SPSS^x-Fan dieses System dem SPSS/PC+ vorziehen. (Allerdings erlaubte

SPSS/PC+ eine deutschsprachige Installation, die Windows-Version wurde ausschließlich als englischsprachige Version ausgeliefert.) Erst nach einigem Experimentieren wird der Kenner in der Fensterumgebung den eigentlichen SPSS-Programmtext wiederfinden; dem Einsteiger mag die Existenz einer „SPSS-Programmiersprache“ verborgen bleiben! Die Anschaffung eines Satzes Handbücher empfehle ich für Institutsbibliotheken auch dann, wenn bereits Handbücher zum SPSS^x oder SPSS/PC+ vorliegen.

Als wesentliche Neuerung sind die Integration der Präsentationsgrafik (anstelle einer Schnittstelle zu einem Fremdprodukt) und des interaktiven Dateneditors (anstelle DataEntry) ins Basisprodukt zu nennen. Während der Sitzung werden zur Orientierung angeboten ein Hilfe-Menü, in das aus der jeweiligen Dialogbox gezielt eingestiegen werden kann, eine Liste der Syntaxdiagramme, eine Liste der Tastenbelegungen (hot keys) für die unterschiedlichen Fenstertypen und ein Glossar der statistischen Fachwörter.

Categories unter SPSS/PC+

Bei der Erneuerung der SPSS-Landeslizenz konnte der Umfang der MS-DOS-Version SPSS/PC+ um die Prozeduren der Option Categories erweitert werden. Wir werden dafür im VAMP zwei zusätzliche Disketten C1, C2 und C3 zum Nachkopieren bereitstellen, eine besondere Beantragung ist für dieses Zusatzprodukt nicht notwendig.

SPSS^x bzw. SPSS am Großrechner

Am Großrechner (vgl. *inforum* Nr. 1/1992, S. 9) stellen wir SPSS unter dem Betriebssystem MVS/ESA im Stapelbetrieb zur Verfügung, angebunden ist LISREL 7 mit PRELIS. Der Aufruf erfolgt unter der katalogisierten Prozedur SPSS, bei Bedarf kann im Einzelfall eine interaktive Nutzung vereinbart werden. Die Version 9 stellen wir nur noch auf Nachfrage hin zur Verfügung, wir möchten uns endgültig von ihr trennen.

SPSS unter Unix

Wir bereiten vor, SPSS auf einem Unix-Rechner im Netz anzubieten. Die Lehrveranstaltung im Sommersemester 93 zum SPSS wird voraussichtlich außer in die Windows-Version auch in eine Unix-Version einführen.

Zentrale Sicherung von Workstation-Daten

von

R. Mersch

Zu den Dienstleistungen, die **caesar**, der MVS-Rechner im Rechenzentrum, wahrnehmen soll, gehört auch die Sicherung (Backup) und Archivierung der im heterogenen LAN der Universität verteilten Daten. Die z. Zt. in der Industrie stattfindenden Entwicklungen auf diesem Gebiet werden vom Rechenzentrum aufmerksam verfolgt, grundlegende Funktionen sind aber bereits vorhanden. Dieser Artikel gibt einige Hinweise zu deren Nutzung.

Im LAN der Universität befindet sich eine Vielzahl von Rechnern, sowohl Arbeitsplatz-Rechner als auch Server, mit unterschiedlichen Betriebssystemen, wie DOS, OS/2, VMS und Unix in seinen verschiedenen Varianten. Für die Benutzer und Verantwortlichen dieser Rechner stellt sich bezüglich der dort residierenden Daten die Frage, wie Sicherungskopien (Backups) zum Zwecke des Schutzes vor unbeabsichtigter Zerstörung angelegt und – wegen des lokalen Platzmangels häufig auch – wohin Daten ausgelagert (archiviert) werden können.

Es bietet sich an, Backups und zu archivierende Daten auf **caesar**, unserem MVS-Rechner, abzulegen. Dort sind sowohl ein gewisser Vorrat an Plattenplatz für diese Zwecke, als auch die für die Verwaltung großer Datenmengen nötigen Hilfsmittel vorhanden. Jeder Benutzer, der über eine universitätsweite Benutzerkennung verfügt, kann seine Dateien mittels FTP nach **caesar** transferieren. Zwei Aspekte sind dabei zu beachten:

- Wenn mehr als nur einige wenige Dateien kopiert werden sollen, empfiehlt es sich, diese zunächst zu einer Datei, im folgenden *Backup-Datei* genannt, zusammenzufassen.
- Da diese Backup-Dateien u. U. nie wieder gebraucht werden, sollten sie im MVS so abgelegt werden, daß sie keine teuren Ressourcen verschwenden.

Diese Aspekte werden in den folgenden Abschnitten näher erläutert, wobei der Schwerpunkt auf der Sicherung von Benutzerdaten liegt. System-Backups erfordern in der Regel weitere Überlegungen, die den Rahmen dieses Artikels sprengen würden.

Erstellung einer Backup-Datei

Es gibt diverse Programme, die das Zusammenpacken von Dateien erlauben, für unterschiedliche Rechner-typen und Betriebssysteme. Ein in der Unix-Welt verbreitetes Programm ist **tar**, das es mittlerweile auch für DOS-Rechner gibt. So ist beispielsweise in der Version 2.05 von PC/TCP, also der derzeit aktuellen TCP/IP-Version im Universitäts-LAN, ein **tar** enthalten. Viele Systeme, z. B. Unix, VMS und OS/2, bieten ein backup-Programm, mit dem ebenfalls Backup-Dateien (im VMS-Bereich *Container-Dateien* genannt) erstellt werden können. Weitere Alternativen sind die im Public Domain anzutreffenden Archivierungsprogramme, insbesondere **zip**.

Bei der Auswahl sind ggf. Besonderheiten des jeweiligen Systems zu berücksichtigen. Im OS/2 muß das Programm beispielsweise in der Lage sein, mit erweiterten Attributen umzugehen. Im Zweifel sollte man zunächst einen Test durchführen oder im Rechenzentrum nachfragen.

Beispiel: Ein Unix-Benutzer will seine in den Directories Daten, Buch und allen deren Subdirectories enthaltenen Dateien zu einer Backup-Datei zusammenfassen, wobei Daten und Buch Subdirectories seiner Home-Directory seien. Dazu mache er zunächst seine Home-Directory zur aktuellen Directory:

```
cd
```

und befehle dann:

```
tar cvf mydat.tar Daten Buch
```

Die Optionen **cvf** steuern, was zu tun ist:

- c Es soll eine Backup-Datei erzeugt werden (create).
- v Der Name jeder Datei, die bearbeitet wird, soll angezeigt werden (verbose).
- f Das Backup soll in eine Datei geschrieben werden, deren Name dieser Option folgt (hier: **mydat.tar**).

Hinter den Optionen und dem Namen der Ausgabe-Datei werden die Namen der zu bearbeitenden Dateien und Directories angegeben, im obigen Beispiel also Daten und Buch. Bei Angabe einer Directory wird der gesamte von ihr aufgespannte Dateibaum eingeschlossen.

Will man aus einer mittels **tar** erzeugten Backup-Datei Dateien wiederherstellen, so ruft man **tar** mit der Option **x** (extract) auf. So kann man aus der oben erzeugten Back-Datei mittels

```
tar xvf mydat.tar
```

alle Dateien, mittels

```
tar xf mydat.tar Buch/kap1.tex
```

die Datei `kap1.tex` in der Directory `Buch` restaurieren.

Obacht ist geboten hinsichtlich der beim Aufruf von `tar` angegebenen Dateinamen, da beim Wiederherstellen einer Datei der beim Erstellen der Backup-Datei verwendete Name, bezogen auf die aktuelle Directory, verwendet wird. Im obigen Beispiel wurden relative Namen, also Namen, die sich auf die aktuelle Directory beziehen, verwendet. Statt dessen hätte man auch absolute Dateinamen verwenden können, z. B.:

```
tar cvf mydat.tar $HOME/Daten $HOME/Buch
```

Dadurch verliert man aber die Möglichkeit, die Dateien an anderer Stelle des Dateibaumes wiederherzustellen, wir empfehlen es also nicht.

Man sollte auch darüber nachdenken, ob es sich lohnt, die Backup-Datei zu komprimieren.

Ablage der Backup-Datei im MVS

Die Übertragung von Backup-Dateien ins MVS kann mit Hilfe von FTP geschehen. Da sie in der Regel selten oder nie wieder gebraucht werden, sollten sie im MVS nicht den für aktive Dateien bestimmten Plattenplatz belegen. Für Backup-Dateien wurde deshalb eine Verwaltungsumgebung¹ geschaffen, die dafür sorgt, daß sie schon nach kurzer Zeit (ca. 2 Tage) auf Magnetbänder ausgelagert (migriert) werden. Sie sind damit aber weiterhin genauso ansprechbar wie nicht migrierte Dateien auch, können also z. B. mittels des FTP-Kommandos `dir` angezeigt und mittels `get` zurücktransferriert werden. Letzteres erfordert einen *Rückruf* (Recall) der Datei: sie wird vom Magnetband zurück auf die Magnetplatte kopiert. Der Rückruf erfolgt automatisch.

Diese Verwaltungsumgebung sorgt ferner dafür, daß eine Backup-Datei gelöscht wird, nachdem sie 180 Tage nicht benutzt wurde.

Entscheidend ist nun, daß das System Backup-Dateien als solche erkennt, damit sie der entsprechenden Verwaltungs-Umgebung zugeordnet werden können. Dies geschieht über den Dateinamen; dessen zweiter Qualifizierer muß lauten: `$BACKUP$`. Am einfachsten erreicht man dies, indem man zu Beginn der FTP-Sitzung befiehlt:

```
cd $backup$
```

Beim Übertragen der Backup-Datei ist folgendes zu beachten:

- In der Regel muß binär übertragen werden. Dies wird erreicht durch das FTP-Kommando `binary` oder `image`.
- VMS-Benutzer sollten die VMS-Version des FTP verwenden – sie erreicht man mit dem FTP-Befehl `VMS` – und die Option `/FDL` beim Übertragen verwenden, also `PUT /FDL datei` bzw. `GET /FDL datei`.
- Die Datei wird im MVS so angelegt, daß sie maximal 11 MB groß werden kann. Wenn das nicht reicht, muß man die Größeneinstellungen im FTP etwa wie folgt ändern:

```
quote site cylinders
quote site primary=2
quote site secondary=10
```

Nach dem Absetzen dieser FTP-Kommandos werden die Dateien im MVS mit einer Größe von 2 Zylindern angelegt (1 Zylinder ist etwa 750 KB groß). Bei Bedarf wachsen sie bis zu 15 mal um 10 Zylinder, können also maximal 152 Zylinder (ca. 114 MB) groß werden. Diese Setzungen passen für kleinere und mittlere Backup-Dateien ganz gut.

Die zum Übertragen einer Backup-Datei nötigen FTP-Kommandos für Benutzer von Unix-, DOS-, oder OS/2-Systemen seien im folgenden noch einmal beispielhaft zusammengestellt. Die FTP-Verbindung bestehe und der Benutzer habe sich bereits angemeldet.

```
binary
cd $backup$
quote site cylinders
quote site primary=2
quote site secondary=10
put mydat.tar mydat.d210193.tar
```

Die Backup-Datei `mydat.tar` wird unter dem Namen `userid.$BACKUP$.MYDAT.D210193.TAR`² im MVS abgelegt. `userid` ist die beim Anmelden angegebene Benutzerkennung. Der Name der Zielfeile wurde so gewählt, daß man ihm entnehmen kann, welche Daten (`MYDAT`) wann (`D210193`) und mit welchem Backup-Programm (`TAR`) hier gesichert wurden. Eine derartige Organisation des Dateinamens ist nicht vorgeschrieben, erleichtert es aber, den Überblick über die Backup-Dateien zu behalten, und wird daher empfohlen.

¹Den Dateien wird vom SMS (Storage Management Subsystem) die Management-Klasse `BACKUP` zugeordnet.

²Ein MVS-Dateiname ist eine maximal 44 Zeichen lange, durch Punkte voneinander getrennte Liste von Qualifizierern, die ihrerseits maximal 8 Zeichen lang sind, aus Buchstaben, Ziffern `$`, `@` und `*` bestehen können und nicht mit einer Ziffer anfangen dürfen.

Zurückholen der Backup-Datei

Eine Backup-Datei kann mit dem FTP-Kommando `get` zurückgeholt werden, wobei zuvor binäre Übertragung eingestellt werden muß!

Es gilt zu berücksichtigen, daß der Zugriff auf eine Backup-Datei in der Regel zunächst den Rückruf der migrierten Datei bedeutet. Dies erfordert das Auflegen eines Magnetbandes, kann also nur erfolgen, wenn Operateure anwesend sind, und nimmt dann einige Minuten in Anspruch. Die meisten der FTP-Programme, die ich bisher ausprobiert habe, warten so lange; nur das PC/TCP-FTP für DOS-Benutzer im Universitäts-LAN bekommt nach kurzer Zeit einen Time-out und gerät in einen undefinierten Zustand, so daß in der Regel nur ein Neustart des PCs hilft. Zwar wird der Rückruf der Backup-Datei trotzdem vollzogen, so daß der nächste Transfer-Versuch erfolgversprechend ist, doch ist dies nun keine Vorgehensweise, die man empfehlen mag. Wir hoffen, dieses Problem bald beheben zu können. Will man es umgehen, so muß man dafür sorgen, daß die Datei vor dem Zugriff durch FTP bereits zurückgerufen ist, etwa dadurch, daß man einen Job mit dem TSO-Kommando `HRECALL` abschickt. So können DOS-Benutzer, und nur für sie ist diese Vorgehensweise z. Zt. empfehlenswert, den Rückruf einer migrierten Backup-Datei mittels eines wie folgt aussehenden Jobs anstoßen:

```
//userid1 JOB (TEST),userid
/*JOBPARM TAPES=1
//RECALL EXEC TSOCMD
//SYSIN DD *
HRECALL $BACKUP$.TEXTE.D280193.TAR
```

Für `userid` ist die Benutzerkennung einzusetzen. Der Rückruf der Backup-Datei `userid.$BACKUP$.TEXTE.D280193.TAR` wird nur angestoßen und ist keineswegs schon fertig, wenn der Job beendet ist, man muß also vor dem Transfer nachschauen, ob die Datei noch migriert ist oder nicht. Diese Information erhält man mit Hilfe des FTP-Kommandos `dir`.

Anmerkungen und Hinweise

- Für Backup-Dateien steht z. Zt. *Plattenplatz* im Umfang von ca. 1,8 GB zur Verfügung. Dies muß für alle noch nicht migrierten sowie zurückgerufenen Backup-Dateien aller Benutzer reichen, so daß es bei intensiver Nutzung durchaus zu Engpässen kommen kann. Sollte dies geschehen, werden wir über eine Vergrößerung des Plattenplatzes nachdenken.

- Migrierte Backup-Dateien liegen auf *Magnetband-Kassetten*. Engpässe sind dort unmittelbar nicht zu befürchten, doch ist ein ausufernder Bestand an Magnetband-Kassetten aus unterschiedlichen Gründen (Platzbedarf, Kosten) durchaus mit Problemen verbunden. Deshalb die Bitte: **Löschen Sie nicht mehr benötigte Backup-Dateien.**
- Backup-Dateien verschlingen nicht nur bei ihrer Speicherung, sondern auch bei ihrer *Übertragung* Ressourcen. Wenn es sich einrichten läßt, sollten Sie sie nachts ins MVS übertragen. Auf dem LAN-Server des Rechenzentrums steht DOS-Benutzern das Kommando `MVSBACK` zur Verfügung, das nach dem Erfragen der benötigten Informationen 10 Stunden wartet und dann den Dateitransfer mit Hilfe von FTP durchführt. Während der Wartezeit kann der Vorgang abgebrochen werden. Das Protokoll der FTP-Sitzung wird an die Datei `MVSBACK.LOG` angehängt. Hilfe erhält man durch Aufruf von `MVSBACK` ohne Parameter.
- Ein prinzipielles Problem der hier beschriebenen Vorgehensweise besteht darin, daß die Backup-Datei zunächst einmal lokal erzeugt wird; auf dem jeweiligen Rechner muß also genügend Platz für sie vorhanden sein. Ist dies nicht der Fall, so kann man daran denken, das Backup in mehrere Teile zu zerlegen oder ein Netz-Laufwerk zu nutzen.

Die mir bekannten Unix-Systeme (AIX und SunOS) bieten einen Ausweg aus diesem Problem: ein Bindestrich (-) als Name der lokalen Datei beim FTP-Kommando `put` oder `get` sowie als Name der Backup-Datei beim `tar` spezifiziert die Standard-Ein/Ausgabe. Man kann dann beispielsweise die vom `tar` erzeugte Backup-Datei über eine Pipe an das FTP weiterreichen, ohne sie lokal ablegen zu müssen. (Rückfragen bei Problemen: Tel. 24 88, E-Mail mersch@uni-muenster.de.)

Das neue T_EX-System unter AIX und SunOS

von

F. Perske

Seit Anfang 1993 steht auf den vom Universitätsrechenzentrum gepflegten RS/6000- und Sun-Systemen ein neues T_EX zur Verfügung.

Die Komponenten des T_EX-Systems liegen in folgenden Versionen vor:

Komponente	Version
T _E X	3.141
METAFONT	2.71
L ^A T _E X	2.09 vom 25.3.92
MakeIndex	2.11
dvips	5.497
xdvi	16
dvi2xx	0.51alpha3

Die Verzeichnis-Bäume von T_EX und METAFONT befinden sich unter /usr/u0/tex bzw. /usr/u0/mf.¹

Der Aufruf der einzelnen Programme erfolgt über die Binaries im Verzeichnis /usr/u0/bin.rs6000 bzw. /usr/u0/bin.sun4. Das entsprechende Verzeichnis muß also im Pfad enthalten sein, um mit den Komponenten des T_EX-Systems arbeiten zu können.

Um T_EX bzw. L^AT_EX unter den verschiedenen Betriebssystemen innerhalb der Universität einigermaßen einheitlich aufrufen zu können, sind folgende Festlegungen getroffen worden:

Aufruf	Beschreibung
n _l atex	L ^A T _E X mit neuem Schriftauswahlmechanismus (NFSS) und deutscher, engl. und franz. Silbentrennung
l _a tex	L ^A T _E X ohne neuen Schriftauswahlmechanismus und mit deutscher, engl. und franz. Silbentrennung
s _l i _T ex	SL ^A T _E X mit deutscher, engl. und franz. Silbentrennung
t _e x	Plain mit deutscher, engl. und franz. Silbentrennung
v _i r _t ex	ohne vorgeladenes Format
i _n i _t ex	Ini _T E _X

Wie bisher wird automatisch die deutsche Silbentrennung eingeschaltet, wenn in der \documentstyle-Markierung des L^AT_EX-Dokumentes der Teilstil german eingetragen ist.

Für englische Texte kann der gleiche Aufruf verwendet werden, da die englische Silbentrennung voreingestellt ist, wenn der Teilstil german nicht angegeben wird.

Der SL^AT_EX-Aufruf funktioniert analog.

Außerdem ist es nun möglich, über die Markierungen

```
\selectlanguage{german}
\selectlanguage{english}
\selectlanguage{french}
```

im Text die Sprache zu wechseln, wobei sowohl die Silbentrennung als auch die von L^AT_EX automatisch generierten Texte, wie z. B. „Kapitel“ oder „Inhaltsverzeichnis“, entsprechend der angegebenen Sprache eingestellt werden. Diese Markierungen stehen nur zur Verfügung, wenn in der \documentstyle-Markierung der Teilstil german angegeben wird.

Allen L^AT_EX-Benutzern empfehlen wir, den n_latex-Aufruf anstelle des latex-Aufrufes zu verwenden. Er bietet mit dem neuen Schriftauswahlmechanismus (NFSS) wesentlich verbesserte Möglichkeiten zur Einbindung neuer Schriften und wird in zukünftigen L^AT_EX-Versionen den alten Mechanismus völlig ersetzen. Schon jetzt gibt es viele zusätzliche L^AT_EX-Stil-Dateien, die nur zusammen mit diesem neuen Schriftauswahlmechanismus verwendet werden können.

Weitere Informationen zum neuen Schriftauswahlmechanismus finden Sie in den gleichnamigen **inform**-Artikeln in Heft Nr. 1/1992, 3/1992 und in dieser Ausgabe und in den Dateien nfss.tex bzw. nfss.dvi im Verzeichnis /usr/u0/tex/inputs/initex/nfss.

AMS-T_EX-Markierungen stehen unter L^AT_EX zur Verfügung, wenn der n_latex-Aufruf verwendet wird und in der \documentstyle-Markierung die entsprechenden AMS-L^AT_EX-Stile eingetragen werden.

Zum Arbeiten mit P_TCT_EX werden im Vorspann des L^AT_EX-Dokumentes folgende Markierungen eingebunden:

```
\input{prepica}
\input{pictex}
\input{postpic}
```

Dann kann dieses Dokument mit dem latex-Aufruf verarbeitet werden. Der Aufruf von n_latex ist in diesem Fall leider noch nicht möglich.

Für die Verarbeitung von dvi-Dateien stehen im Moment folgende Treiber zur Verfügung:

- xdvi als Bildschirm-Treiber für das X-Window-System,
- dvips für die Umwandlung in das PostScript-Format und
- die dvi2xx-Druckertreiber-Familie.

¹Das Verzeichnis /usr/u0 des Rechners obelix ist über NFS exportiert. Es kann vom Systemverwalter eines im Netz eingebundenen Unix-Rechners über den mount-Befehl verfügbar gemacht werden.

In der dvi2xx-Familie sind folgende Druckertreiber enthalten:

- dvi1j2 konvertiert dvi-Dateien nach HP PCL zum Drucken auf einem HP LaserJet⁺, HP LaserJet II und kompatiblen Druckern.
- dvi1j2p erzeugt ebenfalls HP PCL, allerdings zur Ausgabe auf HP LaserJet IIP, HP LaserJet III, HP LaserJet 2000 und kompatiblen Druckern. Diese Version benutzt neuere PCL-Kommandos und unterstützt das Drucken im Querformat.
- dvi1jp und dvi1j sind identisch zu dvi1j2p und dvi1j2, aber sie unterstützen nur „128 character fonts“. Sie sollten für LaserJet-Kompatible benutzt werden, die keine „256 character fonts“ unterstützen.
- dvi3812 konvertiert nach PMP, die Kommando-Sprache des IBM 3812-Druckers.

Viele nützliche Informationen zu den Programmen finden sich auch in den Manual-Pages. Dies betrifft unter anderem auch die Optionen beim Aufruf dieser Programme.

Falls sich irgendwelche Probleme bei der Benutzung der neuen Installation ergeben sollten, wenden Sie sich bitte an mich (perskef@uni-muenster.de) oder an Herrn Kaspar (kaspar@uni-muenster.de, Tel. 2473).

Anwendungssoftware in Unix

von

B. Süselbeck

Im folgenden soll eine kurze Zusammenfassung der vom Rechenzentrum unter Unix angebotenen Software aus dem Bereich des „scientific computing“ gegeben werden.

Numerik

Im Bereich der Numerik sind die beiden großen Fortran-Bibliotheken

- NAG
- IMSL

zu nennen. Sie stehen für die RISC-Systeme der Firmen IBM, Sun und DEC zur Verfügung. Speziell für die Sun-Welt ist auch eine C-Version der IMSL-Bibliothek vorhanden. An dieser Stelle sollte auch der Fortran-90-Compiler von NAG erwähnt werden (nur Sun).

Grafik

Sammlungen grafischer Verfahren in Form von Programm-bibliotheken sind

- IMSL Exponent Graphics Library
- NAG Graphics Library

Sie stehen ebenfalls auf den oben genannten Rechnerplattformen zur Verfügung. Dies gilt auch für das mächtige, in erster Linie dialogorientierte Grafiksystem

PV~WAVE

der Firma PVI. In Kürze wird hierzu noch die menügesteuerte Variante

PV~WAVE Point & Click

angeboten.

Symbolik

Der sehr zukunfts-trächtige Bereich des symbolischen Rechnens wird durch eine ganze Reihe von Produkten unterstützt:

- MATHEMATICA
- MAPLE
- REDUCE
- AXIOM

AXIOM ist zur Zeit nur in der IBM-Welt verfügbar. Die anderen Systeme werden im Moment ebenfalls nur auf den RS/6000-Rechnern angeboten. Hier sind jedoch Versionen in einzelnen Fachbereichen verfügbar. Das Rechenzentrum bemüht sich derzeit um eine Zusammenfassung von verstreuten Lizenzen.

Statistik

Ein System, das nur unter Unix eingesetzt werden kann, ist

- S-PLUS

Es wird zur Zeit in der Sun-Umgebung angeboten. Das in der Universität schon länger im Einsatz befindliche Produkt

- BMDP

ist vom MVS ins Unix verschoben worden und steht dort für Sun- und IBM-Rechner zur Verfügung. Die Systeme SPSS und SAS sind unter Unix noch nicht im Einsatz. Ihre Beschaffung befindet sich noch in der Planungsphase.

Der Aufruf der einzelnen Produkte und Informationen zu Lizenzbedingungen werden in Kürze im Zusammenhang mit einem Konzept zur Bereitstellung von Anwendungssoftware in heterogenen Unix-Netzen im einzelnen vorgestellt.

Neues vom SAS

von

S. Zörkendörfer

Mit dem „Developer's Release“ unter Windows steht erstmals SAS in einer aktuellen Version mit allen Optionen zur Verfügung.

Zum Statistical Analysis System SAS ist erfreulicherweise zu berichten, daß die Hochschullandeslizenz auch 1993 Gültigkeit hat. Damit ist SAS auf PCs unter MS-DOS und Windows angemietet – allerdings konnte die für Januar '93 erwartete Auslieferung der Windows-Version 6.08D aus Diskettenmangel nicht rechtzeitig erfolgen. Auch nach Auslieferung wird nicht gesichert sein, ob wir zu Präsentationsszwecken eine vollständige Referenzinstallation bereithalten können, mir wurden hierzu (für eine vollständige Installation mit unkomprimierten Koordinatendateien) 179.1 MB als erforderliche Plattenkapazität angegeben.

Angekündigt sind folgende Optionen: BASE SAS mit Programmiersprache, integrierter Oberfläche, Dateiverwaltung, Macrosprache; SAS/ACCESS mit Zugriff auf dBASE-DBF- und Lotus-DIF-Dateien; SAS/AF zur Entwicklung menügesteuerter Anwendungen; SAS/ASSIST als Prototyp zu menügesteuerten Anwendungen; SAS/CALC als Tabellenkalkulation; SAS/CONNECT zur Kooperation mit SAS auf

anderen Plattformen (z. B. zum MVS oder CMS, falls dort auch lizenziert); SAS/EIS als Managementsystem; SAS/ENGLISH zur Formulierung für Abfragen und Berichte; SAS/ETS für Zeitreihenvorhersagen; SAS/FSP zur maskenorientierten Dateipflege und für Serienbriefe; SAS/GRAPH als Präsentationsgrafik, inklusive umfassender Koordinatendateien zu thematischen Karten; SAS/IML als interaktive Matrixsprache nicht nur für Numeriker/Statistiker, sondern auch etwa zum inkrementellen Aufbau einer Grafik; SAS/INSIGHT mit interaktiver statistischer Grafik; SAS/LAB zur geführten interaktiven Laborauswertung; SAS/OR als Operations Research mit Project Management; SAS/PH-Clinical für pharmazeutische und biotechnische Berichte zu klinischen Erhebungen; SAS/QC zur Qualitätskontrolle und SAS/STAT eben als das Statistische Analyse-System.

Unter denjenigen obiger Optionen, die wir auf den bisher benutzten Plattformen MVS/ESA, VM/CMS und MS-DOS nicht angemietet haben, wird uns als erstes SAS/INSIGHT zur ausführlichen Erprobung reizen. Da die Mikrorechner im CIP-Pool des Rechenzentrums im Gebäude Einsteinstraße, Raum 107, nun mit lokalen Platten ausgestattet sind, können Sie die MS-DOS-Version SAS 6.04 auf diesen Rechnern zu Übungszwecken unter dem Aufruf SAS604 erproben (grafische Ausgabe am Laserdrucker im Übergang mit device=hp1js2), für Produktionen sollte das Produkt im Rahmen dieser Landeslizenz auf instituts-eigenen PCs angemietet werden.

Am IBM-Großrechner steht SAS in der Version 6.07 zur Verfügung. Neu ist etwa der grafische Editor bei unserer SAS/GRAPH-Installation unter CMS; unter MVS haben wir neben dem joborientierten Aufruf zusätzlich (vorrangig für unsere Lehrveranstaltungen) einen interaktiven Zugang eingerichtet, Sie gelangen in diese Umgebung, indem Sie (z. B. nach tn caesar) im Logon-Schirm unter „Procedure =>“ den Eintrag „SASKURS“ eintippen. Die Lehrveranstaltung in den Semesterferien wird diesen Zugang zum SAS benutzen.

RUM-Tutorial

E-Mail und NetNews mit Unix und VM/CMS

von

B. Neukäter

Die rechnervermittelte Kommunikation (engl. computer mediated communication) gewinnt auch für den gelegentlichen DV-Anwender immer mehr an Bedeutung. Daraus erwächst dem Universitätsrechenzentrum die Verpflichtung, den Zugang zu diesen Diensten so einfach wie möglich zu gestalten und den aktuellen Stand der Entwicklung immer wieder so zu beschreiben, daß auch der DV-Laie diese Dienste nutzen kann. Dieser Artikel soll ein Beitrag dazu sein.

Rechnervermittelte Kommunikationsdienste dienen der zwischenmenschlichen Kommunikation wie Telefon, Telefax, Rundfunk und Fernsehen; sie sind jedoch vielfältiger und flexibler als diese konventionellen Medien. Die elektronische Datenverarbeitung bietet neue Möglichkeiten zum Austausch von Gedanken und Meinungen zwischen Partnern in der ganzen Welt. Zwei Dienste, die diesem Ziel dienen, sollen hier beschrieben werden: die Elektronische Post (E-Mail) und das Konferenzsystem NetNews. Dabei sollen diesmal besonders die Realisierungen dieser Dienste im Betriebssystem Unix im Vordergrund stehen.

Sie können einen elektronischen Brief an einen oder mehrere Partner schicken. Er wird innerhalb weniger Minuten ausgeliefert. Dazu müssen Sie natürlich die E-Mail-Adressen Ihrer Partner kennen. Ihre eigene E-Mail-Adresse können Sie in der Form

kennung@uni-muenster.de

Ihren Partnern mitteilen, wobei für **kennung** Ihre Benutzerkennung einzusetzen ist.

Der Briefpartner schaut von Zeit zu Zeit in seinen Briefkasten (mailbox) und beantwortet seine elektronische Post. Anstelle eines menschlichen Briefpartners kann auch ein Briefverteilungsdienst (mailing list) treten, der die Korrespondenz an alle Teilnehmer verteilt, deren E-Mail-Adressen in einer Liste eingetragen sind. Die elektronische Post wird auf diese Weise zu einem Medium für die Gruppenkommunikation. Es können elektronische Foren eingerichtet werden, in denen sich Teilnehmer mit gleichen Interessen zusammenfinden und über ihre Probleme diskutieren – ohne Terminschwierigkeiten und Parkplatz-

probleme. Entfernungen spielen keine Rolle und die Umwelt wird geschont.

Diesem Ziel dient auch das weltweite Konferenzsystem NetNews, auch bekannt als News oder unter dem Namen des Netzes USENET. Die Beiträge der Teilnehmer an einem NetNews-Forum zu einem bestimmten Thema werden in Form von NetNews-Artikeln gespeichert und – sofern es sich um ein globales Forum handelt – weltweit verteilt. Die Artikel eines Forums werden zu einer Gruppe zusammengefaßt, einer *newsgroup*. Sie können die Artikel lesen, neue Artikel oder zu vorhandenen Beiträgen Nachfolge-Artikel schreiben, die dann ebenfalls allen Teilnehmern zugänglich gemacht werden. Sie haben auch die Möglichkeit, dem Verfasser eines Artikels mittels E-Mail zu antworten, wobei Ihre E-Mail-Adresse und die des Empfängers automatisch eingetragen werden.

NetNews-Foren (*newsgroups*) sind hierarchisch nach Themen geordnet. Unter dem Oberbegriff (*high level index*) *sci* für science gibt es z. B. Foren wie *sci.archaeology* oder *sci.geo.geology* oder *sci.geo.meteorology*.

Zu den „klassischen“ *newsgroups* zählen Foren unter den Oberbegriffen:

comp	Computer science, Informatik
sci	Wissenschaft (science)
soc	Kultur und Gesellschaft (society)
talk	Debatten, längliche Diskussionen
news	Das NetNews-Netz (USENET)
rec	Hobbies (recreational activities)
misc	Verschiedenes (miscellaneous)

Daneben gibt es weitere Oberbegriffe wie *de* (deutsche Foren), *alt* (alternative Foren), *bit* (BITNET-LISTSERV-Foren).

Eine besondere Rolle spielen für uns die lokalen Foren unter dem Oberbegriff *wwu*. Während Ihre Beiträge zu den anderen Foren weltweit oder in bestimmten Regionen verteilt werden, bleiben Ihre Beiträge zu Foren, die mit *wwu*. beginnen, innerhalb der Universität.

E-Mail mit Unix

Nachdem Sie sich bei einem Unix-System angemeldet haben, können Sie E-Mail versenden und empfangen. Kurze Nachrichten versenden Sie mit

mail local-part@host-part

In der Regel ist local-part die Benutzerkennung Ihres Partners und host-part eine Rechnerkennung oder ein Verwaltungsbereich (domain) des Netzwerks. Nutzt Ihr Partner das gleiche System, so kann „@host-part“ entfallen. Es können mehrere E-Mail-Adressen im Kommando angegeben werden.

Sie werden nach dem „Subject“ gefragt und geben in einer Zeile eine kurze Zusammenfassung des Inhalts Ihrer Nachricht ein. Danach folgt die Nachricht selbst. Mit der Backspace-Taste (←) löschen Sie die letzten Zeichen, wenn Sie sich vertippt haben. Möchten Sie, daß die E-Mail noch nicht abgeschickt wird, so können Sie durch zweimalige Eingabe von <Strg-C> erreichen, daß sie in der Datei dead.letter gespeichert wird. Weitere Möglichkeiten erfahren Sie, wenn Sie ? eingeben.

Sie beenden die Eingabe entweder mit einer Zeile, die nur aus einem Punkt (.) besteht, oder mit <Strg-D>. Anschließend werden Sie gefragt, wem Ihre E-Mail noch „nachrichtlich“ geschickt werden soll (cc:). Sie können E-Mail-Adressen eingeben oder einfach nur die Eingabetaste drücken.

Bei längeren Mitteilungen empfiehlt es sich, vorher mit Hilfe eines Editors (z. B. *vi* oder *sedit*) eine Datei zu erzeugen und danach diese Datei mit

```
mail e-mail-adressen < dateiname &
```

abzuschicken. Sie können das Kommando auch ohne das abschließende Und-Zeichen (&) eingeben. Das hat jedoch den Nachteil, daß Sie auf die Beendigung des Kommandos warten müssen, bevor Sie ein neues eingeben können.

Wenn Sie das Kommando

```
mail
```

geben, so wird Ihnen eine Liste der eingegangenen E-Mail gezeigt. Möchten Sie Ihre Post lesen, so betätigen Sie für jede Nachricht die Eingabetaste. Die gelesene Post wird in einer Datei namens mbox aufbewahrt.

Das Kommando mail versetzt das System in einen Zustand, in dem nur bestimmte Kommandos eingegeben werden können, z. B.:

```
- Vorhergehende Nachricht anzeigen,
more aktuelle Nachricht seitenweise ausgeben,
n n-te Nachricht ausgeben (n: Zahl),
d aktuelle Nachricht löschen,
s dn aktuelle Nachricht speichern (dn: Dateiname),
r aktuelle Nachricht beantworten,
```

```
m adr aktuelle Nachricht als E-Mail weiterleiten
(adre: E-Mail-Adresse),
```

```
h Liste der Nachrichten (headers) ausgeben,
```

```
? Hilfe anfordern,
```

```
q Mail-Kommando beenden.
```

Mit dem Kommando

```
mail -f
```

lesen Sie die in der Datei mbox aufbewahrte Post. Es stehen die oben beschriebenen Mail-Kommandos zur Verfügung. Haben Sie mit dem Mail-Kommando

```
s e.briefe
```

Nachrichten gespeichert, so können Sie diese mittels

```
mail -f e.briefe
```

bearbeiten.

Oft haben Ihre Partner lange und komplizierte E-Mail-Adressen. Damit Sie diese nicht jedesmal eintippen müssen, legen Sie sich eine Datei namens .mailrc an und tragen Kurzbezeichnungen ein, z. B.

```
alias kurt
```

```
zy7fx10@rechner.institut.uni-xyz.de
```

```
alias susanne KLSK078@XYZVM.BITNET
```

Von nun an geben Sie statt der E-Mail-Adresse einfach kurt oder susanne ein. Sie können auch mehrere Adressen für eine Kurzbezeichnung angeben.

Möchten Sie, daß Ihre mit kennung@uni-muenster.de adressierte Post an ein anderes System (z. B. DMSWWU1A) weitergeleitet wird, so erzeugen Sie eine Datei .forward, die nur Ihre E-Mail-Adresse auf diesem System enthält.

Weitere Informationen können Sie sich mit dem Kommando

```
man mail
```

verschaffen.

NetNews mit Unix

Um an NetNews-Konferenzen teilzunehmen, brauchen Sie einen *news reader client*. Es gibt für Unix verschiedene *clients*: rn, xrn, nn, tin usw. Welcher am geeignetsten ist, hängt von den Möglichkeiten des Terminals ab und von Ihrem persönlichen Geschmack.

Während xrn nur an X-Window-Terminals zu gebrauchen ist, reicht für rn, nn oder tin eine VT100-Emulation.

Da `tin` an vielen Terminals zu gebrauchen ist und relativ leicht zu handhaben, möchte ich Ihnen dieses Programm kurz beschreiben.

Sie geben das Kommando

`tin`

und erhalten beim erstmaligen Aufruf auf einer Seite eine Kurzbeschreibung. Wenn Sie die Eingabetaste drücken, so erscheint das Gruppenauswahlmenü. Sie bewegen mit den Pfeiltasten den Zeiger (Balken), der das aktuelle Forum anzeigt, und wählen mit `s` (select) die Foren aus, an denen Sie sich beteiligen möchten. Mit `u` (unselect) machen Sie die Auswahl rückgängig. Mit `y` schalten Sie zwischen allen erreichbaren Foren und den von Ihnen ausgewählten hin und her. Mit der Eingabetaste erreichen Sie, daß Ihnen für das angezeigte Forum eine Liste der Artikel gezeigt wird. Mit Hilfe der Pfeiltasten und der Eingabetaste lassen Sie sich einen Artikel zeigen. Mittels Leertaste (space) blättern sie weiter, mit `b` zurück. Der Tabulatorsprung (`→|`) führt zum nächsten Artikel und `h` verhilft Ihnen zu weiteren Informationen. Mit `q` gelangen Sie vom Artikel zur Artikelliste, von der Artikelliste zum Gruppenauswahlmenü und von dort schließlich zur Kommandoebene.

In der Artikelliste sind die Themen (subjects) am linken Rande numeriert. Rechts daneben finden Sie eine Zahl, die angibt, wieviel Antworten (follow-ups) zu einem Thema vorliegen. Die „Gesprächsfäden“ werden also gebündelt.

Möchten Sie selbst etwas zur Diskussion beitragen, so können Sie entweder mit `f` zu einem Artikel, den Sie gerade lesen, einen Nachfolgeartikel schreiben oder mit `w` eine neue Diskussion beginnen. Für erste Versuche eignet sich das Forum `www.test`. Bevor Sie das tun, sollten Sie sich allerdings ein wenig mit dem Editor `vi` vertraut machen ...

E-Mail mit VM/CMS

Sie versenden E-Mail mit dem Kommando

`mail local-part@host-part`

In der Regel ist `local-part` die Benutzerkennung Ihres Partners und `host-part` eine Rechnerkennung oder ein Verwaltungsbereich (*domain*) des Netzes. Nutzt Ihr Partner das gleiche System, so kann `@host-part` entfallen. Es können mehrere E-Mail-Adressen im Kommando angegeben werden.

Sie werden in der Regel nach dem Namen Ihres Partners und nach dem „Subject“ gefragt, also einer kurzen Zusammenfassung des Inhalts Ihrer Nachricht. Danach erscheint der Eingabebildschirm für

Ihre Nachricht. Beim Schreiben Ihres Textes stehen Ihnen die Pfeiltasten und alle Korrekturmöglichkeiten der Tastatur natürlich zur Verfügung. Mit der Funktionstaste `F5` (Send) senden Sie Ihre Nachricht ab. Bevor die Nachricht abgeschickt wird, werden Sie aufgefordert, diese Aktion zu bestätigen. Dazu können Sie ebenfalls die Funktionstaste `F5` verwenden.

Eine Liste der eingegangenen E-Mail wird Ihnen gezeigt, wenn Sie das Kommando

`mail`

eingeben. Am unteren Rand des Bildschirms werden Ihnen die Funktionstasten kurz erklärt. Mit dem Positionszeiger (cursor) kennzeichnen Sie die E-Mail, die Sie lesen möchten, und betätigen die Funktionstaste `F2` (Open). Mittels `F11` (Save) fügen Sie die gelesene Nachricht in die Datei `ALL NOTEBOOK A1` ein und mit `F3` (Quit) kehren Sie zum Ausgangsmenü zurück. Mit `F9` (Delete) löschen Sie die Nachricht aus der Eingangsdatei (`UNREAD NOTEBOOK A1`) und beenden das Mail-Kommando mit `F3`. Sie werden noch gefragt, ob Sie die gelöschten Nachrichten auch wirklich nicht mehr brauchen, und wenn Sie das durch Drücken der Eingabetaste bestätigen, so stellt das Mail-Kommando seine Tätigkeit ein.

Ihre in `ALL NOTEBOOK A1` abgelegte Post lesen Sie mit

`mailbook all`

Die Benutzerführung von `mailbook` und `mail` sind weitgehend identisch.

Gelegentlich werden Sie durch Systemmeldungen (messages) belästigt. In diesem Falle verhilft Ihnen die Funktionstaste `F12` (Cancel) wieder zu einem übersichtlichen Bildschirm.

Fragen stellt Ihnen das Mail-Kommando, indem es auf dem Bildschirm ein Fenster öffnet, wo Sie entweder einen Text eintragen können oder mittels Tabulatorsprung (`→|`) ein Feld auswählen können. Sie schließen das Fenster mit der Entertaste oder einer Funktionstaste.

Am oberen Rand des Bildschirms können Sie eine Menüleiste entdecken. Mit der Funktionstaste `F10` (MenuBar) bringen Sie den Positionszeiger in die Menüleiste. Nachdem Sie mit dem Tabulatorsprung eine Auswahl getroffen haben, drücken Sie die Entertaste, und das Menü öffnet sich. Mit dem Positionszeiger wählen Sie aus und führen mit der Entertaste die angesteuerte Funktion aus. Auf diese Weise können Sie Ihre Post in verschiedenen Dateien (notebooks) ablegen, drucken, weiterleiten, beantworten,

löschen, als gelesen markieren usw. Es gibt Edierhilfen wie *cut* und *paste* und allerlei Möglichkeiten, die Benutzeroberfläche an persönliche Bedürfnisse anzupassen. Probieren Sie es doch einfach einmal aus. Wenn Sie Hilfe benötigen, so steuern Sie entweder das Hilfe-Menü an, oder Sie kommandieren

`help mail`

oder

`help mailbook`

Die E-Mail-Adressen Ihrer Partner verwalten Sie mit dem Kommando

`names`

Es zeigt Ihnen ein Bildschirmformular. Dort tragen Sie Kurznamen (nickname), persönliche Daten (name) sowie local-part (userid) und host-part (node) der E-Mail-Adresse ein und betätigen die Funktionstasten F2 (add) und am Schluß F3 (quit). Anstatt der E-Mail-Adressen können Sie jetzt in den Kommandos `mail` und `mailbook` die Kurznamen verwenden.

Möchten Sie Ihre an DMSWWU1A geschickte E-Mail nicht mittels VM/CMS lesen, so teilen Sie dem Dienstprogramm MAILER durch

`tell mailer forward add local-part@host-part`

Ihren Wunsch nach Weiterleitung mit. Das Kommando

`help mailer`

liefert Ihnen weitere Informationen.

NetNews mit VM/CMS

Im CMS gibt es zur Zeit noch zwei Systeme und deshalb auch zwei Kommandos zum Lesen der NetNews: `netnews` und `nnr`. Mit dem Kommando `netnews` erreichen Sie im wesentlichen nur die News, deren Einspeisung in Unix technische Schwierigkeiten bereitet. Die lokalen Foren (wwu) und die meisten globalen newsgroups sind mit

`nnr`

erreichbar. Die Handhabung ist einfach. Sie sehen zunächst ein Menü, in dem nur die höchsten Indizes einer jeden Gruppenhierarchie angezeigt werden, also z. B. `wwu` oder `comp`. Sie entscheiden sich für eine Hierarchie (mittels Positionszeiger und Funktionstaste F2 (All News) oder F4 (New News), siehe Hinweise am

unteren Bildschirmrand) und sehen nun sämtliche Foren oder nur die Foren, die neue Beiträge enthalten. Nun wählen Sie, ob Sie sofort die Artikel sehen wollen (F2) oder erst die Überschriften (F4). Mit F3 treten Sie den Rückweg an.

Sie beteiligen sich an bestimmten Foren, indem Sie diese im Gruppenauswahlmenü anwählen und lesen. Sie sind dann optisch hervorgehoben und Ihnen werden nur die Artikel angeboten, die Sie noch nicht gelesen haben. Mit der Funktionstaste F9 (Sub/UnS) können Sie sich wieder abmelden.

Das *posting* ist ein zweistufiger Vorgang. Nach Auswahl mit dem Positionszeiger betätigen Sie F5. Der Effekt dieser Taste hängt davon ab, welches Menü gerade angezeigt wird. In der Regel müssen Sie den Editor aufrufen (F5), ihren Text verfassen, sichern (F5), in die erste Stufe zurück (F3) und abschicken (F6).

Weitere Einzelheiten erfahren Sie, wenn Sie die Taste F1 (Help) betätigen oder auf der Kommandoebene `help nnr` eingeben.

Ein neuer Schriftauswahlmechanismus für L^AT_EX – letzter Teil (3)

von

W. Kaspar

In diesem letzten Teil der Artikelreihe¹ werden Änderungen und Erweiterungen, die den Satz von mathematischen Formeln betreffen, besprochen. Außerdem werden Markierungen vorgestellt, mit denen alle bisher besprochenen Merkmale von Schriftzeichen direkt angewählt werden können.

In der bisherigen L^AT_EX-Version war es für eine Formel völlig egal, ob sie innerhalb eines normal, kursiv oder fett gesetzten Textes stand – auf den Schriftschnitt und auch die Schriftfamilie innerhalb der Formel hatte dies keinen Einfluß.

Auch unter dem neuen Schriftauswahlmechanismus ist dies so beibehalten worden. Die Trennung zwischen dem normalen Text und einer Formel ist hier jedoch noch deutlicher, da die im normalen Text verwendeten Markierungen, wie z. B. `\sf` oder `\bf`, im Formelsatz nicht mehr zugelassen sind, bzw. dort einfach ignoriert werden.

¹Die beiden vorhergehenden Artikel sind in *inforum* Nr. 1/1992 und 2/1992 erschienen.

Um trotzdem Zeichen mit anderem Schriftschnitt und aus anderen Schriftfamilien in Formeln verwenden zu können, wurde im neuen Schriftauswahlmechanismus der Begriff des *Math-Alphabets* eingeführt.

Bevor wir näher auf diese Math-Alphabete eingehen, wollen wir uns zunächst einem weiteren neuen Begriff, dem der Math-Version, zuwenden, der eng mit den Math-Alphabeten verbunden ist und der aus einer Weiterentwicklung von `\boldmath` bzw. `\unboldmath` hervorgegangen ist.

Math-Versionen

Eine Math-Version bestimmt das Aussehen aller Schriftzeichen in einer Formel. Über die folgende Markierung, die nur außerhalb von Formeln verwendet werden darf, wird die Math-Version für alle weiteren Formeln festgelegt.

`\mathversion{Name einer Math-Version}`

Standardmäßig wird die Math-Version `normal` verwendet. Die folgende Formel wurde hiermit gesetzt:

$$\delta(x-t) = \sum_{n=0}^{\infty} \varphi_n^*(x) \varphi_n(t)$$

Ändern wir nun die Math-Version, indem wir z. B. die Markierung

`\mathversion{bold}`

vor der Formel einfügen, so wird sie in der Math-Version `bold` (fett) gesetzt:

$$\delta(x-t) = \sum_{n=0}^{\infty} \varphi_n^*(x) \varphi_n(t)$$

Eine ähnliche Wirkung konnten wir auch in der bisherigen $\text{\LaTeX}$ -Version erzielen, indem wir vor der Formel `\boldmath` angegeben haben. Ohne den neuen Schriftauswahlmechanismus hätte das Ergebnis dann so ausgesehen:

$$\delta(x-t) = \sum_{n=0}^{\infty} \varphi_n^*(x) \varphi_n(t)$$

Der Vergleich der beiden letzten Formeln zeigt, daß unter dem neuen Schriftauswahlmechanismus auch hoch- und tiefgestellte Zeichen sowie mathematische Symbole in dem fetten Schriftschnitt gesetzt werden.

Die alten Markierungen `\boldmath` und `\unboldmath` stehen auch weiterhin zur Verfügung und haben unter dem neuen Schriftauswahlmechanismus die gleiche Wirkung wie `\mathversion{bold}` und `\mathversion{normal}`.

$\text{\LaTeX}$ kennt standardmäßig nur die beiden Math-Versionen `normal` und `bold`. Weitere Math-Versionen werden normalerweise über Zusatzstile hinzugefügt. Ein Beispiel hierfür ist der `concrete`-Stil, der in [2] kurz beschrieben ist.

Math-Alphabete

Mit Hilfe der Math-Alphabete werden Zeichen mit anderem Schriftschnitt und aus anderen Schriftfamilien für den Formelsatz bereitgestellt.

So ist z. B. `\cal` die Markierung für ein Math-Alphabet, mit dem wir im Formelsatz über `\cal{A B C}` die kalligraphischen Zeichen *ABC* setzen können.

Weitere standardmäßig zur Verfügung stehende Markierungen für Math-Alphabete sind `\mit` und `\mathrm`, mit denen wir z. B. kursive griechische Großbuchstaben und nicht-kursive Formelzeichen setzen können. Die Eingabe

`\( S \in \cal{S}(M) \)`,
`\( \mit{\Gamma}(x) \)` und
`\( \mathrm{H}_2\mathrm{O} \)`

erzeugt also die Formelzeichen

$$S \in \mathcal{S}, \Gamma(x) \text{ und } \text{H}_2\text{O}$$

Die beiden Markierungen `\cal` und `\mit` standen schon immer in $\text{\LaTeX}$ zur Verfügung. Unter dem neuen Schriftauswahlmechanismus hat sich allerdings die Aufrufsyntax dieser Markierungen geändert. Mußte bisher z. B. `\cal A` geschrieben werden, so lautet die Eingabe nun `\cal{A}`, d.h. der zu setzende kalligraphische Buchstabe wird der Markierung `\cal` als Argument übergeben.²

Die bisher erwähnten Math-Alphabete können auch in der Math-Version `bold` verwendet werden. Das folgende Beispiel zeigt, wie sich dabei die Merkmale der Math-Alphabet-Zeichen verändern. Wir ergänzen das obige Beispiel

`\mathversion{bold} \( S \in \cal{S}(M) \)`
`...`

und erhalten

$$S \in \mathcal{S}, \Gamma(x) \text{ und } \text{H}_2\text{O}$$

²Über die Angabe des Teilstils `nomargid` kann die bisherige Aufrufsyntax auch unter dem neuen Schriftauswahlmechanismus wieder eingeführt werden (siehe auch Fußnote 2 im ersten Teil dieser Artikelreihe in *informatik* Nr. 1/1992).

Der Schriftschnitt unserer Math-Alphabete wurde also durch die Math-Version **bold** automatisch auf fett gesetzt.

Zusätzliche Math-Alphabete

Weitere Math-Alphabete stehen uns zur Verfügung, wenn wir in der `\documentstyle`-Markierung den Teilstil `amssymb` angeben.³ Mit `\Bbb` können wir dann „blackboard bold“-Zeichen, mit `\frac` Frakturzeichen und mit `\bold` fette Buchstaben setzen. So liefert z. B. die Eingabe

```
\( \frac{P}{M} \),
\(\mathbb{N}\) und
\(\nabla\boldsymbol{b}\)
```

folgende Formelnzeichen

$$\mathbb{P}(M), i \in \mathbb{N} \text{ und } \nabla \mathbf{b}$$

Auch in der Math-Version **bold** stehen entsprechende Buchstaben zur Verfügung:

$$\mathbb{P}(M), i \in \mathbb{N} \text{ und } \nabla \mathbf{b}$$

Außerdem enthält dieser Teilstil eine große Anzahl von Markierungen, über die wir zusätzliche mathematische Symbole, wie z. B.

$$\circ \rightsquigarrow \mapsto \succcurlyeq * \cup$$

setzen können. Eine vollständige Liste aller neuen Symbole ist in der Dokumentation des Teilstils abgedruckt.

Neue Math-Alphabete definieren

Zusätzlich zu den vorgegebenen können wir auch eigene Math-Alphabete einrichten. Hierdurch haben wir die Möglichkeit, in Formeln alle Schriftzeichen verwenden zu können, die wir bis jetzt nur im normalen Text setzen konnten.

Benötigen wir in einer Formel z. B. Buchstaben aus der Schriftfamilie „Computer Modern Sans Serif“ und wollen wir diese Zeichen über die Markierung `\mathsf` anwählen, so vereinbaren wir zunächst ein neues Math-Alphabet, indem wir die Markierung

```
\newmathalphabet{\mathsf}
```

in den Vorspann unseres $\text{\LaTeX}$ -Textes oder in eine spezielle Stildatei einfügen. Die allgemeine Form dieser Markierung lautet:

```
\newmathalphabet{Math-Alphabet-Markierung}
```

Nun müssen wir das neue Math-Alphabet den einzelnen Math-Versionen – hier in unserem Beispiel **normal** und **bold** – bekanntmachen und ihm dabei die gewünschten Merkmale seiner Schriftzeichen zuordnen. Der neue Schriftauswahlmechanismus stellt hierfür folgende Markierung zur Verfügung:

```
\addtoversion
{Math-Version}
{Math-Alphabet-Markierung}
{Kennung einer Schriftfamilie}
{Kennung einer Stärke und Breite}
{Kennung sonstiger Schriftzeichenformen}
```

Auch diese Markierung darf nur im Vorspann verwendet werden. Als Kennung können wir prinzipiell alle diejenigen verwenden, die wir schon im zweiten Teil dieser Artikelreihe⁴ besprochen haben.

Wählen wir eine Kennung aus, für die kein Zeichensatz existiert, so erzeugt $\text{\LaTeX}$ eine Fehlermeldung oder Warnung und teilt mit, welcher Zeichensatz anstelle des gewünschten verwendet wird. Eine Liste der Kennungen, für die standardmäßig Zeichensätze vorhanden sind, ist ebenfalls im zweiten Teil dieser Artikelreihe abgedruckt.

Für unser Beispiel legen wir nun fest, wie die Buchstaben unseres Math-Alphabets `\mathsf` in einer Formel aussehen sollen, die in der Math-Version **normal** gesetzt wird. Im Vorspann fügen wir hierfür die Markierung

```
\addtoversion{normal}{\mathsf}{cmss}{m}{n}
```

ein und wählen damit die gewünschte Schriftfamilie (`cmss`) in normaler Stärke, Breite (`m`) und Form (`n`) aus. Für die Math-Version **bold** vereinbaren wir mit der Markierung

```
\addtoversion{bold}{\mathsf}{cmss}{sbc}{n}
```

die gleiche Schriftfamilie (`cmss`), diesmal jedoch mit der Stärke halbfett und der Breite eng (`sbc` für semi-bold condensed).

Hiermit steht nun unter den beiden Math-Versionen **normal** und **bold** das neue Math-Alphabet `\mathsf` zur Verfügung. Wir können nun z. B.

```
\(\mathsf{T}_{ikj}\) bzw.
\mathversion{bold}\(\mathsf{T}_{ikj}\)
```

schreiben und erhalten als Ergebnis:

³Der Teilstil `amssymb` ist in der Standardinstallation des Universitätsrechenzentrums enthalten (siehe auch [1, Installation von Zusatzkomponenten]).

⁴siehe *informatik* Nr. 2/1992., Tabelle 1 und 2.

T_{ikj} bzw. T_{ikj} .

Zur Markierung `\addtoversion` gibt es noch eine *-Variante:

```
\addtoversion*
{Math-Alphabet-Markierung}
{Kennung einer Schriftfamilie}
{Kennung einer Stärke und Breite}
{Kennung sonstiger Schriftzeichenformen}
```

Sie unterscheidet sich von der ersten Form dadurch, daß keine Math-Version angegeben wird. Das neue Math-Alphabet wird mit dieser Markierung allen existierenden Math-Versionen bekanntgemacht, wobei dann natürlich die Schriftzeichen dieses Math-Alphabets in allen Math-Versionen die gleichen Merkmale besitzen.

Diese Markierung sollten wir verwenden, wenn die Zeichen für das Math-Alphabet in nur einer Schriftfamilie und einem Schriftschnitt vorliegen und wir diese Zeichen in mehreren Math-Versionen benötigen.

Für einzelne Math-Versionen können wir die Einstellungen aber auch noch nachträglich mit `\addtoversion`-Markierungen (ohne *) ändern.

Direkte Anwahl von Schriftzeichenmerkmalen

Die meisten Anwender werden mit den bisher vorgestellten Möglichkeiten des neuen Schriftauswahlmechanismus auskommen.

Für spezielle Anwendungen und solche $\text{\LaTeX}$ -Benutzer, die in größerem Umfang $\text{\LaTeX}$ -Stildateien ändern und neu entwerfen, werden im neuen Schriftauswahlmechanismus die Markierungen `\family`, `\series`, `\shape` und `\size` zur Verfügung gestellt, mit denen alle Merkmale von Schriftzeichen direkt eingestellt werden können. Für sie gilt folgende Syntax:⁵

```
\family{Kennung einer Schriftfamilie}
\series{Kennung einer Stärke und Breite}
\shape{Kennung sonstiger Schriftzeichenformen}
\size{Schriftgröße}{Grundlinienabstand}
```

Zur Angabe der Kennungen gilt hier das gleiche wie bei der `\addtoversion`-Markierung im letzten Abschnitt. So haben z. B. die beiden Markierungen

```
\family{cmss} \selectfont
```

die gleiche Wirkung wie `\sf`, und anstelle von `\it\bf` könnten wir auch

```
\shape{it} \series{bx} \selectfont
```

schreiben.

In der `\size`-Markierung wird die Schriftgröße *ohne* die Maßeinheit pt angegeben. Normalerweise können Schriftgrößen von 5 bis 12 pt in 1 pt Schritten und von 14, 17, 20 und 25 pt verwendet werden.

Das zweite Argument der `\size`-Markierung, der Grundlinienabstand, muß eine gültige Längenangabe enthalten, d. h. entweder eine Zahl, gefolgt von einer Einheit, oder eine Längenmarkierung. Mit den Markierungen

```
\size{8}{3.5mm} \selectfont
```

schalten wir also auf eine Schriftgröße von 8 pt um und legen den Abstand der Zeilengrundlinien auf 3,5 mm fest.

Normalerweise werden diese Markierungen aber nicht im Text verwendet, sondern es werden mit ihrer Hilfe innerhalb von Stildateien neue Markierungen definiert. So könnte in einer Stildatei mit

```
\newcommand{\bfm}{\series{b}\selectfont}
```

die Markierung `\bfm`⁶ vereinbart werden, um z. B. Ziffern in fetten Schriftzeichen spaltengerecht in Tabellen setzen zu können. Die Summe in der folgenden Tabelle ist links mit `\bf`, rechts aber mit `\bfm` gesetzt worden.

Position 1	3578,46	Position 1	3578,46
Position 2	2983,51	Position 2	2983,51
Summe	<u>6561,97</u>	Summe	<u>6561,97</u>

⁵ In der neuesten Version des Schriftauswahlmechanismus (NFSS), die demnächst auch über VAMP zur Verfügung stehen wird, sind die Namen der hier aufgeführten Markierungen geändert worden. Den bisherigen Namen wurde die Bezeichnung `font` vorangestellt, so daß sie nun `\fontsize`, `\fontseries`, `\fontshape` und `\fontfamily` lauten. Die alten Namen sind allerdings auch noch gültig, sollten aber nicht mehr verwendet werden.

⁶ Eine Stildatei, die diese Markierung schon enthält, ist z. B. `LTALL.STY`, die in der Zusatzkomponente `RESTM` im VAMP-Paket `TEXFONT` enthalten ist (siehe auch [1, Installation von Zusatzkomponenten]).

Abschließende Bemerkungen

Zusammenfassend können wir sagen, daß der neue Schriftauswahlmechanismus dem Autor ein übersichtliches Schema bietet, mit dessen Hilfe er die Schriftfamilien und Schriftschnitte, die er für seine Publikation benötigt, auf einfache Weise auswählen und zusammenstellen kann.

Auch dem Entwickler neuer Zusatzstile wird die Arbeit wesentlich erleichtert, da der neue Schriftauswahlmechanismus Werkzeuge zur Verfügung stellt, die das Einbinden neuer Zeichensätze sehr vereinfachen.

Weitere Informationen und Beispiele finden wir in den Dateien FONTDEF.MAX oder CONCRETE.STY im Verzeichnis `\TEX30\INITEX\FONTSEL7` oder in den Zusatzstildateien im Verzeichnis `\TEX30\FONTDOCU`. Außerdem enthält [4, S. 332–335] einen Abschnitt über dieses Thema.⁸

Literatur

Die Artikel [1] bis [3] sind in unserem TEX-Installationspaket enthalten. [1] befindet sich im Archiv TDR.ZOO (Datei TDRI.DVI) auf der SETUP-1-Diskette. [2] und [3] werden bei der Installation der LATEXFS-Komponente im Verzeichnis `\TEX30\INITEX\FONTSEL7` abgelegt. Der Artikel [2] befindet sich in der Datei FONTSEL.TEX bzw. FONTSEL.DVI⁷.

- [1] WOLFGANG KASPAR: *TEX – Installationshinweise für den PC*, Westfälische Wilhelms-Universität Münster, Universitätsrechenzentrum, Software-Information, 1991.
- [2] FRANK MITTELBACH UND RAINER SCHÖPF: The new Font Family Selection — User Interface to Standard L^AT_EX, *TUGboat* 11 1990, pp. 297–305.
- [3] FRANK MITTELBACH UND RAINER SCHÖPF: Datei FONTSEL.BUG⁷, Bestandteil des Fontsel-Softwarepaketes.
- [4] KOPKA, H., *L^AT_EX – Eine Einführung*, 4. überarb. u. erweít. Aufl., Bonn: Addison-Wesley 1992.

⁷In der neuesten Version ist in allen Namen das Kürzel „FONTSEL“ in „NFSS“ umbenannt worden.

⁸Dieser Abschnitt ist erst in der 4. Auflage hinzugekommen.

RUM-Grafik

„Besser als das BKA erlaubt“

von

E. Sturm

Auf unserem Farbrasterdrucker Canon CLC 300 können jetzt GIF-, PBMPLUS-, PCX-, PostScript-, RUM/GKS-, Targa- und natürlich wie bisher COLPLOT-Bilder ausgegeben werden.

Der Canon CLC 300 ist ein Farbrasterdrucker mit einer variablen Auflösung bis zu 400 dpi, wobei jeder der Bildpunkte eine von ungefähr 16 Millionen Farben hat. In gewissen Kreisen wird er zum Vervielfältigen von kleinen, bunten Scheinchen benutzt, bei uns hat er Anschluß an einen Rechner, und zwar an caesar, das MVS-System. Benutzt werden kann er aber auch von obelix aus, dem Unix-System. Wie, das soll im folgenden beschrieben werden.

Im Augenblick ist die Welt noch zweigeteilt: PostScript- und GIF-Bilder sind von obelix aus abzuschicken, PCX- und COLPLOT- von caesar aus; Targa- und PBMPLUS-Bilder können von beiden Maschinen aus abgesandt werden. Also der Reihe nach:

Möglichkeiten auf obelix

Auf obelix gibt es ein Kommando namens plot, das bereit und in der Lage ist, Dateien der Formate PBMPLUS(P6), GIF, PostScript und Targa umzuwandeln und an den Canon-Drucker weiterzureichen:

```
plot bild.egal
```

Mit PostScript ist nicht eine EPS-Datei (Encapsulated PostScript) gemeint, eine solche müßte man erst durch Hinzufügen des PostScript-Befehls showpage umwidmen. Die Interpretation der PostScript-Datei erfolgt nicht durch den Canon-Drucker selbst, sondern durch das GNU-Programm GhostScript, das sie in das PBMPLUS-Format P6 umwandelt – ein Format, das fast schon dem Geräte-Code des Canon entspricht.

PBMPLUS ist ein von Sun-Rechnern her bekanntes Konvertierungsprogramm, von dessen Rohdatenformaten wir zunächst nur das P6-Format akzeptieren. Hierbei handelt es sich im wesentlichen um die Kodierung der Bildpunkte in jeweils drei Bytes, nämlich für Rot, Grün und Blau. Wer also selbst programmiert und Bilddateien erzeugen will, findet hier ein höchst

einfaches Format, wenn er sein Bild auf Papier sehen möchte.

GIF ist ein Bilddateiformat, das von der Fa. CompuServe zur Versendung von Bildern in Netzen entwickelt wurde und dort weit verbreitet ist. Auch eine GIF-Datei wird zunächst nach P6 umgewandelt und über das Netz nach caesar geschickt. Möglich sind mit diesem Format allerdings nur 256 verschiedene Farben.

Das Targa-Format wird z. B. von Scannern erzeugt. Auch dieses wird vor der Übertragung nach caesar in P6 umgewandelt.

Möglichkeiten auf caesar

Wie man schon ahnt, erfolgt die Umwandlung des P6-Formats in das Canon-Format und die Ansteuerung des Druckers unter MVS. Dort sind, zumindest zur Zeit, die größten und schnellsten Platten vorhanden – ein Canon-Bild kann ja in der jetzigen Ausbaustufe bis zu 48 MB groß sein (bei 3 Byte pro Bildpunkt).

Das PCX-Format ist ursprünglich für Paintbrush entwickelt worden, inzwischen aber auch anderen Programmen bekannt. Hier sind ebenfalls nur 256 Farben möglich. Beim RUM/GKS-Format handelt es sich um unser altes Plotformat, das im Prinzip nur Striche in den Farben schwarz, rot und blau beschreiben kann (näheres siehe unter help rumgks im CMS).

Wenn man davon ausgeht, daß ein Bild als MVS-Datei vorliegt, so würde der folgende Job ein P6-, PCX-, RUM/GKS- oder Targa-Bild in die JES2-Spool schicken, von wo es dann automatisch zum Drucker übertragen würde:

```
//STURM1 JOB (TEST,J07), 'Eberhard Sturm'
// EXEC PLOT,DATASET='STURM.BILD.EGAL'
```

Das PLOT-Programm stellt selbst fest, um was für ein Format es sich handelt, und leitet das Bild entsprechend weiter. Bei den COLPLOT-Formaten blieb alles unverändert – deshalb sei hier nur auf inform Nr. 4/1990 verwiesen. Falls es Verzögerungen geben sollte: Die Implementierung des PCX-, des RUM/GKS- und des Targa-Formats war bei Redaktionsschluß noch in Arbeit.

Am Ende des Artikels die übliche Bemerkung: Natürlich befindet sich alles im Fluß, es werden sowohl weitere Formate als auch Zugriffsmöglichkeiten von weiteren Rechnern aus hinzukommen.

RUM-Grafik

RUM-Lehre

Lehrveranstaltungen im 1. Halbjahr 1993

Beratung zum Lehrangebot durch Herrn W. Bosse jeweils Di, Do 11-12, Tel. 83-2461

Vor Beginn des SS 1993 werden vom Universitätsrechenzentrum einige ganztägige Intensivkurse durchgeführt, in denen Stoffvermittlung und Übungen integriert sind. Diese Veranstaltungen sollen durch entsprechende Betreuung der Teilnehmer eigene Übungen 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 in die Anmelde listen einzutragen, und sollten unbedingt zu dem angekündigten Beginn anwesend sein. Die Teilnehmer dieser Kurse werden gebeten, diese im SS 1993 zu belegen. Dies ist neben der bestandenen Abschlußklausur Voraussetzung für die Aushändigung eines Scheines über die erfolgreiche Teilnahme.

Lehrveranstaltungen in den Semesterferien (Februar bis April 1993)

- | | | |
|--------|--|-----------------|
| 320018 | Statistische Datenanalyse mit dem Programmsystem SAS
vom 1.3. bis 12.3.1993 ganztägig
Hörsaal: M5, Beginn: 1.3.1993, 9 Uhr | Nienhaus, R. |
| 320022 | Textverarbeitung mit WordPerfect
vom 1.3. bis 12.3.1993 nachmittags
Hörsaal: CIP-Raum 405, Bisinghof 24-25
Beginn: 1.3.1993, 13 Uhr | Stöckelmann, D. |
| 320037 | Computerunterstütztes Publizieren mit $\text{\LaTeX}$
für Fortgeschrittene
vom 15.2. bis 26.2.1993 vormittags
Hörsaal: M4, Beginn: 15.2.1993, 9 Uhr | Kaspar, W. |
| 320041 | Einführung in Unix
vom 1.3. bis 12.3.1993 vormittags
Hörsaal: M2, Beginn: 1.3.1993, 9 Uhr | Ost, St. |
| 320056 | Programmieren in C
vom 15.3. bis 26.3.1993 ganztägig
Hörsaal: M4, Beginn: 15.3.1993, 9 Uhr | Mersch, R. |
| 320060 | Programmieren in FORTRAN
vom 15.3. bis 26.3.1993 ganztägig
Hörsaal: M5, Beginn: 15.3.1993, 9 Uhr | Mertz, K.-B. |
| 320075 | Vektorisierung und Parallelisierung mit FORTRAN
vom 1.3. bis 12.3.1993 vormittags
Hörsaal: M4, Beginn: 1.3.1993, 9 Uhr | Neukäter, B. |

Lehrveranstaltungen in der Vorlesungszeit

- | | | |
|--------|---|-----------|
| 320080 | Standardanwendungen auf Mikrorechnern
Do 15-17
Hörsaal: M4, Beginn: 22.4.1993 | Lange, W. |
| 320094 | Textverarbeitung auf Mikrorechnern ¹
Mi 13-15
Hörsaal: M5, Beginn: 21.4.1993 | Kamp, H. |

- | | | |
|--------|--|--|
| 320109 | Computerunterstütztes Publizieren mit $\text{\LaTeX}$
Do 13-15
Hörsaal: M6, Beginn: 22.4.1993 | Göllmann, L. |
| 320113 | Statistische Datenanalyse mit SPSS
Mo 15-17
Hörsaal: M4, Beginn: 19.4.1993 | Zörkendörfer, S. |
| 320128 | Datenvisualisierung mit PV~Wave
Di 15-17
Hörsaal: M2, Beginn: 27.4.1993 | Süselbeck, B. |
| 320132 | Einsatz relationaler Datenbanken mit SQL
Mi 13-15
Hörsaal: M2, Beginn: 28.4.1993 | Hölters, J. |
| 320147 | Einführung in Unix
Mi 15-17
Hörsaal: M2, Beginn: 28.4.1993 | Grote, M. |
| 320151 | Programmieren in FORTRAN
Mi 9-11
Hörsaal: M4, Beginn: 28.4.1993 | Reichel, K. |
| 320166 | Programmieren in C
Di 13-15
Hörsaal: M4, Beginn: 20.4.1993 | Perske, R. |
| 320170 | Programmieren in Pascal
Do 15-17
Hörsaal: M6, Beginn: 22.4.1993 | Pudlatz, H. |
| 320185 | Datenstrukturen und Programmierverfahren in Pascal
Di 15-17
Hörsaal: M4, Beginn: 27.4.1993 | Bosse, W. |
| 320190 | Programmieren unter OS/2
Mi 13-15
Hörsaal: M4, Beginn: 28.4.1993 | Sturm, E. |
| 320204 | Lokale Rechnernetze: Technologie und Dienste ¹
Do 13-15
Hörsaal: M4, Beginn: 29.4.1993 | Richter, G.
Speer, M. |
| 320219 | Rechnernetz-Praktikum ¹
Mo 13-15
Hörsaal: CIP-Raum 107, Rechenzentrum,
Beginn: 26.4.1993 | Kisker, H.-W. |
| 320223 | Themen des Datenschutzes
Di 13-14
Hörsaal: CIP-Raum 107, Rechenzentrum,
Beginn: 27.4.1993 | Kamp, H. |
| 320238 | Kolloquium über Themen der Informatik
Fr 15-17
Hörsaal: M4 | Held, W./
die wiss. Mitarbeiter
des Rechenzentrums |
| 320242 | Anleitung zum DV-Einsatz bei wissenschaftlichen Arbeiten
nach Vereinbarung | die wiss. Mitarbeiter
des Rechenzentrums |

¹Wegen der Begrenzung der Teilnehmerzahl ist für diese Lehrveranstaltung eine frühzeitige Anmeldung im Dispatch des Rechenzentrums erforderlich.

Kommentare zu den Lehrveranstaltungen

Einführung in Unix (320041, 320147)

Unix ist ein weitverbreitetes Mehrbenutzerbetriebssystem. Es ist auf Rechnern verschiedener Hersteller und unterschiedlicher Leistungsklassen ablauffähig. Damit steht dem Unix-Anwender vom Mikrorechner bis zum Großrechner die gleiche leistungsfähige und komfortable Programmier- und Arbeitsumgebung zur Verfügung. Hardware-Unterschiede der einzelnen Maschinen werden weitgehend verdeckt.

GULBINS: *Unix*, Springer

HEROLD: *Unix-Grundlagen – Kommandos und Konzepte*

LEFFLER: *The Design and Implementation of the 4.3 BSD Unix Operating System*

CHRISTIAN: *The Unix Operating System*, Wiley-Interscience

Programmieren in FORTRAN (320060, 320151)

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.

BRAUER: *Programmieren in FORTRAN 77*, Hühlig

BRAUER: *FORTRAN 77 – Ständig im Griff*, Hühlig

KIESSLING/LOWES: *Programmierung mit FORTRAN 77*, Teubner Studienskripten

METCALF: *Effective FORTRAN 77*, Oxford University Press

WEHNES: *FORTRAN 77*, Hanser

VS FORTRAN Release 4.0 (erhältlich im Rechenzentrum)

Programmieren in Pascal (320170)

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.

JENSEN/WIRTH: *Pascal, User Manual and Report*, Springer

MARTY: *Methodik der Programmierung in Pascal*, Springer

OTTMANN/WIDMEYER: *Programmieren mit Pascal*, Teubner

WILSON/ADDYMAN: *Leichtverständliche Einführung in das Programmieren mit Pascal*, Hanser

Alle Lehrbücher zu Turbo-Pascal ab Version 4.0

Datenstrukturen und Programmierverfahren in Pascal (320185)

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.

LIPSCHUTZ: *Datenstrukturen*, McGrawHill

OTTMANN/WIDMAYER: *Algorithmen und Datenstrukturen*, BI

SCHNEIDER/BRUELL: *Advanced Programming and Problem Solving with Pascal*, J. Wiley & Sons

WIRTH: *Algorithmen und Datenstrukturen*, Teubner

Programmieren in C (320056, 320166)

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 kompakte Einführung in die Programmiersprache C gemäß dem ANSI-Standard. Zum Verständnis sollten Kenntnisse einer anderen Programmiersprache vorhanden sein.

KERNIGHAN/RITCHIE: *Programmieren in C, zweite Ausgabe, ANSI C*, Hanser

LOWES/PAULIK: *Programmieren mit C – ANSI Standard*, B.G. Teubner

Computerunterstütztes Publizieren mit L^AT_EX (320109)

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. $\text{\LaTeX}$ steht auf IBM-kompatiblen PCs und auf vielen anderen Rechnersystemen zur Verfügung.

In dieser Veranstaltung werden die Einsatzmöglichkeiten von $\text{\LaTeX}$ im Publikationsprozeß vorgestellt. Es wird gezeigt, wie Texte mit Editoren wie KEDIT unter DOS für $\text{\LaTeX}$ erfaßt, mit $\text{\TeX}$ formatiert, zur Kontrolle am Bildschirm angezeigt und auf unterschiedlichen Druckern ausgegeben werden können.

Die Hörer sollten Grundkenntnisse im Umgang mit PCs besitzen.

KOPKA: $\text{\LaTeX}$ - Eine Einführung, Addison Wesley

LAMPORT: $\text{\LaTeX}$ - A Document Preparation System, Addison Wesley

PARTL/SCHLEGEL/HYNA: $\text{\LaTeX}$ Kurzbeschreibung

WONNEBERGER: Kompaktführer $\text{\LaTeX}$, Addison Wesley

Computerunterstütztes Publizieren mit $\text{\LaTeX}$ für Fortgeschrittene (320037)

Diese Veranstaltung richtet sich an Autoren, die schon erste Erfahrungen mit $\text{\LaTeX}$ gemacht haben und die Erweiterungen und Veränderungen durchführen wollen.

Zusammen mit dem Makropaket $\text{\LaTeX}$ unterstützt das Satzsystem $\text{\TeX}$ 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 kann sich in einem separaten Arbeitsschritt um das Aussehen dieser Textelemente im gedruckten Dokument kümmern.

In $\text{\LaTeX}$ sind eine Reihe von Textelementen definiert, denen mehrere Layout-Strukturen zugeordnet werden können. Ein Dokument, dessen Textelemente mit $\text{\LaTeX}$ -Markierungen versehen sind, kann so in unterschiedlicher Gestalt formatiert und gedruckt werden.

Die in $\text{\LaTeX}$ 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.

ABRAHAMS: $\text{\TeX}$ for the Impatient, Addison Wesley

APPELT: $\text{\TeX}$ für Fortgeschrittene, Addison Wesley

KOPKA: $\text{\LaTeX}$ - Eine Einführung, Addison Wesley

KOPKA: $\text{\LaTeX}$ Erweiterungsmöglichkeiten, Addison Wesley

LAMPORT: $\text{\LaTeX}$: A Document Preparation System, Addison Wesley

SCHWARZ: Einführung in $\text{\TeX}$, Addison Wesley

Vektorisierung und Parallelisierung mit FORTRAN (320075)

Diese Veranstaltung wendet sich in erster Linie an diejenigen, die ihre FORTRAN-Programmierkenntnisse auffrischen oder vertiefen möchten. Im ersten Teil der Vorlesung sollen die FORTRAN-Sprachelemente kurz und systematisch zusammengefaßt werden. Im zweiten Teil soll speziell auf die Möglichkeiten der Effizienzverbesserung durch Vektorisierung und Parallelisierung eingegangen werden.

Textverarbeitung mit WordPerfect (320022)

Die Veranstaltung gliedert sich in folgende Teile:

- Grundfunktionen von WordPerfect 5.1
- Einfache Textbearbeitung
- Druckfunktionen, Texttransfer
- Erweiterte Textgestaltung

Im letzten Teil sollen behandelt werden: Fuß- und Endnotengestaltung, Organisation von Kopf- und Fußtexten, Tabulatoren, Tabellenbearbeitung, Formeldarstellung, parallele Bearbeitung von Dokumenten, Fenstertechnik, Systembefehle zur Dateiverwaltung, Sicherungsmaßnahmen, Makrodefinitionen, Serienbriefbearbeitung.

KAMP: Textverarbeitung mit WordPerfect, Version 5.1, Software-Information 22 (im Rechenzentrum erhältlich)

Einsatz relationaler Datenbanken mit SQL (320132)

Datenbanken gewinnen immer mehr an Bedeutung zur automatischen Gewinnung von Informationen. Insbesondere der Typ der relationalen Datenbank mit der standardisierten Abfragesprache SQL stellt vielfältige Anwendungsmöglichkeiten zur Verfügung, wie z.B. Speicherung großer Datenmengen mit gleichzeitigem Zugriff mehrerer Benutzer, Überprüfung von Integritätsbedingungen bei Veränderung der Daten und einfache Einbettung von Datenbankzugriffen in Anwendungsprogramme.

C.J. DATE: A Guide to the SQL Standard

C.J. DATE: An Introduction to Database Systems

Statistische Datenanalyse mit SPSS (320113)

Das statistische Programmsystem SPSS (Statistical Package for the Social Sciences) wird in einer am Rechenzentrum implementierten aktuellen Produktionsversion vorgestellt. Mit diesem System stehen bequem aufzurufende Programme zu den gebräuchlichen univariaten und multivariaten statistischen Verfahren sowie zur Datenaufbereitung 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.

STEINHAUSEN/ZÖRKENDÖRFER: *Statistische Datenanalyse mit dem Programmsystem SPSS^x und SPSS/PC+*, Oldenbourg (Hörscheine bei den Autoren erhältlich)

Standardanwendungen auf Mikrorechnern (320080)

In der Veranstaltung werden verbreitete Programmpakete für Mikrorechner unter MS-DOS, Windows und OS/2 vorgestellt. Dabei werden Textverarbeitung, Tabellenkalkulation, Datenbanken, Grafik etc. als Standardanwendungen den Schwerpunkt bilden. Die Veranstaltung wendet sich an Hörer ohne Vorkenntnisse, die sich über Leistungsumfang und Erscheinungsbild typischer Anwendungen informieren wollen. Eine Übersicht über die geplanten Themen wird zu Beginn der Vorlesung bekanntgegeben.

Textverarbeitung auf Mikrorechnern (320094)

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, Übungen zur Textverarbeitung mit dem Programm WordPerfect 5.1 auf Mikrorechnern des Universitätsrechenzentrums durchzuführen.

Datenvisualisierung mit PV~Wave (320128)

Datenvisualisierung bedeutet die Kombination grafischer und analytischer Werkzeuge zur Darstellung

komplexer Zusammenhänge. In dieser Vorlesung soll das Produkt PV~Wave vorgestellt werden, das dieses Konzept auf grafischen Unix-Workstations (verschiedener Hersteller) realisiert. Es werden folgende Themenkreise behandelt:

- PV~Wave als interaktive Programmiersprache
- Darstellung zweidimensionaler Daten
- Darstellung dreidimensionaler Daten
- Bildverarbeitung

Handbücher des Herstellers

Lokale Rechnernetze: Technologie und Dienste (320204)

Nach einer orientierenden Vermittlung von Basiswissen werden in dieser Veranstaltung einzelne Aspekte der Einrichtung und Nutzung von Technologien und Diensten in lokalen Rechnernetzen beleuchtet.

Es wird zunächst eingegangen auf technologische Aspekte lokaler Rechnernetze. Im Mittelpunkt steht dabei die im lokalen Rechnernetz der Universität verwendete Technik (Ethernet und FDDI) und deren Installation. Anschließend wird die Einrichtung und Nutzung von Arbeitsplatzrechnern (insb. MS-DOS und Unix) vorgestellt. Auch die Einrichtung verschiedenster Netzdienste auf Server-Rechnern unter Berücksichtigung des Nutzungsspektrums des lokalen Rechnernetzes der Universität wird in der Veranstaltung ausführlich behandelt.

Die Installation und Nutzung ausgewählter Funktionen wird in praktischen Übungen vertieft. Aus diesem Grund ist die Teilnehmerzahl dieser Veranstaltung beschränkt. Die Teilnehmer dieser Vorlesung sollten über Basiswissen in einem relevanten Betriebssystem (z. B. MS-DOS, OS/2, Unix) verfügen.

Statistische Datenanalyse mit dem Programmsystem SAS (320018)

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 in einer anwendungsfreundlichen interaktiven Programmierungsumgebung am Großrechner bzw. am PC 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.

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

Rechnernetz-Praktikum (320219)

Das Praktikum führt in die Installation, Konfiguration und Nutzung von Netzwerkkomponenten und Netzwerksoftware ein. Es ist gegliedert in eine Reihe von Experimenten, die jeweils von zwei bis drei Personen gemeinsam durchgeführt werden. U. a. werden folgende Themen behandelt:

- Einbau von Netzwerkkarten
- DOS-LAN-Requester
- OS/2 LAN-Server
- TCP/IP
- Zugang zu Dialogsystemen, File-Transfer
- Internet

Das Praktikum beginnt mit einer gemeinsamen Vorlesung, die die für die Experimente erforderlichen Grundlagen bereitstellt. Danach werden Gruppen gebildet.

Für das Praktikum ist eine Voranmeldung erforderlich.

Programmieren unter OS/2 (320190)

Die Lehrveranstaltung gliedert sich in drei Teile:

Im ersten Teil wird das neue PC-Betriebssystem OS/2 vorgestellt, wobei der Schwerpunkt nicht auf Interna, sondern auf der Benutzung liegen soll. Die grafische Benutzungsoberfläche Workplace-Shell und das objektorientierte Design seien hier beispielhaft genannt.

Der zweite Teil ist der Kommandosprache REXX gewidmet, die zu OS/2 dazugehört wie die grafische Benutzungsoberfläche. Durch REXX wird es möglich, mit wenig Aufwand Anwendungen auf eigene Bedürfnisse zuzuschneiden.

REXX findet als Interpreter-Sprache seine Grenzen bei komplexeren Anwendungen. Wegen der Ähnlichkeit der Sprachen wird im dritten Teil die Programmiersprache PL/I vorgestellt, für die es jetzt auch auf dem PC einen sehr guten Compiler gibt. Mit PL/I kann man sowohl übersichtliche als auch schnelle Programme schreiben – ohne sich in die Niederungen der Pointer-Programmierung wie in C begeben zu müssen.

FREMUTH: *OS/2 2.0, te-wi*

COWLISHAW: *Die Programmiersprache REXX*, Hanser

STURM: *Programmieren in PL/I*, Vieweg

Themen des Datenschutzes (320223)

Die Veranstaltung möchte die Grundzüge des novelierten Datenschutzrechts anhand der Datenschutzgesetze des Bundes und des Landes Nordrhein-Westfalen erarbeiten. Dabei sollen auch Hinweise auf die praktische Durchführung der gesetzlichen Bestimmungen gegeben und Einzelfälle im Gesprächskreis diskutiert werden. Zu speziellen Aspekten des Datenschutzes, z. B. studentischer Daten, haben Gastreferenten Vorträge zugesagt.

WOHLGEMUTH, HANS: *Datenschutzrecht, eine Einführung mit praktischen Fällen*, Luchterhand

Kolloquium über Themen der Informatik (320238)

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

Stichwörter **inforum** Jahrgang 16

Ansprechpartner			
16,2-33	–		Ansprechpartner im URZ
Ausbildung			
16,2-29	–		Lehr- und Ausbildungsangebot des URZ
Benutzerverwaltung			
16,1-7	St. Ost		Benutzerverwaltung
Benutzungsordnung			
16,1-27	H. Pudlatz		Vorläufige Benutzungsordnung
Betriebsregelung			
16,3-11	–		Betriebsregelung für den Zugang zu Ressourcen des DV-Gesamtsystems der WWU
BMDP			
16,3-20	B. Süselbeck, S. Zörkendörfer		Neue Versionen der Statistik-Programmpakete
CIP			
16,4-4	R. Perske		Neues vom System
Compiler			
16,3-16	R. Perske		Compilerprozeduren im MVS/ESA
Datenbank			
16,3-22	J. Hölters, L. Göllmann		DB2 – ein neuer Dienst für Datenbankanwendungen
DaWIN			
16,4-5	U. Hegge		Überblick DaWIN im November 1992
DB2			
16,3-22	J. Hölters, L. Göllmann		DB2 – ein neuer Dienst für Datenbankanwendungen
Dienste			
16,2-11	–		Dienste zur Unterstützung der DV-Nutzung
16,2-33	–		Ansprechpartner im URZ
16,3-23	G. Richter		Neue Dienste und Funktionen im Rechnernetz
DV-Administratoren			
16,2-20	–		Organisation der DV-Nutzung in der WWU
16,4-3	H. Pudlatz		DV-Administratoren und DV-Moderatoren
DV-Gesamtausstattung			
16,1-3	W. Held, W. Bosse		Die neue DV-Ausstattung
16,2-4	–		Die neue DV-Gesamtausstattung im Überblick
16,2-11	–		Dienste zur Unterstützung der DV-Nutzung

16,2-14	-	Neu beschaffte DV-Komponenten
16,3-10	-	Neue Datenverarbeitung für die Universität Münster -
		DV-Gesamtausstattung in Betrieb
16,3-11	-	Betriebsregelung für den Zugang zu Ressourcen des DV-Gesamtsystems der WWU
DV-Moderatoren		
16,2-20	-	Organisation der DV-Nutzung in der WWU
16,4-3	H. Pudlatz	DV-Administratoren und DV-Moderatoren
E-Mail		
16,3-23	G. Richter	Neue Dienste und Funktionen im Rechnernetz
Festvortrag		
16,3-4	G. Seegmüller	Struktur und Zukunft der wissenschaftlichen Informationsverarbeitung
Fortran		
16,3-28	B. Neukäter	Programme mit großen Speichieranforderungen
16,3-34	St. Ost, W. Schaub	Parallelisierung - erste Erfahrungen
16,4-12	B. Neukäter	Vektorisierung und Parallelisierung in Fortran
FTP		
16,3-16	R. Perske	Der FTP-Server im MVS/ESA
16,4-4	R. Perske	Neues vom System
Geschäftsverteilungsplan		
16,2-32	-	Geschäftsverteilungsplan des URZ
Glosse		
16,1-27	H. Pudlatz	Vorläufige Benutzungsordnung
Grafik		
16,3-35	H. Pudlatz	Neue Plotter im Universitätsrechenzentrum
16,3-36	A. Jankrift	Ausgabe von PLOT-Files auf HP-LaserJet und HP-DeskJet
16,4-9	H. Pudlatz	Eine Grafik-Pipeline
Grafische Oberfläche		
16,1-17	R. Mersch	Das X-Window-System
16,4-10	E. Sturm	Ergreifen und Abwerfen in OS/2
HCL-eXceed Plus		
16,4-7	S. Schindler	PCs als X-Terminal mit HCL-eXceed Plus
Informationsverarbeitung		
16,3-4	G. Seegmüller	Struktur und Zukunft der wissenschaftlichen Informationsverarbeitung
Karten		
16,1-9	S. Zörkendörfer	Thematische Karten unter SAS
Konferenzen, elektronische		
16,3-13	W. Held	Einrichtung elektronischer Konferenzen

Landesgrößtrechner

16,3-12	W. Held	Landesgrößtrechner
---------	---------	--------------------

L^AT_EX

16,1-12	W. Kaspar	Ein neuer Schriftauswahlmechanismus für L ^A T _E X
16,3-26	W. Kaspar	Ein neuer Schriftauswahlmechanismus für L ^A T _E X – Teil 2

Migration

16,3-15	–	Broschüre „Informationen zur Migration“
---------	---	---

MVS/ESA

16,1-5	St. Ost	Neues vom System
16,1-7	St. Ost	Benutzerverwaltung
16,3-16	R. Perske	Compilerprozeduren im MVS/ESA
16,3-16	R. Perske	Der FTP-Server im MVS/ESA
16,3-28	B. Neukäter	Programme mit großen Speicheranforderungen
16,3-34	R. Mersch	Dateien im MVS/ESA
16,4-5	St. Ost	Regeln zur MVS-Paßwortänderung
16,4-12	B. Neukäter	Vektorisierung und Parallelisierung in Fortran

Organisation

16,2-20	–	Organisation der DV-Nutzung in der WWU
16,4-3	H. Pudlatz	DV-Administratoren und DV-Moderatoren

OS/2

16,1-5	St. Ost	Neues vom System
16,1-7	St. Ost	Benutzerverwaltung
16,4-10	E. Sturm	Ergreifen und Abwerfen in OS/2

Paßwort

16,4-4	R. Perske	Neues vom System
16,4-5	St. Ost	Regeln zur MVS-Paßwortänderung

Parallelisierung

16,3-34	St. Ost, W. Schaub	Parallelisierung – erste Erfahrungen
16,4-12	B. Neukäter	Vektorisierung und Parallelisierung in Fortran

Partnerschaftsvertrag

16,3-12	–	Partnerschaftsvertrag mit der Universität Riga
---------	---	--

Plotter

16,3-35	H. Pudlatz	Neue Plotter im Universitätsrechenzentrum
16,3-36	A. Jankrift	Ausgabe von PLOT-Files auf HP-LaserJet und HP-DeskJet
16,4-9	H. Pudlatz	Eine Grafik-Pipeline

Projekte

16,1-7	St. Ost	Benutzerverwaltung
--------	---------	--------------------

Public Domain

16,3-13	H. Pudlatz	Software im Rechnernetz
---------	------------	-------------------------

PV~WAVE

16,3-13	H. Pudlatz	Software im Rechnernetz
---------	------------	-------------------------

Quiz

 16,1-31 E. Sturm **Inform**-Quiz

Rechnerbeschaffung

16,1-3	W. Held, W. Bosse	Die neue DV-Ausstattung
16,1-5	St. Ost	Neues vom System
16,2-4	—	Die neue DV-Gesamtausstattung im Überblick

Rechnernetz

16,2-8	—	Rechnernetze
16,3-23	G. Richter	Neue Dienste und Funktionen im Rechnernetz

S-PLUS

16,3-20	B. Süselbeck, S. Zörkendörfer	Neue Versionen der Statistik-Programmpakete
---------	----------------------------------	---

SAS

16,1-9	S. Zörkendörfer	Thematische Karten unter SAS
16,3-20	B. Süselbeck, S. Zörkendörfer	Neue Versionen der Statistik-Programmpakete

Scanner

16,4-9	H. Pudlatz	Eine Grafik-Pipeline
--------	------------	----------------------

Schadensvermeidung

16,1-10	—	Tips zur Schadensvermeidung bei PCs
---------	---	-------------------------------------

SMS

16,3-34	R. Mersch	Dateien im MVS/ESA
---------	-----------	--------------------

Software

16,3-13	H. Pudlatz	Software im Rechnernetz
---------	------------	-------------------------

Speicher, dynamischer

16,3-28	B. Neukäter	Programme mit großen Speicheranforderungen
---------	-------------	--

Speicherverwaltung

16,3-34	R. Mersch	Dateien im MVS/ESA
---------	-----------	--------------------

SPSS

16,1-9	S. Zörkendörfer	Neues vom SPSS
16,3-20	B. Süselbeck, S. Zörkendörfer	Neue Versionen der Statistik-Programmpakete

Statgraphics

16,3-20	S. Zörkendörfer, B. Süselbeck	Neue Versionen der Statistik-Programmpakete
---------	----------------------------------	---

Statistik

16,1-9	S. Zörkendörfer	Neues vom SPSS
16,3-20	S. Zörkendörfer, B. Süselbeck	Neue Versionen der Statistik-Programmpakete

TCP/IP		
16,3-23	G. Richter	Neue Dienste und Funktionen im Rechnernetz
T_EX		
16,1-12	W. Kaspar	Ein neuer Schriftauswahlmechanismus für L ^A T _E X
16,3-26	W. Kaspar	Ein neuer Schriftauswahlmechanismus für L ^A T _E X – Teil 2
Unix		
16,1-5	St. Ost	Neues vom System
16,1-6	R. Mersch	Ende des UTS-Betriebs
16,1-7	St. Ost	Benutzerverwaltung
16,1-17	R. Mersch	Das X-Window-System
Userid		
16,1-7	St. Ost	Benutzerverwaltung
UTS		
16,1-6	R. Mersch	Ende des UTS-Betriebs
VAMP		
16,3-13	H. Pudlatz	Software im Rechnernetz
Vektorisierung		
16,4-12	B. Neukäter	Vektorisierung und Parallelisierung in Fortran
Vektorrechner		
16,3-12	W. Held	Landesgrößtrechner
Virenschutz		
16,1-10	–	Tips zur Schadensvermeidung bei PCs
VMS		
16,1-7	St. Ost	Benutzerverwaltung
X-Window		
16,1-17	R. Mersch	Das X-Window-System
16,3-23	G. Richter	Neue Dienste und Funktionen im Rechnernetz
16,4-7	S. Schindler	PCs als X-Terminal mit HCL-eXceed Plus