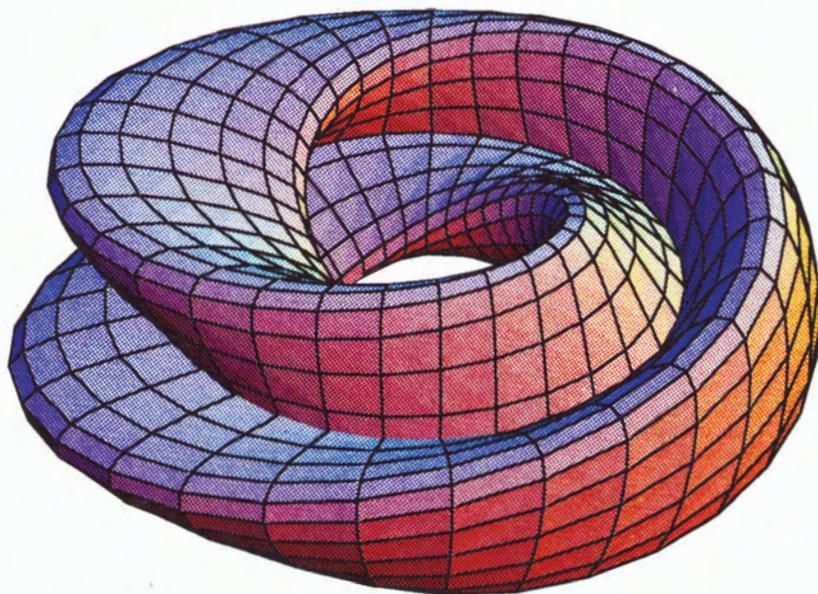


PIPELINE

INFORMATIONEN DES EDV-ZENTRUMS DER TECHNISCHEN UNIVERSITÄT WIEN



Farb-PostScript-Drucker

Zentrales UNIX-Unterstützungssystem

Fachbereichsrechner

Physik und Maschinenbau

Inhalt

Die neuen Rechenanlagen des EDV-Zentrums	3
Der neue Farb-PostScript-Drucker am EDV-Zentrum	4
Technologiesprung im TUNET, Glasfaser und FDDI	6
CONVEX C3220, das neue zentrale UNIX-Unterstützungssystem des EDV-Zentrums	7
Software-Oberflächen der CONVEX-Systeme	8
Abbau der Anlage CDC CYBER 860	10
Störungsmeldungen TUNET	11
Personelle Veränderungen	11
Aktuelle Kurzmitteilungen	
Beratung durch die Abteilung Institutsunterstützung	12
Informationen aus der Bibliothek der Technischen Universität Wien am EMAIL-Server	14
TUWIS-Statistiken	14
NAG und IMSL an der CONVEX	16
NAG am Fachbereichsrechner Maschinenbau	16
Lösung von sehr großen linearen Gleichungssystemen auf der VP50-EX	16
X-Terminal-Software für PC's: HCL-Exceed	17
Neues vom TUNET	18
SEDIT Editor 4.2	18
PC - Windows 3.1	19
A/UX - UNIX für Macintosh	20
AutoCAD	20
Das Computer-Algebra-System Mathematica Version 2.0 - Überblick	22
Der IBM RS/6000 Workstation Cluster am EDV-Zentrum	24
Einbindung der VP50-EX in die verteilte Verarbeitung	26
Wieviel Aufwand erfordert die Nutzbarmachung der Vektorhardware?	27
Simulation	29
Software Engineering der 90er Jahre objektorientiert?	32
Vortragsankündigungen	37
Campus-Lizenzen für die TU Wien	38
Schulungsprogramm	40
Abteilungen und Mitarbeiter (Telefonliste, E-Mail-Adressen)	43

Nächster Redaktionsschluß: 30. März 1992

Editorial

Liebe Leser!

Der neue Farb-PostScript-Drucker hat uns zu einer farbigen Titelseite animiert. Das Interesse von Firmen, in unserer Zeitschrift zu inserieren, ermöglicht die Finanzierung des Farbdruckes.

Diese PIPELINE ist wieder sehr umfangreich geworden. Sie finden gleich zu Beginn alle Informationen über neue Rechner und Geräte, dann aktuelle Kurzmitteilungen über neue Software und Netzdienste. Anschließend folgen allgemeine Beiträge, Berichte und Grundsatzartikel. Den Abschluß bilden Standardinformationen wie Campus-Lizenzen, Kurse, Telefon- und E-Mail-Nummern.

Ein Aufruf an alle PIPELINE-Bezieher an der Technischen Universität Wien: Bitten geben Sie die PIPELINE auch an alle Kollegen an Ihrem Institut weiter, die mit EDV zu tun haben. Wir haben die Erfahrung gemacht, daß viele Kollegen an der TU zu wenig oder gar nichts über das EDV-Zentrum der TU Wien und seine Service-Leistungen wissen, und werden in nächster Zukunft noch weitere Informationsprogramme starten. Sie können telefonisch im Sekretariat (Klappe 5481) weitere PIPELINE-Abonnements anfordern.

Irmgard Husinsky

Zum Titelbild (erstellt von Jadwiga Rogl): Die Klein'sche Flasche, erster Preis in der "Mathematica Graphics Competition 1990" (Ari Lehtonen, University of Jyväskylä, Finnland).

Die Klein'sche Flasche stellt eine geschlossene Fläche dar und kann nicht orientiert werden (hat nur eine Seite). Eine dreidimensionale Darstellung kann nicht ohne Selbstüberschneidung gezeigt werden: auf dem Bild überschneidet sie sich selbst entlang eines Kreises. Siehe auch: "Treatise on Analysis" von Jean Dieudonne (Vol.III, § XVI.25).

Der Code ist sehr einfach:

```
a=2;
f=(a+Cos[u/2]Sin[t]-Sin[u/2]Sin[2t])Cos[u];
g=(a+Cos[u/2]Sin[t]-Sin[u/2]Sin[2t])Sin[u];
h=Sin[u/2]Sin[t]+Cos[u/2]Sin[2t];
ParametricPlot3D[{f,g,h},{t,0,2Pi,Pi/16},{u,0,2Pi,Pi/16}];
```

Offenlegung gemäß § 25 Mediengesetz:

Herausgeber, Inhaber: EDV-Zentrum der Technischen Universität Wien
Grundlegende Richtung: Mitteilungen des EDV-Zentrums der Technischen Universität Wien
Redaktion: Irmgard Husinsky
Druck: kopitu
Adresse: Technische Universität Wien, Wiedner Hauptstraße 8-10, A-1040 Wien
Tel.: (0222) 58801-5481
Fax: (0222) 587 42 11
E-Mail: husinsky@edvz.tuwien.ac.at

Die neuen Rechenanlagen des EDV-Zentrums

Installation und Inbetriebnahme

CONVEX C3220

- das zentrale UNIX-Unterstützungssystem

An der Ausschreibung für ein zentrales UNIX-Unterstützungssystem, die im Sommer 1991 durchgeführt wurde, beteiligten sich 12 Firmen. Nach der Bewertung wurde von der Vergabevorschlagskommission die Firma CONTROL DATA als Anbieter einer CONVEX C3220 als Bestbieter ermittelt.

Die Lieferung des Rechnersystems erfolgte am 5. November 1991, die Installation der Hard- und Systemsoftwarekomponenten sowie die Netzeinbindung konnten ohne Probleme durchgeführt werden.

Das System wurde am 22. November 1991 abgenommen, der Produktionsbetrieb wurde am 9. Dezember 1991 aufgenommen. Eine detaillierte Beschreibung der Hard- und Softwarekomponenten entnehmen Sie bitte dem Artikel auf Seite 7.

Die Installation der Anwendersoftware wird gerade durchgeführt; ein Arbeitsschwerpunkt ist die Implementierung von GKSGRAL (von der Firma MSB) mit SimplePlot, aufbauend auf die Schnittstelle X Window, sowie das Visualisierungspaket AVS.

CONVEX C220

- der Fachbereichsrechner "Physik"

Für dieses Rechnersystem wurde der Firma CONTROL DATA der Zuschlag für eine CONVEX C220 erteilt. Das System wurde Anfang Jänner 1992 geliefert, installiert und am 15. Jänner 1992 abgenommen.

Die CONVEX C220 ist ein Doppelprozessorsystem mit insgesamt 128 MB Hauptspeicher, 8 GB Plattenspeicher und einem 4mm DAT-Tape. Das Betriebssystem - Convex/OS - ist ein UNIX-System mit BSD 4.3 Erweiterungen. Ein Batch-System (CX-Batch basierend auf NQS) und ein Tape-System stehen zur Verfügung.

Ein vektorisierender und parallelisierender FORTRAN- sowie ein C- und Pascal-Compiler sind installiert.

Das System ist in das TUNET eingebunden, der Hostname ist ECXPH; eine Einbindung in das DECNET der TU Wien ist geplant.

An Anwendersoftware werden die Produkte NAG, IMSL, Mathematica, ACSL und GKSGRAL/SimplePlot zur Verfügung stehen.

Der Rechner steht vor allem Benutzern aus der Fachgruppe Physik zur Verfügung, Sonderregelungen für andere Benutzer sind möglich. Ansuchen um Betriebsmittel und Informationsmaterial erhalten Sie im Sekretariat des EDV-Zentrums.

IBM RS/6000-950 und RS/6000-550

- das Rechnersystem für die Fakultäten Maschinenbau und Bauingenieurwesen

Für die Fakultäten Maschinenbau und Bauingenieurwesen wurde der Firma APH der Zuschlag für ein Rechnersystem bestehend aus je einem RISC-Rechner **IBM RS/6000-950 und RS/6000-550** erteilt. Die Installation erfolgte Ende Oktober 1991, die Abnahme des Systems wurde am 27. November 1991 durchgeführt.

Der Rechner RS/6000-950 verfügt über 128 MB Hauptspeicher, 5 GB Plattenspeicher, ein 1/4-Zoll Tape (150 MB) und eine 8mm Bandstation (2,3 GB). Ein CD-ROM Laufwerk und ein Ethernet-Controller sind vorhanden.

Der Rechner RS/6000-550 verfügt über 64 MB Hauptspeicher, 860 MB Plattenspeicher und einen Ethernet-Controller.

Die Systeme sind durch einen Token Ring untereinander verbunden, ein Optischer Link ist geplant.

Beide Rechner laufen unter dem Betriebssystem AIX Version 3.1.5. Für den Batch-Betrieb steht Sterling NQS sowie zum Lastausgleich NQS-EXEC zur Verfügung. Beide Systeme kommunizieren untereinander mit TCP/IP und NFS. Ein FORTRAN- sowie ein C-Compiler stehen zur Verfügung.

Unter NQS stehen folgende (Pipe-)Queues zur Verfügung:

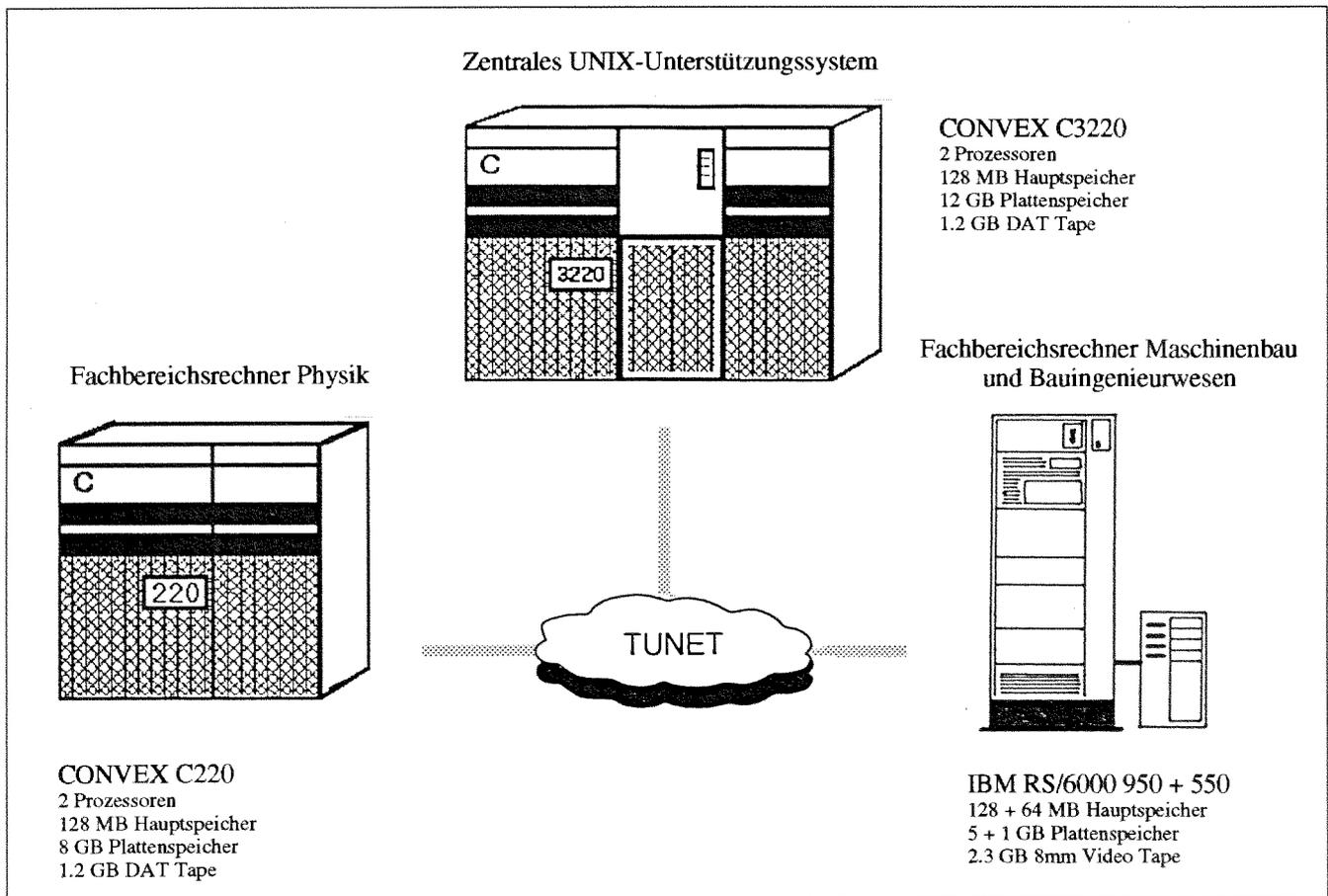
Queue	max. CP-Zeit	Host
short	2000 sec	"Lastausgleich"
long	20000 sec	"Lastausgleich"
short950	2000 sec	rsmb
long950	20000 sec	rsmb
short550	2000 sec	rsmb550
long550	20000 sec	rsmb550

An Anwendersoftware werden die Produkte NAG, IMSL, ACSL und ANSYS sowie eine Reihe von Finite Elemente Paketen installiert.

Das Rechnersystem ist in das TUNET eingebunden (TCP/IP), der Zugang erfolgt ausschließlich über das 950-System; der Hostname ist **RSMB**.

Der Rechner steht vor allem Benutzern aus den Fakultäten Maschinenbau und Bauingenieurwesen zur Verfügung. Sonderregelungen für andere Benutzer sind möglich. Ansuchen um Betriebsmittel und Information erhalten Sie im Sekretariat des EDV-Zentrums.

Peter Berger



Der neue Farb-PostScript-Drucker am EDV-Zentrum

Um eine hochqualitative Farb-Ausgabe zu ermöglichen, wurde von Herrn Dr. Wehrberger eine Marktstudie über Farb-PostScript-Drucker durchgeführt. Nach Vergleichen verschiedener Drucker wurde der Farb-PostScript-Drucker Phaser III von Tektronix ausgewählt, der Anfang Dezember 1991 geliefert wurde.

Technische Beschreibung des Gerätes:

Der Tektronix Phaser III Drucker arbeitet mit Wachsstiften (magenta, gelb, cyan, schwarz), die geschmolzen und mit einem Düsenkopf auf das Papier gespritzt werden. Dadurch ist die Verwendung von normalem Büropapier, Karton und Folien möglich. Der Drucker verfügt über ein Papiermagazin für eine maximale Papiergröße A3.

Die Druckauflösung beträgt 300 dpi. Der Drucker unterstützt nur Adobe's PostScript Language, die zu druckenden Files müssen diesem Standard entsprechen.

Rechneranschluß:

Der Drucker wurde am EVAX-Cluster angeschlossen. Dies hatte mehrere Gründe:

- Die Möglichkeit des Remote-Printing ist sowohl für VAX/VMS-Rechner (über Distributed Queue System DQS) wie auch für UNIX-Rechner oder PCs mit FTP-Software (über Line Printer Dämon LPD) gegeben.
- Das Betriebssystem VAX/VMS kann Druckjobs mit verschiedenen Papier-Sorten (A4, A3, Overhead-Folien) verarbeiten.
- Die Rückmeldung, wieviele Seiten ausgedruckt worden sind, ist mittels eines speziellen PostScript Drucker-Treibers unter dem Betriebssystem VAX/VMS möglich.

Drucker-Standort/-Ausgabe:

Der Drucker befindet sich im Ausgaberaum des EDV-Zentrums (Freihaus, 2. Stock, roter Bereich). Die fertigen Druckjobs können von dort abgeholt werden (Eingang Operatorausgabe).

Betreuung/Beratung:

Der Betrieb des Druckers wird vom Operating des EDV-Zentrums durchgeführt.

Die Beratung von Benutzern, wie diese Queues von Geräten an Instituten (PCs, Workstations) zu erreichen sind, erfolgt im Rahmen der Betreuung von Instituten durch die Abteilung Institutsunterstützung.

Die Beratung von Benutzern über die diversen Möglichkeiten für das Erstellen von Farb-PostScript Files wird im Rahmen der Betreuung der entsprechenden Applikations- und Graphiksoftware geleistet.

Wie kann ich ausdrucken?

Wie schon erwähnt, kann der Drucker Papierformate bis A3 verarbeiten. Derzeit gibt es drei Papierformate, die angeboten werden:

Form-Qualifier	Papierformat
CPS_A4	A4-Papier
CPS_A3	A3-Papier
CPS_OH	Overhead-Folie

Es muß bei jedem Druck Job das Papierformat angegeben werden. Jobs, die kein Papierformat gewählt haben, werden nicht ausgedruckt und gelöscht.

Ausdrucken mit LPR (Unix, PC):

Bei LPR wird das Papierformat mit Hilfe des -C Qualifiers angegeben:

```
lpr -Pcps01ps -Ccps_a4 file.ps
```

Bei PCs muß das Kommando um den Qualifier **-Sevax.tuwien.ac.at** ergänzt werden.

Ausdrucken unter VMS:

Unter dem Betriebssystem VAX/VMS wird das Papierformat mit Hilfe des /FORM-Qualifiers angegeben:

```
PRINT /QUEUE=CPS01PS /FORM=CPS_A4 file.ps
```

In beiden Beispielen wurde A4-Papier ausgewählt.

Was ist vom System Manager zu tun?

Auf einem Rechner, der LPR (Unix-Rechner, PCs) unterstützt:

Hier ist vom System-Manager eine Remote Queue einzurichten:

```
Local Queue Name: cps01ps  
Remote Queue Host: evax.tuwien.ac.at  
Remote Queue Name: cps01ps
```

Auf einem VAX/VMS Rechner über DQS:

Hier werden die FORM-Definitionen benötigt:

```
$ DEFINE/FORM CPS_A4 500 /STOCK=DEFAULT  
$ DEFINE/FORM CPS_A3 501 /STOCK=DEFAULT  
$ DEFINE/FORM CPS_OH 502 /STOCK=DEFAULT
```

Der System-Manager hat eine Remote-Queue mittels der vom DQS System zur Verfügung gestellten Prozedur `dqs$define` einzurichten.

```
$ @sys$manager:dqs$define -  
CPS01PS EVAX CPS01PS 0 -  
"/DEFAULT=(FORM=CPS_A4)/FORM=CPS_A4"
```

Am EVAX-Cluster:

Hier wurde eine Queue CPS01PS definiert. Druck-Jobs können daher jederzeit abgearbeitet werden.

Druck-Berechtigung:

Um am Farb-PostScript-Drucker ausdrucken zu können, ist eine Druck-Berechtigung erforderlich.

Unix-Rechner, sowie VMS-Rechner können mittels des Formulars "Druckerbenutzung / Ansuchen um Betriebsmittel am EDV-Zentrum der TU Wien", welches im Sekretariat des EDV-Zentrums erhältlich ist, als Remote-Print Host berechtigt werden. Bitte beachten Sie, daß jeder User des berechtigten Remote-Print Hosts drucken kann. Eine Einschränkung der Berechtigung auf einzelne User ist nicht möglich.

Am EVAX-Cluster haben Sie Berechtigung, wenn Sie den Service "LASER" beantragt haben.

Sie können dies kontrollieren, indem Sie den Befehl `SHOW PROCESS/PRIVILEGE` verwenden:

Unter dem Titel "Process Rights" sollte sich auch der Service "LASER" befinden.

```
Process rights:  
INTERACTIVE  
REMOTE  
LASER
```

Kosten

Für die Verwendung des Farbdruckers wird ein Kostenersatz eingehoben.

Die derzeitigen Preise sind:

```
$ 12.-    A4 (80 g Papier)  
$ 20.-    A3 (80 g Papier)  
$ 20.-    A4 Overhead-Folie
```

Peter Hoffmann

Technologiesprung im TUNET Glasfaser und FDDI

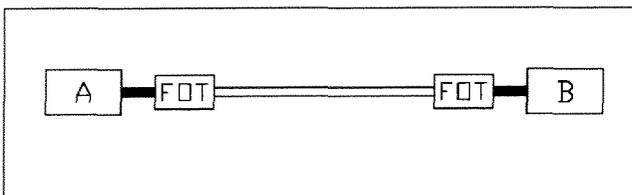
Im Jahre 1982 entstand die Idee, an der TU Wien ein Lokales Netzwerk zu errichten. Als es dann nach der erforderlichen Marktstudie und Bedarfsstudie und ersten Tests im Jahre 1986 zur Installation des ersten Backbonesegments des TUNET kam, standen für die Verbindung der einzelnen Gebäude der TU Wien von der Post verlegte Koaxialkabel zur Verfügung. Diese Koaxialkabel entsprachen jedoch nicht den Erfordernissen zur Verwendung als Ethernet (sie haben einen Wellenwiderstand von 75 Ω). Mit Hilfe des Instituts für Nachrichtentechnik konnten jedoch Standardkomponenten so umgebaut werden, daß die vorhandenen Koaxialkabel verwendet werden konnten. Dadurch war ein provisorischer, den Anfängen des TUNET genügender, Betriebszustand hergestellt.

Im Jahre 1990 wurde vom EDV-Zentrum an die Post der Auftrag erteilt, Glasfaserkabel zwischen den Gebäudekomplexen zu verlegen. Es wurden insgesamt 3,5 km Glasfaserkabel verlegt. Es wurden dabei die Gebäude Karlsplatz-Hauptgebäude, Karlsgasse 11 und 13, Argentinierstraße 8, Gußhausstraße 27-29, Gußhausstraße 28 und 30, Wiedner Hauptstraße 8-10, Wiedner Hauptstraße 7 und Getreidemarkt 9 erfaßt.

Damit war ein bedeutender Entwicklungsschritt im TUNET möglich. Die Backboneverbindungen zwischen den Gebäuden der TU Wien konnten auf Glasfaser umgestellt werden. Der provisorische Betriebszustand, in dem die Verbindungen mit selbstentwickelten Komponenten hergestellt waren, die keinem Standard entsprachen und daher ein Problem bei der Wartung darstellten, wurde durch eine zeitgemäße Lösung ersetzt.

Zur Zeit werden zwei Arten der Verbindung über Glasfaserkabel im TUNET verwendet.

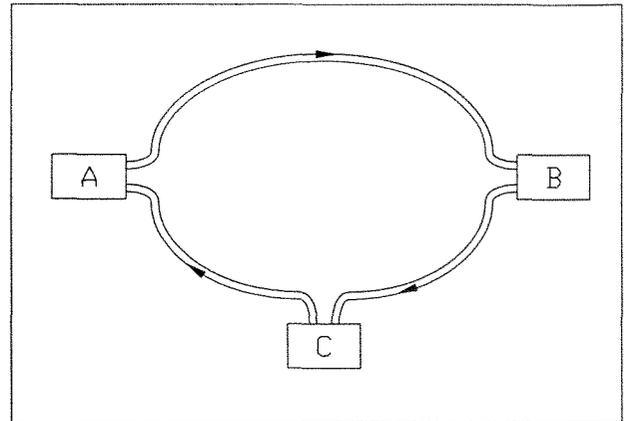
1. Mittels zweier Fiber Optik Transceiver (FOT) können zwei Netzwerkkomponenten mit Transceiveranschlüssen verbunden werden.



Verbindung via Fiber Optik Transceiver

Diese Art der Verbindung wird bei der Anbindung von Karlsgasse 11 und 13, Argentinierstraße 8, Gußhausstraße 28 und 30 sowie des Gebäudekomplexes Getreidemarkt 9 verwendet. Die Geschwindigkeit dieser Verbindungen beträgt 10 MBit/s - so wie auf allen anderen Ethernet-Strecken.

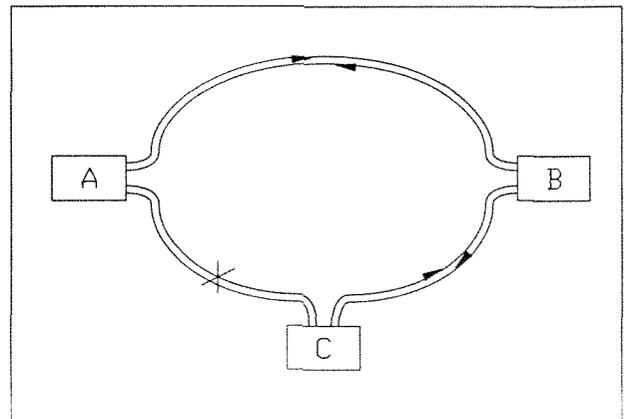
2. Einige im TUNET verwendeten Router enthalten ein Fiber-Distributed-Data-Interface (FDDI).



FDDI-Doppelring

Auf diese Art sind die Router im Freihaus, in der Gußhausstraße 27-29 und am Karlsplatz miteinander verbunden.

FDDI ist eine relativ junge Technologie in der Netzwerktechnik. Sie basiert auf dem "Token Ring" Prinzip. Jedoch wird nicht nur ein einfacher Ring verwendet, sondern ein Doppelring. Damit ist es möglich, eine größere Ausfallsicherheit zu erreichen. Im "primary ring" rotiert das Token. Wenn nun eine Komponente oder eine Strecke zwischen zwei im Ring befindlichen Komponenten ausfällt, erkennen die benachbarten Komponenten diesen geänderten Betriebszustand. Es wird dann der betroffene Bereich außer Betrieb genommen und das Token durchläuft dann die intakten Bereiche des "primary ring" und des "secondary ring".



FDDI-Ring im Fehlerfall eines Kabelbruches

Ein FDDI-Ring arbeitet mit 100 MBit/s. Seit kurzem gibt es auch FDDI-Interfaces für verschiedene Mainframes und Workstations der oberen Leistungsklasse. Mit solchen FDDI-Interfaces ausgestattete Geräte können in den FDDI-Ring integriert werden. Leider ist der Preis dieser Interfaces derzeit noch relativ hoch (öS 100.000,- bis 300.000,-). Daher ist der Einsatz vorerst sicherlich auf Spezialrechner (z.B. Supercomputer, Fachbereichsrechner) beschränkt.

Manfred Siegl

CONVEX C3220

Das neue zentrale UNIX-Unterstützungssystem des EDV-Zentrums

Als zentrales UNIX-Unterstützungssystem wurde Mitte November 1991 ein Rechnersystem CONVEX C3220 von der Firma CONTROL DATA installiert. Die allgemeine Betriebsaufnahme erfolgte am 9. 12. 1991.

Hardware - Konfiguration

Das System CONVEX C3220 ist ein 2-Prozessorsystem mit insgesamt 128 MB Hauptspeicher. Jede CPU verfügt über ein integriertes Skalar- und Vektorsystem, wobei jedes "Processing Subsystem" getrennte "Functional Units" (z.B. für Addition/Subtraktion und dergleichen) enthält. Die Systeme können unabhängig und parallel arbeiten. Jede CPU kann über einen Crossbar-Switch den Hauptspeicher mit einer Übertragungsrate von 200 MB/s ansprechen, d.h. die Gesamtübertragungsrate beträgt bei dieser Konfiguration 400 MB/s zwischen den CPUs und dem Hauptspeicher.

Das Ein/Ausgabesystem besteht aus 2 I/O-Prozessoren, an die 4 unabhängige VME-Bus Control Units angeschlossen sind.

12 GB Plattenspeicher (13 Einheiten à 940 MB), eine 1/2-Zoll Magnetbandeinheit (1600/6250 bpi), ein 1.2 GB 4mm DAT-Tape und ein Ethernet-Controller sind an diesem I/O System installiert.

Ein eigenständiger Serviceprozessor für die Systemüberwachung und Fehlerdiagnose ist vorhanden.

Betriebssystem und systemnahe Software

Convex/OS ist ein UNIX-System mit BSD 4.3 Erweiterungen. Neben dem Standard-UNIX-Kernel stehen ein Batch-System (CX-Batch) und ein Tape-System, in dem auch ANSI-Label verarbeitet werden können, zur Verfügung. Default-Shell ist die C-Shell csh. Für die Batchverarbeitung können mit dem Kommando `qsub` Jobs in drei Queues übergeben werden:

Queue	max. CP-Zeit	max. Anzahl von Jobs
short	2000 sec	2
medium	10000 sec	2
long	20000 sec	2

Bänder werden im Batch und interaktiv mit `tpmount` angefordert, welches einen Link zur entsprechenden Device herstellt. CX-Batch basiert auf dem Network Queuing System (NQS) von Sterling Inc.

An Kommunikationssoftware ist TCP/IP mit den üblichen Anwenderprotokollen (TELNET, FTP, r-Utilities) implementiert. Darüberhinaus stehen NFS und X Window Version 11 Release 4 sowie Motif V 1.1.2 zur Verfügung. Ein vektorisierender und parallelisierender FORTRAN-Compiler `fc` sowie ein C-Compiler `cc` sind vorhanden. Für das interaktive Debuggen kann auf X-Stations der windoworientierte CXdb (Convex Visual Source Debugger) verwendet werden (`fc-Option-cxdb`). Eine Bibliothek mit speziell für CONVEX-Rechner optimierten mathematischen Routinen (VECLIB) ist vorhanden.

Anwendersoftware

Die Applikationspakete NAG (NAG FORTRAN Library, Mark 14) und IMSL sind installiert (siehe Artikel "NAG und IMSL an der CONVEX", Seite 16), BMDP und $\text{T}_{\text{E}}\text{X}$ werden demnächst zur Verfügung stehen.

An Graphikpaketen stehen derzeit im Probebetrieb das Visualisierungssystem AVS von Stardent Computer Inc. und das graphische Kern-System GKSGRAL von GTS-GRAL (installiert von der Firma MSB) zur Verfügung.

Netzzugang

Das Rechnersystem ist über TCP/IP in das TUNET eingebunden. Der Hostname ist `ECX`. Es stehen u.a. die Protokolle TELNET und FTP zur Verfügung.

Bei der Anwahl der CONVEX C3220 von Netzen außerhalb der Domain der TU Wien muß der vollständig qualifizierte Hostname

`ECX.TUWIEN.AC.AT`

angegeben werden.

Organisatorische Hinweise

Das Rechnersystem steht vor allem jenen Benutzern der TU Wien zur Verfügung, die noch über keinen Fachbereichsrechner verfügen. Eine Nutzung des Systems durch andere Universitäten, wissenschaftlich-akademische Institutionen und die Österreichische Akademie der Wissenschaften ist möglich.

User-Nummern

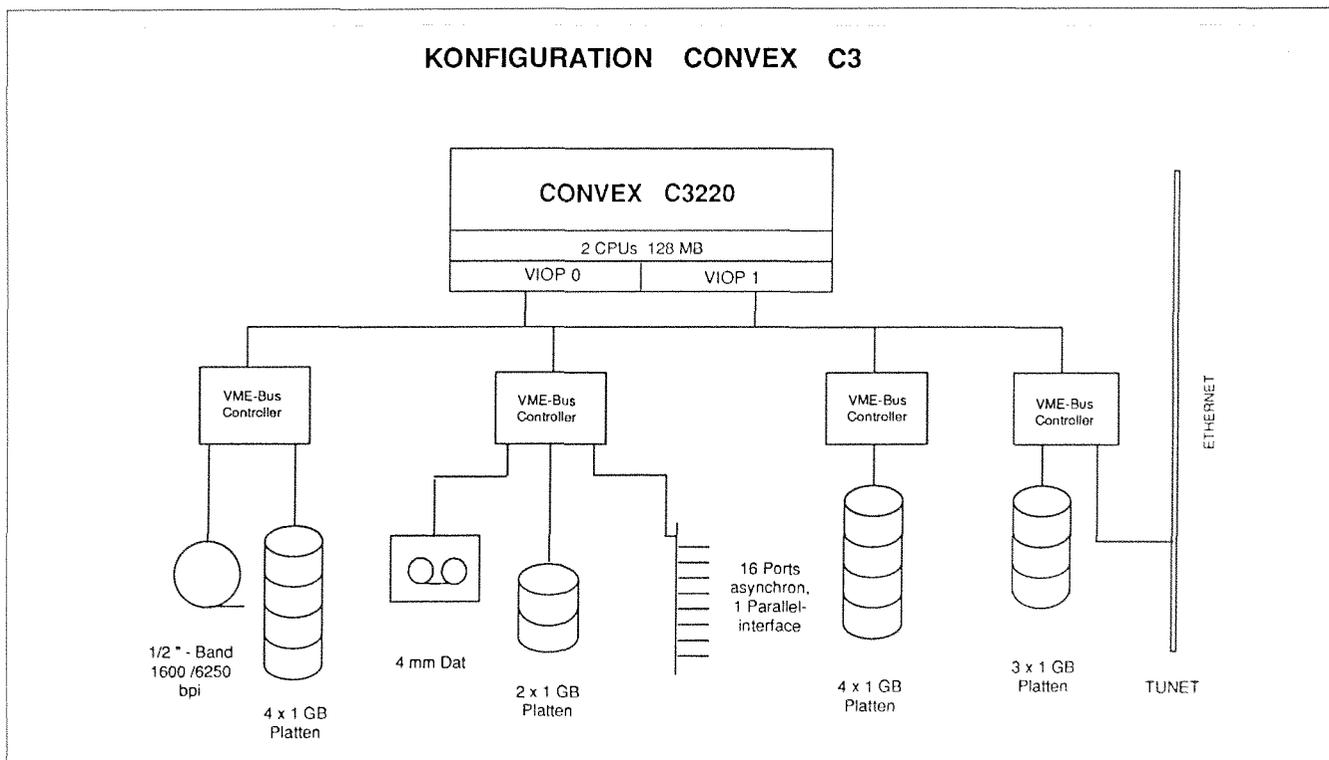
Ein "Ansuchen um Betriebsmittel am EDV-Zentrum der TU Wien" ist im Sekretariat des EDV-Zentrums erhältlich oder wird auf Wunsch gerne zugesandt. Diesem Ansuchen liegen Informationen über das System, CX-Batch, das Tape-System und eine FORTRAN-Kurzbeschreibung bei.

An wen wende ich mich ?

Für Systembetreuung:	J. Sadvsky	Kl. 5819
	H. Mastal	Kl. 5816
Für Betrieb:	P. Berger	Kl. 5815
	Operating	Kl. 5830
Für Accounting:	A. Roza	Kl. 5824
	D. Schornböck	Kl. 5820
Für Anwendersoftware:		
NAG, IMSL, BMDP:	W. Haider	Kl. 5603
$\text{T}_{\text{E}}\text{X}$:	J. Rogl	Kl. 3612
AVS, GKSGRAL, X:	H. Mastal	Kl. 5816
Parallelisierung u. Vektorisierung:		
	I. Macsek	Kl. 5811
	F. Blöser	Kl. 5811
	W. Weisz	Kl. 5818

Peter Berger

KONFIGURATION CONVEX C3



Software - Oberflächen der CONVEX-Systeme

Ein wichtiger Aspekt neuerer Softwareprodukte ist das Anbieten von benutzer- und benutzungsgerechten Oberflächen, die ein problemorientiertes Verwenden von Software ermöglichen, im Gegensatz zur bisherigen implementierungsorientierten Handhabung großer Softwarepakete. Die Erfüllung eines solchen Anspruches führt zwangsläufig zu einer Abstraktion von systemspezifischen Eigenheiten. Die nicht gerade geringen Unterschiede der einzelnen Unix-Systeme auf der Shell-Ebene könnten auf diese Weise gemildert werden.

Der bisher erfolgreichste Vorstoß zur Schaffung einheitlicher Oberflächen war X Window Version 11 zusammen mit dem OSF/Motif V 1.1.2 Motif Window Manager. Es gelang damit, eine einheitliche Oberfläche für Text und Graphik zu schaffen, wobei mindestens ein Client, der die auszugebenden Daten produziert, und ein Server, der das Ausgabemedium zur Verfügung stellt, zusammenarbeiten müssen. Im allgemeinen werden Server und Client auf verschiedenen Systemen verschiedener Hersteller laufen.

Convex besitzt eine lange Tradition im Zurverfügungstellen von Software-Oberflächen. Daher stellt das Produkt CXwindows V 2.1, welches hier auf der Convex C3220 installiert ist, die von Convex verifizierte und mit anderen Convex-Produkten gemeinsam getestete X Window Version 11 Release 4 von MIT zusammen mit OSF/Motif V 1.1.2 Window Manager und der Motif User Interface Development Software dar.

Ein Convex-Produkt entsprechend X Version 11 Release 5 ist in Vorbereitung. CXwindows enthält neben den X-Libraries und *xterm* sowie *mwm* eine Reihe von Test-Clients wie *xclock*.

Vorraussetzung für die Verwendung von X-Clients ist das richtige Setzen der Environment-Variablen DISPLAY. In der C-shell erfolgt das mit:

```
setenv DISPLAY displayname: 0
```

wobei *displayname* der entsprechende TCP/IP-Name des X-Servers ist.

Da mit X Window erst die Basis geschaffen wird, entsprechende Software-Oberflächen zu gestalten, wurden auch noch weitere Produkte, die echte X-Applikationen darstellen, an der Convex C3220 auf X aufbauend implementiert:

- **CXdb**, ein symbolischer Debugger, der windoworientiert auch optimierten Code bearbeiten kann
- **GKSGRAL**, das graphische Kernsystem mit dem X11-Driver
- **AVS**, Application Visualization System

CXdb

CXdb läuft entweder in der X-Umgebung oder unter Maryland Windows sowie im Batch-Mode. Alle für den Debugger analysierbaren Programme müssen mit der Compiler-Option *-cxd* übersetzt sein. Sie können bis zur höchsten skalaren Optimierungstufe *-O1* optimiert sein.

Nach Aufruf des Debuggers mit cxdb werden im Normalfall drei Windows geöffnet:

Command Window: zur Steuerung von CXdb

Source Window: zur Darstellung des gerade durchgeführten Source-Code-Abschnitts

Process Interface Window: für die interaktive Ein- und Ausgabe des laufenden Programms

Weitere Windows werden nach Bedarf geöffnet: Disassembly Window, Stack Window, Examine Window, Register Window und Help Window. Drei Arten von sogenannten Predefined Event Points können gesetzt werden: Breakpoints, Tracepoints und Watchpoints.

Ein auf X Window aufbauender Online Training Guide für CXdb ist vorhanden.

GKSGRAL

GKSGRAL ist die vollständige, portable Implementierung des nach DIN und ISO (ISO 7942) genormten Graphischen Kernsystems (GKS) auf dem Level 2b. Es wurde von GTS-GRAL (Darmstadt) entwickelt und wird von MSB (Wien) vertrieben und installiert. GKSGRAL bietet die Möglichkeit zur Erstellung, Speicherung und Ausgabe von zweidimensionalen Bildern, sowie zur Ein- und Ausgabe auf graphischen Workstations.

Die zentralen GKS-Funktionen sind die graphischen Ausgabeelemente Polygon, Polymarker, Text, Füllgebiet, Zellmatrix und verallgemeinertes Darstellungselement. Ihre wichtigste Eigenschaft ist die Geräteunabhängigkeit; sie

definieren ein "logisches Bild", das auf den graphischen Geräten entsprechend deren Fähigkeiten dargestellt wird.

An Gerätetreiber-Schnittstellen sind derzeit vorgesehen X Window, Tektronix (einschließlich Emulation mit Kermit), PostScript und HPGL. Nach Installation und Test von GKSGRAL wird das Anwendungspaket SimplePlot folgen.

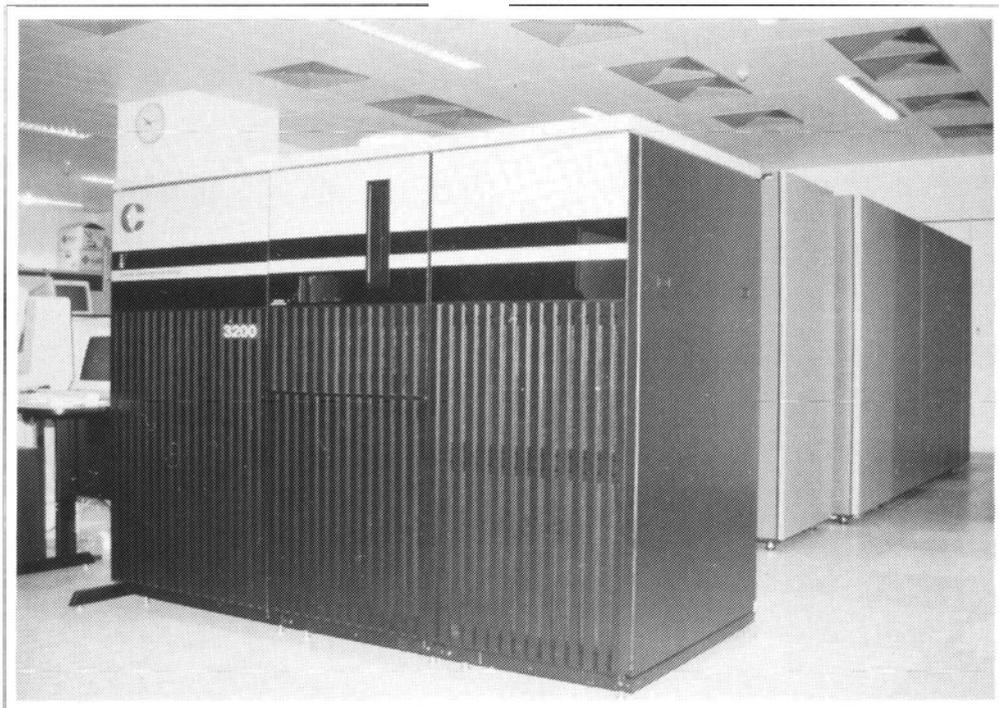
Das Visualisierungssystem AVS

Convex AVS ist die Convex-spezifische Implementierung des auf vielen Rechnerarchitekturen verbreiteten Application Visualization System (AVS) von Stardent Computer Inc. AVS ermöglicht mit seinem modularen Aufbau die Visualisierung von Ergebnissen technisch wissenschaftlicher Aufgabenstellungen, problembezogen und ohne zusätzlichen Programmieraufwand.

Es stehen die Subsysteme Image Viewer, Volume Viewer und Geometry Viewer zur Verfügung.

Neben einer großen Zahl von Standardmodulen für die Eingabe, das Filtern, die Darstellung und die Ausgabe der Daten können eigene Module entwickelt werden. Der Datenfluß wird mit einem Netzwerk-Editor festgelegt, wobei die einzelnen Module auch auf unterschiedlichen Systemen ablaufen können (verteilte Verarbeitung). Da zum jetzigen Zeitpunkt TU-weit noch keine Entscheidung für ein bestimmtes Visualisierungssystem gefallen ist, werden Benutzer, die sich mit AVS beschäftigen möchten, gebeten, sich mit dem EDV-Zentrum diesbezüglich in Verbindung zu setzen.

Helmut Mastal



CONVEX C3220

Abbau der Anlage CDC Cyber 860

Der Abbau der CDC-Rechenanlage Cyber 180-860A wird nach Ablauf des Mietvertrages Ende März 1992 durchgeführt.

Damit steht ab 1. April 1992 an der TU Wien kein allgemein zugänglicher Rechner mit dem Betriebssystem NOS/VE zur Verfügung. Die Verwendung der Cyber 930 als Übergangslösung ist nur in begründeten Ausnahmefällen möglich, da dieser Rechner über sehr wenig Plattenspeicher verfügt, sich das Betriebssystem auf dem höheren Level NOS/VE 1.5.1 befindet, keine Applikationssoftware vorhanden ist und der Abbau Ende 1992 erfolgt.

Welche Rechner stehen zur Verfügung ?

Nach der Installation des neuen zentralen UNIX-Unterstützungssystems und der beiden Fachbereichsrechner stehen als zentrale Rechenanlagen folgende Systeme zur Verfügung:

- CONVEX C3220, zentrales UNIX-Unterstützungssystem
- CONVEX C220, Fachbereichsrechner Physik
- IBM RS/6000-950+550, Fachbereichsrechner MB + BI
- VAX VMS-Cluster, VMS-Unterstützungssystem
- SNI VP50-EX Hochleistungs-Vektorrechner (unter VSP/I)

Alle diese Systeme sind in das TUNET eingebunden und können mit TCP/IP (u.a. TELNET und FTP) angesprochen werden.

Fileübertragung

Wie die Fileübertragung auf diese Rechner am besten durchgeführt werden kann, wurde in der PIPELINE Nr. 4 ausführlich dargestellt.

Wir möchten nochmals darauf hinweisen, daß Binaries, Object-Modules, NOS/VE interne Backup- und Bibliotheks-Files sowie Files, die mit AAM-Utilities erzeugt wurden, auf anderen Rechnern **nicht** mehr verwendet werden können. Das sind also Dateien, die mit Compilern, mit OCU oder BUILD_UTILITY, mit BACPF (das gilt auch für die vom Rechenzentrum durchgeführte Datensicherung) und mit SCU erstellt wurden, sowie keyed/indexed-sequential und mit FMU speziell reformatierte Files.

Eine direkte Migration jener Files ist nicht sinnvoll - es müssen vor der Übertragung mit den jeweiligen Tools die ihnen zugrunde liegenden "Source-Files" (im Plain-Text Format) wiederhergestellt werden, die dann auf andere Rechner transferiert werden können.

Files, die mit SDF (Generierung von Bildschirmmasken) erzeugt wurden, können in keinem Fall weiter verwendet werden.

Welche Services werden weitergeführt ?

* Zentraler CDC-Drucker und die CDC-Drucker in den Benutzerräumen Gußhausstraße und Freihausgründe

- Der CDC Zeilendrucker im Maschinenraum 2. Stock (1000 Zeilen/min) wird an die CONVEX C3220 angeschlossen und kann dann von jedem UNIX-System aus verwendet werden (Übertragung mit LPD/LPR-Protokoll).

- Die CDC Zeilendrucker (600 Zeilen/min) in den Benutzerräumen Freihausgründe (Erdgeschoß) und Gußhausstraße (5. Stock) werden über die bestehenden Druckerkonsolen an die Novell-Server in diesen Räumen angeschlossen und können so von jedem *zentralen* UNIX-Rechner (auf Antrag auch von anderen Systemen aus) angesprochen werden.

* HP-Plotter DIN A3/4 (HP 7550A)

Der Plotter wird an die CONVEX C3220 angeschlossen und kann von jedem UNIX-Rechner aus angesprochen werden (nur HPGL-Files)

* Laserdrucker DIGX und DIGW

Die beiden Laserdrucker stehen ab 1. April 1992 nicht mehr zur Verfügung; als Ersatz sind in den Benutzerräumen "Freihaus-Erdgeschoß" und in der Gußhausstraße 27-29, 5. Stock, Laserdrucker EPSON EPL 7500 installiert, die über das TUNET angesprochen werden können. Weiters steht ein Farb-PostScript-Drucker Tektronix Phaser III zur Verfügung (Achtung: kostenpflichtig), der am EVAX-Cluster angeschlossen ist..

Eine genaue Information, welche Drucker wie angesprochen werden können, wird in Kürze am EDV-Zentrum aufliegen.

Welche Services werden eingestellt ?

* Die beiden CDC Zeilendrucker im Bereich Getreidemarkt stehen nicht mehr öffentlich zur Verfügung.

* Für die Verwendung von Electronic Mail (die Bitnet-Knotenadresse AWITUW01 entfällt), EARN/NTF und PAD ist eine Benutzerberechtigung am Mail-Server (EMAIL) notwendig. Mail kann auf Antrag für maximal 6 Monate auf einen anderen Rechner umgeleitet werden.

* Auf den oben angeführten Rechnern sind ERLGRAPH (Graphikpaket) und SPSSX (Statistikpaket) nicht installiert.

* Die an der Cyber installierten Produkte ACM-Algorithmen, MOTIS, NONSAP, REDUCE, SAP4 und SPICE werden nicht mehr verfügbar sein.

Die ACM-Algorithmen ab 493 sind von einem öffentlich zugänglichen Server abrufbar. Senden Sie Mail an

netlib@ornl.gov

mit dem Auftrag

send nnn from toms

nnn ist die Nummer des Algorithmus, oder

send index from toms

für ein Verzeichnis aller auf dem Server verfügbaren ACM-Routinen. Der Auftrag kann in der Kopfzeile "Subject" oder im Textbereich stehen.

Wir ersuchen dringend um Mitteilung, ob durch den Abbau der Cyber-Anlage Probleme oder Unzulänglichkeiten (z.B. durch Verwendung von Cyber-spezifischen Datenstrukturen, Systemroutinen, Softwarepaketen etc.) auftreten, die durch die oben beschriebenen anderen Rechner nicht gelöst werden können.

Für Fragen und Auskünfte wenden Sie sich bitte an Herrn P. Berger (Kl. 5815) oder an Herrn E. Srubar (Kl. 5826).

Peter Berger, Erwin Srubar

Störungsmeldungen TUNET

Falls Sie Störungen im TUNET melden wollen, gibt es folgende Möglichkeiten:

- **Über electronic Mail** (falls sie bei einer Störung noch funktioniert):

Bitte schicken Sie eine Mail an die Adresse

trouble@tunamea.tuwien.ac.at

in der Sie Zeit, Ort und Art der Störung und eine Telefonnummer für Rückfragen angeben. Diese Mail wird automatisch an alle Mitarbeiter der Abteilung Kommunikation verschickt. Sie erhalten eine automatische Antwort, daß Ihre Mail eingetroffen ist.

- **Über Telefon:**

Für den Bereich TUNET gibt es eine eigene Störungsnummer:

587 56 23

Dort meldet sich normalerweise Herr Kondraschew. Falls er gerade nicht im Zimmer ist, meldet sich ein Anrufbeantworter. Bitte geben Sie immer Zeit, Ort und Art der Störung, sowie Ihren Namen, Institut und Telefonnummer für Rückfragen an.

Falls Störungen vorhersehbar sind (z.B. Arbeiten am Netzwartungstag - jeweils am 2. Montag im Monat von 8.00-12.00 Uhr), werden diese in der Newsgruppe

at.tuwien.tunet

angekündigt.

Johannes Demel

Personelle Veränderungen

Der Bundesminister hat mit Wirksamkeit vom 1. 1. 1992 Frau **Dipl. Ing. Irene Hyna** an das Bundesministerium für Wissenschaft und Forschung dienstverpflichtet, wo sie u.a. für die Campuslizenzen österreichweit zuständig sein wird. Wir wünschen uns, daß Frau Dipl. Ing. Hyna auch in ihrer neuen Funktion die Anliegen der TU Wien mit so viel Engagement und Tatkraft wie bisher unterstützen wird.

Dr. Günther Wehrberger wird ab nun die Funktion eines Leiters der vorzusehenden Gemeinschaftseinrichtung "EDV-Pool" der Fakultät für Raumplanung und Architektur wahrnehmen. Wir hoffen, daß sich dadurch die freundschaftliche Zusammenarbeit mit der Fakultät für Raumplanung und Architektur vertiefen wird.

Beratung durch die Abteilung Institutsunterstützung

Die EDV-Beratung der Mitarbeiter der Institute und Abteilungen an der Technischen Universität Wien stellt eine der wichtigen Dienstleistungen der Abteilung Institutsunterstützung dar. Dafür stehen alle Mitarbeiter der Abteilung zur Verfügung, insbesondere für Fragen aus deren Spezialgebieten bzw. deren Schwerpunkten (siehe Tabelle). Beratung muß schnell, kompetent und zielführend durchgeführt werden. Es zeigt sich, daß je nach Komplexität der Fragestellung bzw. der angestrebten Hilfestellung unterschiedliche Wege der Kontaktaufnahme und der entsprechenden Dienstleistung eingeschlagen werden.

• Telefon

Bei kurz zu beantwortenden Anfragen, bzw. Anforderungen rufen Sie den Mitarbeiter direkt an. Entweder kann er Ihnen gleich helfen, oder er ruft Sie nach Erledigung Ihrer Anfrage zurück. Sollten Sie den Mitarbeiter telefonisch nicht gleich antreffen, wenden Sie sich bitte an das Sekretariat (Kl. 5481) oder an Herrn Schandl (Kl. 5488). Sie werden dann verlässlich zurückgerufen.

• e-mail

Wählen Sie bitte auch die Möglichkeit der e-mail, um Kontakt mit den Mitarbeitern der Abteilung Institutsunterstützung herzustellen, wenn Sie Probleme haben, bzw. können Sie Anfragen auf diesem Weg gleich mit den nötigen technischen Angaben versehen. Auch bei e-mail wird Ihnen der Rückkontakt garantiert. Wir sind bestrebt, Ihre Anfrage spätestens nach 24 Stunden zu beantworten.

• Besuch

Suchen Sie ein längeres Gespräch, bzw. wollen Sie über ein Problem detaillierter mit einem der Mitarbeiter sprechen, so machen Sie sich mit diesem bitte einen Termin über Telefon, bzw. e-mail aus, so daß genügend Zeit für Ihr Anliegen zur Verfügung steht. Je nach Sachlage können Sie den Mitarbeiter im EDV-Zentrum besuchen, bzw. wird es bei konkreten Problemen zweckmäßiger sein, an Ihrem Arbeitsplatz mit ihm zusammenzutreffen.

Machen Sie bitte auch von der Möglichkeit Gebrauch, mit einem oder mehreren Mitarbeitern der Abteilung über Ihre geplanten EDV-Aktivitäten, bzw. -Anschaffungen ein ausführliches Gespräch zu führen, um Ihre Planung optimal durchzuführen, insbesondere auch in Hinblick auf die zu erwartenden Folgekosten bezüglich Software und Wartung, aber auch in Hinblick auf die vom EDV-Zentrum unterstützten Schwerpunkte.

Neben der individuellen Beratung werden wir weiter versuchen, Sie durch unsere Veranstaltungen, sowie durch Schulungen und Workshops über Wissenswertes und Schwerpunkte zu informieren.

Albert Blauensteiner

Schwerpunkt	Name	Klappe	e-mail
AIX	B. Simon	5602	simon@edvz.tuwien.ac.at
Apollo	G. Houdek	3616	houdek@edvz.tuwien.ac.at
Applikationssoftware	G. Wehrberger	3617	wehrberger@edvz.tuwien.ac.at
CAD	G. Wehrberger	3617	wehrberger@edvz.tuwien.ac.at
DEC-Ultrix	B. Simon	5602	simon@edvz.tuwien.ac.at
Distributed Systems	A. Sprinzl	5599	sprinzl@edvz.tuwien.ac.at
Graphik	G. Wehrberger	3617	wehrberger@edvz.tuwien.ac.at
HCI	A. Sprinzl	5599	sprinzl@edvz.tuwien.ac.at
Macintosh / Apple	G. Gollmann	3611	gollmann@edvz.tuwien.ac.at
Objektorientierte Technologien	A. Sprinzl	5599	sprinzl@edvz.tuwien.ac.at
PC-UNIX	W. Selos	3610	selos@edvz.tuwien.ac.at
Software Engineering	A. Sprinzl	5599	sprinzl@edvz.tuwien.ac.at
SUN	G. Houdek	3616	houdek@edvz.tuwien.ac.at
Symbolische Algebra	J. Rogl	3612	rogl@edvz.tuwien.ac.at
T _E X	J. Rogl	3612	rogl@edvz.tuwien.ac.at
VMS	R. Sedlaczek	3611	sedlaczek@edvz.tuwien.ac.at

Informationen aus der Bibliothek der Technischen Universität Wien am EMAIL-Server

Seit Mitte Dezember 1991 ist am EMAIL-Server der TU Wien ein eigenes Verzeichnis mit dem Namen `ub` eingerichtet. Dort befinden sich Listen und Verzeichnisse, die vorher meist nur in gedruckter Form vorhanden waren bzw. in der Bibliothek erfragt werden mußten. Diese Daten können nun ohne besondere Zugangsberechtigungen mittels Anonymous FTP auf den eigenen Rechner übertragen und dort weiterverarbeitet werden.

Die Informationen befinden sich in folgenden Unterverzeichnissen:

<code>cd-rom</code>	Die in der Bibliothek verfügbaren und kostenlos abfragbaren Datenbanken auf CD-ROM
<code>datenbanken</code>	Die an der Informationsvermittlungsstelle der Bibliothek abfragbaren bibliographischen Datenbanken
<code>oeffnungszeiten</code>	Aktuelle Öffnungszeiten der einzelnen Abteilungen der Bibliothek
<code>schlagwortliste</code>	Liste von den im Bibliotheksverbundsystem BIBOS vergebenen Schlagwörtern für die an der Bibliothek vorhandenen Bücher
<code>zeitschriftenliste</code>	Liste der in der Bibliothek und an den einzelnen Instituten vorhandenen Zeitschriften

Genauere Beschreibungen dieser Listen liefert das File `README` im Verzeichnis `ub`, fallweise auch Files mit dem Namen `README` in den einzelnen Unterverzeichnissen.

Die Listen werden in unregelmäßigen Abständen auf den neuesten Stand gebracht. Das Erstellungsdatum der Dateien ist im Dateinamen enthalten (so bedeutet z.B. der Name der im Unterverzeichnis `cd-rom` enthaltenen Datei `cd151291.lis`: Liste der CD-ROMs vom 15.12.1991).

Im folgenden Beispiel sei angeführt, wie das Zeitschriftenverzeichnis auf einen an das TU-Netz angeschlossenen PC übertragen werden kann:

```
ftp email.tuwien.ac.at      oder      (Verbindungs-
ftp ftp.tuwien.ac.at      aufnahme)

Userid: anonymous
Password: ftp

cd ub                        (Wechseln in das Verzeichnis /ub)
dir oder ls                  (liefert eine Liste der Unterverzeich-
                             nisse, s. oben)
cd zeitschriftenliste      (Wechseln in dieses Unterverzeichnis)
get zs161291.lis           (Übertragen der Liste auf den eigenen
                             Rechner)
quit                         (Ausstieg)
```

Bei Fragen wenden Sie sich bitte an die Bibliothek (E-Mail-Adresse: `hhrusa@email.tuwien.ac.at` bzw. Tel.: 58801-5965 Hr. Hrusa, oder 58801-5964 Hr. Neumayer). Allgemeine Fragen zur Bibliothek und deren Benutzung stellen Sie bitte an die während der Öffnungszeiten jederzeit besetzte Zentrale Information (Tel.: 58801-5951).

*Hans Hrusa, Fritz Neumayer
UBTU Wien*

TUWIS-Statistiken

TUWIS (Technische Universität Wien InformationsSystem)

Der Akademische Senat der TU Wien hat am 14.10.1991 beschlossen, alle nicht personenbezogenen Daten der Verwaltung in Form von Statistiken und Listen auf einem über das TUNET zugänglichen Rechner verfügbar zu machen. Der (Lese-)Zugriff auf diese Daten kann von jedem Interessenten mit seiner eigenen Usernummer oder, falls eine solche nicht vorhanden ist, über die Usernummer ANONYMOUS und dem Paßwort FTP erfolgen.

Die am EMAIL-Server im Verzeichnis `/ud` (bzw. `/home/ftp/ud`, wenn man am Mailserver interaktiv arbeitet) abgelegten TUWIS-Statistiken werden in regelmäßigen Abständen vom Verwaltungsrechner auf dem letzten Stand gehalten - das Erstellungsdatum entnehmen Sie bitte dem

Modificationdate der Datei. Für den Inhalt verantwortlich ist die ADV-Abteilung der Universitätsdirektion.

Derzeit sind im Verzeichnis `/ud` folgende Unterverzeichnisse zu finden:

<code>stud</code>	Studentenzahlen, Inskriptions- und Absolventenstatistiken
<code>pers</code>	Personalverzeichnis pro Institut oder alphabetisch, Statistik
<code>mitt</code>	Mitteilungen der Universitätsdirektion (Allgemeine Hinweise, Akademische Funktionäre, Mitteilungsblätter)
<code>lva</code>	Liste der Lehrveranstaltungen mit Ort- und Zeitangaben
<code>stpl</code>	Studienpläne

NAG und IMSL an der CONVEX

An der CONVEX C3220 stehen die beiden mathematisch-statistischen Programmsammlungen NAG (Numerical Algorithms Group) und IMSL (International Mathematical and Statistical Library) zur Verfügung.

Anstelle der BLAS-Routinen (Basic Linear Algebra Subprograms) von NAG bzw. IMSL werden die entsprechenden Routinen aus der VECLIB verwendet. Die VECLIB ist eine speziell für die CONVEX-Rechner geschaffene Library von optimierten mathematischen Basisroutinen.

NAG FORTRAN Library

Installiert wurde die Library-Version Mark 14 in Double Precision im CONVEX Standard Floating Point Format ("native format").

IMSL Libraries

Installiert wurden

- MATH/LIBRARY Fortran Subroutines for Mathematical Applications (Vers. 1.1)
- STAT/LIBRARY Fortran Subroutines for Statistical Analysis (Vers. 1.1)

- SFUN/LIBRARY Fortran Subroutines for Evaluating Special Functions (Vers. 1.2)

Die installierte Library enthält Routinen in Single und in Double Precision im CONVEX Standard Floating Point Format ("native format").

Verwendung

Die Files **readme** in den Directories `/usr/local/nag` bzw. `/usr/local/imsl` enthalten Informationen über die Verwendung der Libraries.

Dokumentation

Die gedruckte Library-Dokumentation liegt bei mir im Zimmer (EDV-Zentrum, Wiedner Hauptstraße 8-10, 2. Stock, gelber Bereich, DB02O20) zur Einsichtnahme auf.

Bei Fragen die beiden Produkte betreffend wenden Sie sich bitte an mich (Klappe 5603, e-mail: haider@edvz.tuwien.ac.at).

Walter Haider

NAG am Fachbereichsrechner Maschinenbau und Bauingenieurwesen

Auf dem Rechnersystem IBM RS/6000-950 und 550 steht die NAG FORTRAN Library (Mark 14) in Double Precision (IEEE-Format) zur Verfügung.

Die BLAS-Routinen (Basic Linear Algebra Subprograms) sind in der Library inkludiert.

Verwendung

Das File **readme** in dem Directory `/usr/local/nag` enthält Informationen über die Verwendung der Library.

Dokumentation

Die gedruckte Library-Dokumentation liegt bei mir im Zimmer (EDV-Zentrum, Wiedner Hauptstraße 8-10, 2. Stock, gelber Bereich, DB02O20) zur Einsichtnahme auf. Bei Fragen die NAG-Library betreffend wenden Sie sich bitte an mich (Klappe 5603, e-mail: haider@edvz.tuwien.ac.at).

Walter Haider

Lösen von sehr großen linearen Gleichungssystemen auf der VP50-EX

Seit kurzem steht auf der VP50-EX ein Out-Of-Core-Solver (Gleichungslöser für Matrizen, die nicht zur Gänze im Hauptspeicher Platz haben) für lineare Gleichungssysteme zur Verfügung.

Mit Hilfe dieses Gleichungslösers ist es möglich - mit nur 100 MB Vektormemory - Gleichungssysteme bis zur Dimension $n = 5500$ in doppelter Genauigkeit zu lösen. Der dabei verwendete Algorithmus beruht auf der Zerlegung der Matrix in einzelne Blöcke, die nicht gleichzeitig im Speicher gehalten werden müssen.

Für die Lösung eines Systems der Dimension $n = 5500$ werden ungefähr 560 CPU-Sekunden, für $n = 4000$ ungefähr 220 CPU-Sekunden benötigt. Auf die benötigten Dateien 'LIB1.OUT-CORE.CNTL', 'LIB1.OUT-CORE.FORT77' und 'LIB1.OUT-CORE.INC' können alle Benutzer der VP50-EX zugreifen. Es stehen sowohl ein Jobbeispiel, ein Programmbeispiel als auch Testdaten zur Verfügung.

Am EDV-Zentrum ist eine ausführliche Beschreibung für die Verwendung dieses Gleichungslösers erhältlich.

Iris Macsek

X-Terminal-Software für PCs : HCL eXceed

Aufgrund der regen Nachfrage auf unsere Bestellaktion wurden 50 Lizenzen des Produkts HCL eXceed in Form einer Sammellizenz für die Institute angeschafft. Einige von den Lizenzen können noch vergeben werden (siehe auch Seite 38). Wir wiederholen hier kurz die wichtigsten technischen Eigenschaften der Software.

Dabei handelt es sich um eine vollständige Implementierung eines X-Window Rel. 11.4 Servers. Es ist eine optionelle Unterstützung von MS-Windows, sowie den Adaptern 8514 bzw. TIGA vorgesehen.

Was benötigen Sie dazu an Hard- und Software?

- einen PC mit den Prozessoren: 80286, 386 oder 486
- mindestens 640k RAM unterhalb von 1MB
- mindestens 1MB RAM über 1MB
- einen EGA oder VGA-Adapter mit entsprechendem Monitor
- eine Microsoft-kompatible Maus
- eine Harddisk
- eine Ethernet-Karte mit dazugehöriger TCP-Transport-Software

Transport-Software

PC/TCP
PC/NFS

Hersteller

FTP (vom EDV-Z unterstützt)
SUN (vom EDV-Z getestet)

LAN-Workplace
LAN-Workplace
PathWay, WIN/TCP
3+Open TCP
HP-ARPA-Services
TCP BNS/PC
NFS od. Telnet Package

Excelan
Novell
Wollongong
3Com
HP
Ungermann-Bass
Beam & Whiteside

Highlights:

- Unterstützung von Windows
- Unterstützung von 8514-Adaptern
- Version für TIGA am EDV-Zentrum erhältlich
- Übersichtliches Konfigurationsprogramm
- Importierung von Fonts im .BDF-Format möglich
- Unterstützung des Internet-Nameservices, so die Transport-Software das kann.

Achtung: Seien Sie bitte sparsam mit Subnetz-übergreifenden X-Window-Sessions, da die Belastung von Routern, Bridges und dem Backbone-Netz sonst sehr schnell ansteigt!

Bei Fragen wenden Sie sich bitte an: Dieter Schornböck, Kl. 5820 (Bestellung, Verteilung, Organisation), Walter Selos, Kl. 3610 (Technische Beratung für Installation und Betrieb).

Walter Selos

Neues vom TUNET

Nach Installation einer neuen Systemversion im CDCNET stehen folgende neue Funktionen zur Verfügung:

1. Zugang zu "beliebigen" TCP/IP-Rechnern vom CDCNET aus. Bisher war für Terminals, die am CDCNET angeschlossen sind, die Möglichkeit, auf TCP/IP-Rechner (in der Regel der TU Wien) eine Verbindung aufzubauen, sehr eingeschränkt, da für sämtliche Rechner, für die ein Zugang möglich sein sollte, eine händische Konfiguration im CDCNET erforderlich war. Mit der neuen Systemversion unterstützt CDCNET das Domain Name Server Konzept. Dadurch entfällt die explizite Konfiguration. Zu einem Rechner im TUNET, der über TCP/IP - TELNET erreichbar ist, kann nun mit dem folgenden Befehl eine Verbindung aufgebaut werden:

```
crec telnet sd='rechnername'
```

also z.B. zum neuen UNIX-System (CONVEX C3) mit dem Befehl

```
crec telnet sd='ecx'
```

oder

```
crec telnet sd='ecx.tuwien.ac.at'
```

2. Ab sofort funktioniert der Zugang zur VP50-EX über TCP/IP und IBM 7171 (für Nicht-3270-Terminals bzw. Emulationen) mit der TCP/IP-Adresse:

```
vp7171.tuwien.ac.at
```

Das bisher aufgetretene Problem mit dem doppelten Echo auf dieser Adresse konnte nach Installation der neuen Release vom CDCNET in den CDC Terminal-Servern (die in

diesem Fall intern zur Umsetzung von TCP/IP auf asynchrone Verbindungen eingesetzt werden) behoben werden.

3. Zugang zur Cyber 860 von einem TCP/IP Rechner aus (z.B. einem PC mit LRZ-Connect).

Bisher konnte man die Cyber 860 am EDV-Zentrum von einem TCP/IP Rechner aus nur über den Befehl

```
telnet diga.tuwien.ac.at
```

erreichen (statt telnet kann je nach Terminalemulationsprogramm auch tn bzw. lrzconn stehen). Dieser Zugang hat jedoch den Nachteil, daß gewisse Funktionen nicht unterstützt sind (z.B. Page Length, Page Wait, Line Length, %-Befehle wie %2, %a, %s, %1, etc.).

Mit der neuen CDCNET Version wurde dieses Manko behoben.

Mit dem Befehl

```
telnet cdcnet.tuwien.ac.at
```

gelangt man in den CDCNET Command Modus direkt über TCP/IP, also mit der vollen Datenrate von 10 MBit/s.

Danach kann man, wie an einem an CDCNET angeschlossenen Terminal gewohnt, den Befehl

```
CREC DIGA
```

geben, um zur Cyber 860 zu gelangen. Es kann aber auch vorher der Befehl DO PWON oder DO PC_CONNECT_13, etc. gegeben werden.

Johannes Demel

SEDt Editor 4.2

Das EDV-Zentrum hat neben einer Source-Lizenz eine Site-Lizenz für die TU abgeschlossen und kann daher den SEDT auch den Instituten kostenfrei zur Verfügung stellen. Zur Installation auf Instituts-VAXen sind das Directory SEDT\$LIBRARY: von der EVAX zu kopieren und die Hinweise im File INSTALL.HOW zu beachten. Die aktuellen Versionen für SUN, ATARI, DOS und ULTRIX können vom VAX-Cluster von TUSKITS:[SEDt.V042...] abgeholt werden.

Am VAX-Cluster wurde die am EDV-Zentrum verbesserte Version 4.2 des Full-Screen Editors SEDT installiert. Informationen über den Editor sind jetzt auch über die normale VMS HELP-Funktion abrufbar: HELP SEDT

Wichtigste Verbesserung der VMS-Version gegenüber der bisher installierten Version 4.0 sind neben einigen Bug-Fixes die Maus-Unterstützung für DECterm- und VWS Terminal-Windows, die bisher nur die PC-Versionen boten.

Besondere Eigenschaften dieses Editors mit EDT-kompatibler Tastenbelegung sind:

- 'Insert' und 'Replace' Modus
- 'Cut' und 'Paste' von rechteckigen Blöcken
- Mehrere Fenster, horizontal oder vertikal teilbar, verschiebbar und in der Größe veränderbar
- Lern-Funktion
- 'Undo' des letzten Befehls
- Textsuche wahlweise sensitiv auf Groß/Kleinschreibung
- Textsuche und -ersetzung wahlweise als 'regular expression'
- Text-Formatierhilfen (Zentrieren, umschaltbare Tabulator- und Randeinstellungen, automatischer oder manueller Zeilenumbruch im Block- oder Flattersatz).

Rudolf Sedlaczek

MS Windows 3.1

- **Das sind die Fakten:**

Eine neue Ära kündigt sich an. Den Anwender erwartet eine Vielzahl von verbesserten Funktionen sowie eine deutlich gesteigerte Performance.

Die neue Version 3.1 wird im kommenden Frühjahr erwartet, doch die aktuelle Beta-Version zeigt, in welche Richtung es gehen wird: deutlich verbesserte Gesamtperformance, die Möglichkeit, mit Icons objektorientiert zu arbeiten und ein völlig neues Wysiwyg-Gefühl, dank TRUETYPE, sind nur einige Highlights aus dem neuen Windows Paket.

- **Der Real Mode ist tot**

Eine gravierende Maßnahme von MS-Windows 3.1 ist der Verzicht auf den Real Mode. Die graphische Benutzeroberfläche wird nicht mehr mit 8086/808er Mikroprozessoren laufen.

Der Real Mode war einer der größten Bremsen für Windows Programme und Programmierer. Ab Windows 3.1 müssen sich Entwickler nicht mehr um diesen Modus kümmern. Man kann diese Mechanismen allein der CPU (80286 aufwärts) und deren Fähigkeiten im Protected Mode überlassen.

- **Verbessertes Wysiwyg dank TRUETYPE**

Windows 3.1 wird über einen voll integrierten Fontmanager für Truetype-Schriften verfügen. Es handelt sich hierbei um ein von Apple lizenziertes Verfahren für die Darstellung von Outline-Fonts.

Windows 3.1 wird wahrscheinlich insgesamt 13 Truetype-Schriften enthalten, sowie die aus früheren Windows-Versionen bekannten Bitmap-Fonts.

- **Die Veränderungen im Programm-Manager sind klein, aber fein**

Auf den ersten Blick hat sich am Programm-Manager nicht viel verändert. Microsoft war bestrebt, nicht den Eindruck zu vermitteln, daß sich von Version zu Version das gesamte Outfit ändert.

Wenn man aber genau hinsieht, findet man doch kleine Unterschiede. Die Icons wirken plastischer und haben keine "Namenslatten", denn die jeweiligen Bezeichnungen können umgebrochen werden.

Auch die Hilfsfunktion wurde verbessert. Wo auch immer man sich im System befindet, eine kontext-sensitive Hilfe ist jetzt fast immer möglich.

Der Programm-Manager hat eine neue Programmgruppe erhalten. Sie heißt STARTUP und enthält alle Programme, die bereits beim Systemstart geladen oder gestartet werden sollen. Das ist ein eleganter Ersatz für LOAD= und RUN= aus der Datei WIN.INI.

- **Der Datei-Manager wurde gründlich überholt**

Zwar ist die Grundform geblieben, doch ist jetzt jedes Fenster teilbar. Ein einfacher Klick auf ein Laufwerksymbol ändert den Fensterinhalt.

So war es auch unter der Version 3.0. Neu ist, daß bei einem Doppelklick jetzt ein neues Fenster mit eigenem Verzeichnisbaum und Inhaltsverzeichnis erscheint und das alte Fenster seine Einstellungen behält.

Die Arbeitsgeschwindigkeit des Datei-Managers wurde erheblich verbessert, indem die Entwickler von Microsoft vor allem den Programmcode des Datei-Managers gestrafft haben. Zusätzliche Power im 386 Mode gibt es über den 32-Bit-Datentransfer. Ermöglicht wird er von zwei neuen Windows-internen Gerätetreibern.

- **Die Systemsteuerung bekommt Zuwachs**

In der Systemsteuerung von Windows ist die feste Balkenform des Control Panel verschwunden. Das Fenster ist jetzt größenvariabel und enthält in der untersten Zeile wie auch der Datei-Manager oder Excel eine Informationszeile.

- **Verbessertes Drag-n-Drop-Konzept**

In der neuen Windows-Version können jetzt mittels Drag-n-Drop Dateien geöffnet und gedruckt werden. Hierfür zieht man die Dateisymbole per Maus auf die entsprechenden Programm-Icons und läßt sie fallen. Der Rest geht von selbst.

- **Die Welt der Objekte hält nun auch bei Windows Einzug**

Aufbauend auf Clipboard und DDE (Dynamic Data Exchange) wird mit Windows 3.1 *Object Linking and Embedding*, kurz *OLE*, eingeführt. Dieser Begriff steht für die Möglichkeiten, Daten und Programme zu Objekten zusammenzufassen und diese wiederum in andere Objekte zu integrieren. Es wird dabei zwischen Embedding (der festen Einbindung) und dem Linking (der losen Verknüpfung) der Daten mit Programmen unterschieden.

In beiden Fällen benötigt man zum Erstellen eines Objekts zunächst einen Daten-Server und einen (oder mehrere) Daten-Client(s). Der Server enthält die Datenquelle - der Client ist das Datenziel, er fordert die Daten vom Server an.

- **Warmstart für jedes Fenster**

Für abgestürzte Programme gibt es unter Windows 3.1 mit einer Tastenkombination drei Möglichkeiten fortzufahren, mit den Zielen:

- 1.) Windows kehrt zu dem Programmpunkt zurück, wo es gerade unterbrochen wurde,
- 2.) die aktive Anwendung wird aus dem Speicher geworfen und alle zugehörigen Speicherfragmente hinterher,
- 3.) Windows wechselt nicht in das aktuelle Fenster, sondern bootet den PC neu.

Ich hoffe, Ihnen damit einige Neuerungen der kommenden Windows Version 3.1 mitgeteilt zu haben. Wie schon anfangs gesagt, wird die neue Version im kommenden Frühjahr erwartet und sie soll wiederum Bill Gates Meisterstück werden - lassen wir uns überraschen.

Lorenz Gisch

A/UX - UNIX für Macintosh

Was ist A/UX ?

A/UX ist Apples Version von UNIX für den Macintosh. Basis dieser Implementation ist AT&T System V mit vielen Berkeley Extensions. Das Besondere an A/UX ist die enge Integration von MacOS und UNIX. Eine speziell angepaßte Version von MacOS (derzeit 6.0.7) läuft dabei als UNIX Prozeß. Über die gewohnte "Desktop" Benutzeroberfläche sind beide Welten zugänglich, Dokumente können frei zwischen MacOS und UNIX Volumes kopiert werden.

Erste Eindrücke

Für Testzwecke wurde A/UX 2.0.1 auf einem Macintosh IIfx eingespielt. Die Installation von einer CD verlief weitgehend problemlos, bei der Konfiguration des Systems (Netzwerk etc.) ist allerdings doch ein deutlicher Unterschied zum Komfort des MacOS zu bemerken. Hier ist noch einige Arbeit für die Entwickler offen, denn MacOS zeigt mit MacTCP oder AppleShare, daß es auch einfacher geht. Die nächsten Monate werden zeigen, welche Seite ihre Schwächen schneller beseitigen kann. So fehlen etwa unter MacOS bisher Kommandoprozeduren für Operationen, wie die Verwaltung von Benutzern auf Fileservern. System 7 legt aber mit AppleEvents und AppleScript die Grundlage für Abhilfe auf diesem Gebiet. Unbefriedigend ist auch, daß unter der

gegenwärtigen Version von A/UX nur MacOS 6.0.7 zur Verfügung steht.

Einsatzbereich

Für die meisten Macintosh Benutzer bringt A/UX wohl keine Vorteile. Auch unter MacOS ist die Kommunikation mit der UNIX-Welt über Telnet, FTP, NFS, X-Windows, NetNews etc. möglich. Etliche dieser Produkte sind sogar Public Domain oder Shareware und von ausgezeichneter Qualität. A/UX ist daher dort sinnvoll, wo an einem Arbeitsplatz sowohl UNIX als auch MacOS gefahren werden soll - etwa um die Textverarbeitungsmöglichkeiten des Macintosh zu nutzen und gleichzeitig UNIX-Programme zu fahren.

Ausblick

Für das erste Quartal 92 ist die Version 3.0, die sich derzeit im Betatest befindet, angekündigt. Neben einer Reihe von Detailverbesserungen wird sie insbesondere Kompatibilität mit System 7 bringen. Langfristig soll A/UX in einer Gemeinschaftsentwicklung mit AIX (IBM) aufgehen. Ob, wann und in welcher Form diese Pläne allerdings realisiert werden, ist heute noch nicht abzuschätzen.

Georg Gollmann

AutoCAD

AutoCAD ist ein sehr leistungsfähiges CAD-Paket und bringt mit der neuen Release 11 viele Verbesserungen und Erweiterungen. Schon mit dem Basispaket können die vielfältigsten Probleme gelöst werden. Für spezielle Anwendungen, wie für Architekten und Bauingenieure oder Maschinenbauer gibt es Zusatzpakete, die durch vordefinierte Objekte die Arbeit wesentlich erleichtern. Der Preis für das Basispaket (mit AME) beträgt für Universitäten öS 9000.-, für AutoShade öS 7000.- o. MwSt.

Neue Funktionen in AutoCAD Release 11:

- *Plotten von Mehrfachansichten:*
Damit können verschiedene Ansichten einer AutoCAD-Zeichnung auf nur einem Blatt Papier ausgeplottet werden - genau so wie diese auf dem Bildschirm gestaltet wurden.
- *Prüfung und Wiederherstellung der Datenbank:*
Zeichnungsdateien werden auf Fehler überprüft und beschädigte Dateien wiederhergestellt.
- *Netzwerkunterstützung:*
AutoCAD kann so konfiguriert werden, daß es mit einem Server läuft. Eine Dateisperre schützt dabei auch auf heterogenen Netzwerken die Daten, indem nur ein Anwender auf einmal eine Zeichnung bearbeiten kann.

- *Integrierte Schnellschattierung:*
Damit können Oberflächen- und Festkörpermodelle dargestellt werden.
- *Erweiterte Bemaßung:*
Es gibt neue Bemaßungsfunktionen wie Koordinatenbemaßung, verbesserte Durchmesser- und Radiusbemaßung sowie die Möglichkeit, jede festgelegte Kombination der Bemaßungseinstellung zu speichern und wiederherzustellen.
- *Verbesserte Texterfassung:*
Neue Textfunktionen sind: eine Dialogbox für die Textbearbeitung, mehrere neue Optionen für die Textausrichtung sowie Sonderzeichen für deutsche, französische und andere fremdsprachige Versionen von AutoCAD.
- *Externe Referenzdateien:*
Damit können Referenzzeichnungen mit einer aktiven Zeichnung verbunden werden, ohne daß diese ständig zu der Zeichnungsdatei hinzugefügt werden muß.
- *AutoCAD Development System (ADS):*
Eine leistungsfähige Programmierschnittstelle für C und andere höhere Programmiersprachen, mit der eng integrierte Applikationsprogramme erstellt werden können.
- *Festkörper-Primitive:*
Festkörper-Primitive wie Keile, Zylinder, Kugeln,

Kegel, Quader und Tori sind im standardmäßigen AutoCAD-Paket.

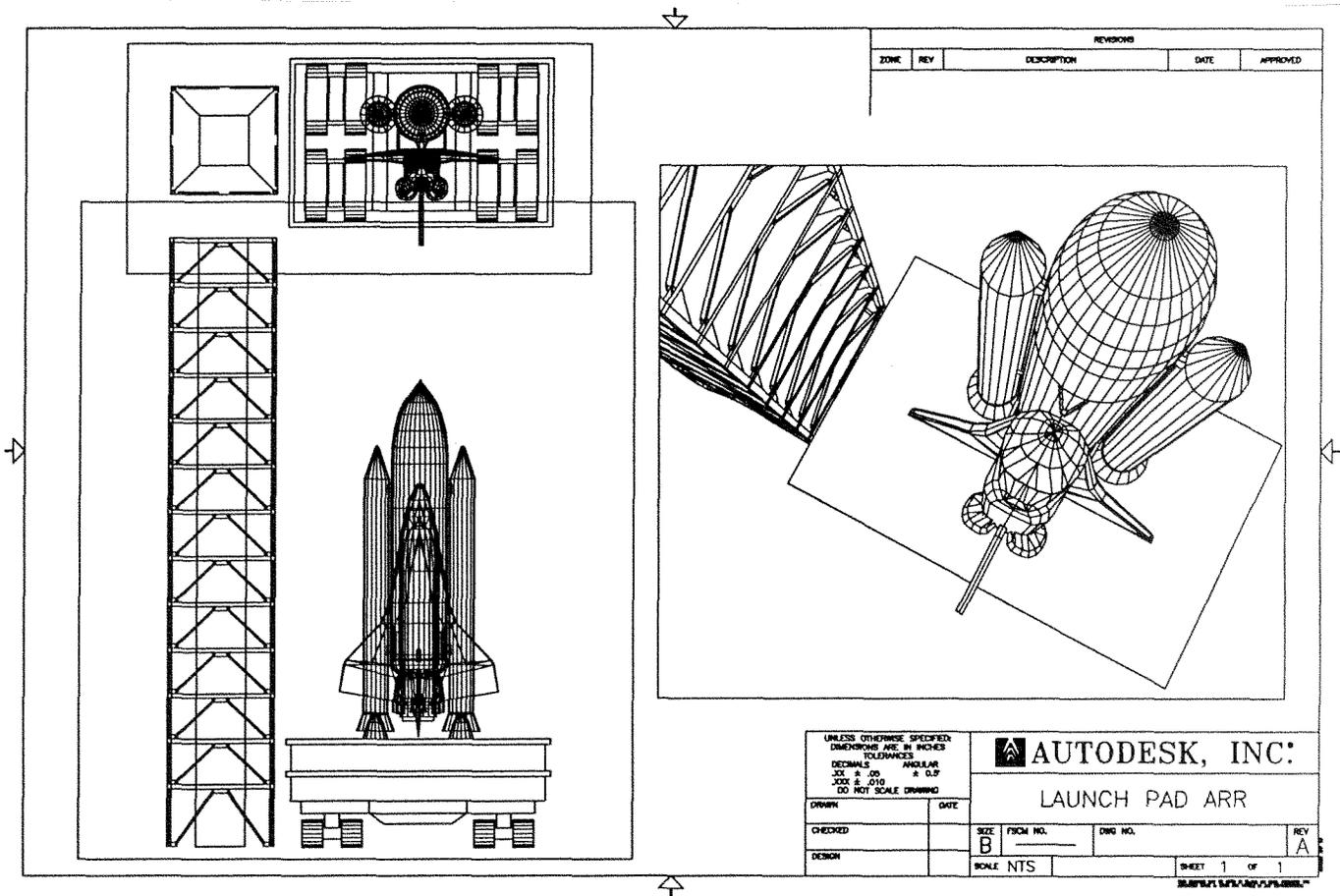
- *AutoCAD Advanced Modeling Extension (AME)*: Ein als Option lieferbares, in AutoCAD voll integrierbares Zusatzpaket, das zahlreiche Möglichkeiten bietet, Festkörper innerhalb von AutoCAD zu modellieren.

Die AutoCAD-Speicherverwaltung kann bis zu 4 GB Hauptspeicher adressieren. Was also an Memory nicht vorhanden ist, wird in 400 KB-Segmenten auf Festplatte ausgelagert. Die Festplatte sollte je nach Umfang der Zeichnungen 100 bis 300 MB groß sein. Eine parallele Schnittstelle ist für den Hardwarelock erforderlich.

Systemanforderungen für PC:

PC mit 386er Processor und Coprocessor oder mit 486er Processor, mindestens 2 MB Hauptspeicher, besser 4-8 MB.

Helmut Mayer



AutoCAD Demonstrationsbeispiel

CAE Einführungskurse mit VISULA

ganzjährig jeden Mittwoch Vormittag:

8 Uhr bis 10 Uhr: Vortrag

10 Uhr bis 12 Uhr: Übung

Anmeldung:

Sekretariat für Industrielle Elektronik,

Frau Ferner

Das Computer-Algebra-System *Mathematica*

Version 2.0 - Überblick

Trotz ihres jungen Alters (sie ist soeben 3 geworden), hat *Mathematica* die 'desk tops' der Mathematiker und Wissenschaftler der ganzen Welt revolutioniert. Dem immer breiteren Kreis der Benutzer wurde ein umfangreicher Satz neuer 'Features' in einigen Bereichen zur Verfügung gestellt: String Manipulation und Support von fremden Zeichensätzen, dreidimensionale Graphik und Graphikzusammenstellung, Ton, Differential-Gleichungen und Integration (symbolische und numerische), lineare Algebra und lineares Programmieren, direkte Interaktion mit den externen Programmen, Error Detection und Debugging wie auch Notebook Front End. Viele interne Änderungen sind, zwecks Effizienz, im 'Kernel' der Software durchgeführt worden, wie auch kleine Verbesserungen und 'bug fixes' aus der vorigen Version (1.2). Eine komplette Referenz zu Version 2.0 ist in der zweiten Auflage des Buches "*Mathematica: A System for Doing Mathematics by Computer*" (Addison-Wesley, 1991) von Stephen Wolfram erschienen.

Im folgenden Beitrag möchte ich die wesentlichen Features der Version 2.0 präsentieren:

Allgemeine Eigenschaften:

- Ein Compiler beschleunigt das Evaluieren numerischer Ausdrücke. Der Aufruf des Compilers erfolgt durch viele numerische und graphische Funktionen automatisch.
- Zusätzliche funktionale Programmierkonstrukte wie: `ReplacePart`, `Delete`, `Fold`, `FixedPointList`, `MapIndexed`, `MapThread` und `FlattenAt`.
- Block wurde durch `Module` und `With` ergänzt. Dies erlaubt das Lokalisieren der Variablennamen und Konstanten mit Hilfe des 'lexical scoping'. Diese Art des 'scoping' wird jetzt auch in den reinen Funktionen, Regeln und Zuweisungen verwendet.
- Reine Funktionen, Regeln und Zuweisungen werden jetzt als sog. 'scoping Konstrukte' behandelt, sodaß die darin vorkommenden Variablennamen 'local' verwendet werden. Iterations-Funktionen wie `Table`, `Sum` und `Do` verwenden jetzt `Block`, um die Iterationsvariablen zu lokalisieren.
- Patterns wurden verallgemeinert, um beliebige Bedingungen einzuschließen, z.B. $((x_+y_-; N[x+y]>0))$ ist jetzt ein gültiger Ausdruck.
- Informationen über diejenigen Berechnungen, die mit Maschinenzahlen durchgeführt wurden, sind jetzt auf einfache Art zugänglich; `MachineNumberQ` testet die Maschinenzahlen und die Variablen: `$MachinePrecision`, `$MachineEpsilon`, `$MachineEpsilon`, `$MinMachineNumber`, `$MaxMachineNumber` liefern als Resultat die Parameter der Maschinenarithmetik des verwendeten Rechners.
- Mit Hilfe der Funktionen `Evaluate`, `Unevaluated`, `Hold`, `ReleaseHold`, `ToHeldExpression` und `HoldForm` kann der Auswertungsgrad in speziellen Fällen besser kontrolliert werden.
- `Position`, `Case` und `Count` - akzeptieren jetzt auch 'level' Spezifikation .
- Eine neue Funktion `NDSolve` findet numerische Lösungen von gewöhnlichen Differentialgleichungs-Systemen. Der symbolische Differentialgleichungs-Löser `DSolve` kann eine größere Zahl von Gleichungstypen behandeln.
- `Integrate` bildet eine vollständige Implementation des Risch-Algorithmus für unbestimmte Integrale und Algorithmen für bestimmte Integrale, die auf den allgemeinen hypergeometrischen Funktionen basieren. Das heißt, es können fast alle Integrale, die in Integral-Tabellen aufgelistet sind, gelöst werden. `NIntegrate` wurde verbessert und hat mehr Optionen.
- `Limit` kann noch mehr Funktionen behandeln und ist imstande, die 'directional-limits' zu finden. Bei einer unbekannten Funktion wird der nicht-evaluierte Zustand beibehalten.
- Interpolationsfunktionen können mit Hilfe einer Menge von 'samples' definiert werden. Das Ergebnis einer numerischen Berechnung hat dann diese Form.
- Neue Funktionen lösen allgemeine lineare Programmierprobleme und das Minimalisierungs- und Maximalisierungs-Problem.
- Numerische lineare Algebra wurde effizienter gestaltet. Neue Funktionen beinhalten: `MatrixPower` und `MatrixExp` (symbolisch und numerisch), `QRDecomposition` und `SchurDecomposition` (nur numerisch).
- `Faktor`, `Expand` und auch andere Funktionen haben jetzt eine Option zur Behandlung von trigonometrischen Ausdrücken. Spezielle Funktionen wie `ComplexExpand` unterstützen symbolische Manipulation der Ausdrücke mit komplexen Variablen. Zusätzlich kann mit `Expand` spezifiziert werden, welcher Teil eines Ausdruckes entwickelt werden soll.
- Die trigonometrischen Funktionen `Csc`, `Sec` und `Cot` wie auch die hyperbolische Funktion `Sinh` etc., werden in ihrer nicht-evaluierten Form behalten.
- Die Fakultät kann mit Hilfe von Gauß'schen Zahlen spezifiziert werden.
- Eine Anzahl von zusätzlichen speziellen mathematischen Funktionen ist implementiert worden: `SinIntegral`,

RiemannSiegelZeta und viele der statistischen Verteilung verwandte Funktionen.

Graphik und Ton

- Beliebige Typen von Graphiken können zusammengelegt werden, um eine graphische Matrix zu bilden.
- Volle Spezifikation der Optionen, die als Automatic gesetzt worden sind, wie auch explizite Aufzählung von Graphik-Grundelementen (definiert durch Graphik-Optionen) erlauben die Kommands: FullOption und FullGraphics.
- Direkter Zugriff zum PostScript wird unterstützt; Liniendicke und Punktgrößen können absolut spezifiziert werden.
- Dreidimensionale Graphik unterstützt jetzt Text, beliebige Perspektiven, Farbe der Innenflächen, Überlappung (Supperposition) der Objekte, Bounding-Box Clipping, parametrisches Plotten der Flächen und Konversion zu zweidimensionaler Graphik.
- Eine Menge von neuen Optionen kam dazu, sowohl für die zwei-, wie auch für die dreidimensionale Graphik, wodurch man mehr Einfluß auf Achsen, Rahmen, Hintergrund, Anzahl und Abstand der Konturen und Farben gewonnen hat.
- Die Fähigkeit, Ton zu generieren, ist auf den meisten Systemen implementiert worden. Play und ListPlay verwenden entsprechende Funktionen, um Daten um eine klingende Wellenform hören zu lassen.

String und File Manipulation

- Fremde Zeichensätze und mehrsprachige Messages werden unterstützt.
- Beliebige Ein- oder Zweibytezeichen können verwendet werden. Ein allgemeiner Mechanismus, der zwischen externen und internen Zeichen unterscheidet, wurde eingebaut.
- Eine Reihe von Funktionen behandelt String Search, Substitution und Reversal, Konversion der Klein- und Großbuchstaben wie auch Überprüfung der Syntax. Es gibt Tests für unterschiedliche Zeichentypen.

- Neue File-Operationen beinhalten Directory, SetDirectory, FindList (Suchmöglichkeit in den externen Files), FileNames und FileByteCount.

- Sogenannte Streams wurden eingeführt (z.B.: Verwenden des UNIX-Pipe-Mechanismus). Die Strings können zu Streams konvertiert werden, sodaß Standard-Fileoperationen über diese ausgeführt werden können.

- ReadList ist mit neuen Optionen ausgestattet (Behandeln von zusätzlichen Datenformaten und das Überspringen des Textes ist möglich). Die Typen Record und Word sind dazu gekommen.

Debugging und Error Detection

- Es existiert eine allgemeine Debugging Möglichkeit: die Funktion Trace liefert eine 'execution history'.
- *Mathematica* Shells können innerhalb einer Berechnung rekursiv aufgerufen und verlassen werden, wonach das System zu seinem ursprünglichen Zustand zurückkehrt.
- Bei Syntax Error werden erweiterte Error Meldungen ausgegeben. Mögliche Input Errors (Tippfehler inklusive) werden erkannt und gemeldet.
- Der Befehl MessageList gibt eine Liste all jener Messages aus, die während der Evaluierung einer bestimmten Eingabezeile gefunden wurden.

Externe Interaktion

- An UNIX-Systemen ermöglicht das MathLink-Protokoll eine high-level Kommunikation zwischen *Mathematica* und externen Programmen. Es können z.B. externe Befehle wie der C-Compileraufruf eines durch *Mathematica* generierten C-Programms und dessen anschließende Ausführung von *Mathematica* aus durchgeführt werden. Umgekehrt ist es auch möglich, daß ein externes Programm - durch Verwendung bestimmter Libraries - *Mathematica* aufrufen kann.
- Die Macintosh Benutzer-Oberfläche beinhaltet eine Reihe von Neuigkeiten: übersichtliche kleine Menüs, ein Notebook-basierendes Help, neue Editier- und Layout-Funktionen, die Möglichkeit, Notebooks in RTF oder TeX-Format abzuspeichern, Text-Kovertierung nach PICT, und vieles mehr.

Jadwiga Rogl

Der IBM RS/6000 Workstation Cluster am EDV-Zentrum

Mitte Oktober des vergangenen Jahres wurde ein Cluster mit neun IBM RS/6000 Workstations am EDV-Zentrum installiert. Ziel dieses Artikels ist, den Zweck und die Verwendungsmöglichkeiten darzustellen. Zuvor werden jedoch die Hardware und die eingesetzte Software etwas genauer beschrieben.

Hardware

Der Cluster besteht aus neun IBM RS/6000-320H Workstations, wobei eine (im folgenden Mastersystem genannt) etwas besser ausgestattet ist als die anderen.

Jede Workstation enthält als CPU den POWER Chipsatz von IBM, der mit 25 MHz getaktet wird (dies entspricht dem Modell 530 mit dem Unterschied, daß das Modell 320H im Desktop Gehäuse untergebracht ist, außerdem ist beim Modell 320H der Datencache nur halb so groß, nämlich 32 KBytes), 16 MBytes (32 MBytes) Speicher und 400 MBytes (1,4 GBytes) Plattenkapazität (wobei die in Klammer stehenden Werte für das Mastersystem gelten).

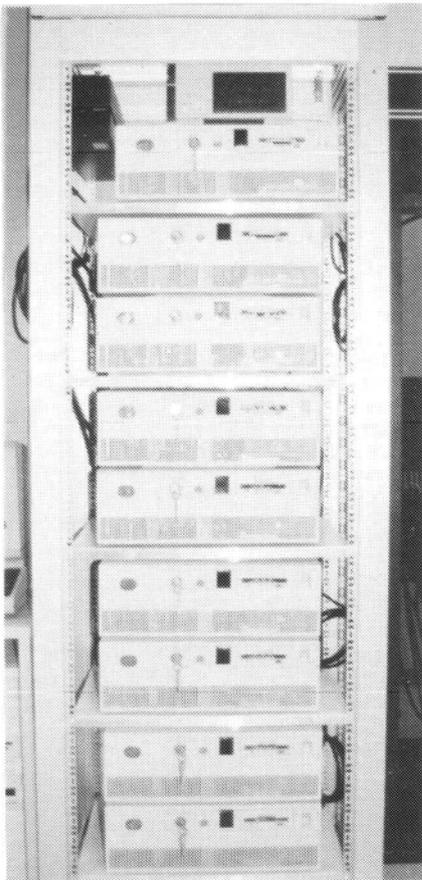


Abbildung 1

Allen Workstations fehlt jedoch Tastatur und Monitor und nur das Mastersystem ist zur Systemadministration mit einem Terminal verbunden. Die Workstations selbst sind via Token-Ring (16 MBit/s) miteinander vernetzt. Das Mastersystem stellt die Verbindung zum TUNET mit einem

zusätzlichen Ethernet Interface her. Auf beiden Transportmedien wird TCP/IP als Protokoll verwendet. Zusätzlich werden die Filesysteme /u (enthält die Home Directories der User) und /local (worauf sich die zusätzlich zum System installierte Software befindet) des Mastersystems via NFS von den anderen Workstations gemounted. Auf den lokalen Platten befinden sich jeweils das komplette Betriebssystem (AIX 3.1), temporäre Daten und ausreichend Swap Space. Die Workstations sind zu einem Turm aufeinandergestellt (siehe Abb. 1).

Software, Parallelisierungswerkzeuge

Die Initiative zu diesem Workstation-Projekt ging von Prof. Weinberger aus, der vom Gewinn des Gordon Bell Awards für das "Beste Preis/Leistungsverhältnis" durch G.A. Geist (1991) und Jack Dongarra (LINPACK!) erfahren hat. Erreicht wurde dieser Erfolg mittels PVM (Parallel Virtual Machine). Dieses von Vaidy Sunderam von der Emory University (Atlanta) entwickelte und von Adam Beguelin und Robert Manchek (Oak Ridge National Laboratory und University of Tennessee) weiterentwickelte Softwaretool ist frei erhältlich und dient zum Parallelisieren von Software auf heterogenen vernetzten Systemen.

Die Idee ist, (frei nach dem Client/Server Modell) unterschiedliche Architekturen (Workstations, Superminis und Supercomputer), als einzelne virtuelle Parallelcomputer einzusetzen (siehe Abb. 2). Derartige heterogene Systeme können aus DECstations, VAXstations, SUN Sparcs, IBM RS/6000 Workstations, Sequent Symmetries, Alliant FX/8 und Cray Supercomputer bestehen. Weitere Portierungen werden in Zukunft erfolgen.

PVM besteht aus einem Daemon (pvmd) und einer Library. Parallele Anwendungen bestehen aus Library-Calls, die dann die Kommunikation zur Laufzeit über den Daemon abwickeln. Die so erstellten Programme verwenden das Message-Passing Paradigma. Selbstverständlich ist PVM auch auf dem IBM RS/6000 Workstation Cluster des EDV-Zentrums installiert.

HeNCE, ebenfalls in Tennessee entwickelt, bildet nun die graphische Ergänzung zu PVM. Damit ist es möglich, anhand von Abhängigkeitsgraphen der einzelnen Module eines Programms die notwendige Kommunikation zu erzeugen (in PVM Syntax).

HeNCE unterstützt außerdem das Erzeugen und Verteilen der ausführbaren Programmfiles. Auch die Konfigurierung der dazugehörigen Parallel Virtual Machine wird durch HeNCE graphisch unterstützt. Die dabei spezifizierte Cost-Matrix bestimmt die Verteilung der Prozesse auf die einzelnen Systeme. Danach kann, ebenfalls mittels HeNCE, PVM mit dem Anwenderprogramm gestartet werden. Während des Programmablaufs werden Events, die danach mittels einer Trace Funktion analysiert werden können, abgespeichert.

Nach Kontaktaufnahme mit Adam Beguelin und Robert Manchek konnte eine Prototyp-Version von HeNCE auf

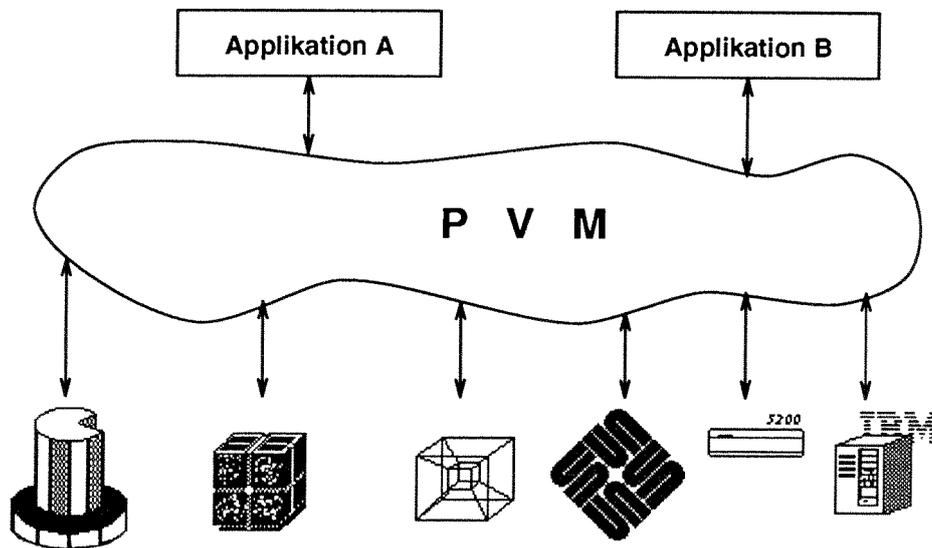


Abbildung 2

dem IBM RS/6000 Workstation Cluster des EDV-Zentrums installiert werden (zum erstenmal auf IBM RS/6000 Workstations) wodurch das EDV-Zentrum zur Beta-Site für die Entwicklung dieses Systems wurde.

TCGMSG von Robert J. Harrison (Abteilung für theoretische Chemie, Argonne National Laboratory) entwickelt, ist nahezu äquivalent zu PVM, nur, daß kein Daemon eingesetzt wird, sondern die Prozesse mittels eines Konfigurationsfiles im Vorhinein gestartet werden. Prof. Lischka vom Institut für Theoretische Chemie, Universität Wien, wird TCGMSG am Workstation Cluster einsetzen.

Ferner ist die von David Gelernter und Nick Carriero in Yale entwickelte parallele Programmiersprache Linda (als Network-Linda Version) inklusive dem Debugging Tool TupleScope installiert. Linda basiert auf dem Tuple Space, der ein assoziatives virtuelles Shared Memory darstellt. Entsprechende Operationen zum Schreiben und Lesen des Tuple Space (in, out und read) sind zur Programmiersprache C beigelegt worden.

CirL, von Cogent Research Inc., verwendet ebenfalls das Linda Sprachparadigma, bietet jedoch zusätzlich ein komplettes Prozeß-Environment an. Das bedeutet, daß ein durch CirL erzeugter Prozeß genau soviel an Environment-Information bekommt, wie es der Prozeß, der ihn generierte, bestimmt hat. CirL Prozesse kommunizieren über ein Netz von Dictionaries (äquivalent zum Tuple Space). Ein Entry-Dictionary gehört auch zum Environment. Durch entsprechende Kontakte wird der Workstation Cluster zur Beta-Test Site in Europa für RS/6000 Systeme.

PCN (Program Composition Notation) ist ein High-Level Language Ansatz von Caltech (Chandy, Taylor und Kesselmann). Es besteht große Ähnlichkeit zur Programmiersprache Strand, PCN ist jedoch frei verfügbar. Da es auch eine Version für RS/6000 Cluster gibt, ist PCN auch auf dem IBM RS/6000 Workstation Cluster des EDV-Zentrums installiert, allerdings noch nicht konfiguriert.

Anwendungen

Wie schon erwähnt, ging die Initiative zu diesem 1.6 Mio. Schilling teuren Projekt von Prof. Weinberger (Institut für

Technische Elektrochemie) aus. Daher ist sein Programm auch unter jenen, die auf dem Cluster erprobt werden sollen. Auch Prof. Schwarz (ebenfalls Institut für Technische Elektrochemie) zeigt Interesse.

Prof. Steinhauser und Prof. Lischka vom Institut für Theoretische Chemie der Universität Wien wollen ihre parallelisierten Codes zur Simulation des dynamischen Verhaltens von Molekülen auf dem Cluster ausprobieren.

Aber nicht nur Chemiker werden dieses System verwenden. Zum Beispiel möchte Doz. Faber (Institut für Kernphysik) seinen QCD Code auf den Cluster portieren und Doz. Betz (Institut für Allgemeine Physik) Simulationen an Einkristallen auf dem Cluster durchführen.

Das Institut für Theoretische Geodäsie plant eine parallele Schwerefeldmodellierung zur Reduktion von geodätischen Messungen.

Doz. Überhuber vom Institut für Numerische Mathematik schließlich wird eine Evaluierung des Clusters und der installierten Tools, sowie Algorithmen zur parallelen numerischen Quadratur entwickeln.

Alle diese Anwender können natürlich den Workstation Cluster nicht gleichzeitig zur Produktion von wissenschaftlichen Daten einsetzen. Dazu wäre es sinnvoller, wenn jeder seine eigene Workstation verwenden würde. Es ist vielmehr eine Testumgebung geschaffen worden, um solche Systeme ausprobieren zu können. Deshalb fiel die Wahl auch auf ein homogenes System, dessen CPU-Performance durchaus attraktiv ist (eine IBM RS/6000-320H leistet ca. 30 ESSL-MFLOPS).

Zur Unterstützung werden die ESSL (von IBM erweiterte Basic Linear Algebra Subroutines (BLAS)) und die Library von NAG installiert werden.

Sollten Sie, geehrter Leser, nun Interesse bekommen haben, ebenfalls gemeinsam mit dem EDV-Zentrum ein Projekt auf dem Workstation Cluster durchzuführen, dann wenden Sie sich bitte an Josef Fritscher (Klappe 5505, E-Mail: fritscher@edvz.tuwien.ac.at).

Josef Fritscher

Einbindung der VP50-EX in die verteilte Verarbeitung

Das 4-Jahreskonzept des EDV-Zentrums sieht ein hierarchisch aufgebautes Rechnernetz vor, in dem Computer verschiedener Leistungsklassen entsprechend ihrer jeweiligen Auslegung eingesetzt werden. Die Durchführung einfacher Berechnungen, die Vorbereitung aufwendiger Analysen sowie die Auswertung von Ergebnissen werden in einem solchen Konzept von kleineren Rechnern (Superminis, Workstations) übernommen, während die Durchführung numerisch aufwendiger Berechnungen am besten im Batch-Betrieb auf einem möglichst leistungsfähigen Rechner (Supercomputer mit hoher Vektor- und Skalarleistung) erfolgt.

Die VP50-EX als Compute Server

Dieses Prinzip der verteilten Verarbeitung konnte anhand einer konkreten Problemstellung am Institut für Festigkeitslehre der TU Wien (Vorstand Prof. H. Mang) unter Einbindung des Vektorhochleistungsrechners SNI VP50-EX zur Anwendung gebracht werden. Im Zuge einer wissenschaftlichen Arbeit, die umfangreiche numerische Berechnungen umfaßt, ist auch die Lösung eines großen linearen Gleichungssystems erforderlich. Die Dimension der vollbesetzten Koeffizientenmatrix liegt in der Größenordnung von 3000 bis 4000. Ausgehend von der Tatsache, daß die Lösung dieses Gleichungssystems auf dem institutseigenen HP-Minicomputer ca. 4½ Stunden Rechenzeit benötigt, sollte in Zusammenarbeit mit dem zuständigen Projektbearbeiter, Herrn DI C. Kropik, eine Möglichkeit gefunden werden, die VP50-EX für die Lösung des Gleichungssystems einzusetzen. Die dafür notwendigen Schritte sind die Berechnung der Koeffizientenmatrix am Institutsrechner, die Übertragung der Matrix und der rechten Seite vom Institutsrechner auf die VP50-EX, die Lösung des Gleichungssystems auf der VP50-EX sowie die Rückübertragung der Lösung zur Weiterverarbeitung am Institutsrechner. Diese Schritte sollen überdies ab dem Starten der Berechnung der Koeffizientenmatrix ohne Benutzereingriff ablaufen, denn nur dann ist die eingeschlagene Vorgangsweise auch sinnvoll.

Datenübertragung vom Institutsrechner zur VP50-EX

Dank der einheitlichen Kommunikationssoftware ist eine Übertragung von Dateien zwischen Rechnern mit unterschiedlichen Betriebssystemen (in diesem Fall Unix am Institutsrechner und VSP/I an der VP50-EX) ganz einfach. Die Koeffizientenmatrix wird mittels FTP auf die VP50-EX übertragen. Aufgrund der großen Datenmenge kommt dafür nur eine binäre Übertragung in Frage, was neben dem geringeren Speicherbedarf und der kürzeren Übertragungszeit auch den Vorteil der höheren Genauigkeit hat. Für eine Koeffizientenmatrix und rechte Seite der Dimension n in doppelter Genauigkeit sind dabei $8 \cdot (n^2 + n)$ Bytes zu übertragen. Die Übertragung für ein Beispiel der Dimension $n=3387$ dauert je nach Netzbelastung etwa 15 Minuten. Die auf der VP50-EX in einer binären Datei gespeicherten Daten sind dann allerdings noch im Darstellungsformat des Institutsrechners. Im konkreten Fall ist die Satzstruktur die des HP-Rechners und die Daten sind im IEEE-Format. Für die Weiterverarbeitung wurden vom EDV-Zentrum Routinen

entwickelt, die die HP-Datensätze einlesen und die Zahlen vom IEEE- ins IBM-Format der VP50-EX konvertieren.

Lösung des Gleichungssystems auf der VP50-EX

Anschließend an die Übertragung der Originaldaten in eine Datei der Anlage VP50-EX wird das Programm zum Konvertieren der Daten und zur Lösung des Gleichungssystems einfach dadurch gestartet, daß die Datei mit den dazu nötigen Steuerzeilen ebenfalls mittels FTP (diesmal im ASCII-Modus) vom Institutsrechner zur VP50-EX übertragen wird, wobei als Name für die Zieldatei INTRDR anzugeben ist. Für FTP unter VSP/I ist INTRDR (Abkürzung für "internal reader") der symbolische Name für die Batch-Eingabewarteschlange der VP50-EX. Für das konkrete Beispiel mit $n=3387$ liegt die Rechenzeit bei etwa 7 Minuten (im Vergleich zu 4½ Stunden), wobei dank dem speziell für die VP50-EX optimierten Gleichungslöser der Vektoranteil an der Gesamtrechenzeit nahezu 100 Prozent beträgt.

Rückübertragung der Lösung zum Institutsrechner

Anschließend an die Berechnung erfolgt die Rückübertragung der Lösung zum Institutsrechner ebenfalls mittels FTP, wobei natürlich je nach Wunsch auch FORTRAN- und Loader-Meldungen, wenn sie auf eine Datei geschrieben wurden, übertragen werden können. Die Übertragung wird durch Aufruf von FTP innerhalb desselben Batchjobs gestartet, der wie zuvor beschrieben in die Warteschlange INTRDR übertragen worden ist und die Berechnungen durchgeführt hat. Je nachdem, ob die Berechnung der Lösung fehlerfrei erfolgte oder irgendein Fehler auftrat, wird zum Schluß eine kurze Datei mit dem Namen OK.VP bzw. ERROR.VP zum Institutsrechner übertragen. Ein dort laufender Daemon wartet auf eine Datei mit diesen Namen und veranlaßt die automatische Weiterverarbeitung des Lösungsvektors bzw. die Verständigung des Benutzers im Fehlerfall. Die Zeit für die Rückübertragung der Meldungen und des Lösungsvektors ist aufgrund der geringen Datenmenge (Größenordnung $21 \cdot n$ Bytes, da formatierte Daten im ASCII-Modus rückübertragen werden) vernachlässigbar.

Zusammenfassung

Anhand eines konkreten Beispiels konnte gezeigt werden, daß die VP50-EX durchaus für große Berechnungen, die nur einen Teil des zu bearbeitenden Problems ausmachen, eingesetzt werden kann. Neben der schnelleren Berechnung auf der VP50-EX hat diese Art der verteilten Verarbeitung auch den Vorteil, daß am Institutsrechner die beträchtliche Rechenzeit, die sonst für die Lösung des Gleichungssystems aufgewendet werden müßte, für andere Aufgaben zur Verfügung steht. Überdies kann durch Verlagerung eines Teils der Berechnung die Bearbeitung eines Problems erst möglich werden, für das die Kapazität des Institutsrechners allein nicht ausreichend ist. Wenn auch Sie an derartigen Problemlösungen interessiert sind, so wenden Sie sich bitte an mich (Klappe 5811).

Friedrich Blöser

Wieviel Aufwand erfordert die Nutzbarmachung der Vektorhardware?

Seit der Installation des ersten Hochleistungsrechners mit Vektorarchitektur am EDV-Zentrum sehen sich die Mitarbeiter mit dem hartnäckigen Gerücht konfrontiert, daß Vektorprozessoren nur unter Aufbietung eines beträchtlichen persönlichen Aufwands des Anwenders genutzt werden können; es ist manchmal sogar vom Programmieren in Assembler die Rede. Um es gleich vorwegzunehmen: Es stimmt nicht. Beim Rechner VP50-EX kann der Vektorassembler gar nicht bestellt werden.

Wie man vorgeht, um an entsprechenden Rechnern, wie den vom EDV-Zentrum betreuten Modellen SNI VP50-EX, CONVEX C3220 und C220, die Vektorhardware ansprechen zu können, soll in diesem Artikel kurz skizziert werden.

Die Nutzung des Vektorprozessors soll kein Selbstzweck sein, sondern die Abarbeitung eines numerisch aufwendigen Algorithmus merklich beschleunigen. Nicht der Anteil der Vektorzeit zur gesamten Prozessorzeit sollte als Maß für die Sinnhaftigkeit der Vektorisierung dienen, sondern die Reduktion der Gesamtrechenzeit durch Nutzung der Vektorbefehle.

Verwendung fertiger Programme

Hat man den Lösungsansatz für ein Problem festgelegt, ist der effizienteste Weg zum Ergebnis, sich eines fertigen Programms zu bedienen, das für den ins Auge gefaßten Rechner in einer optimal angepaßten Version zur Verfügung steht. Solche Programme gibt es für eine Vielzahl von Anwendungsgebieten: Strukturmechanik und Strömungsdynamik (Finite Elemente-Programme wie ABAQUS, ADINA, FIDAP, NISA2 und ähnliche), theoretische Chemie (z.B.: Molekülberechnungen), Lösung von elliptischen und parabolischen Differentialgleichungen und andere.

Um zu klären, welche Programme bereits installiert sind und welche beschafft werden können, wenden Sie sich bitte an die Mitarbeiter der Abteilung Hochleistungsrechnen.

Einbindung vorhandener vektorisierter Unterprogramme

Wenn man ein Programm selbst erstellen muß, sollte man immer nur jene Teile selbst kodieren, für die es keine geeigneten, fertigen Unterprogramme gibt. Im allgemeinen enthalten die Unterprogrammbibliotheken der Hardware-Hersteller und der renommierten Software-Entwickler (wie NAG, IMSL und einige Universitäten) Routinen, die algorithmisch stabile und leistungsmäßig sehr effiziente Implementierungen darstellen, deren Qualität in den dabei abgedeckten Applikationsbereichen von Eigenentwicklungen selten übertroffen werden.

Insbesondere enthalten die kritischen Programmteile Entscheidungsanweisungen, die verhindern, daß die Vektorbefehle auch dann ausgeführt werden, wenn ein äquivalenter skalarer Code schneller abgearbeitet wird (z.B. bei zu kleinen Vektorlängen).

Die zweiteinfachste Methode ist also, für ein Programm nur den notwendigen Steuerteil, die Ein- und Ausgabe und den Aufruf eines Unterprogramms zur Durchführung des Algorithmus zu schreiben. Diese Vorgangsweise ist nicht nur theoretisch von Interesse, sondern erweist sich in nicht wenigen Fällen als sehr praktikabel: das Programm, von dem im Artikel über den Einsatz des Rechners VP50-EX in der verteilten Datenverarbeitung die Rede ist, hat so eine Struktur.

Das Vektorisieren eigener Programme

Bei der Verwendung eigener Programme oder dem Erstellen neuer Programme, deren Rechenteil (teilweise) nicht durch vorhandene Unterprogramme abgedeckt wird, findet die Vektorisierung im Zuge der Übersetzung durch den Compiler statt. Dabei werden Schleifen, in denen, abhängig von den Laufvariablen, Elemente von ein- oder mehrdimensionalen Feldern verarbeitet oder berechnet werden, auf eine Folge von Vektorbefehlen abgebildet. Nicht alle Schleifen sind (voll) vektorisierbar; es gelten einige Bedingungen, die anzuführen hier nicht genug Platz ist. Welche Konstrukte in Vektorbefehle umgesetzt werden können, ist Inhalt des Vektorisierungskurses des EDV-Zentrums, kann aber auch in den Vektorisierungshandbüchern der Computerhersteller nachgelesen werden.

Der erste Schritt ist also das Compilieren des FORTRAN-Programms, wobei die Vektoroption eingeschaltet ist. Gleichzeitig sollte man sich vom Compiler auch die maximal möglichen Vektorisierungsmeldungen auf der Liste ausgeben lassen. Im allgemeinen wird nur ein Teil der Schleifen vektorisiert und manche ineinandergeschachtelte Schleifen werden vielleicht nach einer anderen Laufvariablen vektorisiert, als man es sich vorgestellt hat. Die begleitenden Meldungen geben den Grund für dieses Verhalten an und oft finden sich Hinweise auf Verbesserungen.

Bevor man sich jedoch genauer mit den Listen auseinandersetzt, sollte man feststellen, in welchen Programmteilen bei skalarer Verarbeitung die meiste Rechenzeit verbraucht wird. Dazu gibt es Unterstützungsprogramme (FORTUNE auf VP50-EX und CXpa auf CONVEX), die als Zeitnehmer fungieren und die Zeitverteilung ausgeben, ohne daß dabei der Quelltext des eigentlichen Programms verändert wird. Die Zeitnehmung verursacht allerdings zusätzlichen Verarbeitungsaufwand, wodurch sich die Rechenzeiten verlängern; aber die Verhältnisse stimmen bis auf geringe Abweichungen.

Bei einem ersten Durchlauf mit einem Datensatz, der so repräsentativ wie möglich sein soll, beschränkt man sich auf die Verteilung der Rechenzeit auf die einzelnen Unterprogramme. In einem zweiten Lauf wird die Aufteilung der Rechenzeit auf die Befehle der wenigen - im Idealfall einer - Routinen, die den größten Teil der Verarbeitungszeit verbrauchen, aufgezeichnet. Programmteile (Einzelbefehle,

Schleifen oder Programme), die nur für wenige Prozent der Rechenzeit verantwortlich sind, bleiben beim weiteren Vorgehen unberücksichtigt, da ihre Beschleunigung sich kaum auf die Verringerung der Gesamtrechenzeit auswirkt. Anhand der Zeitverteilung erkennt man, in welchen Bereichen eine Optimierung des Gesamtprogramms durchgeführt werden soll.

Bisher wurden nur Schritte beschrieben, die ohnehin bei der Programmierung auch skalarer Rechner durchgeführt werden sollten, aber oft nicht geschehen. Bei der Vektorisierung kommt nur dazu, daß die Optimierung auch darin besteht, so effizient wie möglich die höhere Verarbeitungsgeschwindigkeit der Vektorhardware einzusetzen.

Nachdem nun bekannt ist, an welchen Stellen die Optimierung lohnt, benutzt man die entsprechenden Bereiche der Compiler-Liste und die dazugehörigen Meldungen, um Verbesserungen anzubringen.

Einige davon bestehen im Einfügen von Compiler-Direktiven in das Quellprogramm, um dem Compiler Eigenschaften des Programms mitzuteilen, die aus der statischen Analyse des Programmtexts nicht ersichtlich sind - oder nur aufgrund von Beschränkungen in der Logik des Compilers nicht erkannt werden: Anzahl der Schleifendurchläufe bei variablen unteren oder oberen Grenzen, Wahrscheinlichkeiten, daß die Bedingung einer IF-Anweisung wahr ist, kein tatsächliches Auftreten von Regressionen (auch Rekursionen genannt), auch wenn der Programmtext dies zuließe usw.

Manche Verbesserungen bestehen auch im Umstellen von Codeteilen, ohne daß sich die Gesamtlogik des Programms ändert. Oft erweisen sich solche Umbauten auch bei skalarer Abarbeitung als günstiger, da dadurch nicht selten auch andere Optimierungsmöglichkeiten des Compilers ermöglicht werden: Die Vektorisierung ist oft nur eine Folge günstig angelegter Programmstrukturen, was immer zu besseren Maschinenprogrammen führt.

Wenn nach dieser ersten Iteration die vektorisierten Programmteile zwar beschleunigt wurden, aber noch immer die meiste Rechenzeit verbrauchen, kann mit der Optimierung aufgehört werden, da die nicht behandelten Teile ohnehin wenig zum Gesamtverbrauch beitragen. Wenn aber jetzt andere Bereiche des Programms die meiste Rechenzeit verbrauchen, so sind diese zu untersuchen, bis eine stabile Rechenzeitverteilung erreicht wird. Wie weit man dem Op-

timium zustrebt, hängt nur vom noch zu erwartenden Gewinn und dem voraussichtlichen Aufwand ab.

Am Rechner VP50-EX gibt es noch ein weiteres, interaktives Unterstützungsprogramm, VECTUNE, das aufgrund der Analyse von FORTUNE und einer statischen Analyse des Programmtexts Vorschläge zur Optimierung in Richtung Vektorisierung macht. Falls man keinen FORTUNE-Lauf machen kann (z.B. weil noch keine sinnvollen Eingabedaten zur Verfügung stehen oder das Programm noch nicht vollständig ist), versucht VECTUNE, nur anhand der Untersuchung des FORTRAN-Codes Vorschläge für Verbesserungen zu machen.

Die Verwendung der vektorisierenden Compiler und der anderen Vektorisierungshilfen werden gemeinsam mit den Regeln für vektorisierbare Schleifen im Vektorisierungskurs des EDV-Zentrums besprochen.

Neue Algorithmen zur effizienten Nutzung von Vektorprozessoren

Die bisher beschriebene Vorgangsweise führt jedoch zu keinem Erfolg, wenn sich die programmierten Datenstrukturen nicht für eine Vektorisierung eignen (z.B. WERT1, WERT2 bis WERT10 statt DIMENSION WERT(10)). Solche Programme mit schlecht strukturierten Daten, wie sie leider auch heute noch häufig vorhanden sind, sollten im Interesse einer besseren Wartbarkeit und zum Teil auch effizienteren skalaren Umsetzbarkeit schleunigst umgeschrieben werden.

Manche Algorithmen eignen sich jedoch prinzipiell nicht für eine Vektorisierung. Für eine große Anzahl von Anwendungsgebieten wurden aber neue Ansätze entwickelt, die Datenstrukturen aufweisen, die gut vektorisierbare Programme ermöglichen. Die Suche nach vorhandenen oder die Entwicklung neuer Algorithmen ist sicher am aufwendigsten, aber auch am lohnendsten. Eine eingehende Erörterung würde jedoch den Rahmen dieses Artikels sprengen. Bei entsprechender Beteiligung würde die Abteilung Hochleistungsrechnen die Einrichtung eines Arbeitskreises "Vektoralgorithmen" einrichten und fördern.

Willy Weisz

Simulation

Im folgenden soll ein kurzer Überblick über die Aktivitäten des EDV-Zentrums auf dem Gebiet der Simulation gegeben werden.

Software

Das EDV-Zentrum besitzt Lizenzen, Campuslizenzen und Demo-Versionen von Simulationssoftware für verschiedene Rechnerarten.

ACSL (Advanced Continuous Simulation Language) ist die an der TU Wien am meisten verwendete Sprache zur Simulation und Analyse kontinuierlicher Systeme. Neben dem Einsatz in der Lehre und Ausbildung (z.B. mit ACSL ausgestattete PC-Räume) wird ACSL an TU-Instituten z.B. zur Ventiltriebssimulation, Optimierung von Regelungssystemen, Schaltersimulation, Robotersimulation, Simulation von biologischen Systemen und Kompartmentsystemen, von neuropharmakologischen Prozessen, von elektrischen Antrieben, von instationären dynamischen Problemen, von nichtlinearen Schwingungen etc. verwendet.

Die meisten Anwender arbeiten mit der PC-Version (das EDV-Zentrum besitzt eine Campus-Lizenz für die TU Wien), bisher Level 9. Eben hat das EDV-Zentrum das Update für Level 10 im Rahmen des Software-Pools erworben. Level 10 läuft unter Windows und verwendet MS FORTRAN 5.1. Einige der Level 10 Features sind: keine Beschränkung in der Modellgröße, Double Precision Arithmetik, IF THEN ELSE, Verbesserungen im Run-Time-

Interpreter (Abkürzungen für Befehle, Plotten während des Simulationslaufs etc.).

Benutzer an der TU Wien, die bereits eine ACSL/PC Lizenz vom EDV-Zentrum bezogen haben, können das Level 10 Update um S 300.- erwerben. Das Manual kostet S 200.-. Neue Lizenzen sind zum Preis von S 600.- erhältlich. Auch die Nicht-Windows-Version (Level 9, Update 9A1/9C1) ist noch erhältlich.

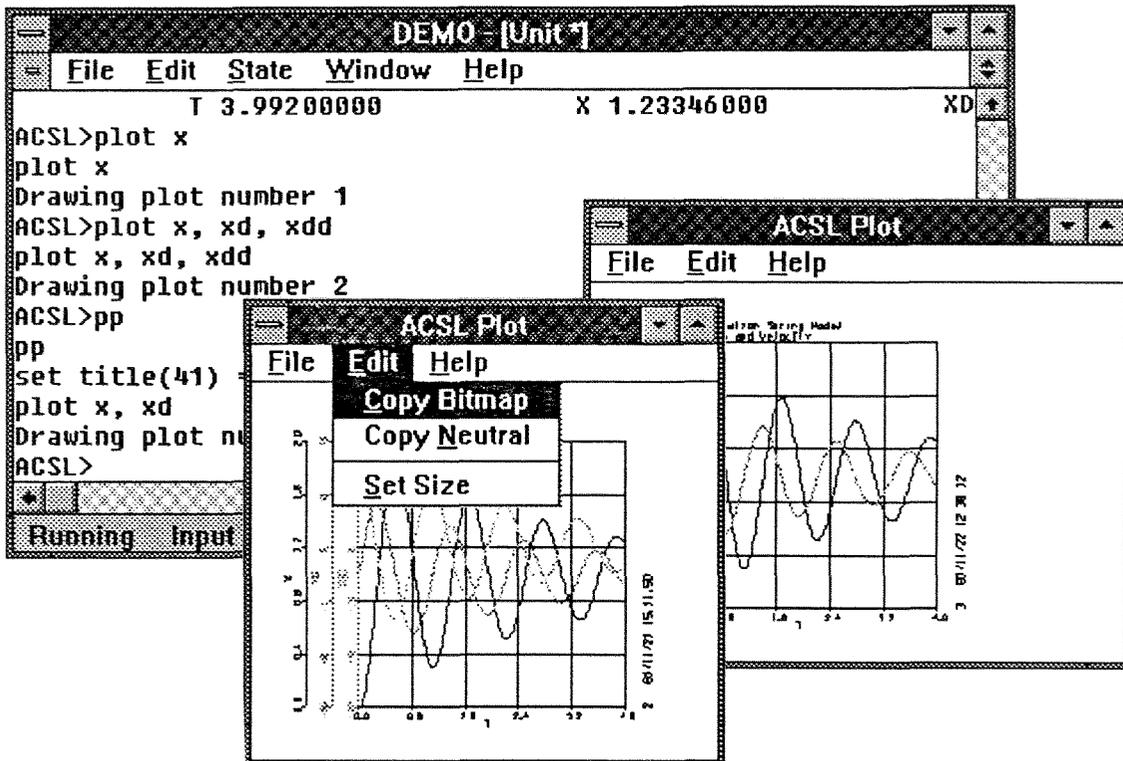
Zum Einsatz in der Ausbildung ist das ACSL/PC Classroom Kit gedacht (kostenlos, beschränkte Variablenanzahl).

Für Anwendungen im Workstation-Bereich existiert eine Network-Lizenz für UNIX-Workstations (DECstations und HP-Stations).

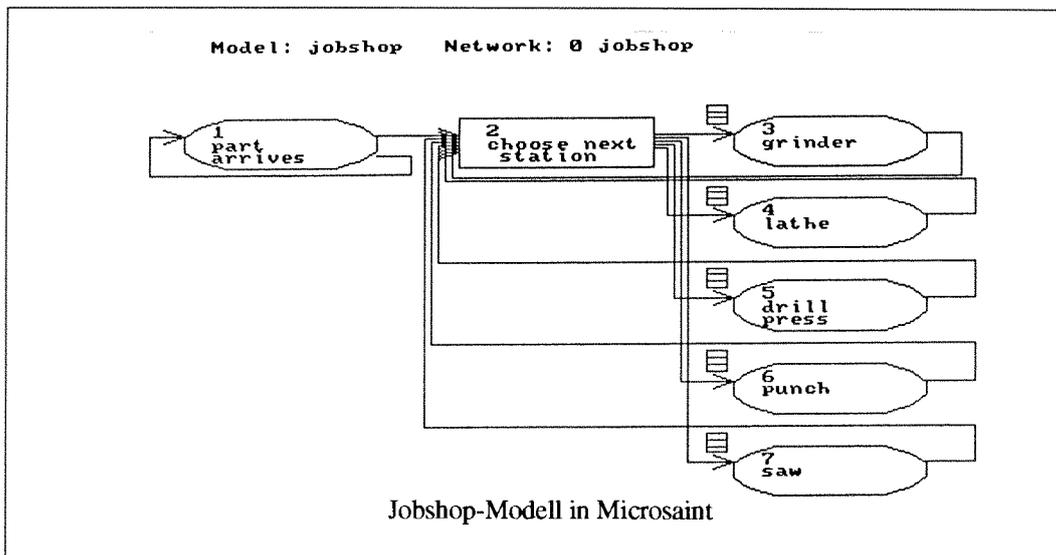
Für große rechenintensive Modelle wurde ACSL für die Fachbereichsrechner Physik und Maschinenbau bestellt (Installation im Frühjahr).

Die deutschsprachige *ACSL User Group*, die von der TU Wien aus organisiert wird, bietet regelmäßigen Informationsaustausch durch ein Mitteilungsblatt und durch Veranstaltungen. Das nächste Treffen der User Group findet am 9. März in Paderborn statt.

Für Benutzer der ACSL-Lizenzen des EDV-Zentrums bietet das EDV-Zentrum regelmäßige Information über User Reports, Firmenneuigkeiten, Hilfestellung bei auftretenden Problemen, Installationsberatung.



ACSL for Windows



Microsaint ist eine diskrete prozeßorientierte Simulationssprache mit halbgraphischer Oberfläche und halbgraphischer Animation. Das Classroom Kit für PC (Anzahl der Blöcke beschränkt) kann kostenlos an interessierte Institute weitergegeben werden.

Auf dem Gebiet der diskreten Simulation ist **SIMAN**, mit der **CINEMA** Animationssoftware, sehr verbreitet. Das EDV-Zentrum hat für das Professor's Package für PC ein Update auf SIMAN IV bestellt. Das Laboratory Package (Studentenversion) ist an der Abt. Simulationstechnik vorhanden.

Eine Lizenz für **PC SIMSCRIPT II.5** (eine allgemeine Simulationssprache für diskrete und kontinuierliche Simulation mit eingebauter Graphik) ist vorhanden, jedoch zur Zeit nicht installiert.

Auskünfte über Lizenzen und Demo-Versionen gibt Frau Husinsky (Klappe 5484).

Simulationsseminare

Auf dem Gebiet der Simulation arbeitet das EDV-Zentrum eng mit Dr. Breitenacker von der Abt. Simulationstechnik der TU zusammen. Die Abt. Simulationstechnik bietet Spezialvorlesungen und Problemlberatung bei der Modellbildung und der Auswahl von Simulationssoftware an.

Das EDV-Zentrum veranstaltet gemeinsam mit der Abteilung Simulationstechnik eine Reihe von Simulationsseminaren. Es sind dies ganztägige Veranstaltungen, bei denen jeweils eine Firma eine Produktgruppe vorstellt und Anwender über ihre Erfahrungen berichten.

1991 haben drei Seminare, mit zum Teil internationalen Vortragenden, großes Interesse gefunden:

Seminar unter Mitwirkung der Firmen Rapid Data Ltd. und Mitchell & Gauthier Ass. über die Produkte ACSL, PROBLOCK, OPTDES, PRO-MATLAB und SIMUSOLV.

Seminar unter Mitwirkung von Scientific Computers GmbH über CTRL-C, Model-C und XANALOG.

Seminar unter Mitwirkung von SIMUTECH über SIMUL_R, PROSIMUL_R und SIMUL_TR.

Als nächste sind Seminare über MATLAB, SIMULAB und SPICE sowie über GPSS/H und Petrinetzsimulatoren geplant.

Im Rahmen dieser Seminare erhielten die Veranstalter eine Reihe von **Studenten- und Demo-Versionen** verschiedener Simulationssoftware:

kontinuierliche Simulation:

SIMUL_R	compilerorientierte Sprache, für PC und Workstation
XANALOG	blockorientiert, für PC, mit graphischer Oberfläche
CTRL-C, Model-C	blockorientiert, für Workstation, mit graphischer Oberfläche
MATLAB, SIMULAB	mathematisch orientiert, für PC

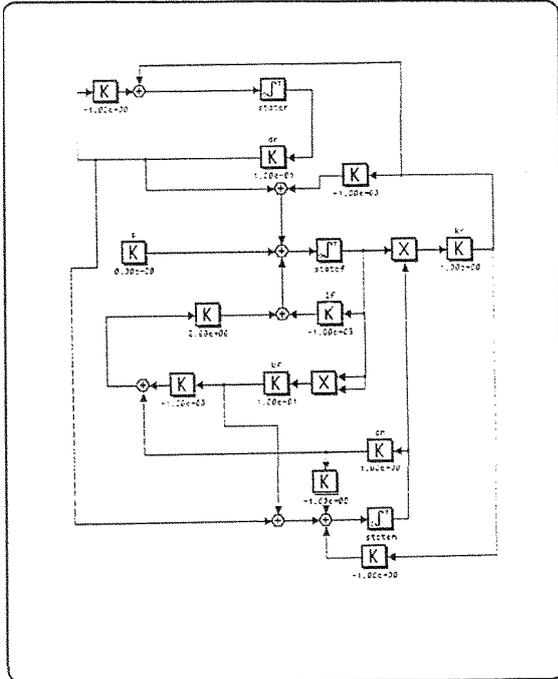
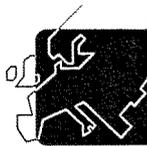
diskrete Simulation:

PROSIMUL_R	diskreter Part zu SIMUL_R, für PC und Workstation
GPSS/H	sehr verbreitete universelle Sprache, mit Proof Animation, für PC
PROMODEL	Demo einer in den USA sehr verbreiteten Sprache

Europäische Kontakte

In Europa sind seit vielen Jahren nationale und regionale Simulationsgesellschaften aktiv. Sie widmen sich der Förderung und Weiterentwicklung von Modellbildung und Simulation in Grundlagen und Anwendung sowie der Verbesserung der Kommunikation zwischen Theoretikern und Praktikern.

ASIM ist die deutschsprachige Simulationsvereinigung, sie umfaßt Anwender in Deutschland, Schweiz und Österreich. Das EDV-Zentrum war lange Zeit durch Dr. Kleinert im ASIM-Vorstand vertreten. Zur Zeit ist Frau Husinsky Vorstandsmitglied. Im Jahre 1990 wurde an der TU Wien die ASIM-Jahrestagung veranstaltet.



EUROSIM ist der Dachverband aller europäischen Simulationsgesellschaften. EUROSIM gibt eine Mitgliederzeitschrift heraus, *EUROSIM - Simulation News Europe*, sie erscheint dreimal jährlich. Die Redaktion dieser Zeitschrift liegt in den Händen von Dr. Breitenecker und Frau Husinsky. Besonders erfreulich ist, daß seit der Öffnung des ehemaligen Ostblocks über diese Zeitschrift auch rege Kontakte mit diesen Ländern Mitteleuropas möglich sind. Neben aktuellen Nachrichten der einzelnen Simulationsgesellschaften, Grundsatzartikeln, Industry News, Vorstellungen von Simulationsrechenzentren etc. werden in jedem Heft eine Reihe von Simulationswerkzeugen im Vergleich vorgestellt. Von den Editoren vorgegebene Beispiele (Software Comparison) werden von Lesern gelöst. Die Beiträge kommen von Benutzern aus ganz Europa und zeigen Vor- und Nachteile der verschiedenen Simulationssprachen (kontinuierlich und diskret). Demnächst ist auch ein Vergleich der skalaren, vektorisierten oder parallelisierten Performance von Simulationsläufen geplant.

Abbau des EAI SIMSTAR

Abschließend soll noch erwähnt werden, daß der Simulationsmultiprozessor EAI SIMSTAR, der 1984 installiert wurde, mit Beginn des Jahres 1992 außer Betrieb genommen wurde. Er wurde in letzter Zeit nicht mehr verwendet.

Irmgard Husinsky

ANZEIGE

Software Engineering der 90er Jahre objektorientiert ?

„Language is the dress of thought.“
Johnson (*Lives of the English Poets*)

Einleitende Betrachtungen

Das Aufkommen leistungsfähiger Hardware, insbesondere im Laufe der letzten Jahre, wird zunehmend von einer allmählichen, schwerpunktmäßigen Verlagerung der Interessen in Richtung Software-Konstruktion begleitet.

Diese generelle Entwicklung ist nicht überraschend. Die hardwaremäßige Leistungssteigerung bewegt sich seit geraumer Zeit immer noch im Bereich von Zehnerpotenzen, die softwaremäßigen Steigerungszahlen hingegen sind beinahe vernachlässigbar. Im Mittelpunkt des Interesses stehen aber nützliche, im allgemeinen aus mehreren Komponenten¹ bestehende Informationssysteme. Sowohl die qualitativen, als auch die quantitativen Forderungen an derzeitige Informationssysteme sind, wie man sich leicht überzeugen kann, beachtlich gestiegen. Dabei ist die mit den Forderungen eng zusammenhängende **Systemkomplexität**, vor allem aus *ökonomischen* Gründen, in den flexiblen Systemteil, in die Software, verlagert.

Software-Systeme nach *ökonomischen Grundsätzen* zu konstruieren, die den geforderten, vielfältigen Qualitätsforderungen der Funktionsangemessenheit, der Zuverlässigkeit, der Robustheit, der Betriebsökonomie, der Betriebssicherheit u.a.m. genügen müssen, scheint gegenwärtig zu einem zentralen Thema der kommenden Jahre zu werden.

Ein Minirückblick

Die Multidimensionalität der Konstruktion von Software wurde mit aller Deutlichkeit Ende der 60er Jahre erkannt. Die natürliche Reaktion auf diese seither datierte „Software-Krise“ war, nach systematischen Wegen zu suchen, um diese Tätigkeit in einen überschaubaren und kontrollierbaren Rahmen zu bringen.

Ausgegangen wurde von der Erkenntnis, durch umfassende Methodik, die später durch adäquate Werkzeugunterstützung ergänzt werden sollte, den Software-Erstellungsprozeß in den Griff zu bekommen. Im Laufe der Zeit kam es dabei zur Verlagerung der Schwerpunkte von der Programmierungstätigkeit hin zum architektonischen Design und später zur Problemstellungs-Spezifikation. Im Mittelpunkt dieser methodischen Ansätze stand das **funktions-**, später dann das **datenausgerichtete Organisationsprinzip** („Orientierungsprinzip“). Diese zwei Software Engineering-Prinzipien stellten bisher die meist angewandte Art der Fokussierung bei der architektonischen Gestaltung der Software-Systeme dar.

Mit weiterer Steigerung der Problemstellungskomplexität, insbesondere im Blickwinkel der sich verschärfenden ökonomischen Wettbewerbsbedingungen, scheinen diese beiden Orientierungsprinzipien einer zunehmenden Kritik ausgesetzt zu sein.

Software-Pflege: „Sorgenkind“ des Software Engineering

Was sind die *Motivationsgründe*, die den Bestrebungen nach alternativen Paradigmata auf dem Software Engineering-Gebiet zugrunde liegen ?

Eines der größten Probleme des Software Engineering liegt im Bereich der **Software-Pflege** (Software-Wartung). Unter Software-Pflege werden Maßnahmen verstanden, die ergriffen werden müssen, um ein erstelltes Software-Produkt innerhalb eines gedachten Zeitrahmens sinnvoll nutzen zu können.

Die Ursachen für diese unerfreuliche Situation sind vielfältig². Zum einen wird häufig, vor allem aus verkaufspolitischen Gründen, eine möglichst rasche Produkt-Erstellung betrieben, ohne sich viel den Kopf darüber zu zerbrechen, was mit dem Produkt in weiterer Zukunft geschieht. Zum anderen lassen sich eventuelle, mit der Produktbenutzung neu gewonnene Erfahrungen und Erkenntnisse a priori nicht voraussehen, die dann nachträglich, in Form unterschiedlicher Anpassungsforderungen und Wünsche während der Pflegephase berücksichtigt werden sollen. Hinzu kommt noch die Tatsache, daß die Erstellung eines **fehlerfreien**, komplexen Software-Systems aufgrund derzeit spärlich bekannter, und noch spärlicher angewandter, im allgemeinen unzureichender Prüfverfahren im Bereich der Illusion liegt.

Betrachtet man nun den auf die Existenz von Software bezogenen Gesamtaufwand, fällt einer derzeit geltenden Erfahrungsregel zufolge

1/3 Erstellung + 2/3 Pflege,

d.h. ein Drittel für die Software-Erstellung und zwei Drittel für die Software-Pflege zu. Anhand dieser Beziehung ist leicht zu erklären, warum die Nachfrage nach neuartigen Software-Systemen trotz des rasanten Entwicklungsfortschrittes dem derzeitigen Angebot davonläuft. Infolge der massiven Bindung von Ressourcen im Pflegebereich gibt es nämlich kaum freie Ressourcen für die Konstruktion von *novitätsartigen* Systemen.

Es wird deshalb intensiv nach Wegen gesucht, durch geeignete Maßnahmen die derzeit herrschenden Verhältnisse zumindest im Sinne 2/3 Software-Erstellung + 1/3 Software-Pflege zu beeinflussen,

¹ Zu diesen zählt u.a. Hardware und Software.

² Erfahrungsgemäß wird der Pflegeaufwand wie folgt aufgeschlüsselt: 50% fortsetzende Entwicklung, 25% technologische Anpassung, 21% Fehler-Korrektur, 4% sonstiges.

Objektorientierung : neue Hoffnung für die 90er ?

Ausgehend von den bisher gewonnenen Erfahrungen mit beiden bereits erwähnten Orientierungsprinzipien der **gleichzeitigen** Betrachtung von Datenentitäten sowie den auf diesen operierenden Funktionen scheint es einen besseren Weg zu geben.

Ein besonderer Vorzug einer neuartigen organisatorischen Fassung von *Daten (Zustand) + zugehörigen Funktionen (Verhalten)* zu sogenannten **Objekten** (abstrakten Datentypen, ADT's) liegt in der Kapselung *abstrakter, genau abgegrenzter Sachverhalte*. Ein solches (Organisations-) Kompositionsprinzip führt zur Konstruktion von weitgehend unabhängigen Objekten **spezifischer, qualitativ angebar** Beschaffenheit.

Daraus lassen sich einige für das Software Engineering wesentliche Vorteile ableiten. Zum einen haben wir vom Standpunkt des System Engineering mit klar abgegrenzten abstrakten Entitäten (Systemobjekten) zu tun, was zur *Systemüberschaubarkeit* wesentlich beiträgt³. Zum anderen ist es die potentielle Möglichkeit der mehrfachen Verwendung (reuse) einmal konstruierter Objekte, die vom **ökonomischen Standpunkt** aus von einer enormen Bedeutung ist. Diese **Wiederverwendbarkeit** von Objekten führt nicht nur zur **Aufwandsreduktion** bei der Konstruktion eines Software-Systems, sondern aufgrund der Verwendung einer bereits durch mehrmaligen Einsatz erprobten, hochwertigen Menge von *Elementarbausteinen* zu einer **Fehlerquellen-Reduktion**. Dies führt wiederum zu einer signifikanten **Qualitätssteigerung** der konstruierten Software-Systeme, die aufgrund des zunehmenden Einsatzes in relevanten, gesellschaftskritischen Bereichen des täglichen Lebens (Transportwesen, Energieversorgung, u.a.m.) rapide in den Mittelpunkt des öffentlichen Interesses rückt.

Die kurz angedeuteten Vorteile sind zum einen nicht zum „Nulltarif“ zu bekommen, zum anderen können bei genauer Betrachtung der objektorientierten Sachverhalte unterschiedliche Inkonsistenzen im Ansatz sowie „Schattenseiten“ entdeckt werden. Der Versuch scheint daher gerechtfertigt, einige der Vor- und Nachteile des objektorientierten Ansatzes näher unter die „Lupe“ zu nehmen.

„Objektorientierte“ Euphorie und ihre Gründe

Was sind nun die Gründe, die dazu beitragen, die Objektorientierung als den neuen, universellen Lösungsansatz des Software Engineering anzusehen ?

Im Marketingbereich scheinen unterschiedliche, mit der Objektorientierung zusammenhängende Begriffe sich als ein Synonym für „gut, besser, zeitgemäß“ etabliert zu haben. Über den durchschlagenden Erfolg des objektorientierten Software Engineering gibt es im Augenblick allerdings nur spärlich veröffentlichtes Material.

Die Objekt-Begriffswelt befindet sich derzeit in einer Phase der Präzisierung, der (Re-)Definition von Konzepten, der forschungsmäßigen Auseinandersetzung. Objektklassen, Metaklassen, Instanzen, Vererbung, Methoden, Messagen, Polymorphismus, statisches, dynamisches Binden u.v.a.m. sind Begriffe, mit denen häufig und gerne operiert wird (siehe auch Literaturhinweise).

Im Zusammenhang mit objektorientiertem Software Engineering werden häufig folgende Gesichtspunkte hervorgehoben, die für die Konstruktion von Software zweifellos von enormer Bedeutung sind (Aufstellung unvollständig):

Wiederverwendbarkeit:

Einmal erstellte Objekte lassen sich für die Konstruktion ähnlicher Systeme *ohne Modifikation* wiederverwenden („SW-IC's“). Dieser Gesichtspunkt ist von besonderer ökonomischer Bedeutung.

Modularisierung:

Die Beherrschung des steigenden Komplexitätsgrades von Problemstellungen zwingt zur Anwendung komplexitätsreduzierender Maßnahmen, zu denen eine sinnvolle, u.a. durch das Software-Design stark beeinflusste **Untergliederung** der logischen Sachverhalte gehört. Dadurch wird infolge einer eindeutigen Spezifikation jedem Objekt eine klare Zuständigkeit (zu erbringende Dienstleistung) zugeordnet.

Produzent/Konsument-Architektur:

Durch diese Architektur wird der System-Überschaubarkeit, der System-Skalierbarkeit, technologischen Anpaßbarkeit, der Robustheit, der Ökonomie der Ressourcennutzung, der Performance u.a.m. in besonderem Maße Rechnung getragen.

Systemtransformation:

Ein objektorientiertes *Problemstellungs-Modell* läßt sich „nahtlos“ unter Einhaltung der **Systemisomorphie** (Problemstellungstreue) über alle nötigen Zwischenstufen bis zum *Software-System-Modell* (Programm) „hinunter“ transformieren. Die Systemtransformation im Falle einer „klassischen“ Orientierung (siehe oben) ist nicht nahtlos und bereitet oft Probleme⁴.

Projektmanagement:

Projektteamarbeit läßt sich nach den zu erstellenden Teilobjekten überschaubarer organisieren, überblicken, kontrollieren, prüfen.

Software-Wiederverwendbarkeit: Thesen und Praxis

Die primär ökonomisch ausgerichtete Motivation der Wiederverwendbarkeit von Software-Komponenten (Objekten) ist innerhalb des objektorientierten Ansatzes von einer zentralen Bedeutung. Es erscheint daher angebracht, noch weitere Überlegungen bezüglich der Grundthesen anzustellen, auf denen der Wiederverwendbarkeits-Ansatz beruht.

- a) Es gibt (ökonomisch interessante) Anwendungsgebiete, wo es möglich ist, eine wohldefinierte (am besten dis-

³ Dies ist eine überaus wichtige Systemforderung; man denke nur an die steigende Systemkomplexität.

⁴ Der Übergang Problemstellung Software-System->Architektur.

junkte) Menge von Elementarobjekten zu spezifizieren (sogenannte Applikations-Frameworks), aus denen sich unterschiedliche Applikationen einer Domäne konstruieren lassen.

- b) Die Elementarobjekte sind als standardisierte Bausteine zu verwenden, d.h. ohne interne (noch so geringe) Eingriffe. Anderenfalls hat man sich sofort mit den „alten“ „Pflege“-Problemen herumzuschlagen, die gerade vermieden werden sollten.
- c) Objekte können qualitativ, quantitativ beschrieben werden und als solche katalogisiert werden, so daß sie im Bedarfsfall (leicht) gefunden werden können.
- d) Objekte können nach dem Baukastenprinzip miteinander problemlos kombiniert werden („SW-Lego“⁵).

Wie ist es nun um die Software-Wiederverwendbarkeit in der Praxis bestellt? Dazu möge im folgenden jede der angeführten Thesen mit einigen Bemerkungen bedacht werden.

Zu a): Schaffung von Frameworks

Zu einer derartigen Anwendungsdomäne, wo die angeführte These zuzutreffen scheint, gehört der Bereich der Mensch - Maschine - Kommunikation (HCI, Applikations-schnittstellen, Oberflächen). Im übrigen gibt es gegenwärtig aber wenig ähnlich ausgeprägte Domänen, die sich in dem angedeuteten Sinn **systematisieren** lassen. Der durchschlagende Erfolg des objektorientierten Paradigmas wird vermutlich entscheidend davon abhängen, inwiefern es gelingt, **standardisierte, domänenspezifische**⁶ nichttriviale Frameworks (Objektbibliotheken) **höherer Abstraktionsebenen** zu schaffen, die von breitem Publikum (!) akzeptiert werden. Im Augenblick zeigt sich jedenfalls mit zunehmender Deutlichkeit, daß es zu den schwierigsten Aufgaben gehört, ein umfassendes, domänenspezifisches Applikations-Framework gewünschter Generalität (und Qualität !) überhaupt zu spezifizieren. Die Praxis brachte ferner die Erfahrung mit sich, daß nur in seltenen Fällen Objekte, die während eines konkreten Projektes erstellt wurden und als „Abfallprodukt“ zugunsten der Wiederverwendung in anderen Projekten eingesetzt werden sollten, auch tatsächlich ohne einen Zusatzaufwand eingesetzt werden konnten. Im Namen der Wiederverwendbarkeit mußten vielmehr künftige Vorhaben berücksichtigt und nicht vernachlässigbare finanzielle Mittel zusätzlich aufgebracht werden.

Zu b): Standardisierung von Frameworks

Die Notwendigkeit der Standardisierung von Applikations-Frameworks wurde von der Datenverarbeitungs-Industrie erkannt, liegt aber im wesentlichen noch in Form einer Idee vor.

Ein beachtliches Unbehagen tritt in dem Augenblick auf, wenn auf der einen Seite versucht wird, Objektklassen zum Zweck einer problemlosen Pflege derart auszulegen, daß keine Modifikation notwendig ist (eine dabei auftretende unangenehme Begleitfolge: der Anstieg der Klassenanzahl).

Auf der anderen Seite zeigt sich in der Praxis, Objektklassen infolge stets spezifischer, vom Standardfall in der Regel abweichender Bedürfnisse, zumindest im kleinen Umfang modifizieren⁷ zu müssen.

Führt man den Gedanken der „geringen Modifikationen“ weiter, indem man von einer „gering“ zu modifizierenden Menge von Objektklassen in der Größenordnung von 10^4 ausgeht (z.B. im Fall eines *realen Projektes*), so haben wir die Wunde der alten „Wartungsprobleme“ wieder „aufgerissen“.

Zu c): Framework - Verwaltung

Die qualitative und quantitative Beurteilung sowie Beschreibung der Beschaffenheit von Framework-Komponenten, die natürlich von der Verfügbarkeit einschlägiger Methoden abhängt, stellt aufgrund der derzeit sehr unbefriedigenden Situation auf diesem Gebiet keinesfalls eine leicht zu erledigende Aufgabe dar.

Bezüglich der Framework-Verwaltung können in der Praxis Schwierigkeiten auftreten, die nicht untergeordneter Bedeutung sind. Die Verwaltung kleinerer Frameworks (grober Richtwert : einige Hundert Objektklassen) bereitet unter bestimmten *Voraussetzungen* relativ wenig Probleme.

Die Schwierigkeiten der geeigneten Organisation der Komponenten - Katalogisierung im Sinne einer leichten Wiederverwendung treten massiv auf, wenn es sich um umfangreiche Frameworks handelt. Liegt die Anzahl der verfügbaren Komponenten im Bereich 10^4 bis 10^5 , so ist es keineswegs ersichtlich, nach welchen Gesichtspunkten nun eine geeignete Objektklasse in dem „Sandhaufen“ lokalisiert werden soll. Die Problematik verschärft sich um einiges, wenn zwischen den Objektklassen infolge der oft gepriesenen, während des **Software-Designs** herangezogenen **Vererbung** multiple Abhängigkeiten bestehen.

Es gibt genügend Forschungsprojekte, die darauf abzielen, zu dieser Problematik eine befriedigende Lösung zu finden.

Zu d): Zusammensetzung von Bausteinen

Die problemlose Zusammensetzung hängt primär von der „cleveren“ Spezifikation der Baustein-Schnittstellen (domänenabhängig, Forderung: „nicht zu viel, nicht zu wenig“) sowie ihrer „Einfrierung“ durch entsprechende Standardisierung (siehe auch oben) ab.

Objektorientierung, Distribution und Parallelismus

Die Preisentwicklung der letzten Jahre auf dem Datenverarbeitungs-Sektor (CPU's, Speicher, ...) hat dazu wesentlich beigetragen, daß wir im Laufe der 90er Jahre zunehmend mit der ökonomischen Nutzung verteilter Ressourcen zu tun haben werden⁸. Das objektorientierte Software Engineering kommt daher nicht umhin, der Problematik der verteilten Nutzung von Ressourcen auch entsprechend Rechnung zu tragen. In diesem Zusammenhang zeigt sich aber, daß die

5 Lego is a reg. trademark of the Lego System A/S, Denmark.

6 Transportwesen, Handel, Verwaltung, Finanzwesen; in der Forschung und Lehre Simulation, u.v.a.m.

7 Modifikation durch Spezialisierung und Generalisierung wird oft als ein besonderer Vorzug der objektorientierten Programmierungspraxis hervorgehoben.

8 Siehe auch meinen Beitrag: Verteilte Computersysteme - Trend der 90er Jahre, PIPELINE Nr. 3, Juni 91.

„Vermählung“ dieser Themenbereiche durch keinen magischen Trick von heute auf morgen erledigt werden kann.

Eine besondere Bedeutung kommt diesbezüglich der Object Management Group (OMG), einer kürzlich ins Leben gerufenen, um die 150 Mitglieder zählenden Vereinigung, zu, der eine Reihe renommierter Computerfirmen angehört. Durch die Spezifikation des Object Management Broker (OMB) verfolgt diese Vereinigung die generelle Zielsetzung, einen umfassenden de-facto Standard für die objektorientierten 90er Jahre zu schaffen.

Mit Hilfe von OMB soll eine *Plattform sowie Netzarchitektur-unabhängige* Inanspruchnahme von standardisierten, verteilten Objektcomponenten ermöglicht werden. Es handelt sich um ein revolutionierendes Konzept, das die Praxis des Software Engineering in der Zukunft maßgeblich beeinflussen wird.

Objektorientiertes Instrumentarium, Bemerkungen

Zum Instrumentarium des Software Engineering zählen generell Prinzipien, Techniken, Methoden, Ausdrucksmittel (Spezifikations-, Programmiersprachen), unterstützende Software-Werkzeuge. Einige ergänzende Bemerkungen zu den vorangegangenen Überlegungen scheinen angebracht.

Objektorientierte Methoden

Unzählige Veröffentlichungen dokumentieren intensive Bestrebungen, jede der Entwicklungsphasen des Erstellungsprozesses durch neu konzipierten methodischen Rahmen zu unterstützen⁹. Im wesentlichen geht es dabei um die Fragestellung ... „Wie kann man die *relevanten* Objekte (Klassen) finden?“ und „Wie hängen die Objekte zusammen?“ Abgesehen von einigen Teilerfolgen gibt es derzeit keine systematische, *allgemein anwendbare* Vorgangsweise, die zur methodischen Beantwortung der skizzierten Fragestellung führt.

Unumstritten bleibt aber, daß die Aufstellung eines objektorientierten *Problemstellungs-Modells* (PSM) den (objektorientierten) architektonischen Entwurf sowie die (objektorientierte) Implementation wesentlich erleichtert. Das PS-Modell „liefert“ sozusagen die zu implementierenden Objekte (Klassen), die während des architektonischen Entwurfes um *implementationspezifische* Objekte, nach-Bedarf, ergänzt werden.

Somit bleibt die Aufstellung des objektorientierten PS-Modells eines der **zentralen Themen**. Massive Schwierigkeiten können dann auftreten, wenn Anwender, Analytiker, Designer, „Geldgeber“, die sich an der **Problemfassung** beteiligen, sich nicht der gleichen, **objektorientierten Denkweise** (und dieser entsprechenden Sprache) bedienen.

Haben wir im Fall des „klassischen“ Software Engineering mit grundlegend anderen Problemen zu tun?

Objektorientierte Programmiersprachen

Im allgemeinen gibt es viele Gründe, die für die Wahl der einen oder anderen Programmiersprache sprechen, wie beispielsweise ...

- vorhandene Kenntnis,
- Verbreitung,
- Compiler-Verfügbarkeit,
- Effizienz des erstellten Codes (Umfang, ...),
- Angemessenheit,
- Kompaktheit, Ausdruckseleganz, Lesbarkeit, Überschaubarkeit,
- Prüfbarkeit u.a.m.

Es ist auffällig (und offensichtlich der Tradition entsprechend), daß am häufigsten die *Verbreitung, Compiler-Verfügbarkeit, Code-Effizienz*, also durchwegs **Implementationsaspekte**, als primärer Wahlgrund für eine Programmiersprache betrachtet werden. Eher als zweitrangig, wenn überhaupt, werden Aspekte der *Angemessenheit, Kompaktheit, Ausdruckseleganz, Lesbarkeit, Prüfbarkeit* u.d.gl. berücksichtigt. Diese Sicht der Dinge wurde durch den technologischen Stand der 70er Jahre geprägt, ja diktiert und kann somit als nicht mehr zeitgemäß angesehen werden.

Die stetig zunehmende Systemkomplexität **zwingt** zum Umdrehen der Wahlprioritäten. Wenn man von ausgesprochen „exotischen“ Programmiersprachen des Forschungsumvironments absieht, so ist es nur in Ausnahmefällen notwendig, den Implementationsaspekten einer Programmiersprache eine dermaßen überbetonte Aufmerksamkeit zu schenken.

Die Diskrepanz ist auf dem Gebiet der objektorientierten Programmierung besonders ausgeprägt, insbesondere im Zusammenhang mit den Programmiersprachen „C++ und die anderen“.

Auf der einen Seite hat man mit einer „**neuen**“ Systemkomplexität, abstrakten Objekten und Beziehungen zu tun, mit einer **systemischen** Sicht der Dinge. Im Vordergrund müssen *Systemtransparenz, Überschaubarkeit, Lesbarkeit, Prüfbarkeit* usw. stehen. Wenn es andererseits aber darum geht, ein adäquates Ausdrucksmittel zu wählen, das ja gerade in dieser Hinsicht sich unterstützend auswirken soll, geht man eher die „**alten**“, „bewährten“ Wege.

Zu den derzeit häufig diskutierten und in Anwendung befindlichen Programmiersprachen gehören u.a. ART, C++, CLOS, Eiffel, KEE¹⁰, Modsim, Objective-C, ObjectPascal, Sather, Self, Simula, Smalltalk.

Im folgenden wird eine kurze Gegenüberstellung einiger wesentlicher Merkmale geboten. Die Auswahl wurde hier aus Platzgründen beschränkt. Es handelt sich um keinen Vergleich. Eine Vollständigkeit wird nicht angestrebt.

C++:

Vorteile: vorhandene Kenntnis und Verbreitung von C

Nachteile: eine hybride „low level“ Sprache, mangelnde Unterstützung „sauberer“ Konzepte der Objektorientie-

⁹ Gemeint sind insbesondere die Abschnitte ... OO-Analyse und Spezifikation, OO-Design.

¹⁰ ART und KEE werden vor allem in Artificial Intelligence Kreisen eingesetzt.

rung, vom Standpunkt des *Systems Engineering* schwer überschaubar und lesbar, Parallelismus ? sprachspezifische Objektbibliothek ? garbage collection ? Typenparametrisierung (Generizität) ? Objektpersistenz ?

Eiffel:

Vorteile: Lesbarkeit, Überschaubarkeit, Prüfbarkeit, Generizität, teilweise Objektpersistenz, autom. garbage collection, Parallelismus in Aussicht

Nachteile: bescheidene Objektbibliothek ?

Modsim:

Vorteile: Lesbarkeit, Überschaubarkeit, Spezifikationsmöglichkeit von Objekttypen, Parallelismus, Objektbibliothek für Simulationszwecke und HCI- sowie graphischer Support

Nachteile: garbage collection ? Persistenz ?

Simula:

Vorteile: Objektbibliothek für Simulationszwecke, Parallelismus

Nachteile: Persistenz ? Generizität ?

Smalltalk:

Vorteile: Gut ausgebautes Entwicklungsumgebung, gute HCI-Objektbibliothek, Objektpersistenz in Aussicht

Nachteile: Typenprüfung ? Parallelismus ? Performance ? Generizität ?

Objektorientierte Entwicklungsumgebungen

Am Markt gibt es derzeit kaum objektorientierte Umgebungen, die einem gewissen Grundkatalog an zeitgemäßen Forderungen seitens des (theoretischen) Software Engineering genügen. Der Umstand ist auf vielfältige Ursachen zurückzuführen.

Zum einen hat sich die „objektorientierte“ Fachwelt noch nicht darüber einigen können, wie ein „Algorithmus“ des objektorientierten Vorgehens aussehen könnte. Jeder weiß es: einen Automatismus kann man nur dann konstruieren, wenn man die zu automatisierende Vorschrift genau kennt. Die menschlichen Fähigkeiten wie Kognition, Assoziation, Akkommodation, ökonomische Anwendung von Heuristiken, Beurteilungsvermögen u.v.a.m. haben für das Software Engineering eine zentrale Bedeutung. Die bisherigen Erfolge der Artificial Intelligence, diese Fähigkeiten

„algorithmisieren“ zu wollen, halten sich, wie sich jeder überzeugen kann, „in Grenzen“.

Die Computerindustrie hat es auch nicht leicht. Die qualitativen Forderungen an Entwicklungsumgebungen, die seitens der Software-Konstrukteure laufend steigen¹¹, sollen in den Software-Tools ihren Niederschlag finden. Hinzu kommt nun der in die Tiefe gehende, seit Ende der 80er Jahre sich langsam abzeichnende „Paradigmenwechsel“ zur Objekt-Orientierung. Die Computerindustrie ist bestrebt, in erster Linie die in die klassischen Tools bisher getätigten Investitionen zurückzuverdienen. Die wesentlich gestiegene Komplexität sowie der damit verbundene aufzubringende, enorme Aufwand sind auch Ursachen dafür, die Konstruktion neuer objektorientierter Umgebungen eine gewisse Zeit auf sich warten zu lassen.

Ausblick

Forschung auf dem Gebiet der objektorientierten Technologien ist gegenwärtig voll im Gange. Es gibt eine Fülle spezifischer Themen, die einzeln, kombiniert untersucht werden. Mit sehr großer Hoffnung wird die nahtlose Vermengung von objektorientierten Ansätzen und der verteilten Verarbeitung erwartet, wie sie beispielsweise von der OMG verfolgt werden.

Objektorientiertes Paradigma läßt sich schon heute mit Erfolg einsetzen. Die Grundvoraussetzung dafür ist die gute Kenntnis der Vor- und Nachteile dieses neuen „Denkschemas“ sowie die Hintergründe, die zu seinem Entstehen geführt haben.

Literatur

Booch G. : Object Oriented Design with Applications, Benjamin Cummings, 1991

Bulman D. : A modest survey of OOD approaches, Hotline on OO technology, Oct. 1991

Cobat6 F.J. : On Building Systems That Will Fail, CACM, Sept. 1991

Meyer B. : Object-oriented Software Construction, Prentice Hall, 1988

Winblad A. L., Edwards S. D., King D.R. : Object-Oriented Software, Addison Wesley, 1990

Antonin Sprinzl

¹¹ Support der Methodenvielfalt, Integrationsaspekte, Offene Systeme, Flexibilität, Anwenderfreundlichkeit, u.a. (siehe auch meine Vortragsunterlagen: Einführung in Software Engineering, Anliegen, Probleme, Instrumentarium, 1991).

Vortragsankündigungen

Die Abteilung Institutsunterstützung des EDV-Zentrums lädt zu folgenden Vorträgen ein:

Einführung in Software Engineering

Vortragender: A. Sprinzl

30., 31. März und 1. April 1992, 9 Uhr bis 12 Uhr

Themen:

Scope, Definition, Grundbegriffe
Problemstellung, Erstellungsprozeß, Artefakt, Instrumentarium
Qualitätsbetrachtungen
Problembereiche, praktische Hinweise
Entwicklungstendenzen

Objektorientierte Technologien in Software Engineering

Vortragender: A. Sprinzl

2. April 1992, 9 Uhr bis 12 Uhr

Themen:

Klassisches Software Engineering in Übersicht
Paradigmenwechsel, Gründe?
Klassisches vs. objektorientiertes Software Engineering
Grundlegende Konzepte, Programmiersprachen, Environments
Problembereiche, Zukunftstendenzen

Graphische Oberflächen, derzeitiger Entwicklungsstand

Vortragender: A. Sprinzl

27. April 1992, 9 Uhr bis 12 Uhr

Themen:

Applikationsstrukturen im Wandel der Zeit
HCI-Architekturen, Vor-, Nachteile
toolkitbasierte Oberflächen, User Interface Management Systems
derzeitiges Angebot, Beurteilung
Problembereiche, Zukunftsentwicklung

Einführung in die Konstruktion graphischer Applikationsoberflächen

Vortragender: A. Sprinzl

1. und 2. Juni 1992, 9 Uhr bis 12 Uhr

Themen:

Human Computer Interaction, Forderungen, Problembereiche
Konstruktion von Oberflächen "gestern und heute"
Oberflächen zwischen Wunsch und Realität
Rapid Prototyping mit Hilfe eines Interface Builders

Auskünfte und Anmeldung zu allen Vorträgen bei A. Sprinzl (Klappe 5599) oder sprinzl@edvz.tuwien.ac.at

Campus-Lizenzen für die TU Wien

Für die folgenden Produkte bestehen bisher Verträge:

Produkt	Charakteristik	Status	Gebühr	unterstützte Geräte	Art der Verteilung	Dokumentation	Produkt-Betreuer
ACSL	Simulationssprache zur Simulation kontinuierlicher Systeme	Bestellungen laufend möglich	600.- öS pro PC, 5000.- öS pro WS	PC, DECstation, HP/Apollo	Abholung	wird mitgeliefert	Husinsky (Kl. 5484)
Erlgraph	FORTRAN-Unterprogramme für Graphik-Ausgabe	Bestellungen laufend möglich	kostenlos	PC, Workstations	Abholung (6 Disketten mitbringen)	im Sekretariat um 60.- öS	Mayer (Kl. 5603)
HCL eXceed	X-Window Emulator	nur noch wenige Restexemplare	2000.- öS pro Gerät	PC (MS/DOS)	SW-Server	bei Lehrmittelzentrum/Fachbuchhandlung um 490.- öS	Selos (Kl. 3610)
LIT	Software zur Verwaltung von Literatur-Zitaten	Bestellungen laufend möglich	kostenlos	PC	Abholung	wird mitgeliefert	---
NAG FORTRAN-Library	mathematische Unterprogramme in FORTRAN	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Geräte-Type	praktisch alle	je nach Geräte-Typ verschieden	direkt bei NAG bestellen	Haider (Kl. 5603)
NAG Graphics Library	FORTRAN Unterprogramme für graphische Ausgabe	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Geräte-Type	fast alle	je nach Geräte-Typ verschieden	direkt bei NAG bestellen	Haider (Kl. 5603)
NAG Online Supplement	maschinlesbare Dokumentation für NAG Library	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Geräte-Type	für diverse Workstations	je nach Geräte-Typ verschieden	keine	Haider (Kl. 5603)
ORACLE	Datenbanksystem für komplexe Anwendungen	Bedarfsmeldungen ab sofort	300.- öS (PC), 1000.- öS bis 5000.- öS (WS)	PC und div. Workstations	je nach Geräte-Typ verschieden	direkt bei ORACLE bestellen	---
PC/TCP Plus inkl. NFS	Netzwerk-Software für TCP/IP	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Gerät	PC (MS-DOS) mit div. Ethernet-Boards	SW-Server	bei LMZ/Fachbuchhandlung um 200.- öS	Kainrath (Kl. 5604)

SPSS/PC+	statistisches Programmpaket (Grundpaket und div. Moduln)	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Gerät	PC (MS-DOS, OS2), Macintosh	SW-Server	bei LMZ/Fachbuch- handlung bestellen Einf.schrift um 25.- öS im Sekretariat	Mayer (Kl. 5603)
T_EX	Textsatzsystem für wissenschaft- liche Dokumente	Zugriff laufend möglich	kostenlos (Public Domain Version)	PC, Workstations	über FTP (Adresse: tex.tuwien.ac.at)	im Buchhandel erhältlich	Rogl (Kl. 3612)
Virus-Utilities	Entdecken und Entfernen von Computer-Viren sowie laufende Überwachung	Bestellungen laufend möglich, Auslieferung läuft	400.- öS pro Institut	PC	Abholung	wird mitgeliefert	Gisch (Kl. 5488)
WordPerfect	Textverarbeitungs- system	Bestellungen laufend möglich, Auslieferung läuft	300.- öS pro Geräte-Type	PC (MS-DOS, MS-Windows), Macintosh, IBM RS 6000, SUN Sparc, UNIX *)	SW-Server	bei LMZ/Fachbuch- handlung um 850.- öS	Mayer (Kl. 5603)

*) WordPerfect ist zwar für eine Reihe von UNIX-Systemen verfügbar, allerdings derzeit meist in älteren Versionen.

Von folgenden Sammelbestellungen sind noch einige Exemplare verfügbar:

CorelDraw	leistungsfähiges Zeichenprogramm	Restexemplare	3000.- öS pro Gerät	PC (MS-Windows)	Abholung	wird mitgeliefert	---
ESSL- Routinen	Unterprogramme zur Leistungs- steigerung bei numerisch inten- siven Programmen	Restexemplare	3000.- öS pro Gerät	IBM RS 6000/320	Abholung	wird mitgeliefert	---

Bestellformulare für die einzelnen Produkte sendet Ihnen die Abteilung Benutzerbetreuung auf Anfrage zu (Kl. 5821). Die Vertragsbedingungen mit den einzelnen Firmen weisen teilweise erhebliche Unterschiede auf. So gibt es Campus-Lizenzen mit unbeschränkter Anzahl von Einzelbenutzern und andere, die auf eine bestimmte Anzahl limitiert sind. Dementsprechend sind auch die Schutzgebühren entweder pro Institut oder pro Gerät vorgesehen. Die Vertragsbedingungen (Academic Use, keine Weitergabe, Copyright ...) sind auch von den Einzelbenutzern einzuhalten, was durch Unterschrift auf einer Verpflichtungserklärung zu bestätigen ist.

Je nach den Lieferbedingungen der einzelnen Hersteller erfolgt die Ausgabe der Software entweder auf Originaldisketten, auf kopierten Datenträgern oder über File-Server (siehe Spalte "Art der Verteilung"). Für organisatorische Fragen zu den Lizenzen steht Herr Schornböck (Klappe 5820) zur Verfügung, Fragen zu den Produkten beantworten die in der Spalte "Produktbetreuer" angegebenen Mitarbeiter des EDV-Zentrums.

Irene Hyna

Schulungsprogramm

Hinweise

Bei den mit ¹⁾ gekennzeichneten Kursen ist die Teilnehmerzahl beschränkt. Für diese Kurse wird um rechtzeitige Anmeldung bei Frau Poremba (Klappe 5821) mindestens eine Woche vor Kursbeginn ersucht. Bei mit ²⁾ gekennzeichneten Kursen ist bei der Anmeldung eine Kautions von öS 300,- zu hinterlegen. Bei der Anmeldung wird der Kursort bekanntgegeben. Für die mit ³⁾ bezeichneten computerunterstützten Kurse (CAI) findet jeden Mittwoch um 15 Uhr c.t. eine kurze Einführung statt. Zu dieser ist eine Anmeldung bis spätestens Montag 12 Uhr erforderlich. Bei allen anderen Kursen werden Kursort und Kurszeiten durch eine separate Aussendung bekanntgegeben.

Kurse, die von EDV-Zentrums-Personal gehalten werden, sind für Angehörige österreichischer Universitäten und für Studenten kostenlos. Für Kurse, die von anderen Personen gehalten werden, wird ein Kostenbeitrag eingehoben.

Bei der Vergabe von Kursplätzen werden Angehörige der Technischen Universität Wien bevorzugt. Nach Maßgabe freier Plätze können auch Hochschulfremde an den Kursen teilnehmen. In diesem Fall ist ein Kostenbeitrag zu entrichten.

1. Einführung in die Datenverarbeitung für Nicht-Techniker ^{1) 2)}

Vortragender: Schmitt

Dauer: 2 Tage mit je 4 Unterrichtseinheiten (à 45 min) + Vorführungen am Nachmittag

Kosten: 2.000,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-03-12 und 13
92-05-06 und 07
92-06-11 und 12

Vorkenntnisse: keine

Einführung in die Aufgabenstellung der Datenverarbeitung; Vergleich Personal Computer, Workstation und Groß-EDV; Was sind Computernetze; Ergonomie eines Bildschirmarbeitsplatzes; Aufgaben des Betriebssystems (DOS); Graphische Benutzeroberflächen (Windows 3.0, Apple); Vorführung von Anwendungsprogrammen

2. Einführung in Word für Sekretariate ^{1) 2)} (Word 5.5, Winword 1.1 am PC; Word 4.0 am Apple)

Vortragender: Schmitt

Dauer: 2 Tage mit je 4 Unterrichtseinheiten (à 45 min) + Übungen am Nachmittag

Kosten: 2.000,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-02-24 und 25
92-04-08 und 09

Vorkenntnisse: Kurs 1 oder gleichwertige Kenntnisse

Aufrufen und Beenden von Word; Texteingabe, Besonderheiten von Tastatur und Maus; Formatieren von Absätzen, Seiten, Dokumenten, Tabellen; Ausgabe von Dokumenten

3. Einführung in WordPerfect 5.1 am PC ^{1) 2)}

Vortragende: Kimbacher (ÖAW)

Dauer: 3 Tage mit je 4 Unterrichtseinheiten (à 45 min) + Übungen

Kosten: 3.000,- (für Angehörige einer Universität und Studenten 1.800,-)

Termine: 92-03-09 bis 11
92-05-11 bis 13

Vorkenntnisse: Kurs 1 oder gleichwertige Kenntnisse

Aufrufen und Beenden von WordPerfect 5; Texteingabe, Besonderheiten von Tastatur und Maus; Formatieren von Absätzen, Seiten, Dokumenten, Tabellen; Ausgabe von Dokumenten; Erstellen von Serienbriefen; Macros

4. Serienbriefe mit Word for Windows ^{1) 2)}

Vortragender: Schmitt

Dauer: 1 Tag mit 3 Unterrichtseinheiten (à 45 min) + Übungen

Kosten: 1.000,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-03-25
92-06-03

Vorkenntnisse: Kurs 2

Serienbriefe; Etikettendruck

5. Große Dokumente mit Word for Windows ^{1) 2)}

Vortragender: Schmitt

Dauer: 1 Tag mit 3 Unterrichtseinheiten (à 45 min) + Übungen

Kosten: 1.000,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-03-30
92-06-04

Vorkenntnisse: Kurs 2

Druckformatvorlagen; Fußnoten, Inhaltsverzeichnis, Index; Formelsatz

6. Große Dokumente mit WordPerfect ^{1) 2)}

Vortragende: Kimbacher (ÖAW)

Dauer: 1 Tag mit 3 Unterrichtseinheiten (à 45 min) + Übungen

Kosten: 1.000,- (für Angehörige einer Universität und Studenten 600,-)

Termin: 92-04-06

Vorkenntnisse: Kurs 3

Styles; Fußnoten, Inhaltsverzeichnis, Index; Formelsatz

7. Einführung in EXCEL für Sekretariate ^{1) 2)}

Vortragender: Schmitt

Dauer: 3 Unterrichtseinheiten (à 45 min)

Kosten: 500,- (für Angehörige einer Universität und Studenten gratis)

Termin: 92-04-27

Vorkenntnisse: Kurs 1 und 2 oder gleichwertige Kenntnisse

Aufrufen und Beenden von EXCEL; Grundlagen der Tabellenkalkulation; Business-Graphiken

8. Zugriff auf die Services von TUNET

Vortragender: Schmitt

Dauer: 3 Unterrichtseinheiten (à 45 min)

Kosten: 500,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-03-23
92-05-18

Vorkenntnisse: Kurs 1 oder gleichwertige Kenntnisse

Organisation von Lokalen Netzen; TUNET Ausdehnung und Protokolle; Zugriff auf einen Rechner; (Name-, Time- und Mail-) Server, Gateways

9. E-Mail im TUNET (PC, UNIX, Apple) ¹⁾

Vortragende: Angebrandt, Schmitt

Dauer: 3 Unterrichtseinheiten (à 45 min)

Kosten: 500,- (für Angehörige einer Universität und Studenten gratis)

Termine: 92-03-05 (nur Mail am PC)
92-04-03 (nur Mail am Apple)
92-05-04 (nur Mail unter UNIX)

Vorkenntnisse: Kurs 1 und 8 oder gleichwertige Kenntnisse

E-Mail Konzept an der TU Wien; Software-Beschaffung und Installation; Benutzeroberfläche; E-Mail Adressen

10. Einführung in das VAX/VMS-Betriebssystem

Vortragender: Sprinzl

Dauer: 2 Vormittage

Termin: 92-02-24 und 25

Vorkenntnisse: keine

Highlights der VAX/VMS-Systemarchitektur, Komponenten; fundamentale Systemkonzepte, relevante Begriffe; DCL-Interpreter, Kommando-Prozeduren; Grundzüge der Ressourcenallokationsmechanismen; Filesystem, Filemanipulation, logische Namen; Systemmeldungen; Dokumentation, Hilfseinrichtungen; Grundlegende Werkzeuge

11. Virtuelles Memory Management, Prozeß Management im VAX/VMS-System

Vortragender: Sprinzl

Dauer: 1 Vormittag

Termin: 92-02-26

Vorkenntnisse: Kurs 10

VMS-Betriebssystem, Aufbau, grundlegende Komponenten; Virtueller Adreßraum, Prozeß-, System- Arbeitsspeicher, Process Working Set, Parameter; Verwaltungs-

tabellen, Aufbau; Image-Ablauf; Pager, Swapper; Page Faults, System-Caches; VMS-Systemverhalten bei unterschiedlicher Laststruktur; Voraussetzungen für optimale Systemperformance; Systemmonitoring; Scheduler, Aufbau, Funktionsweise, Process Queues

12. Shareable Images im VMS-Environment

Vortragender: Sprinzl

Dauer: 1 Vormittag

Termin: 92-02-27

Vorkenntnisse: Kurs 11

Sprachprozessoren im VMS-Environment; Program Sections, Objektmodule; Linker und seine Funktion; Shareable Images, Bedeutung im VMS-Environment und Aufbau; Anwendungsbeispiele

13. Programmieren im VAX/VMS-Environment

Vortragender: Sprinzl

Dauer: 2 Vormittage

Termin: 92-03-02 und 03

Vorkenntnisse: Kurs 12

Grundsätzliches zur Konstruktion von Software; logische Namen, Dateien (Organisation, Manipulation, Verwaltung); IO-Subsysteme in Übersicht; Grundlegende Werkzeuge der Programmentwicklung, Eigenschaften, Anwendung; Interpretation-Kommunikation, Procedure Calling Standard, Systemmeldungen; Run Time Library, System Services in Übersicht; Maßnahmen zur Erhöhung der Programm-Ab-laufeffizienz; System Queues, Anwendung

14. Einführung in das Betriebssystem UNIX

Vortragender: Houdek

Dauer: 2 Vormittage

Termin: 92-05-14 und 15

Vorkenntnisse: keine

Zugang zu einem UNIX-System; Paßwortschutz; Einordnung eines Benutzers in Benutzerklassen; allgemeine Kommandosyntax; Dokumentation; Dateien; Dateistruktur; Dateizugriff; Dateiorganisation; Dateiattribute; Bedeutung der Zugriffsrechte; Setzen und Ändern von Zugriffsrechten; Prozesse; Bourne-Shell; Prozeßkenndaten; Hintergrundprozesse; Hilfreiche UNIX-Kommandos; Einführung in vi

15. Paralleles Programmieren auf Shared Memory Parallel Computern

Vortragender: Fritscher

Dauer: 1 Vormittag

Termin: 92-04-07

Vorkenntnisse: Grundkenntnisse in UNIX

Einführung in die Parallel Computer Architekturen; Strategien zum parallelen Programmieren; Shared Memory Programmieren auf der Sequent Balance; paralleles Programmieren mit Linda; Übungen auf der Sequent Balance

16. Programmieren auf Distributed Memory Parallel Computern

Vortragender: Fritscher

Dauer: 1 Vormittag

Termin: 92-04-28

Vorkenntnisse: Grundkenntnisse in UNIX

Einführung in die Parallel Computer Architekturen; Strategien zum parallelen Programmieren; Message Passing Programmieren auf dem IBM Workstation Cluster; Übungen auf dem Workstation Cluster

17. Einführung in die Programmiersprache C³⁾

Vortragender: Computer Aided Instructions (CAI)

Dauer: 10 Unterrichtseinheiten (à 90 min)

Kosten: 1.000,- (für Angehörige einer Universität und Studenten gratis)

Termin: immer

Vorkenntnisse: keine

Datentypen; Operatoren; Kontrollstrukturen; Funktionen und Speicherklassen; Ein-/Ausgabe; Zeiger- und Datenstrukturen

18. Einführung in die Programmiersprache FORTRAN 77 (Structured Programming in FORTRAN 77)³⁾

Vortragender: Computer Aided Instructions (CAI)

Dauer: ca. 15 Unterrichtseinheiten (à 90 min)

Kosten: 1.000,- (für Angehörige einer Universität und Studenten gratis)

Termin: immer

Vorkenntnisse: keine EDV-Kenntnisse, aber Englisch

Struktogramme; Grundbegriffe von FORTRAN 77; einfache Programme

19. Programming in COBOL³⁾

Vortragender: Computer Aided Instructions (CAI)

Dauer: ca. 20 Unterrichtseinheiten (à 90 min)

Kosten: 1.000,- (für Angehörige einer Universität und Studenten gratis)

Termin: immer

Vorkenntnisse: keine EDV-Kenntnisse, aber Englisch

Einführung in die Datenverarbeitung; Grundbegriffe von COBOL; Erstellen einfacher Programme

20. Der PC als Arbeitsplatzrechner (Vorlesung)

Vortragender: Schmitt

Dauer: 2 Unterrichtseinheiten (à 45 min) pro Woche mit Übungen

Termin: Sommer-Semester

Vorkenntnisse: keine

Einführung in das Betriebssystem auf PCs (MS DOS, MS Windows), Macintosh (Finder) und Workstations (UNIX, X-Window)

21. Effiziente Nutzung der Vektorrechner (Vektorisierung)¹⁾

Vortragende: Weisz, Macsek

Dauer: 3 Nachmittage

Termin: 92-03-17 bis 19

Vorkenntnisse: FORTRAN

Vektorisierbare Konstrukte; Eigenschaften der Vektorprozessoren; Vektorisierende FORTRAN-Compiler; Unterstützungswerkzeuge für die Vektorisierung

22. Vektorisierungswerkzeuge an der Convex¹⁾

Vortragende: Weisz, Macsek

Dauer: 1 Nachmittag

Termin: 92-03-24

Vorkenntnisse: FORTRAN, eventuell Kurs 21

Spezifika des Vektorprozessors der Convex; Vektorcompiler und andere Werkzeuge

23. Einführung in die Benutzung des Vektorrechners SNI VP50-EX (Batch und Time Sharing, Vektorisierungswerkzeuge)¹⁾

Vortragende: Weisz, Macsek

Dauer: 3 Nachmittage

Termin: 92-03-31 bis 04-02

Vorkenntnisse: allgemeine EDV-Kenntnisse, ev. Kurs 21

Interaktive Benutzeroberfläche; Batch-Steuerbefehle; interaktive Prozeduren; Spezifika des Vektorprozessors der VP50-EX; Vektorcompiler und andere Werkzeuge

24. PATRAN

Vortragender: Computer Aided Instructions (CAI)

Dauer: 15 Unterrichtseinheiten (à 60 min)

Kosten: 2.000,-

(Begründung der Kosten siehe PIPELINE 2)

Termine: nach Vereinbarung

Vorkenntnisse: keine EDV-Kenntnisse, aber Englisch

Videokurs für die Verwendung von PATRAN für Pre- und Postprocessing insbesondere zum Finite Elemente Programm ABAQUS

25. Einführung in die Benutzerarbeitsplätze des EDV-Zentrums¹⁾

Vortragende: Schmitt, Tutoren

Dauer: 2 Stunden

Termine: ab Semesterbeginn alle 14 Tage

Vorkenntnisse: keine

Einführung in die Hardware-Ausstattung der Benutzer Räume (Rechner und Drucker); Organisatorisches; Software-Ausstattung; Graphische Benutzeroberfläche (Maus, Menüs, Ikonen, Fenster, Rollbalken); Vernetzung, Zugang zu anderen Rechnern; Bedienung der Drucker

Gerhard Schmitt

**Abteilungen
und Mitarbeiter
(Telefonliste,
E-Mail-Adressen)**

*EDV-Zentrum der
Technischen Universität Wien
Wiedner Hauptstraße 8-10
A - 1040 Wien
Tel.: (0222) 58801-5481
Fax: (0222) 587 42 11*

**Vorstand des EDV-Zentrums:
Prof. K. Schwarz (5188)**

vorstand@edvz.tuwien.ac.at
schwarz@edvz.tuwien.ac.at

Stellvertreter:

Prof. H. Grünbacher (8150)

herbert@vlsivie.tuwien.ac.at

**Leiter des EDV-Zentrums:
W. Kleinert (5480)**

kleinert@edvz.tuwien.ac.at
leiter@edvz.tuwien.ac.at

**Administrationsreferat
(Sekretariat): 5481**

administration@edvz.tuwien.ac.at
sekretariat@edvz.tuwien.ac.at

E. Beck 5489
S. Huber 5481
A. Müller 5485

mueller@edvz.tuwien.ac.at

Vertragsassistent J. Fritscher 5505
fritscher@edvz.tuwien.ac.at

Störungsnummern:

58801-5830 Zentrale Server
Operating
587 56 23 TUNET

Außenanschlüsse:

Datex-P (CDCNET) 26231060 102
EMAIL 26231060 4
asynchron,
300 - 2400 bd 587 46 92 Serie
300 - 9600 bd 587 46 95 Serie

**Abteilung Benutzerbetreuung
Leiter: D. Schornböck (5820)
schornboeck@edvz.tuwien.ac.at**

I. Poremba	5821	poremba@edvz.tuwien.ac.at
A. Roza	5824	roza@edvz.tuwien.ac.at
G. Schmitt	5600	schmitt@edvz.tuwien.ac.at
E. Schörg	5821	schoerg@edvz.tuwien.ac.at
E. Widmann	5486	widmann@edvz.tuwien.ac.at

**Abteilung Institutsunterstützung
Leiter: A. Blauensteiner (5493)
blauensteiner@edvz.tuwien.ac.at**

L. Gisch	5488	gisch@edvz.tuwien.ac.at
G. Gollmann	3611	gollmann@edvz.tuwien.ac.at
W. Haider	5603	haider@edvz.tuwien.ac.at
G. Houdek	3616	houdek@edvz.tuwien.ac.at
H. Mayer	5603	mayer@edvz.tuwien.ac.at
J. Rogl	3612	rogl@edvz.tuwien.ac.at
M. Schandl	5488	schandl@edvz.tuwien.ac.at
R. Sedlaczek	3611	sedlaczek@edvz.tuwien.ac.at
W. Selos	3610	selos@edvz.tuwien.ac.at
B. Simon	5602	simon@edvz.tuwien.ac.at
A. Sprinzi	5599	sprinzi@edvz.tuwien.ac.at
H. Stallbaumer	5494	stallbaumer@edvz.tuwien.ac.at
G. Wehrberger	3617	wehrberger@edvz.tuwien.ac.at

**Abteilung Kommunikation
Leiter: J. Demel (5829)
demel@edvz.tuwien.ac.at**

M. Angebrandt	5834	angebrandt@edvz.tuwien.ac.at
H. Kainrath	5604	kainrath@edvz.tuwien.ac.at
J. Kondraschew	5483	kondraschew@edvz.tuwien.ac.at
F. Matasovic	5605	matasovic@edvz.tuwien.ac.at
M. Rathmayer	5834	rathmayer@edvz.tuwien.ac.at
M. Siegl	5604	siegl@edvz.tuwien.ac.at
Walter Weiss	5605	weiss@edvz.tuwien.ac.at

**Abteilung Hardware - Planung und Betrieb
Leiter: P. Berger (5815)
berger@edvz.tuwien.ac.at**

W. Altfahrt	5819	altfahrt@edvz.tuwien.ac.at
J. Beiglböck	5815	beiglboeck@edvz.tuwien.ac.at
P. Deinlein	5830	deinlein@edvz.tuwien.ac.at
H. Eigenberger	5830	eigenberger@edvz.tuwien.ac.at
H. Fichtinger	5825	fichtinger@edvz.tuwien.ac.at
P. Hoffmann	5487	hoffmann@edvz.tuwien.ac.at
H. Mastal	5816	mastal@edvz.tuwien.ac.at
J. Pfennig	5830	pfennig@edvz.tuwien.ac.at
J. Sadovsky	5819	sadovsky@edvz.tuwien.ac.at
E. Srubar	5826	srubar@edvz.tuwien.ac.at
G. Vollmann	5825	vollmann@edvz.tuwien.ac.at
Werner Weiß	5830	weisswer@edvz.tuwien.ac.at

**Abteilung Hochleistungsrechnen
Leiter: Willy Weisz (5818)
weisz@edvz.tuwien.ac.at**

F. Blöser	5811	bloeser@edvz.tuwien.ac.at
H. Flamm	5823	flamm@edvz.tuwien.ac.at
I. Husinsky	5484	husinsky@edvz.tuwien.ac.at
J. Macsek	5811	macsek@edvz.tuwien.ac.at
G. Petschl	5823	petschl@edvz.tuwien.ac.at
P. Torzicky	5823	torzicky@edvz.tuwien.ac.at