Zeitschrift: Visionen : Magazin des Vereins der Informatik Studierenden an der ETH Zürich

Herausgeber: Verein der Informatik Studierenden an der ETH Zürich

Band: - (1995)

Heft: 11

Heft

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Siehe Rechtliche Hinweise.

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. Voir Informations légales.

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. See Legal notice.

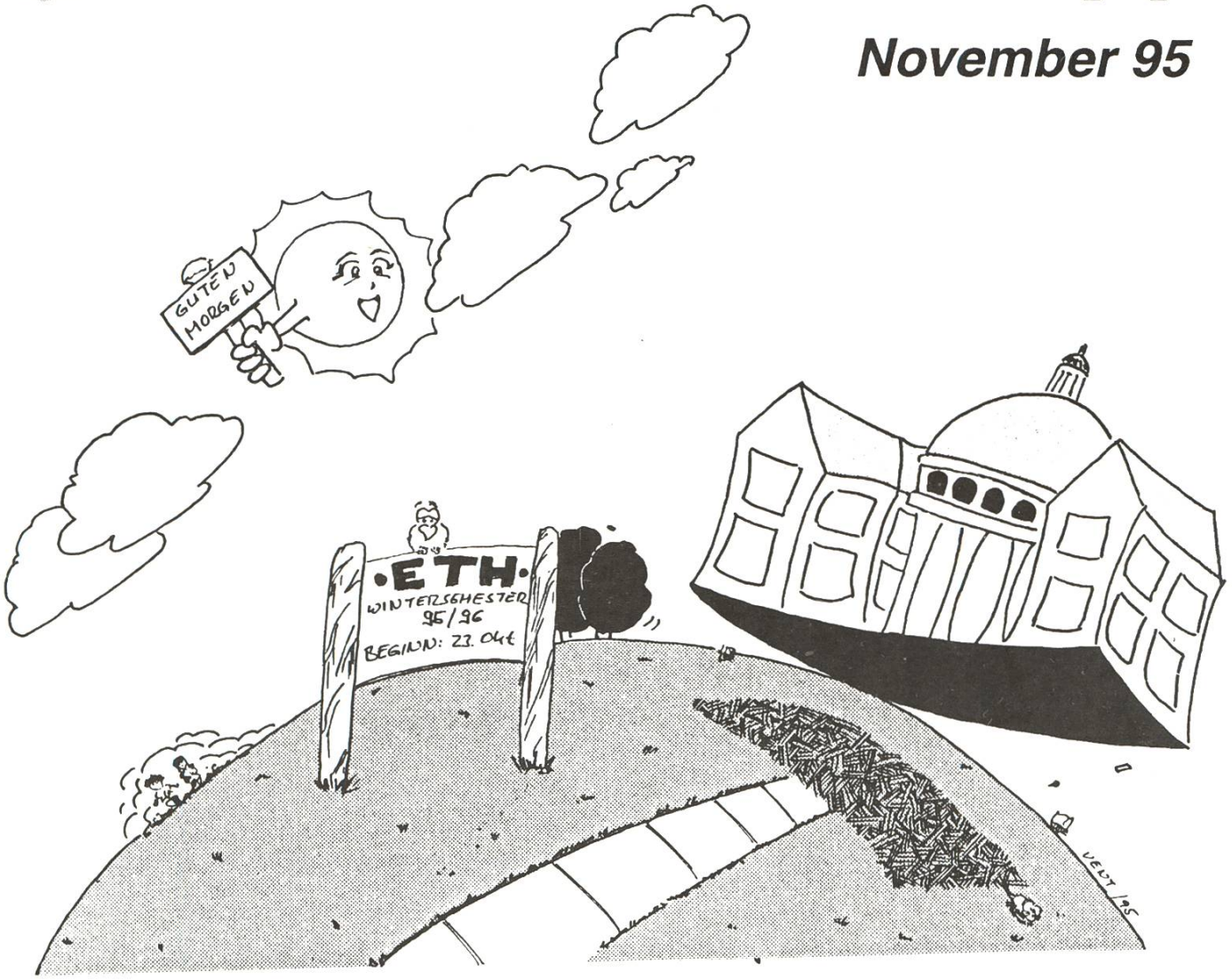
Download PDF: 20.05.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Visionen

11

November 95



Telecom '95
Ressourcensparen
Fächertest: GDV II

Unix

VISIONEN Herausgegeben vom Verein der Informatikstudierenden an der ETH Zürich

Adressen

Aktuarin: Nathalie Weiler
Guggachstrasse 10, 8057 Zürich
Tel.: 01 / 364 23 91
e-mail: nathalie@vis.inf.ethz.ch

Feste & Kultur: Gregor Battilana
Eschenhaustrasse 34, 8053 Zürich
Tel.: 01 / 381 66 24
e-mail: lothian@vis.inf.ethz.ch

Präsident: Daniel Kluge
Irringersteig 3, 8006 Zürich
Tel.: 01 / 252 04 14
e-mail: dankluge@iiic.ethz.ch

Prüfungen und Unterricht:
Michael Bischof
Bergstrasse 128, 8032 Zürich
Tel.: 01 / 251 07 11
e-mail: mbischof@iiic.ethz.ch

Quästor: Michael Baumer
Im Schachenhof 26, 8906 Bonstetten
Tel.: 01 / 701 11 91
e-mail: baumi@vis.inf.ethz.ch

Visinfo(Infosystem):
Michel Müller
Rheinländerstr. 15, 4056 Basel
Tel.: 061 / 321 81 23
e-mail: mimuelle@iiic.ethz.ch

Impressum

Herausgeber:
Verein der Informatikstudierenden
an der ETH Zürich.

Redaktor: Christian Limpach
email: chris@vis.inf.ethz.ch

Adresse Verlag & Redaktion:
VIS
Verein der Informatikstudierenden
ETH Zentrum, IFW B29
8092 Zürich

Tel.: 01 632 72 12 (Mo–Fr, 12.15–13.00)
Fax: 01 632 11 72
e-mail: vis@iiic.ethz.ch
URL: http://vis-next.iiic.ethz.ch/
Postkonto: 80–32779–3
Präsenzzeit: Mo–Fr: 12.15–13.00
Jahresabonnement: SFr. 20.–
Auflage: 1400
Inseratepreise:

1 Seite	s/w	SFr. 500.–
1 Seite	+1 Farbe	SFr. 750.–
1/2 Seite	s/w	SFr. 250.–

Sonderwünsche auf Anfrage

Redaktions- und Anzeigeschluss für die nächste Ausgabe:
Donnerstag, 23. November 1995

Die Beiträge geben die Meinung des jeweiligen Autors wieder. Offizielle Äusserungen des VIS oder des Vorstandes sind als solche gekennzeichnet.

©1995 by
Verein der Informatikstudierenden

Sallü zäme

Ich hoffe, dass seit dem Studienbeginn keine Neueingetretene oder Neueingetretener sich an der ETH verlaufen hat, so ganz alleine, ohne Tutor.

Da jetzt schon die 1.Semestrigen einen INOX-Account bekommen, und dieser Tatsache im Survival-Guide nur beschränkt Rechnung getragen wurde, wollen wir es in den folgenden Visionen nachholen, da es auch gestandene Professoren gibt, welchen ein solcher Crashcourse gut tun könnte.

Da gerade in den vergangen Visionen einige Möglichkeiten von UNIX, resp. einige Programme erklärt wurden, können wir schlecht E-mail und Netnews nochmals vorstellen. Darum werden wir einen Nachdruck der entsprechenden Artikel zusammenstellen, sofern diese nicht Eingang in den Survival-Guide gefunden haben. Dieser Nachdruck wird dann im VIS-Büro erhältlich sein.

Die Vordiplomsprüfungen haben auch im Vorstand für Aufregung gesorgt, gingen doch gleich drei Vorständler an ein solches. Unserem Quästor darf man gratulieren, er hat es geschafft, im ersten Anlauf das 1.VD zu passieren. Unser Festminister wird leider noch ein zweites Mal antreten müssen, zu seinem 2.VD.

Leider ist für unseren VISNFO-Verantwortlichen Michel die IT-Karriere zu Ende, ihn ereilt das gleiche Schicksal wie sein Vorgänger. Am zweiten Versuch am 2. VD nur um Haaresbreite Schiffbruch erlitten, und jetzt trotz langjährigem Studium vor der Türe gelandet.

Am Montag dem 13.November wird die MV im GEP-Pavillon um 1900 Uhr stattgefunden haben. Für alle die's nicht gefunden haben, der GEP-Pavillon ist der Bunker, gegenüber dem Eingang der Mensa. Leider konnten 'Traci and the Lords' nicht wie angekündigt auftreten; Prüfungen und dergleichen verunmöglichten dies.

-daniel

Telecom 95 Ingenieurtag vom 10. Oktober 95

Die Telecom ist die weltweit grösste Messe im Bereich der Telekommunikation und findet alle vier Jahre in den Palexpo-Hallen von Genf statt. Die Pro Telecom hat dazu allen interessierten IIIc-Studierenden einen Gratiseintritt und eine Übernachtungsmöglichkeit in Genf angeboten.

Von der Übernachtungsmöglichkeit machten Christian Fritz und ich Gebrauch und führen schon am 9. Oktober nach Genf, um auch am Ingenieurtag fit zu sein. Zum Glück hatten wir die VD-Prüfungen schon hinter uns und konnten so unbeschwert die Exkursion geniessen. Genf zeigte sich am Anreisetag von seiner schönsten Seite; die Stadt wollte erforscht werden. Bald wurde es Abend und wir machten uns auf die Suche nach der (modernen) Zivilunterkunft, in der wir die Nacht mit vielen HTL-Studenten aus Basel verbrachten. Am Morgen wurden wir zu Landis&Gyr chauffiert, wo ein grosses Morgenbuffet bereit stand. Dort konneten wir möglichst viel Energie für den Tag tanken. Anschliessend ging es per Direktbus zu den Eingangstoren der Palexpo-Hallen, wo wir auch die anderen Studis trafen, die erst morgens anreisen wollten. Wir hatten nun ganze 8 Stunden Zeit uns in den Hallen auszutoben.

Die Stände der grossen Firmen waren schon vom Anblick her überwältigend. Bis zu vierstöckige Bauten, in die bis zu 11 Millionen Dollar investiert wurden, sollten die Firmen im besten Licht zeigen. Gerade heute ist der Telekommunikationsmarkt einer, der den grössten Wachstum noch vor sich hat und in den jeder einsteigen will. Somit waren alle die Rang und Namen haben da, aber auch viele, von denen man noch nie was gehört hat, jedoch mit ihren innovativen Ideen bald in aller Munde sein könnten. Viele Firmen präsentierten Lösungen im Bereich interaktives Fernsehen mit Settop-Boxen. Drahtlose Kommunikation und ATM-Technologien waren weitere Hauptthemen. Andere marktreife Technologien, wie z.B. Spracherkennung, wurden auch gezeigt: Ein Chinese sprach vor mir ins Telefon und der Computer führte seine Befehle aus, danach durfte ich per Stimme im WWW navigieren. Der Rechner verstand mich auf Anhieb. Das Internet war natürlich auch vertreten, darunter auch die Möglichkeit, Programme in einer Page zu integrieren und der 3D-Standard. Man könnte natürlich

hier sehr lange weiterfahren mit den vielen Attraktionen. Leider war man als Student oft nicht so sehr erwünscht, wenn man sich mehr Informationen beschaffen wollte. Entweder man bleibt hartnäckig, kleidet sich entsprechend oder man hat eine Visitenkarte einer bekannten Firma. Obwohl die Telecom für jede zugänglich ist, bestand das Publikum praktisch nur aus Geschäftsleuten aus aller Welt.

Der Tag war sicherlich ein interessantes Erlebnis. Für mich als Beobachter ist es sicherlich ein anderes als für jemanden der Kontakte suchen wollte oder bestimmte Technologien unter die Lupe nahm. Auf jeden Fall finde ich, war es eine einmalige Möglichkeit, diese gigantische Messe zu besuchen, in der alles präsentiert wird, was heute aktuell ist. Andere Städte wie Berlin wollen diese Messe auch, doch findet die nächste Telecom '99 nochmals in Genf statt. Ein interessanter Nebeneffekt der Telecom: Die Hotels waren bis nach Neuenburg ausgebucht.

Reto Grob
rgrob@iiic.ethz.ch

BFI Bebag Finanz Informatik AG

Wir entwickeln Applikationen im Bereich Portfoliomanagement (Handelsunterstützung, Wertpapierverwaltung, Buchhaltung und Reporting) und suchen eine(n)

Software- bzw. Informatik-Ingenieur(in)

mit der Fähigkeit, einerseits teambezogen zu arbeiten und andererseits auch weitgehend selbständig in neue Gebiete (Informatik und Finanz) einzusteigen.

Die Entwicklungsumgebung basiert auf UNIX und C++ (Zielumgebung zusätzlich Windows NT). Eingesetzt werden X Window System (Motif), FrameMaker, Informix, GNU (g++, gcc, gdb, make, emacs usw.), LISP, Smalltalk, SNiFF+, Pure Software (Purify, Quantify, Pure Coverage), sowie weitere Werkzeuge (CVS, tcl/tk, perl usw.).

Unsere Mitarbeiterinnen und Mitarbeiter stammen aus den Bereichen Informatik, Elektrotechnik, Mathematik, Betriebswirtschaft und Finanzen, und haben u.a. an der ETH abgeschlossen bzw. promoviert.

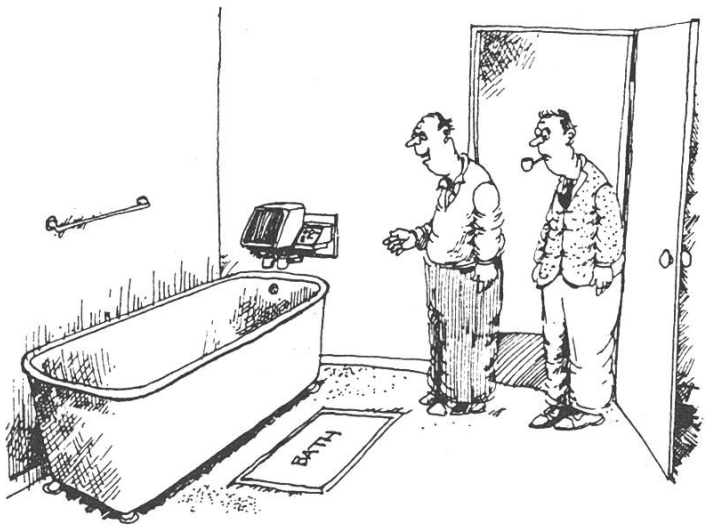
Weitere Infomationen: Hanspeter Wachter, wachter@bfi.bebag.ch, Tel. 01 / 735 81 71

BFI Bebag Finanz Informatik AG
In der Luberzen 1, CH-8902 Urdorf, Tel. 01 / 735 81 81, Fax 01 / 735 81 88

ACM-Wettbewerb – Reminder

Damit sicher niemand vergisst, sich für die ETH-interne Ausscheidung anzumelden, hier nochmals das Wichtigste zum nächsten ACM-Wettbewerb:

Wann?	ETH-intern: Samstag, 2.12.95 von ca. 0800-1600 SWERC: Freitag/Samstag, 8./9.12.95
Wo?	ETH-Zürich, Hauptgebäude
Anmeldeschluss:	***Montag, 27.11.95, 1200***
Anmeldung:	Elektronisch jederzeit über WWW/E-mail; oder auf Papier beim VIS (IFW B29) während den Präsenzzeiten.
Weitere Infos:	Web: `http://www.acm.inf.ethz.ch/` (oder `http://lkt-1.ethz.ch/SWERCHome.html`) e-mail: `acm@vis.inf.ethz.ch`



'There are hundreds of sensors built into the bath surface, so that when you lose the soap it can tell you where it is.'

Schweizerischer Bankverein

IT-Camp

IT-Camp ist eine im Aufbau begriffene Abteilung des Schweizerischen Bankvereins in Basel, in der neue Informationstechnologien untersucht und in Pilotanwendungen umgesetzt werden. Die Projekte werden in enger Zusammenarbeit mit schweizerischen Hochschulen und Universitäten durchgeführt.

Für Projekte im Bereich

Multimedia-Kommunikation

bieten wir einer/einem ***Dipl. Informatik-Ing. ETH*** oder ***Dipl. Elektrotechnik-Ing. ETH*** die Gelegenheit, in einem gleichwohl interessanten wie zukunftsorientierten Gebiet mitzuarbeiten und neue Ideen einzubringen.

Im Zentrum steht die Umsetzung aktueller Entwicklungen im Bereich multimediafähiger Hochgeschwindigkeitsnetze sowie in den Arbeitsplatz integrierter Konferenzsysteme. Die Erfüllung hoher Sicherheitsanforderungen ist dabei eine entscheidende Rahmenbedingung.

Haben wir Ihr Interesse geweckt? Für Auskünfte steht Ihnen Herr Dr. Thomas Gutekunst (Tel. 061 / 288 32 65; gutekunst.thomas@ch.swissbank.com) gerne zur Verfügung. Ihre vollständigen Bewerbungsunterlagen senden Sie bitte an den Schweizerischen Bankverein, Frau Regina Regenass, Personaldienst Schweiz, 4002 Basel.

Was ist Unix ?

Warum Unix?

Warum soll man Unix benutzen? Diese Frage lässt sich kaum sinnvoll beantworten, weil es wohl soviele Gründe für, als auch gegen Unix gibt (Es sind meist dieselben...). Einer ist wohl der, dass Unix für Hochschulen relativ billig ist und da der Staat eh immer sparen will...
Wie dem auch sei: Ihr werdet wohl oder übel Unix benutzen (müssen), da es auf den meisten Rechnern installiert ist.

Gross- & Kleinschreibung

Unix ist case-sensitive, d.h. es unterscheidet Gross- und Kleinbuchstaben. Alle Kommandos, die nachfolgend beschrieben werden, müssen also genauso eingetippt werden, wie sie dargestellt sind.

Das Filesystem

Unter Unix werden wesentlich mehr Attribute zu Dateien abgespeichert, als beispielsweise unter MS-DOS. Die wichtigsten sind die Dateizugriffsrechte. Jede Datei besitzt unterschiedliche Zugriffsrechte für den Benutzer, seine Gruppe und für den Rest der Welt. Es gibt dabei das Schreibrecht, welches Schreiben in eine Datei erlaubt. Desweiteren gibt es das Leserecht, das es einem Benutzer erlaubt die Datei zu lesen und damit auch sie zu kopieren. Die Kopie gehört dann demjenigen, der die Kopie erstellt hat. Als drittes gibt es das Ausführrecht, das erlaubt eine Datei auszuführen.
Die Attribute lassen sich mittels list-command mit dem Schalter `l` anzeigen: `r` steht für Read, `w` für Write und `x` für eXecute.

```
>ls -l

total 2895
drwx------  2 baumi       1024 Jun 23 10:11 Mail/
drwxr-xr-x  5 baumi       1024 Oct 30 18:04 Mailboxes/
-rw-r--r--  1 baumi       9043 Jun 20 21:21 readme.text
-rw-rw-r--  1 baumi       1593 Oct 30 16:11 log
...
```

Die ersten beiden Einträge sind Unterverzeichnisse (sichtbar am 'd'-Attribut). Das Erste hat keine Zugriffsrechte (Permissions) für jemand anderes als den User, dem die Datei gehört (hier 'baumi'). Das bedeutet, dass nur dieser User den Inhalt des Verzeichnisses lesen oder beschreiben kann. Das Zweite kann hingegen von allen gelesen werden.

Die erste Datei hat für alle User die r-permission gesetzt, kann also von allen gelesen werden, hingegen kann nur der User selbst die Datei verändern. Die letzte Datei zeigt, wie man die Rechte zur Arbeit benutzen kann: Jeder der in der gleichen Gruppe wie baumi ist, kann das log-File verändern, so dass alle die an einem Projekt arbeiten, Änderungen vermerken können. Es kann aber dennoch kein Aussenstehender einfach die Datei löschen. Die Rechte werden mit `chmod` geändert.

Zum Filesystem: Ausgehend vom Wurzelverzeichniss `/` werden alle Dateien auf der Festplatte und anderen Geräten angesprochen. Es gibt also normalerweise keine Zeichen für verschiedene Laufwerke. Man muss beispielsweise das Diskettenlaufwerk mit `/dev/fd0/` ansprechen.
Das Filesystem von Unix erlaubt jedoch symbolische Links (Verweise) auf andere Dateien und Unterverzeichnisse. Einen Link erzeugt man mit dem Befehl `ln -s x y` der eine Datei y erstellt, die auf die Datei `x` zeigt.

Wer übrigens die Diskette wiederhaben will muss den Befehl `eject` benutzen.

Die man-pages

Wie kommt man als Unix-Neuling zu Informationen, welche Befehle wie ausgeführt werden? Man benutzt die man-pages! '`man x`' ist ein Befehl, der die Online-Hilfe von Unix aufruft. Dabei werden die wichtigsten Informationen zum Befehl *x* ausgegeben. Im Prinzip findet man in den man-pages alles, was es über Unix zu wissen gibt (leider ist das mit Prinzipien so eine Sache...). Eine andere Methode: Fragt die, die es schon wissen. Das ist die vermutlich einfachste Vorgehensweise, solange die anderen nicht entnervt aufgeben... Die dritte Methode ist sich das 'Wurmbuch' zu kaufen (Der Titel kommt daher, dass auf dem silbrigen Umschlag ein Wurm gezeichnet ist, der auch im Buch selber vorkommt). Es ist das erste und beste Buch, das komplett in Deutsch zum Thema erschienen ist.

Shell's

Eine Shell ist ein Programm, das vereinfacht gesagt das Starten von Programmen und das Verwalten von Daten ermöglicht. Was man mit den Shells alles anstellen kann lest Ihr am besten bei Daniels Artikel nach.

X11

X11 steht für das X-Windows System, das auf den Suns installiert ist. Es stellt für Programme Fenster zur Verfügung und ermöglicht den

Usern ein angenehmeres Arbeiten. Es ist allerdings nicht vergleichbar mit MS-Windows, weil es für andere Aufgaben konzipiert ist. Es ermöglicht zum Beispiel einem Programm auf dem Server via Netzwerk ein Fenster auf dem eigenen Bildschirm anzuzeigen. Die eigentliche Fensterverwaltung ist in einem eigenen Programm integriert, so dass sich diese bei Bedarf auswechseln lässt.

rlogin, telnet

Mit `rlogin` und `telnet`' kann man auf einen anderen Computer zugreifen, als ob man direkt an diesem sitzen würde (naja fast...). Mit `rlogin`' muss man dabei nicht für jeden Rechner wieder ein Passwort eingeben, falls man dies so konfiguriert hat (im File `~/.rhosts`, siehe Manpage: `man rhosts`).

Ausloggen

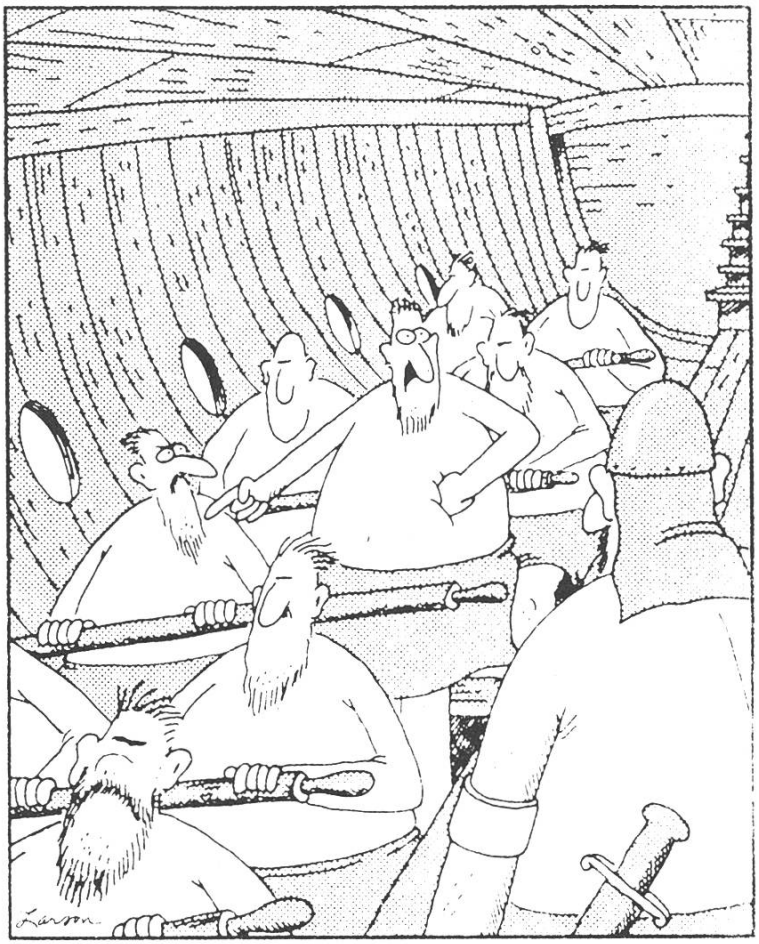
Nach dem Beenden der Arbeit kann man unter Unix nicht einfach aufstehen und den Computer abschalten. Man muss sich abmelden, damit keiner an Deinen Daten rumfummeln oder sogar mit Deinem Account weiterarbeiten kann. In einer Shell geht das mit dem Befehl `exit`, im X11-System muss man den Befehl `exit` aus dem Popup-Menü (rechte Maustaste) auswählen. Übrigens: Man kann nie sicher sein, dass nicht noch weitere User den Computer via Netzwerk benutzen, daher darf man unter keinen Umständen den Rechner ausschalten!

Weitere Quellen

Wie oben angemerkt ist das 'Wurmbuch' (also der richtige Titel ist schlicht 'Unix') eine hervorragende Quelle (da neu aufgelegt). Hilfreich sind (hoffentlich) auch die Artikel in den Visionen und vor allem für die Erstsemestrigen auch die im neuen Survival Guide.
Es lässt sich zum Thema Unix vor allem ein Tip geben: Lest die verfügbaren Texte!!! Gerade Informatiker sollten nicht den gleichen Fehler machen wie ihre Kunden: Nämlich zuerst mal die Handbücher verbrennen und anschliessend heulen: 'Es geht nicht...'

Michael Baumer
baumi@vis.inf.ethz.ch

It's your turn for the window seat!



"Mr. Mathews! Mr. Mathews! I just came back from the restroom and Hodges here took my seat!...It's my turn for the window seat, Mr. Mathews!"

Wir haben einen Platz für Dich freigehalten!

Wir sind ein junges Informatik-Ingenieurbüro im Zentrum von Zürich (Niederdorf). Zur Verstärkung unseres kleinen, dynamischen Teams (alles ETH-Ingenieure) suchen wir ab sofort:

Informatiker/in

Wir bieten:

- herausfordernde Aufgabe in modernster Entwicklungsumgebung
- inspirierendes Arbeitsklima, flexible Arbeitszeiten, auch Teilzeitstelle möglich
- überdurchschnittliches Salär

Bewerbungen oder Nachfragen bitte an:

Unitek Engineering AG
z.Hd. Matt Ackermann
Münstergasse 12
8001 Zürich
Tel. 01 - 261 55 44

UNITEK
ENGINEERING

Ressourcensparen als Informatikstudent

Halt! Überspring nun diesen Artikel nicht einfach mit dem Gedanken "Schon wieder so ein Grüner, der uns die Computerbenutzung verbieten will". Dem ist nicht so, denn Ressourcensparen heisst nicht nur Energiesparen, sondern vielfach auch, etwas schneller tun zu können, wenn man nur weiss wie. Und schliesslich dürfen wir nicht vergessen, dass es nicht selbstverständlich ist, dass wir von vielen Dienstleistungen an der ETH gratis profitieren dürfen. Internet für Studis? War nicht immer so... Gratis VPP? Bleibt vielleicht nicht immer so...

Dieser Artikel soll aber keine Einführung in all die verschiedenen Programme sein, die uns die Computer effizienter benutzen lassen. Er soll nur denjenigen, die sie schon kennen, zeigen, wie sie sie besser benutzen können.

Save a bit: Sparen mit WWW

Die populärste Anwendung im Internet ist sicher das World Wide Web (nicht umsonst spricht man von zweistelligen Wachstumsraten), insbesondere auch für diejenigen, die dieses Jahr ihren ersten Internetzugang bekommen.

Zuerst einmal ist es nicht sinnvoll, wenn für jeden Web-Benutzer an der ETH ein und die selbe Seite von weit weg extra übertragen werden muss. Dafür gibt's schliesslich Caches. Damit alle Benutzer den selben Cache verwenden können, muss dieser auf einem zentralen Rechner plaziert sein. Alle Web-Zugriffe laufen dann über diesen sogenannten `Proxy` ab. Wenn nun jemand auf eine Seite zugreift, die erst kürzlich von jemand anderem abgerufen wurde, so braucht diese nicht noch einmal vom entfernten Server geholt zu werden, sondern sie wird einfach aus dem Cache des Proxys geholt. Dadurch wird Bandbreite gespart und es geht erst noch viel schneller! Wie man's macht? Ganz einfach: Trage im Netscape im Menu *Options→Preferences* unter *Proxies* den HTTP-Proxy `proxy-inf.ethz.ch` mit Portnummer `8080` ein und setze unter *No Proxy for:* die Domain `ethz.ch`. Letzteres ist deshalb sinnvoll, weil es sich für lokale Zugriffe nicht lohnt, sie zu cachen, da man die gewünschten Seiten besser direkt vom lokalen Server holt, zumal dann Änderungen auch unmittelbar bemerkt werden.

Auf *proxy-inf.ethz.ch* stehen rund 75 MB cache zur Verfügung, was für's erste reichen sollte. Übrigens gibt's auch eine Seite, wo

noch andere Vor- und Nachteile von Proxies aufgeführt sind: `http://www.inf.ethz.ch/house_style/`.

Zusätzlich zu diesem "Allgemeincache" kann man sich auch noch einen lokalen Cache auf dem Rechner einrichten, an dem man gerade sitzt, bzw. auf dem Netscape läuft. Dies hat besonders dann Vorteile, wenn man sich über viele Links hinwegklickt und später diesen Weg wieder zurück geht. Die alten Seiten können dann einfach und schnell aus dem lokalen Cash geladen werden. Im Preferences-Menu (zu finden unter *Options*) reserviert man sich unter *Cache and Network* je etwa 5 MB als *Memory Cache* und *Disk Cache*. Um keine quota-Probleme zu bekommen, setzt man den Pfad für den Disk-Cache auf `/tmp/`*`username`* und damit Netscape dieses Verzeichnis auch immer findet, baut man noch etwas in der folgenden Art in's `~/.login` ein (Folgendes Beispiel ist in csh-Syntax gehalten. *Username* muss natürlich durch den eigenen Login Namen ersetzt werden, sonst gibt's erst recht ein Chaos!)

```
if ( ! -d /tmp/username ) then
   mkdir /tmp/username
endif
```

Ein weiterer Schritt, um die langsamen und überbenutzten internationalen Leitungen zu entlasten, besteht darin, für eine bestimmte Information den nächstmöglichen Web-Server anzufragen. Es gibt zum Teil auch für WWW-Server Mirrors (zum Thema Mirror folgt weiter unten noch mehr), so unterhält zum Beispiel Silicon Graphics (deren Server zweifelsohne von vielen Studenten ab und zu mal durchstöbert wird) eine europäische Kopie ihres Hauptservers in USA (für Interessierte: Die Adresse lautet `http://www-europe.sgi.com/`). Es bleibt die Frage: Wie findet man sowas heraus? Die einfache Antwort: Mit einer Web-search engine. Wer zu einem bestimmten Thema Informationen sucht, hat gute Chancen, mit Lycos eine grosse Zahl Verweise zum gesuchten Thema zu finden. Man wähle `http://www.lycos.com/`, gebe ein paar Stichworte ein und folge den darauf zurückgelieferten Links. Auf diese Art findet man auch gleich, wo sich der nächste Server mit der gesuchten Information befindet.

Save some more bits: Sparen mit ftp

Mit ftp (File Transfer Protocol) lassen sich bekanntlich auch grosse Files bequem um die halbe Welt transportieren. So spassig es auch sein mag, ein File aus dem entfern-

ten Japan zu saugen, so unnötig ist das in vielen Fällen. Von den meisten grossen ftp-Archiven gibt es nämlich sogenannte "Mirrors". Das sind Archive, die eine vollständige Kopie des Originals enthalten und diese täglich auf den neusten Stand bringen. Die Vorteile sind klar: Der Server mit dem Originalarchiv wird entlastet und durch das Ausweichen auf ein näher gelegenes Archiv geht auch der download schneller. Auch hier stellt sich die Frage: Wie findet man die Mirrors? Ebenso einfach ist die Antwort: Mit Archie oder auch einer Web-search engine. Archie-Server enthalten die Inhaltsverzeichnisse von bekannten ftp-Servern. Kennt man den Namen des gesuchten Files (zumindest ungefähr), startet man `xarchie &` (sitzt man an einem Terminal kann man sich auch mit `telnet archie.switch.ch` und login `archie` helfen. Gesucht wird dann mit `prog` *`filename`*). Gibt man dort diesen Namen ein, liefert einem xarchie eine Liste der Server, wo sich das File befindet. Mit *File→Get...* kann man es sich dann auch gleich noch holen (vorher aber in der Liste den nächsten Server anklicken, sonst hat's nicht viel gebracht).

Übrigens unterhält Switch (der Netzwerkbetreiber, an dem alle Schweizer Hochschulen angeschlossen sind) einen grossen Mirror, sodass man viel gesuchte Files äusserst bequem gleich um die Ecke bekommt und die internationalen Netze dabei nicht gross belastet (es ist wohl unnötig zu sagen, dass das Herunterladen von ftp.switch.ch um ein vielfaches schneller ist als zum Beispiel von einem Server im Edu-Land).

Es gibt auch ftp-Server, die ihre Dienste zusätzlich über den WWW anbieten. Deshalb findet man ein gesuchtes File manchmal auch mit Lycos.

Save a tree: Sparen beim Drucken

Zuerst einmal sollte klar sein, dass nur kleinere Dinge auf die normalen Laserdrucker in den Sun- und Mac-Räumen ausgedruckt werden sollen (also beispielsweise einzelne Übungen). Druckt man sich kurz vor Semesterende die gesammelten Musterlösungen aus, so sollte man dafür VPP (Verteiltes Printen und Plotten) benutzen, da die VPP-Drucker wesentlich leistungsfähiger sind. Ich sollte eigentlich nicht erwähnen müssen, dass Bücher zum privatgebrauch nicht über VPP ausgedruckt gehören! Bei offensichtlichem Missbrauch ist vielleicht bald Schluss mit dieser bequemen Gratis-Dienstleistung.

Aber Achtung: Wer arglos `vpp largefile.ps` eingibt, kann eine papiergewaltige Überraschung erle-

ben. VPP ist nämlich nicht intelligent genug, um selbstständig zwischen ASCII- und PostScript-Texten unterscheiden zu können. Bei falscher Handhabung kann dann eine einfache Musterlösung von drei Seiten zu einer mittleren Bibliothek mit mehreren hundert Seiten ausarten, weil der Drucker den PostScript-Source ausdruckt, anstatt das Programm auszuführen. Man merke sich also: PostScript-Texte (erkennbar an der Fileendung .ps oder an den ersten beiden Zeichen im File (`%!`)) über VPP immer mit der Option `-dev=PTS` ausdrucken! Damit lässt sich schon einmal eine Menge unnötiges Papier sparen.
Zumindest auf den HP Laserdruckern (`-dev=HPLASER`; hier braucht `-dev=PTS` nicht mehr angeben zu werden, weil dieser nur PostScript druckt) lässt es sich zudem auch noch doppelseitig drucken (mit `-form=DS`, wie DoppelSeitig), was auch die Masse des zu schleppenden Ausdrucks reduziert. Für Listings (in ASCII, Option `-dev=PRINTER`) sind manchmal auch die Parameter LISTQ und LISTH interessant, die den Ausdruck zweispaltig (quer oder hoch) erzeugen. Mit dem Kommando `psnup` lassen sich mehrere PostScript-Seiten auf einem Blatt unterbringen. Beispielsweise fasst `psnup -2 infile.ps outfile.ps` jeweils zwei Seiten zu einer zusammen. Druckt man dieses File nun auch noch doppelseitig (`-dev=HPLASER -form=DS`), so hat sich der Papierausstoss bereits auf 25% reduziert.

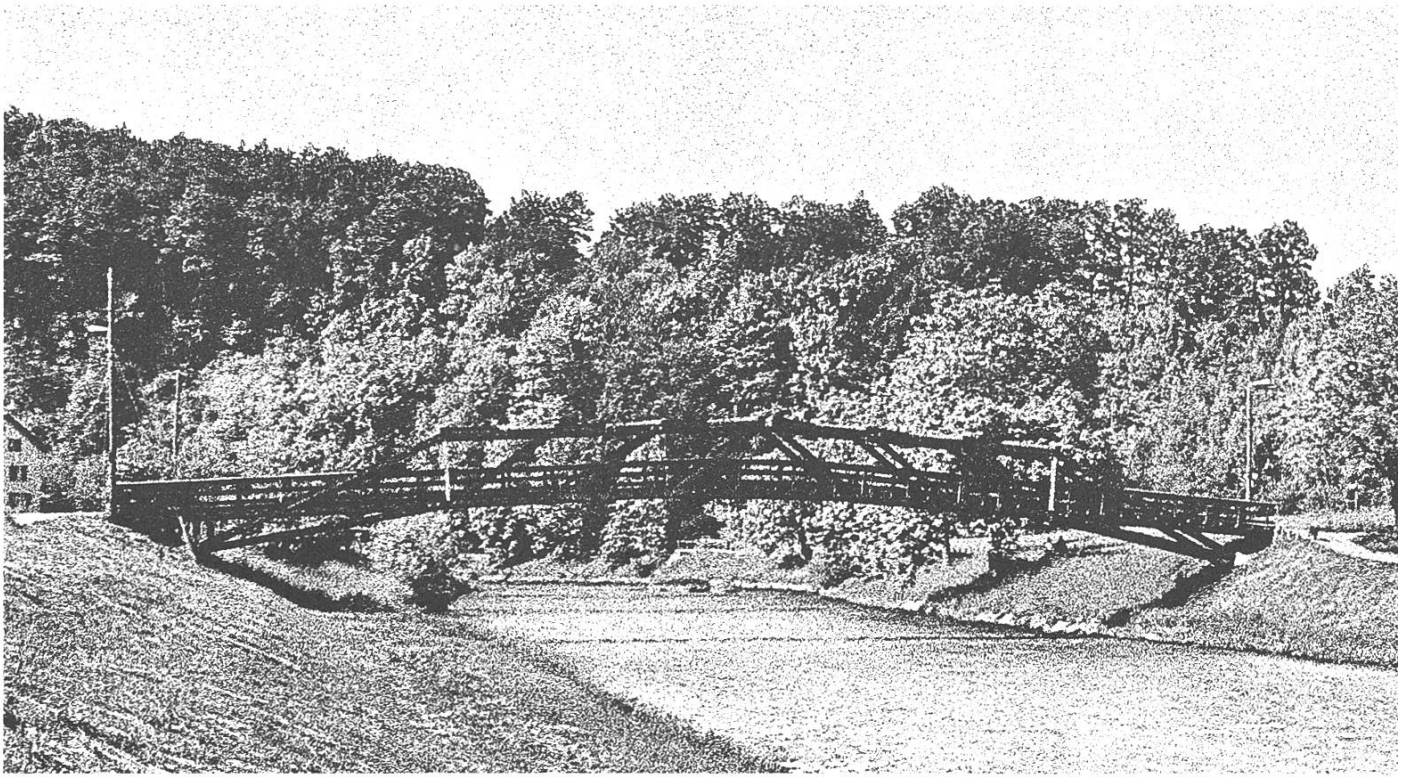
Neben den VPP-Stationen liegen übrigens Infoblätter, die als Erinnerungshilfen ganz praktisch sind und die man sich auch sonst einmal anschauen sollte, um besser über VPP Bescheid zu wissen. Wer sich die Infos lieber per WWW anschaut, der probiere `http://www.rz.id.ethz.ch/vpp/vpp.html`.

Übrigens brauchen PostScript-Tests nicht unbedingt ausgedruckt zu werden, denn mit `ghostview` lassen sie sich am Rechner schon mal bequem im Voraus anschauen. Und da PostScript eine Seitenbeschreibungssprache ist, gilt hier tatsächlich WYSIWYG (What You See Is What You Get).

Save an electron:
Bildschirme abschalten

Die Sun-Computer sollte man nie ausschalten (!), schliesslich erledigen sie auch noch Aufgaben, wenn gerade niemand vor ihnen sitzt. Anders ist es mit den Bildschirmen. Diese brauchen auch dann eine Menge Strom, wenn sie niemand braucht, deshalb sollte man sie nach Gebrauch abschalten. Das ist besonders im IFW am Abend wichtig,

Warum sich nach einem Mittagsspaziergang auch komplexe Informatik-Probleme viel flüssiger lösen lassen.

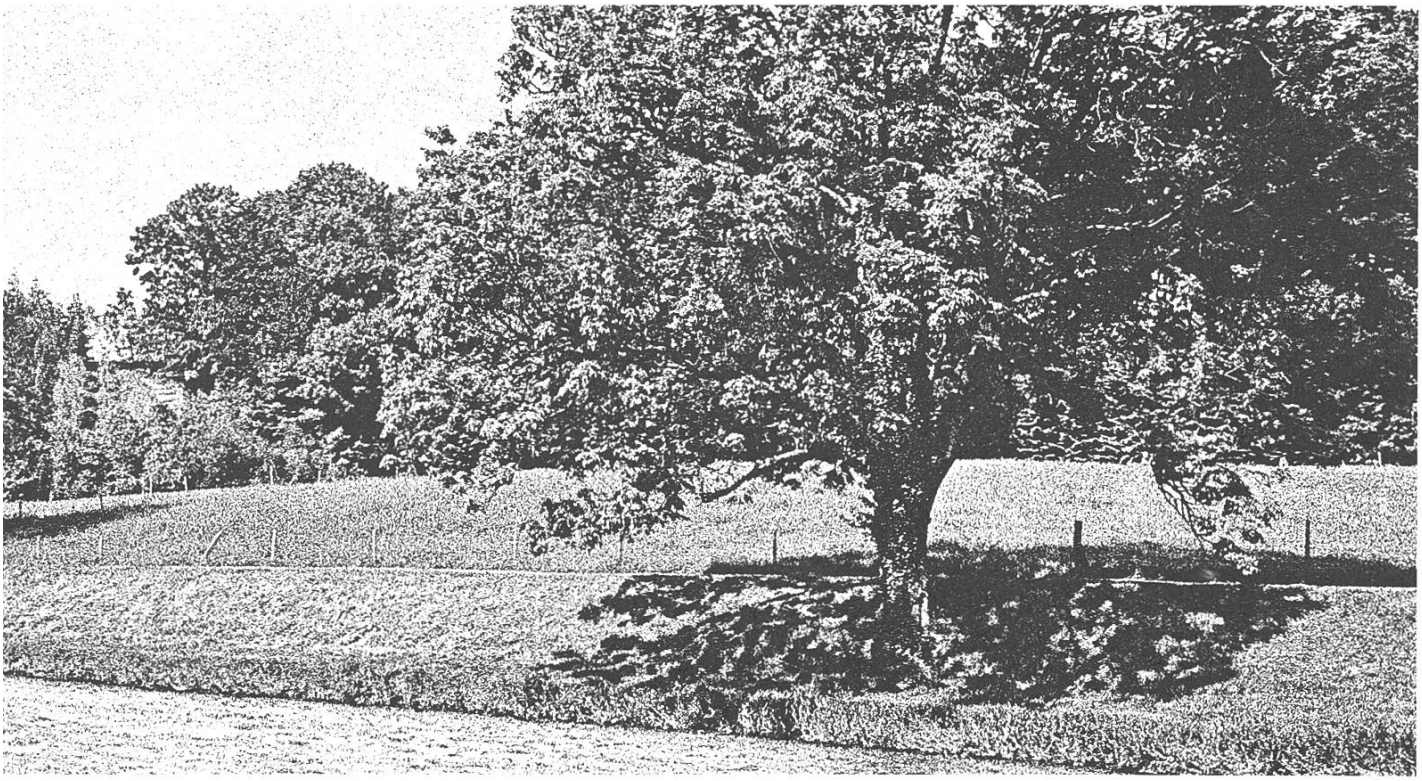


Rentenanstalt
Swiss Life

Es gibt Situationen, in denen man bei der Arbeit gedanklich steckenbleibt – man kommt einfach nicht mehr weiter.
Die Möglichkeit, an einem solchen Tag während der Mittagspause an der Sihl spazieren zu gehen und dabei Ablenkung und Entspannung zu finden, kann wahre Wunder bewirken. Entspannung von der EDV für die EDV.
Als grösste Lebensversicherungsgesellschaft der Schweiz möchten wir unseren Mitarbeiterinnen und Mitarbeitern nicht nur interessante Aufgaben und Verantwortung übertragen, sondern auch für Lebensqualität am Arbeitsplatz sorgen.

In unserem Dienstleistungs- und Rechenzentrum Sood/Adliswil steht unserer versierten Informatik-Crew ein grossartiges Angebot zur Verfügung: modernste Infrastruktur, wunderschöne Umgebung, direkte Lage an der S-Bahn-Station – ein attraktiver Arbeitsplatz im Grünen, knappe 15 Minuten vom Hauptbahnhof.
Falls Sie mehr über Ihre Einsatzmöglichkeit als Informatikspezialisten bei der Rentenanstalt/Swiss Life wissen möchten, steht Ihnen Reto Handschin, Personaldienste, Tel. 01/711'45'32, für weitere Informationen gerne zur Verfügung.

Adliswil – Informatik in nächster Zukunft



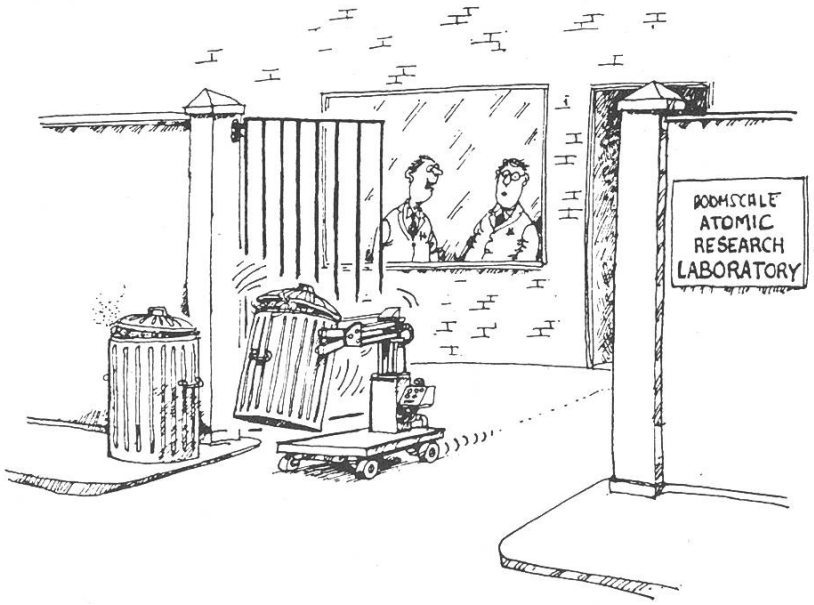
weil dies bei den dortigen Monitoren nicht automatisch geschieht. Im HG G26 übernimmt ein Zeitschaltrelais diese Funktion. Und wenn man schon dabei ist, kann man auch gleich den Bildschirm des Nachbarrechners ausschalten (natürlich nur, wenn er nicht gerade benutzt wird).

Zusammenfassung

Zum Schluss nochmals das Wichtigste in Kürze:

- **Netscape**: Proxy setzen und lokalen Cache benutzen
- **Ftp**: Mit Archie nächsten Server (mirror) suchen
- **VPP**: PostScript nicht als ASCII ausdrucken (Option -PTS merken), für Previews ghostview verwenden und möglichst viel doppelseitig drucken.
- **Bildschirme**: Nach Gebrauch abschalten

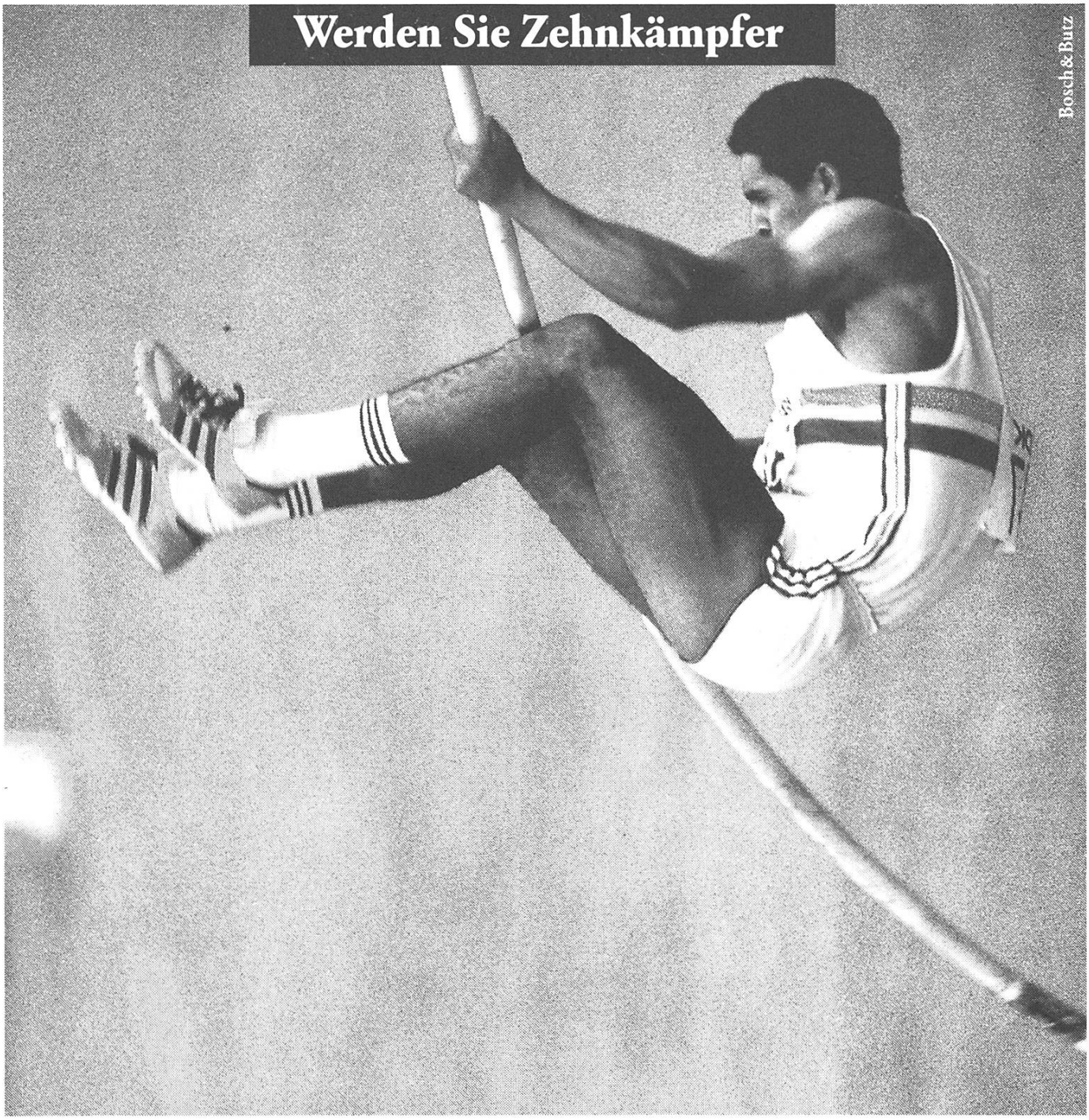
Felix Rauch (IIIC/7)
frauch@iiic.ethz.ch



'Thanks to computers we don't have to manhandle radioactive waste at all now.'

Der Karrieretip der SBG:

Werden Sie Zehnkämpfer



UBS PROFESSIONAL Setzen Sie an zu einem grossen Sprung in die Zukunft. Mit einem **Traineeship Informatik und Telekommunikation** der SBG. Starten Sie Ihre Informatik-Karriere bei einem der grössten und fortschrittlichsten EDV-Anwender der Schweiz. Hier können Sie Ihre Vielseitigkeit in den technischen Disziplinen Applikations- und Systementwicklung ausspielen. Oder Ihre Ausdauer in Automations- und Infrastrukturprojekten beweisen. Nehmen Sie die erste Hürde mit Schwung. Rufen Sie uns an.

SBG-JobInfo, Frau Irene Fuetsch: 01/236 73 01.

Wir machen mit.

Schweizerische Bankgesellschaft

Fächertest: Visualisierungstechniken (Methoden der graphischen Datenverarbeitung II)

Typ: Vertiefung
Dozenten: Prof. M. Gross
Krediteinheiten: 6

Vorlesung

Die Vorlesung ist im wesentlichen eine Fortsetzung von "Methoden der graphischen Datenverarbeitung I", sie kann aber auch einzeln besucht werden. Grob gesagt geht es um die Berechnung und Eigenschaften von Kurven, Flächen und Volumendarstellungen, wobei das Schwergewicht auf den Kurven lag. Das Tempo der Vorlesung war angemessen, zur besseren Veranschaulichung wurden häufig Videos eingesetzt. Die mathematischen Voraussetzungen waren nicht überaus hoch, aber doch eindeutig höher als bei GDV I.

Skript

Da das Skript (wie auch die Vorlesung) total neu war, enthielt es noch vereinzelt Fehler, sie waren jedoch nicht gravierend und werden nächstes Jahr bestimmt behoben sein. Es enthält in einer vernünftigen Abstimmung sowohl Text, als auch Formeln und Bilder. Im Vergleich zum letzten Skript aus GDV I ist es deskriptiver, was ein eindeutiges Plus darstellt. So brauchen während der Vorlesung kaum Notizen gemacht zu werden.

Übungen

In den ersten praktischen Übungen ging es darum, in einem teilweise vorgegebenen Framework einzelne Programmteile zu schreiben, sodass verschiedene Arten von Kurven gezeichnet werden konnten. Dies war zwar weniger interessant als der Raytracer aus GDV I, dafür vernünftiger (sowohl vom Nutzen als auch vom Aufwand her) und bringt doch auch einiges für das Verständnis des Stoffes. Später sollten dann mit dem Visualisierungstool AVS einfache Volumen dargestellt und auch einige Flächendarstellungen selbst programmiert werden. Zusätzlich zu den praktischen gab es auch noch zwei theoretische Übungen, die es einem erlaubten, sich selbst in den eher mathematischen Bereichen zu testen.

Prüfung

Die Prüfung findet in einem angenehmen Rahmen statt. Von vier gezogenen Themengebieten kann eines gestrichen werden. Zu den anderen drei muss man je rund fünf Minuten lang Rede und Antwort stehen. Prof. Gross hilft einem auf

die richtige Lösung, wenn man etwas Falsches gesagt hat oder nicht weiter weiss. Die Atmosphäre ist angenehm, man muss sich aber auch in den mathematischen Bereichen einigermassen sicher sein.

Fazit

Während es bei GDV I noch eher um grundlegende Algorithmen ging, kommt in dieser Vorlesung vermehrt die Mathematik in's Spiel. Das macht die Sache zwar ein Bischen schwerer, allerdings lassen sich nun auch ganz andere Arten von Daten visualisieren, die nicht minder interessant sind.

Hier findest Du die schönen Seiten des Studiums.

BÜCHER-LADEN
Die Non-Profit-Organisation der Studentinnen und Studenten der Universität Zürich

Musterlösungen gesucht !

Wie jedes Jahr suche ich auch diesmal wieder Studenten, die bereit sind, eine Musterlösung für diverse Fächer des 1. und 2. Vordiploms zu erstellen. Natürlich nehme ich auch Musterlösungen für Kernfächer und Fragestellungen von mündlichen Aufgaben gerne entgegen.

Haltet euch bei den Vordiplom Musterlösungen an folgende Spielregeln:

- Nehmt bis spätestens vor Weihnachten mit mir Kontakt auf. (E-Mail bevorzugt!) Es stellt sich immer wieder heraus, dass bereits jemand anders eine Musterlösung geschrieben oder ein Assistent mir eine Lösung angeboten hat.
- Ich möchte rechtzeitig die Vordiplomsammlung herausgeben. Deshalb werde ich nur Lösungen berücksichtigen, die vor dem 20.1. zu mir gelangen.

Als 'Lohn' winkt wie immer eine Einladung zum Mitarbeiteressen zu Semesterende. Ansonsten wünsche ich Euch und besonders den gut 200 Informatikstudenten, die sich mit Physik intensiver beschäftigen müssen, noch ein interessantes Wintersemester.

M. Bischof

Go!

Inserate in Uni-Zeitungen 1996

Jetzt reservieren für Trainee-Stellen, Dienstleistungen und Produkte.

Ihr Partner für umfassende Universitäts-Kommunikation.
Verlangen Sie unverbindlich die Dokumentations-Diskette!

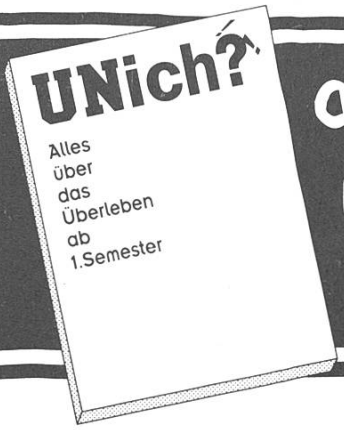
Go! Uni-Werbung AG
9014 St. Gallen
Tel. 071/272 777

Go! Uni-Publicité SA
1201 Genève
tel. 022/741 13 41

Liberty Campus

"Bekomme ich mit guten Noten mehr Stipendien?" ... ?

" ... !



aufschlagen und auf Seite 95 nachlesen!"

Alles, was Sie über die Wahl der Uni, die Formalitäten, Wohnungs- und Jobsuche, Studienpläne, Auslandsemester oder das Studentenleben generell wissen möchten, steht im UNich?!, der Überlebenshilfe für Studienanfänger. Jetzt gratis am SBG-Schalter für alle Liberty Campus-Kontoinhaber und solche, die es werden wollen. Oder für Fr. 19.80 im Buchhandel.

Liberty Campus, das Ausbildungskontopaket für Studierende bis 30 Jahre: mit mehr Zinsen, Kontoführung und Zahlungsverkehr kostenlos, Überzugslimite auf Anfrage bis zu 3000 Franken, gratis ec-Karte.

Weitere Informationen kostenlos unter Tel. 0800-88 00 88.

Wir machen mit.

Schweizerische Bankgesellschaft

Get ready for UNIX, the greatest Adventure in Computer Science

UNIX wird neben Oberon DAS Betriebssystem sein, welches für diverse Vorlesungen als Fallbeispiel oder als Arbeitsplattform eingesetzt wird.
UNIX ist vor rund 25 Jahren als Hack an der Forschungsabteilung der AT&T Bell-Labs entstanden, und hat sich seither weiterentwickelt, eigentlich eher weiter mutiert.

Der Anfänger sollte sich bei UNIX vor allem über 3 Dinge klar sein:

- Es gibt keinen undelete resp. unrm Befehl.
- Konsistenz ist ein Begriff aus einem anderen Universum.
- Es gibt keinen undelete resp. unrm Befehl.

Meist wird man heute UNIX an einem graphikfähigen Bildschirm betreiben, mit der X Windows Oberfläche. Man beachte dabei, dass auch X11 mal ein kleiner Hack an einer Uni war. Draw your conclusions.

Auch wenn unter SUN-Openwindows mit seinem File-Browser alles fast aussieht, wie auf einem Mac oder unter Windows, so wird man feststellen, dass man für vieles nicht um die Shell herumkommt. Eine Shell ist unter Openwindows übrigens ein "Cmdtool", warum man sich die Vokale von "Command" sparen wollte ist mir unklar, aber es ist typisch für UNIX.

The shell game

Ich habe vorhin kurz von der Shell gesprochen, das ist natürlich eine Vereinfachung, denn jeder Benutzer kann ein ihm gefälliges Programm benutzen, und es "shell" nennen.
Einen Standard gibt es nicht, es gibt deren zwei, denn es existieren unter UNIX zwei Shellfamilien, und von diesen wieder einige Abkömmlinge. Am Anfang war **sh** (auch bourne-shell gennant), darauf kam **csh**, und von diesen beiden entwickelten sich die verschiedenen Shells fort. sh und csh werden eigentlich nur noch von Skripts gebraucht, aus dem einfachen Grund, dass die beiden auf jedem System zu finden sind. Skripts sind Programme, welche in Wahrheit Text-Dateien sind, mit Shell Kommandi.

Die zur Zeit gebräuchlisten Shells sind bash, tcsh und zsh.

- **bash** steht für "bourne again shell" und ist demnach ein sh-Abkömling, mit einigen Erweiterungen aus dem csh Lager.

- **tcsh** ist dem Namen nach eine csh Weiterentwicklung, Unterstützung der Cursor-Tasten, und andere Verbesserungen.
- Bei **zsh** handelt es sich um einen Zwitter, es ist eine sh, welche alle Vorzüge aus tcsh auch besitzt.

zsh ist zur Zeit meine bevorzugte Shell, wobei zu sagen ist, dass ich natürlich voll UNIX-geschädigt bin, und mich mit Maus und Menus nur mühsam auf einem Computer zurechtfinde, respektive ich erachte es in den meisten Fällen als pure Zeitverschwendung, da ich weiss, was ich tun will, und wie.

Eine Shell interpretiert alle Kommandos, welche man eintippt, eine gute Übersicht hiervon findet Ihr auf der nächsten Seite.

Jedes Kommando besteht mal aus einem Namen, und beliebig vielen Argumenten, welche durch Leerzeichen voneinander getrennt werden.

```
Bsp.:
% ls -l Mail
received sent
```

The Metasyntactic Zoo

Jede Shell hat einige sogenannte Meta-Zeichen, welche nicht als Zeichen interpretiert werden, sondern eine spezifische Funktion haben. Dies sind bei den meisten Shells die folgenden:

Literal-Escape:	' " \
Filecompletion:	* ? ~ []
Redirection:	| > < `
Jobcontrol:	& %
Execution:	; ( )
History:	! ^
Variables:	$

So ich glaube, das sind alle!
Naja...je nach Shell gibt es noch mehr, oder gewisse Zeichen haben plötzlich auch eine andere Bedeutung, there's always a surprise waiting.

Escape from Absolom

Wichtig ist natürlich immer wieder, eines dieser Meta-Zeichen als Zeichen an irgend ein Programm zu übergeben, und dafür ist die erste Familie zuständig, man nennt sie im englischen "Escape-Characters".
Die Shell wird alles zwischen Hochkommatas ' ' Zeichen für Zeichen an das Programm verschenken, und sich nicht drum kümmern. Zwischen Anführungszeichen " " gesetzte Zeichenketten, werden fast wie zwischen Hochkommatas behandelt, Ausnahme bildet das Dollar-Zeichen, dieses wird als Variablen-Indikator behandelt, siehe Variablen weiter unten.

Syntax	Erklärung	Beispiele
`apropos` *`stichwort`*	Listet alle Programme auf, die etwas mit dem Stichwort zu tun haben	`apropos printer`
`man` *`befehl`*	Zeigt Manual (Erklärungsseiten) zum angegebenen Befehl	`man ls`
`cat` *`dateiname`*	Zeigt den Inhalt einer Textdatei am Bildschirm an	`cat Beispiel.Txt`
`more` *`dateiname`*	Wie cat, wartet aber, wenn der Bildschirm voll ist, auf einen Tastendruck	`more Beispiel.Txt`
`ls` *`[dateinamen]`*	Listet die Datei-Namen des aktuellen Verzeichnisses auf	`ls` `ls *.Txt`
`cp` *`data datb`*	Kopiert Datei (datb := data)	`cp A.Save A.Txt`
`mv` *`data datb`*	Benennt Datei um (move)	`mv alt.Txt neu.Txt`
`rm` *`dateiname`*	Löscht Datei (remove)	`rm Unnoetig.Txt`

`cd` *`verzeichnis`*	Wechselt das aktuelle Verzeichnis. Besondere Formen: "cd" wechselt ins Home-Verz., "cd .." ins nächsthöhere Verz.	`cd`
		`cd ..`
		`cd /bin`

`mkdir` *`verzeichnis`*	Erstellt ein Unterverzeichnis	`mkdir Uebungen`
`eject`	Wirft Diskette aus dem externen Laufwerk aus	`eject`
`alias` *`[defin]`*	So lassen sich Abkürzungen für häufig getippte Befehle definieren oder ansehen	`alias` `alias c 'ls *.c'`
`quota`	Gibt Überblick über den Dir zustehenden Festplattenspeicher Deines Accounts	`quota -v`
`logout oder exit`	Beendet Deine Unix-Session	`logout`

Der Backslash \ schützt nur gerade das Zeichen hinter ihm, ist also dazu da, einzelne Zeichen zu escapen. Man kann natürlich auch den Backslash mit sich selbst escapen.

```
Bsp.:
% echo '$PATH'
$PATH
% echo "$PATH" "&&"
/usr/bin:/usr/ucb:/bin:. &&
% echo \"
"
```

sure you want to delete all the files?

Wie auch in MS-DOS, kann man in UNIX bei Befehlen anstatt aller Dateinamen, einen kryptischen Ausdruck eintippen, welcher dann von der Shell ausgewertet wird, und so durch alle Namen, welche darauf passen, ersetzt wird.

Zuerst einmal die Regeln:

?	ersetzt genau ein beliebiges Zeichen.
*	ersetzt beliebig viele Zeichen, wobei beliebig auch 0 Zeichen einschliesst.
[...]	ersetzt eines der Zeichen, welches zwischen den eckigen Klammern steht.

```
Bsp.:
% ls -d /???
/bin /dev /etc /lib /tmp /usr
% ls /v*
/vmunix
% ls -d /[bd]??
/bin /dev
```

smoking pipes

Eine der herausragensten Möglichkeiten von UNIX ist die, dass Programme miteinander verkettet werden können, und die Ausgabe von dem einen

Programm die Eingabe des nächsten ist. Anstatt dass die Eingabe eines Programmes von der Tastatur kommt, und die Ausgabe auf den Bildschirm geht, kann man beides auch auf Dateien umlenken. Die Varianten:

`% ls /bin | more`	lenkt die Ausgabe des Programmes `ls` auf die Eingabe des Programmes `more`.
`% ls > bla`	schreibt die Ausgabe von `ls` auf die Datei `bla`.
`% more < bla`	lässt `more` die Eingabe von der Datei `bla` lesen.
`% ls >> bla`	hängt die Ausgabe von `ls` an die Datei `bla` an.
`% ls 'cat bla'`	übergibt die Ausgabe von `cat bla`, oder was immer zwischen den Apostrophen ' ' steht, als Argumente.

Dies sind die wichtigsten Methoden, aber es gibt noch weitere Kombinationen von Sonderzeichen, welche auch ihre Bedeutung haben, selbstverständlich in Abhängigkeit von der verwendeten Shell.

no job

Da es sich bei UNIX um ein Multiuser Multiprozessing System handelt, kann man Befehle auch im Hintergrund ausführen. Ein Befehl mit einem Ampersand & am Schluss der Kommandozeile wird im Hintergrund ausgeführt, und man bekommt die Prozess-Nummer und die Job-Nummer zurück. Dieser Prozess läuft dann im Hintergrund weiter, und verschenkt allfällige Ausgaben dem Benutzer, sofern dieser die Ausgabe nicht wie oben umgeleitet hat.
Die csh hat dann auch Möglichkeiten eingeführt, jobs zu kontrollieren:

`% fg %1`	bringt den Job mit der Job-ID 1 in den Vordergrund, d.h. so als ob man ihn ohne & gestartet hat.
`% bg %1`	lässt den gestoppten Job Nummer 1 wieder im Hintergrund weiterarbeiten.
`% jobs`	listet alle zur Zeit existierenden Jobs einer Shell.

Falls man irgend ein Programm gestartet hat, welches z.B. mehr Zeit braucht, als man erwartet hat, und es im Hintergrund weiterlaufen lassen will, kann man das sehr einfach tun: Durch drücken von ^Z (Control-Taste und z gleichzeitig drücken) kann man das Programm suspendieren. Mit `bg` kann man den gestoppten job im Hintergrund weiterlaufen lassen.

```
Bsp:
% xterm
^Z
Stopped
```

G L A N C E

Software Engineering

Als eigenständiges Schweizer Ingenieurunternehmen mit 24 Mitarbeitern versteht sich Glance AG vor allem auf

- das Erbringen von Entwicklungs- und Beratungsdienstleistungen im Bereich innovativer Informationssysteme, sowie
- die beratende Unterstützung bis hin zur schlüsselfertigen Realisierung von technischen Software-Projekten wie Sanierung, Überarbeitung und Erneuerung bestehender Systeme, das Re-Engineering.

Wir arbeiten für namhafte Kunden aus der Industrie und dem Dienstleistungssektor mit einem vielseitigen Projektportfolio, welches unter anderem die Bereiche Electronic Document Management, Anlage-Portfolio-Management, Medizinische Informationssysteme und Gebäudeleittechnik umfasst.

Zur Verstärkung unseres Teams suchen wir initiative, selbständig arbeitende

Entwicklungs-Ingenieure

mit Interesse an anspruchsvoller Arbeit im Umfeld von grafischen Benutzeroberflächen (MS Visual C++, XVT DSC++), relationaler sowie objektorientierter Datenbanken (ODBC, Oracle, Objectivity) und offenen Plattformen (Unix, Windows).

Sie arbeiten mit bei der Erstellung von Spezifikationen, Entwurf und Konzeption, Realisierung, Test sowie Dokumentation unter Anleitung eines erfahrenen Projektleiters.

Wir bieten moderne Arbeitsmittel in kollegialer Atmosphäre sowie ein interessantes und breites Betätigungsfeld mit Freiraum für Initiative und Eigenverantwortung – und nicht zuletzt Weiterbildung. Der Arbeitsplatz liegt im Grünen und ist 2 Minuten von der S-Bahn (S5) entfernt.

Rufen Sie uns an, wenn Sie mehr über diese Stelle wissen möchten, oder senden Sie Ihre Unterlagen an Frau E. Käser.

GLANCE AG
Software Engineering
Gewerbestrasse 4, 8162 Steinmaur, Telefon 01 853 39 49

```
% bg
[1] xterm &
% jobs
[1] + Running xterm
```

you're history

Seit der Existenz der csh wurde die Möglichkeit eingeführt mittels weiterer Sonderzeichen auf bereits eingetippte Befehle zuzugreifen, und so nachträglich Tippfehler auszubessern.
Falls die Shell auf ein Ausrufezeichen ! stösst, wird sie annehmen, dass die darauffolgenden Zeichen der Anfang eines Befehls ist, an welchen sich die Shell erinnern kann. Es gibt auch weitere Modifiers zur sogenannten History-Completion, auf welche ich nicht eingehen will, nur darauf, dass die Folge !! den letzten Befehl nochmals ausführt.

```
Bsp.:
% ls
Mail bin
% !! -Fs
ls -FS
total 2
1 Mail/ 1 bin/
```

Variablen Inkonsistenz

Die Variablen-Geschichte ist in den Shells inkonsistent gelöst, und hat unterschiedliche Syntax, je nach Shell-Familie. Daneben darf man noch unterscheiden zwischen Variablen, welche intern zur Shell gehören, und zu Umgebungs-Variablen, dem sogenannten Environment, auf welches nicht nur die Shell zugreift, sondern welches jedem Kommando der Shell verschenkt wird.
Auf Environment Variablen zugreifen ist einfach, die Shell wird einfach alles was von der Form $variablename ist, durch den Wert der Variablen ersetzen.
Das Setzen der Variablen geht auf verschiedene Weise:

```
csh:  setenv VARIABLE wert
 sh:  VARIABLE=wert
      export VARIABLE
zsh:  export VARIABLE=wert
```

Dies ist eine kleine Einführung, welche auf nicht alle Details eingeht, Fortsetzung folgt in den nächsten Visionen. Viel Spass !!!!!

Daniel G. Kluge
daniel@vis.inf.ethz.ch

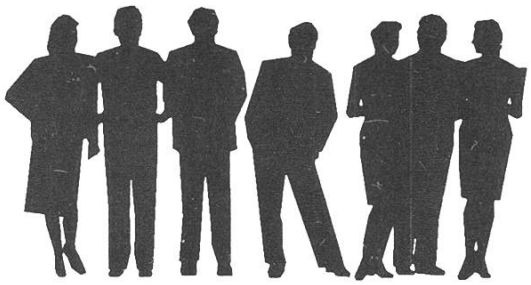
Referenzen:

- Manpages von csh, sh, zsh
- The UNIX-Haters Handbook, Simson L. Garfinkel
- Learning by doing, jeder selbst

„TAG DER OFFENEN TÜR“

DIE RENTENANSTALT/SWISS LIFE LÄDT EIN!

AM FREITAG 24. NOVEMBER 1995
von $13.^{30}$ bis ca. $17.^{00}$ Uhr
Informatikcenter, Soodmattenstr. 4, 8134 Adliswil



Die Teilnehmerzahl ist auf ca. 25 Leute ***(ab 5. Semester)*** beschränkt!

Das Programm sieht folgendermassen aus:

$13.^{30}$ Begrüssung durch Herrn Dr. E. Marmier
mit anschliessendem Referat zu den Themen:
- RA/SL-Gruppe
- Einbindung/Aufgaben RA/SL-Informatik
- zukünftige Informatikstrategien

$14.^{15}$ Vorstellung einzelner Projekte durch die betreffenden Projektleiter in den Bereichen
- Kommunikation- & Netzwerkprogramme
- Infrastrukturprogramme
- Kommerzielle Softwareentwicklung

Ihr erhält einen Einblick ins „daily business“ der RA/SL-Informatik, wie z.B. die Realisierung der einzelnen Projektphasen und welche technischen Umgebungen zur Verfügung stehen.

$16.^{00}$ gemeinsamer Apéro im Personalrestaurant, dabei bietet sich Euch die Gelegenheit, mit Projekt- und Abteilungsleitern/innen in ungezwungener Atmosphäre zu diskutieren.

Anmelden könnt Ihr Euch ab Semesterbeginn im VIS-Büro (IFW B29) während den Öffungszeiten jeweils von Mo-Fr von $12.^{15}$-$13.^{00}$ und zu den üblichen Bedingungen, d.h. mittels Hinterlegung eines Depots von Fr. 20.--.

G.A.B. 6648 Minusio

Falls unzustellbar bitte zurück an:

Verein der Informatikstudierenden
IFW B29
ETH-Zentrum

CH-8092 Zürich

Inhalt

Salü zäme **S. 3**
Telecom 95: Ingenieurtag vom 10.10.95 **S. 4**
ACM-Wettbewerb – Reminder **S. 6**
Was ist Unix ? **S. 8**
Ressourcensparen als Informatikstudent **S. 12**
Fächertest: Visualisierungstechniken (GDV II) **S. 20**
Musterlösungen gesucht ! **S. 22**
Unix Adventure **S. 24**