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

Herausgeber: Verein der Informatik Studierenden an der ETH Zürich

Band: - (1995)

Heft: 5

Heft

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. 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. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

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. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

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. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

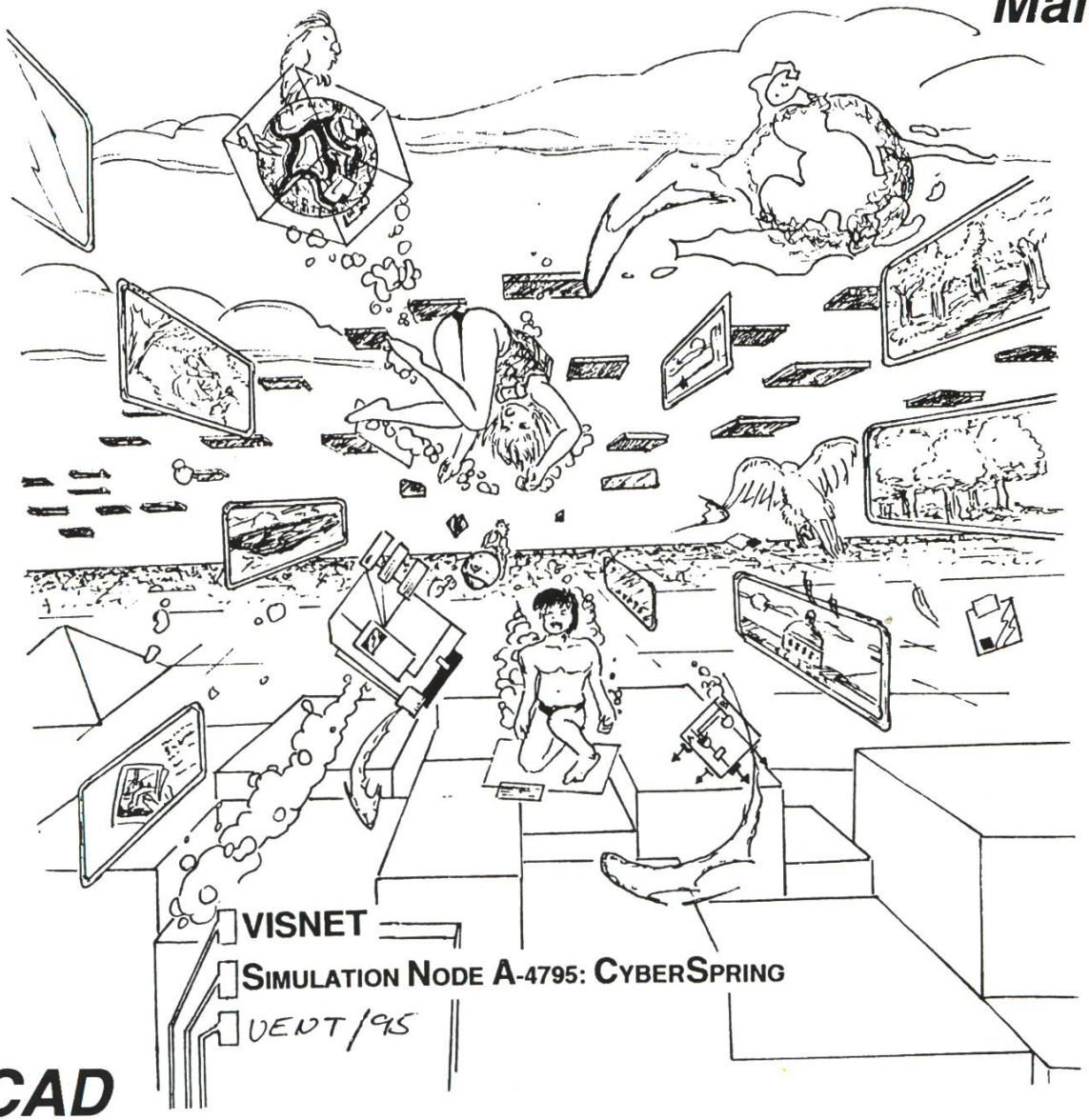
Download PDF: 20.08.2025

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

Visionen

5

Mai 95



CAD
USENET und tin
Diplomandenumfrage (Teil2)

VISKAS

Visionen Herausgegeben vom Verein der Informatikstudierenden an der ETH Zürich

Adressen

Aktuar: Christian Limpach
Mainaustrasse 44, 8008 Zürich
Tel.: 01 / 422 27 13
e-mail: chris@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

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

Verleger: Christian Widmer
St. Gallerstrasse 30, 9403 Goldach
Tel.: 071 / 41 13 73
e-mail: rezi@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.

Verleger: Christian Widmer
Redaktor: Nathalie Weiler

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: 1200
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, 18. Mai 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

Salü zäme

Die Prüfungen sind vorbei, und für die meisten hat das Semester wieder voll eingeschlagen: Vorlesungen, Übungen und was der Studentenfreuden mehr sind. Der VIS hat auch die ordentliche Mitgliederversammlung bereits hinter sich, mit einem rekordverdächtigem Tiefstand was die Anwesenden betrifft. Da hat auch alles Locken mit Spielabend nichts gefruchtet. Die Ankündigunsfrist von 5 Tagen über einem verlängerten Wochenende waren wohl etwas zu kurz, damit die Information auch den Hinterletzten erreicht.

Die MV wird seit diesem Semester jeweils am Anfang des Semester stattfinden, v.a. deshalb weil wir es leid sind beim Rekrutieren von Mitarbeitern immer "Jaaaaaa, nach den Prüfungen vielleicht"zu hören.

Mit dieser MV sind leider auch wieder zwei Vorstandsmitglieder von uns gegangen: Christian der 1., seines Zeichens Aktuar und Redaktor wird in seinem Heimatland Luxenburg das Praktikum absolviern, und deshalb seinen Vorstandspflichten nicht mehr nachkommen können. Unser noch taufrischer Verleger, Christian zum 2., hat mit Studium und anderen Engagements genug um den Kopf, so dass der Vorstandsjob über die Klippe springen musste.

Ich möchte den beiden für ihr Engagement ganz herzlich danken! Es waren beides Vorständler, welche sich immer wieder mal über ihr Ressort hinaus für den VIS eingesetzt haben.

Zum Glück haben sich wieder zwei mehr oder weniger bekannte Gesichter für den Vorstand entschieden. Christian Berger nimmt einen zweiten Anlauf für den Vorstandsjob, er war ja letztes Semester für einige wenige Wochen im Vorstand, und hatte sich nach dem ACM-Stress verabschiedet dann; welcome back. Dann konnten wir dem AMIV ihren Festminister abjagen. Gregor Battilana hat sich nach seinem Wechsel zum Informatikstudium nun auch zu einem Mitwirken im "richtigen' Fachverein entschieden, welcome aboard. Er wird die Feste von VIS und AMIV etwas koordinieren, denn anscheinend herrscht auch drüben ein Motivations- und Personalnotstand. Da die Feste einen fast identischen Kundenstamm haben, macht eine Zusammenarbeit sicher Sinn.

Apropos Feste, hier noch die wichtigsten Daten für aktive Studenten: Das allerseits beliebte VISKAS findet am Donnerstag 15. Juni (Fronleichnam ? SO WHAT !?) und das

AMIV Waldfest am 3. Juli statt.

War das jetzt wieder "das trockenste Vorwort das ich je gelesen habe" (Orginalzitat, und nicht von mir!)

Urteile selbst!

Bis sonstwann,

-daniel

Aus der Abteilung

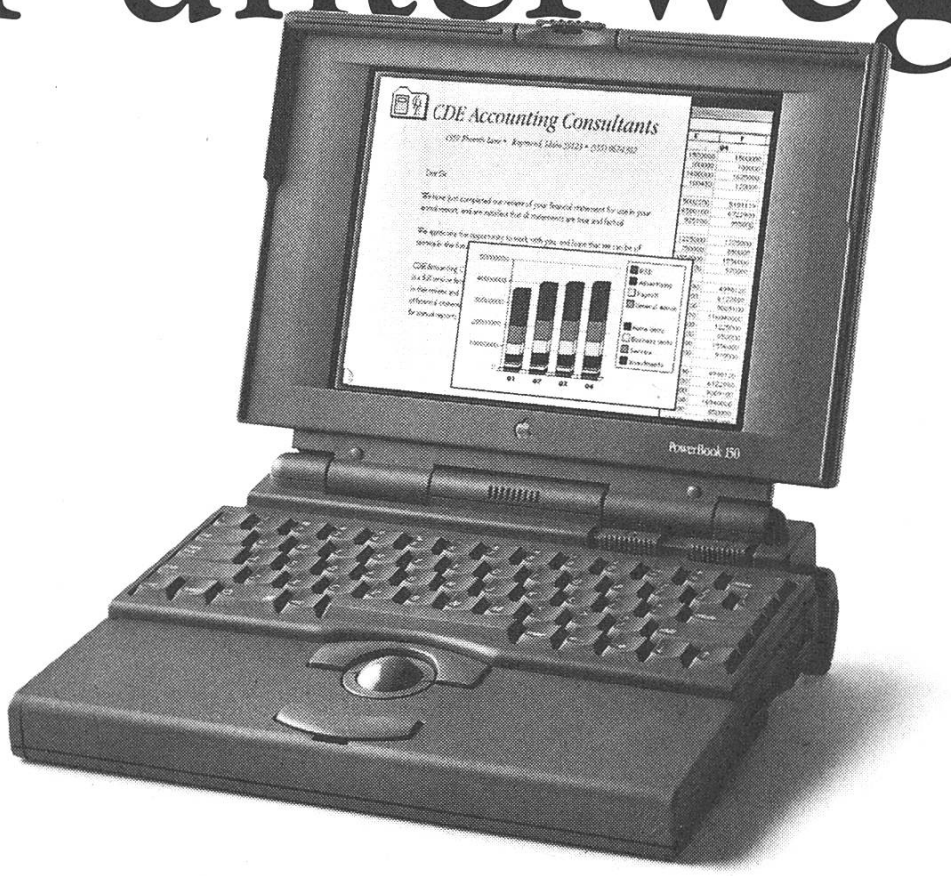
Die Prüfungsanmeldekarten für die Prüfungssession Herbst 1995 können in der Woche **vom 8. bis 12. Mai** beim Abteilungssekretariat bezogen werden.

Letzter Anmeldetermin beim Rektorat,**Freitag, 2. Juni 1995**

Abteilungsekretariat für Informatik

Ein paar Chips für unterwegs.

SCHMINKE, GABRIEL & TEAM



PowerBook 150

Das Gelernte in die Welt hinaustragen – nichts leichter als das. Und nichts günstiger als das. Denn das Apple PowerBook 150 kostet für Studenten weniger als 1590.– Franken. Das sind ein 68030 Prozessor mit 33 MHz Taktfrequenz, 120 MB auf der Festplatte, 4 MB Arbeitsspeicher und ein umfassendes Softwarepaket ... gleichmässig verteilt auf 2.5 kg original Apple Macintosh. Absolut tragbare Zustände im Hörsaal, in der Bibliothek und in der Badi. Wer nicht den Doktor machen will, holt sich das Ding beim nächsten offiziellen Apple Vertreter schnell ab.

Apple

Generalvertretung für die Schweiz und Liechtenstein:
Industrade AG, Apple Computer Division, Education Team, Hertistrasse 31, 8304 Wallisellen, Tel. 01-832 81 11

Aus dem Ressort Unterricht und Prüfungen

Zuerst einmal "Salut im neuen Semester!". Diese Prüfungssession ist vorbei, doch die Nächste kommt nur allzu schnell. Deshalb suche ich schon jetzt Studentinnen und Studenten,

- die bereit sind, eine Musterlösung zu erstellen. Zwar habe ich noch keine Angabeblätter, aber ihr könnt jetzt schon mit mir Kontakt aufnehmen. Als Belohnung gibt es wie gewohnt die Einladung zum Mitarbeiteressen des VIS am Semesterende.
- die zu mündlichen Prüfungen angetreten sind. Schreibt oder mailt mir doch einen kurzen Prüfungsbericht! Dieser muss nicht unbedingt praezise alle Fragen enthalten. Schon eine kurze Beschreibung (Prüfungsatmosphäre, Art der Fragen, Reaktion auf Fehler, etc.) kann für eine Prüfungsvorbereitung dienlich sein. Auch Prüfungsberichte über nicht gerade zu exotischen Nebenfächer sind willkommen. Ich bin gerade dabei, den ganzen Schlussdiplomordner neu zu strukturieren. Neu: Ihr könnt auf dem WWW-Server des VIS nachsehen, ob es zu einem Vertiefungs- oder Nebenfach bereits eine Prüfungssammlung gibt.
- Selbstverständlich publiziert der VIS weiterhin Fächertests (auch über Nebenfächer) und ist dankbar, solche von Euch zu erhalten.

Michael Bischof
Ressort Unterricht und Prüfungen

Für Studentinnen und Assistentinnen:

Mittagessen in der IFW-Infobar

Jeden 1. Mittwoch im Monat, 12 Uhr

3. Mai, 7. Juni, 5. Juli

Nora Sleumer (IFW E 45.2, sleumer@inf.ethz.ch)

Ulrike Stege (IFW A 48, stege@inf.ethz.ch)

Fächertest: CAD: Geometrie und Datenstrukturen

Typ: Vertiefung
Umfang: 2V, 1U, 6 Kreditpunkte
Dozent: Prof. M. Engeli
Besuchte Vorlesung: WS 94/95

Vorlesung

Die Vorlesung behandelt im wesentlichen die folgenden Themen:

- Aufbau eines CAD-Systems und Schnittstellen mit anderen Systemen (insbesondere Fertigung, CAM)
- Allgemeine geometrische Probleme (z.B. Durchstosspunkt Gerade-Ebene, Schraffur einer Fläche, Kegelschnitte, Transformationen)
- Geometrie im Computer
- Mathematische Darstellung von Kurven, Flächen und Körpern
- Beispiel eines CAD-Systems: EUKLID

Der Kern der Vorlesung behandelt die mathematische Darstellung von Kurven und Flächen, die sich zu einem Körper zusammensetzen lassen. Folgende Darstellungen werden behandelt: Bezier, rationale Bezier, B-Spline und rationale B-Spline (insbesondere NURBS (Non-Uniform Rational B-Spline)).

Prof. Engeli ist in der Gestaltung der Lektionen flexibel und reagiert sehr gerne auf Fragen der Studenten. Da Prof. Engeli über eine ausgeprägte Erfahrung in der Industrie verfügt, ist er auch der richtige Ansprechpartner für Fragen, die nicht nur den Stoff betreffen, wie z.B. welche Probleme bei der Zusammenarbeit zwischen Konstruktion und Fertigung in einem Betrieb auftreten können.

Der Besuch der Vorlesung erfordert keine speziellen Vorkenntnisse. Der Student sollte aber über räumliches Vorstellungsvermögen und eine gewisse Resistenz gegenüber mathematischen Formeln verfügen.

Skript

Es existiert kein Skript. Zu einzelnen Themen werden Blätter abgegeben. Es empfiehlt sich, in den Lektionen anwesend zu sein, und seine eigenen Notizen zu machen. Zusätzlich gibt es zahlreiche Literatur (z.T. sogar in der Info-Bibliothek) zu den Themen.

Übungen

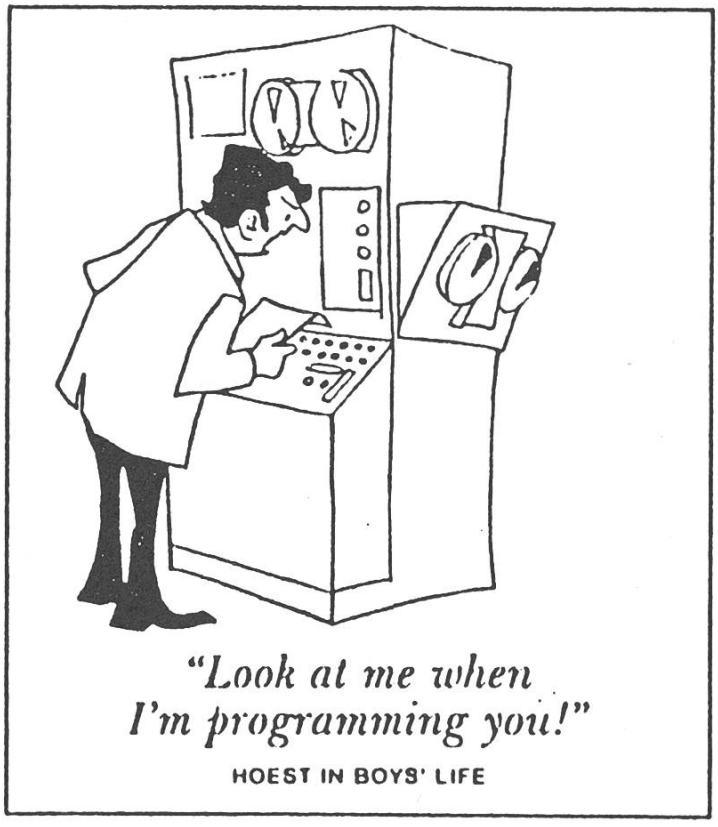
Es gibt sowohl Programmier- als auch Papierübungen, für die man jeweils zwei Wochen Zeit hat. Die Übungen sind recht knifflig und verlangen meistens zusätzlich zum

Vorlesungsstoff einen Trick für die Lösung. Zu allen sieben Übungen dieses Semesters wurden aber aufschlussreiche Musterlösungen abgegeben.

Prüfung
Die Prüfung ist mündlich und dauert 15 Minuten. Die Fragen verlangen neben Kenntniss des Stoffes vorallem auch eigene Überlegungen. Prof. Engeli prüft fair und hilft, wenn es nicht mehr weitergeht.

Bemerkungen
Im SS hält Prof. Engeli die Vorlesung CAM Fertigung. Beide Vorlesungen können unabhängig voneinander besucht werden, es empfiehlt sich jedoch der Besuch beider Lehrveranstaltungen.

Leider sind CAD und CAM neuerdings nicht mehr auf der Liste der Lehrveranstaltungen der Abteilung IIIc. Ich kann mir keinen Grund für die Streichungen vorstellen. Beide Vorlesungen beinhalten eine gesunde Mischung zwischen Theorie und Praxis, was an unserer Abteilung sicher nicht immer behauptet werden kann.



Kommentar zur VLSI-Prüfung

In der letzten Prüfungssession absolvierte ich im Rahmen des zweiten Teils meiner Schlussdiplomprüfungen unter anderem eine Doppelprüfung im Fach "VLSI"; genauer gesagt liess ich die Vorlesungen "Architektur von VLSI-Schaltungen" und "Entwurf von VLSI-Schaltungen" mündlich prüfen.

Zu Beginn der Prüfung schauten mich die beiden Prüfer (Herr Käslin und Herr Felber) schräg an und meinten: "Du hast bei uns keine Semesterarbeit gemacht!". Dass ich geduzt wurde, machte mir keine Probleme, ich duzte einfach zurück! Doch das Wort "Semesterarbeit" setzte in mir Erinnerungen frei an die Vorlesungen, wo in fast jeder Stunde gesagt wurde, dass man in der Semesterarbeit (Chip-Designprojekt) auf dies und jenes achten soll. Offenbar ist es üblich, dass man auch eine Semesterarbeit dort macht, wenn man die Vorlesungen besucht. Ich hatte aber meine Semesterarbeit in einem anderern Institut absolviert, und hoffte zu Beginn der Prüfung, dass dies keine Auswirkung haben würde! Als nächstes sagten die Prüfer: "Du bist von der Abteilung IIIc, oder?" und schauten mich noch schräger als zuvor an! Ich erinnerte mich da sofort an die zahlreichen Bemerkungen der Dozenten, die während den Vorlesungen über die Informatiker gemacht wurden. Die Hauptzielgruppe dieser Lehrveranstaltungen sind wohl die Elektrotechniker, denn auf dem Titelblatt des Skripts steht unter anderem "Abteilung für Elektrotechnik" und kein Wort über die Informatiker.

Der erste Teil der Prüfung handelte über Vor- und Nachteile von ASICs und mikroprogrammierbaren Prozessoren. Die Frage war allerdings so formuliert, dass ich erst nach 10 Minuten wusste, was die beiden von mir wissen wollten. Das meiste hab ich dann auch gesagt (leider nicht ganz alles).

Im zweiten Teil der Prüfung wurde ich über etwas gefragt, was gar nie behandelt wurde (ich besuchte jede Vorlesung!) und auch nicht im Skript steht. Die Prüfer behaupteten zwar, es stünde im Skript. Möglicherweise steht es drin für die Vorlesungen im SS94,WS94/95. Ich besuchte aber die beiden Vorlesungen im SS93,WS93/94. Trotzdem wusste ich einiges über dieses Thema (es ging um Ausgangs-Treiber und damit verbundene Probleme), auch wenn Herr Käslin und Herr

Felber viel nachfragen mussten, bis ich sagte, was sie hören wollten.

Diese Doppelprüfung ergab dann die Note 3.5.
Natürlich fragte ich mich, warum ich in VLSI eine so schlechte Note bekommen hatte. An meiner Unfähigkeit lag es wohl kaum, da ich in den übrigen Prüfungen einen Schnitt von 5.29 hatte. Am Fleiss lag es auch nicht, da ich überdurchschnittlich viel für dieses Fach gelernt hatte. Auch die Vorlesungen hatte ich immer besucht und die (aufwendigen) Übungen fast alle (bis auf eine!) gemacht!
Ich vermute, dass Studenten, die keine Semesterarbeit in diesem Fach gemacht haben, in den Prüfungen ziemlich benachteiligt werden. Was der Grund dafür sein könnte, ist mir schleierhaft! Und wenn man dazu noch Informatiker ist, und kein Elektroingenieur, dann kann man sich eine allfällige gute Note sowieso in den Kamin schreiben!
Ich möchte nun allen zukünftigen Absolventen einer VLSI-Prüfung raten, die Semesterarbeit in diesem Fach zu machen oder Elektrotechnik zu studieren!

Albin Meyer
IIIc/DS

Anmerkung des VIS:
Wir wären froh, wenn wir weitere Personen die diese Prüfung abgelegt haben, uns ihre Erfahrungen (positive und negative) mitteilen würden. Ihr erreicht uns während der Präesenzstunden im VIS-Büro oder via e-mail an vis@inf.ethz.ch.

"I liked it better when you read the paper at breakfast."

JOSEPH FARRIS IN SUCCESS

Eine Einführung in USENET und tin

Begriffe wie USENET, netnews oder schlicht news kennt wohl jeder. Der richtige Umgang mit diesen Dingen und die effiziente Bewältigung der enormen Informationsmengen ist aber nicht unbedingt einfach. Insbesondere dann nicht, wenn man keinen guten Newsreader kennt und beherrscht. Diese Probleme sollen hiermit beseitigt werden.

Was ist das USENET überhaupt?
Am ehesten könnte man das USENET mit einem verteilten bulletin board system vergleichen. Es ist unterteilt in verschiedene Themenbereiche, sogenannte Gruppen (groups, oder auch newsgroups), von welchen es im Moment rund dreitausend geben dürfte, und umspannt die ganze Welt. Innerhalb einer Gruppe werden verschiedene zum Namen der Gruppe passende Themen diskutiert, sogenannte Threads. Ein Thread hat einen Titel (der Titel des ersten zu ihm gehörenden Artikels) und besteht meistens aus einer Anfrage und eventuell einigen (1 bis in seltenen Fällen über hundert) Antworten und kann sich in mehrere Subthreads aufspalten. Die meisten Newsreader (Programme, um die news möglichst komfortabel lesen zu können) sortieren die Artikel einer Gruppe leider nur nach ihrem Datum, dabei ist es viel angenehmer die Artikel, die zu einem bestimmten Thread gehören, gerade nacheinander lesen zu können. Diese Möglichkeit bieten sogenannte threaded newsreader", wie z.B. trn oder eben tin.

Vorbereitung von tin
Zuerst einmal muss man tin sagen, woher die News zu beziehen sind. Für Studenten liegen sie auf neptune.ethz.ch, also ist folgender Befehl in einer shell (csh, tcsh) auszuführen:

```
setenv NNTPSERVER neptune
```

Die Syntax für sh, bash und zsh ist:

```
set NNTPSERVER=neptune export NNTPSERVER
```

Damit tin weiss, dass es die News nicht auf dem lokalen Rechner findet, startet man das Programm mit der Option **-r** (remote) oder verwendet den Namen rtin anstelle von tin. Der Pfad von tin ist `/usr/local/bin`, der komplette Befehl zum starten von tin ist also: `/usr/local/bin/rtin` Wenn man `/usr/local/bin` in seiner PATH-Variable schon eingefügt hat (z.B. in der Datei `/.cshrc`), genügt **rtin** Zudem empfiehlt es sich, auch die obigen Zeilen zum Setzen des NNTPSERVERs in die jeweilige startup-Datei der eige-

nen Lieblingsshell einzutragen, damit man sie nicht jedesmal von Hand wieder eingeben muss. Nachdem tin erfolgreich gestartet wurde, heisst einen das Programm ersteinmal herzlich willkommen. Nach dem Drücken einer beliebigen Taste erscheint ein mehrheitlich leerer Bildschirm. Falls man nur einige ganz bestimmte Newsgroups lesen möchte, deren Namen man schon kennt, kann man diese nun direkt mit **'g'** anwählen. Dazu jedoch später mehr. Für all diejenigen, die sich im USENET noch nicht auskennen und somit auch noch nicht wissen, welche Gruppen sie überhaupt lesen sollen, empfiehlt es sich, tin sogleich wieder mit **'q'** zu verlassen. Das Programm hat nämlich eine Datei namens `/.newsrc` angelegt, in der alle im Moment auf diesem Server verfügbaren Newsgroups drin stehen.

Die Qual der Wahl

Nun lädt man sich die Datei `/.newsrc` in einen Editor und schaut sie sich einmal an. Die ersten paar Zeilen sehen dann etwa so aus:

```
alt.3d!
alt.abortion.inequity!
alt.activism!
alt.activism.d!
...
```

Zuerst steht jeweils der Name der Newsgroup, dann ein Kennzeichen dafür, ob es sich um eine vom Benutzer abonnierte Gruppe handelt (':') oder um eine, die ihn nicht interessiert ('!'). Danach kommen noch die Nummern der bereits gelesenen Artikel. Da bis jetzt noch keine Gruppe angewählt wurde, steht nach den Ausrufezeichen nichts und da erst recht auch noch nichts gelesen wurde, sind auch keine Artikelnummern vorhanden. Um nun aus den über 2300 Gruppen die auf neptune gespeichert sind, die interessanten zu finden, sucht man am besten mit der Suchfunktion des Editors nach bestimmten Wörtern, z.B. nach 'mac', 'unix' oder nach was es einem auch immer gelüstet (einen etwas besseren Überblick erhält man, wenn man die Datei vorher sortiert. Sie sieht zwar sortiert aus, ist es aber nur am Anfang). Meistens findet man auf diese Art mehr Gruppen, als einem lieb ist. Folgende Aufstellung der verschiedenen Gruppen-Hierarchien soll eine kleine Hilfe sein, damit man sich in der riesigen Vielfalt leichter zurechtfindet:

alt.*: Alternatives. In dieser Hierarchie ist es besonders leicht, neue Groups zu installieren, deshalb gibt es einige etwas merkwürdige oder ungewöhnliche Namen. Hier findet man häufig Fans von Themen, die in keine der anderen Gruppen passen (z.B. alt.fan.james-bond).

bit.listserv.*: Bitnet-Mailinglisten.

biz.*: kommerzielle Angebote.
ch.*: Schweiz-spezifische Themen.
comp.*: Hier wird es für Informatiker interessant, denn es geht um alles rund um Computer. Einige der wichtigeren Unterhierarchien sind: **comp.lang.*** (Computersprachen), **comp.os.*** (Betriebssysteme), **comp.sys.*** (verschiedenste Systeme wie Amiga, Atari, PC, DEC, SUN, HP, SGI, NeXT,...), **comp.unix.*** (allgemeines zu verschiedenen UNIX-Versionen).
de.*: Alles nochmals in etwas kleinerem Rahmen für den deutschsprachigen Raum.
ethz.*: nicht sehr viele, aber für ETH-Angehörige wichtige Angehörige.
gnu.*: Wichtiges rund um die GNU-Utilities (emacs, gcc, ...)
misc.*: Miscellaneous. Verschiedenes wie z.B. Verkaufs- und Job-Angebote.
news.*: Netnews über sich selber (Organisation des USENET, Ankündigungen, Missbräuche, Statistiken). Für Neulinge dürfte insbesondere `news.newusers.questions` von besonderem Interesse sein. Wenigstens am Anfang sollte man hier ein paar Mal hereinschauen, um etwas mehr über's USENET zu erfahren.
rec.*: Hier findet man vor allem Diskussionen über nicht Computer-abhängige Hobbies wie Kunst, Autos, Spiele, Musik und Sport.
sci.*: Science. Wissenschaftliche Themen.
soc.*: Society, d.h. Kulturelles aus den verschiedensten Ländern und religiöse Themen
talk.*: Hauptsächlich Politik und Philosophie.
inf.*: Einige lokale Groups, die nur innerhalb der ETH gelesen werden können und vor allem für Informatiker wichtig sind. Studenten sollten wenigstens `inf.students` und `inf.info` regelmässig lesen, um über Änderungen an den Systemen rechtzeitig informiert zu sein.
Um die gefundenen Themenbereiche nachher im tin lesen zu können, ersetzt man das Ausrufezeichen am Ende der entsprechenden Zeile durch einen Doppelpunkt. Jetzt tin wiederstarten und es geht richtig los!

Erste Schritte mit tin

Hat man vorher (wie oben beschrieben) einige Gruppen in der Datei `/.newsrc` mit einem Doppelpunkt angewählt, so sieht der Hauptbildschirm von tin nun schon etwas interessanter aus. Zuoberst steht, wo man sich gerade befindet ("Group Selection"), sowie die Anzahl angezeigter Gruppen. Ein eventuell vorhandenes 'R' bedeutet, dass nur die Gruppen angezeigt werden, in denen sich ungelesene Artikel befinden. Darunter sind Zeilenweise die

abonnierten Gruppen in folgender Form:

```
Nummer Artikel Name
[Beschreibung]
```

Die einzelnen Spalten bedeuten:

`Nummer`: Nummer der Gruppe (für direkte Anwahl, siehe weiter unten).

`Artikel`: Anzahl der noch nicht gelesenen Artikel. Diese Zahl dürfte jetzt bei einigen Groups noch erschreckend hoch sein. Allerdings werden dabei auch Artikel mitgezählt, die schon längst gelöscht wurden.

`Name`: Der Name der Gruppe.

`Beschreibung`: Eine ganz kurze Beschreibung, worum es in der Gruppe geht. Zum Teil fehlt diese Beschreibung.

Im unteren Teil des Bildschirms befindet sich ein kleines Hilfsmenu. Es ist übrigens überall im Programm möglich, mit 'h' eine Liste aller momentan gerade möglichen Befehle und ihre Bedeutung anzuzeigen. Die wichtigsten davon im Hauptmenu sind: Pfeil hoch und Pfeil runter (bzw. 'k' und 'j') zum Verschieben des Cursor-Balkens und 'Return' zum Anwählen einer Gruppe. Hat man mehr Gruppen abonniert als auf einer Seite Platz haben, so kann man entweder, wie gerade beschrieben, durch die Liste scrollen, mit 'Space' seitenweise vorwärtsgehen, oder mit 'g' und dem genauen Namen einer Gruppe direkt zu ihr hin springen. Ist endlich eine interessante Gruppe gefunden, kommt man mit 'Return' zum Group-Level, wo man einen Überblick über die gewählte Gruppe erhält. Zuerst muss tin allerdings noch alle Artikel indizieren, damit sie nachher leichter zu ordnen sind. Die auf den rifrafs installierte Version von tin ist leider nicht auf dem neuesten Stand, deshalb kann sie die auf dem Server bereits vorhandene Vorsortierung der Artikel nicht ausnützen. Besonders beim ersten Anwählen einer Gruppe kann das ein Weilchen dauern (Es ist auch möglich, tin einmal mit **'rtin -u -v'** aufzurufen, um alle Gruppen auf einmal zu indizieren oder diesen Befehl in der privaten crontab einzutragen; die Indizierung wird dann, z.B. über Nacht erledigt). Später wird dieser Vorgang nicht mehr so lange dauern, da jeweils nur noch die neuen Artikel bearbeitet werden müssen.

Das Menu auf Gruppen-Stufe sieht ähnlich aus wie das Hauptmenu: Zuoberst steht der Name der Gruppe, sowie einige Zahlen. Die wichtigen sind die vor dem 'T' und vor dem 'A'. Sie zählen die Anzahl angezeigter Threads bzw. Artikel. Ein allfälliges 'R' zeigt wiederum an, dass nur die ungelesenen Artikel aufgeführt sind. Darunter folgen zeilenweise die einzelnen Threads mit folgendem Zeilenformat:

```
Nummer [+] [Antworten] Titel
Autor
```

Die Bedeutungen sind:

`Nummer`: Die Thread-Nummer zum direkten Anwählen.

`+`: Hier hat es noch ungelesene Artikel.

`Antworten`: Anzahl ungelesene Artikel, falls es mehr als einen in diesem Thread gibt. Wird keine Zahl angezeigt, so handelt es sich um einen einzelnen Artikel ohne (ungelesene) Antworten.

`Titel`: Der Titel des ersten Artikels in diesem Thread (die nachfolgenden unterscheiden sich meist nur durch ein vorangestelltes 'Re: ').

`Autor`: Der Name oder die e-mail Adresse des Autors.

Darunter ist wieder ein kleines Hilfsmenu, diesmal mit geringfügig anderen Befehlen. Auch hier kann man mit Pfeil hoch und Pfeil runter (bzw. 'k' und 'j') die zur Auswahl stehenden Threads anwählen oder mit 'Space' seitenweise durch die Liste blättern. Mit 'Return' zeigt man den ersten Artikel eines Threads an, und zwar auch dann, wenn man ihn schon gelesen hat. Auf diese Art wird man zuerst nocheinmal daran erinnert, worum es in dem Thread geht. Mit 'Tab' überspringt man den ersten Artikel eines Threads (falls man ihn bereits gelesen hat) und kommt direkt zum ersten ungelesenen. Zur besseren Übersicht über einen Thread kann man ihn auch anwählen und sich dann mit 'l' eine Liste aller darin enthaltenen Artikel ausgeben lassen.

Auf der Artikel-Stufe sieht der Bildschirm etwas anders aus als vorher. In der obersten Zeile steht das Datum und die Zeit, zu der der Artikel abgeschickt wurde (bereits umgerechnet in die lokale Zeitzone), der Name der aktuellen Newsgroup und die Nummer des Threads, zu dem der Artikel gehört. Die nächste Zeile enthält die Anzahl Linien des Artikels, seinen Titel (invers) und wieviele Antworten darauf bereits geschrieben wurden. Der unterste Teil des Seitenkopfes ist dem Autor des Artikels vorbehalten, er enthält dessen e-mail Adresse, seinen Namen und die Organisation, von wo der Artikel abgeschickt wurde (z.B. 'ETH Zürich'). In der Mitte ist ein Ausschnitt des Artikels und zuunterst wieder das kleine Hilfsmenu (welches sich übrigens mit 'H' aus- und einschalten lässt). Ganz unten wird angezeigt, wieviel Prozent des Artikels bereits gelesen wurden und das Verhältnis gelesene/totale Anzahl Bytes. Mit 'Space' und 'b' scrollt man eine Seite weiter bzw. zurück. Am Ende eines Artikels gelangt man mit 'Space' zum nächsten Artikel eines Threads, oder, falls es der letzte Artikel war, direkt zum nächsten Thread. Die Tabulator-Taste führt immer (nicht

nur am Ende eines Textes) zum nächsten ungelesenen Artikel. Diesen sucht tin ebenfalls zuerst im aktuellen, dann im nächsten Thread.

Beantworten von Artikeln

Beim Lesen eines Artikels gibt es noch zwei weitere wichtige Funktionen, die man kennen und nicht verwechseln sollte: Persönliches (reply, Taste 'r') und öffentliches (followup, 'f') Beantworten eines Artikels. Eine persönliche reply wird als e-mail direkt an den Autor geschickt, während ein followup in allen Gruppen landet, in denen auch der aktuelle Artikel verschickt wurde (meistens ist das nur eine, man kann Artikel aber auch in mehreren Gruppen verschicken). Der Text geht dann durch Tausende von Rechnern auf der ganzen Welt (ausser bei lokalen Gruppen) und belastet dabei sowohl Netzwerk-Kapazitäten als auch unzählbare CPUs und Festplatten. Man sollte sich deshalb gut überlegen, ob es nicht auch genügen würde, eine Antwort als e-mail zu geben. Viele Fragesteller im USENET bieten jeweils an, man solle die Antworten direkt an sie schicken, sie würden dann das Ergebnis, falls gewünscht ein paar Tage später in einem followup zusammenfassen. Dies ist häufig eine gute Methode, da e-mails wesentlich weniger Resourcen brauchen als USENET-Artikel. Nachdem man sich mittels 'r' oder 'f' zum Beantworten einer Frage oder zum Ergänzen einer Antwort entschlossen hat, wird sofort der Editor gestartet, der in der Environment-Variable EDITOR der shell steht. Im Normalfall ist es angenehmer für die anderen Leser, wenn noch ein paar Zeilen des vorherigen Artikels im Text stehen, damit sie den Zusammenhang besser sehen. Beantwortet man mit den Tasten 'R' oder 'F', so wird der alte Text gequotet, d.h. ganz in den Artikel übernommen, aber zusätzlich wird vor jeder Zeile ein Doppelpunkt (lässt sich mittels Voreinstellungen auch in das gebräuchlichere '¿' ändern) eingefügt, damit man die zitierten Stellen auch als solche erkennt. Aber Achtung: Quoten will gelernt sein. Es ist sinnlos und gilt im USENET als unhöflich (gelinde gesagt), einen langen Artikel vollständig zu quoten und am Schluss einen winzigen Kommentar wie z.B. "me too" anzufügen. Man sollte sich gut überlegen, welche Stellen für's Verständnis wirklich wichtig sind und den Rest wieder löschen.

Ist man fertig mit der Antwort, hat den Text abgespeichert und den Editor verlassen, so steht man vor drei Möglichkeiten:

Abbruch ('q' wie quit), wenn man

sich's anders überlegt hat und die Antwort doch nicht abschicken möchte,
Editieren ('e'), falls einem noch etwas eingefallen ist und man den Text nachträglich noch ändern möchte, und
Abschicken ('s' wie send bei einem reply und 'p' wie post bei einem followup).
Im Inhaltsverzeichnis einer Gruppe wird ein eben abgeschickter Artikel allerdings noch nicht erscheinen. Es dauert meist einige Minuten, bis der Artikel auf dem Server in die Gruppe aufgenommen und weitergeschickt wurde.

Starten eigener Threads

Bis jetzt habe ich nur beschrieben, wie man auf bereits existierende Artikel antwortet. Wie aber beginnt man einen eigenen Thread? Eigentlich ist es recht einfach: Man sucht sich die passende Gruppe (mit scrollen oder 'g' im Gruppen-Menu) und tippt ein 'w' (write). Dann muss man einen Titel (Subject) eingeben, der möglichst passend und kurz ist. So aussagekräftige Überschriften wie "Help me"werden wohl von vielen Leuten gar nicht erst gelesen, weil sie nur diejenigen Artikel anwählen, die sie vom Titel her interessieren und den grossen Rest schlicht überspringen. Zu lange Titel sind auch nicht gut, weil sie in der Liste der Threads sonst nicht mehr vollständig zu lesen sind. Ist der Titel eingegeben, wird der Editor gestartet, wobei ein Teil des Headers bereits vorgegeben ist.
An der ersten Zeile (Subject) gibt es eigentlich nichts mehr zu ändern, es sei denn, es ist einem in der Zwischenzeit ein besserer Titel eingefallen. In der zweiten Header-Zeile steht die Gruppe, in der der Artikel verbreitet wird. Man kann auf dieser Zeile auch noch weitere Gruppen angeben, aber man sollte das nur tun, wenn es unbedingt nötig ist! Einen Artikel, der auf diese Weise in mehreren Gruppen verschickt wird, nennt man "Cross-posting". Massives cross-posting von Artikeln (oder gar wiederholtes posten eines Artikels in jeder Gruppe einzeln) in unpassenden Gruppen ist im USENET äusserst unbeliebt und kann zu ziemlich bösartigen Reaktionen führen. Wenn man zum Beispiel seinen PC gegen einen Mac eintauschen möchte, ist es durchaus angebracht, diesen Artikel in eine PC- und eine Mac-Gruppe zu crossposten (aber nicht in alle PC und alle Mac-Gruppen!). Cross-postings sind auf jeden Fall besser, als den gleichen Artikel mehrmals in verschiedenen Gruppen von neuem zu verschicken, da cross-postings nur einmal übertragen und gespeichert werden (in den Gruppen der Newsserver werden einfach Pointer auf den Artikel eingetragen).

Zudem werden cross-postings in allen Gruppen, in denen sie verschickt wurden, als gelesen markiert, wenn man sie in einer davon gelesen hat.
Die dritte Header-Zeile ist für das Gebiet der Verbreitung des Artikels zuständig (Distribution). Meistens kann man hier einfach "world" eintragen, wodurch der Text weltweit überalldorthin verschickt wird, wo die jeweilige Gruppe auch empfangen wird. Möchte man einen Artikel nur gerade auf dem benutzten Newsserver lokal speichern und nicht weiterverbreiten, kann man hier auch "local" eintragen. Die letzten beiden Zeilen ("Summary:" und "Keywords:") sind freiwillig und werden nicht von allen Newsreadern unterstützt. Es emfpiehlt sich vielleicht trotzdem, hier ganz kurz etwas passendes einzutragen. Nachdem der Header richtig vervollständigt wurde, kann man den eigentlichen Text schreiben. Nach dem Speichern und Verlassen des Editors kommt wieder die bereits oben beschriebene Frage, was nun mit dem Text geschehen soll.

Nun müsste eigentlich das wichtigste klar sein. Wer seine neu erlernten Fähigkeiten mit einem Test-Artikel ausprobieren möchte, der kann das in der Gruppe **alt.test** tun. Falls im Titel nicht "ignore" steht, wird man in den nächsten Tagen von Mails aus aller Welt überschwemmt werden, die einem bestätigen, dass der Artikel richtig verschickt wurde. Da die News-Installation aber eigentlich schon funktioniert, genügt es, einen Test-Artikel nur lokal zu verbreiten.

So, dies muss für einen ersten Einstieg ins USENET und den Newsreader tin reichen. In einer der nächsten Visionen werde ich noch ein paar fortgeschrittene Themen behandeln. Für Fragen, Kritik und Anregungen stehe ich selbstverständlich gerne zur Verfügung.

Felix Rauch (IIIc/6)
frauch@iiic.ethz.ch

Zen oder die Kunst, ein Netzwerk zu verwalten

Mit der praxisorientierten Realität wird man ja im Informatikstudium denkbar selten konfrontiert. Da bieten sich aber, neben dem obligatorischen Praktikum, verschiedene Möglichkeiten, das nachzuholen. Eine zum Beispiel ist, während dem Semester oder in den Ferien in einer Firma zu arbeiten. Das kann zwar interessant sein, bringt Geld, aber dieser Stress während man eigentlich Übungen lösen oder Prüfungen vorbereiten sollte? Eine andere Möglichkeit wäre...

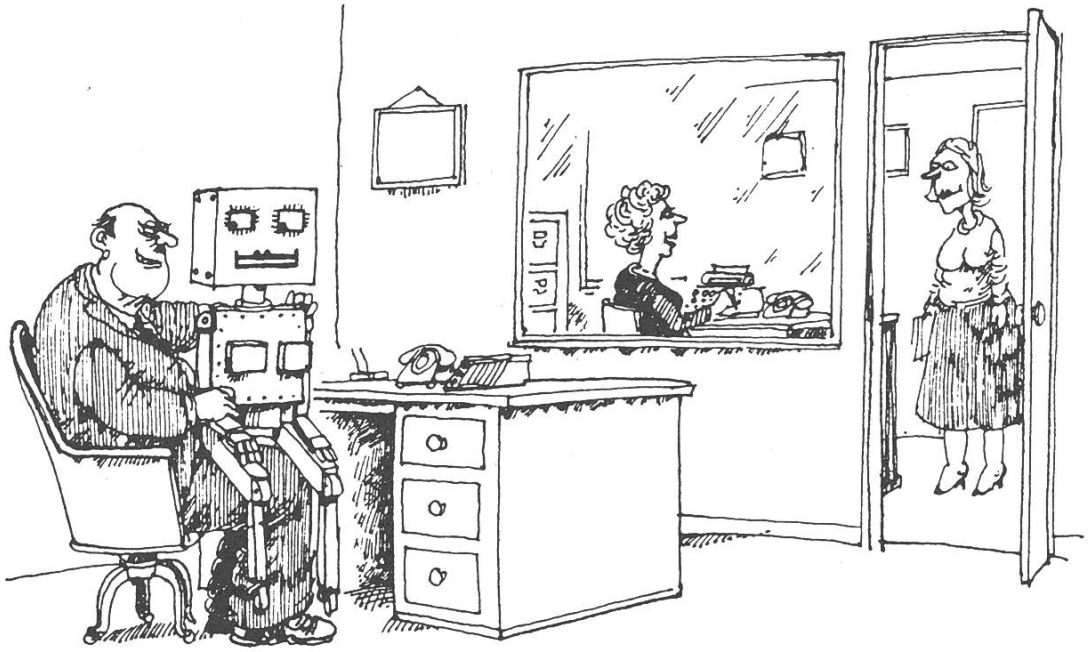
And now for something completely different. Der Betrieb an der ETH ist ziemlich stur, leistungsbetont, aufs Studium ausgerichtet, ohne viel Raum für das Leben nebenher. Da bieten sich aber diverse Möglichkeiten, das nachzuholen. Eine zum Beispiel ist, jeden Abend im Bonnie Prince zu hocken, oder im Kino, vor dem Computer oder im E 23, oder jede organisierte Party zu konsumieren. Das kann zwar ganz amüsant sein, kostet meistens Geld, aber immer so auf die eher einsame und passive Tour? Eine andere Möglichkeit wäre...

And now for something completely different. Bei unserer Ausbildung kommt ein wichtiger Aspekt viel zu kurz: Lernen, wie man im Team arbeitet, wie man sich in eine heterogene Gruppe eingliedert und gemeinsam Lösungen schafft. Da bieten sich aber mehrere Möglichkeiten, das nachzuholen. Eine zum Beispiel ist die Arbeit in einem der Studi-Gremien in Sachen Politik, Mensa, SAB, Krankenkasse usw. Ganz bestimmt sehr lehrreich, aber naja, ist doch alles ein bisschen trocken, oder? Eine andere Möglichkeit wäre...

And now for something really completely different. KOSTA ist die Gruppe von ETH- und Uni-Studis (sowie Ehemaligen), die all das Erwähnte vereinigt: Eine inhomogenere Gruppe ist kaum denkbar, und trotzdem schaffen wir es laufend, gemeinsam Dinge auf die Beine zu stellen, die jeder oder jedem einzelnen von uns prächtigste Alpträume verschaffen würden, müsste er oder sie alleine dafür geradestehen. Was für Dinge? Ganz verschieden, von den Gamenights für ein paar hundert bis zum Polyball für einige tausend. Aktivitäten sind also wichtig bei uns, wer für sich, seine Ideen, handwerklichen und organisatorischen Talente eine Spielwiese sucht, ist bei der KOSTA genau richtig. Wir sind die, die machen. Aber ins Kino gehen oder ab und zu eine Party als Gäste genies-

sen, das tun wir natürlich auch. Und die Informatik-Praxis? Nun, unser EDV-Chef denkt an Pensionierung (sein Inf.-Ing.-Diplom winkt am zeitlichen Horizont), unser PC-Netz und seine BenutzerInnen brauchen also demnächst eineN neueN AufpasserIn. Die Erfahrungen in Sachen Administration, Hard- und Software-Taktiken (für Strategien sind wir noch zu klein), Benutzerunterstützung und -ansprüche, die Du hier auf einem überschaubaren und einigermassen unkritischen System (mit sehr toleranten Usern) machen kannst werden Dir später nicht nur zukünftige Arbeitgeber hoch anrechnen. Also pack die Chance! Und wenn Dich KOSTA auch sonst interessiert: Wir erzählen Dir gerne mehr über uns, sobald Du Dich bei uns meldest, telefonisch (252 77 20), per Fax (252 70 64), E-Mail (info@kosta.vseth.ethz.ch) oder Schnäggeposcht (KOSTA, Leonhardstrasse 25a, 8001 Zürich). Bis bald!

KOSTA/Polyballkommission



'I'm all for computers, they do all the dull, boring jobs for you.'

Diplomandenumfrage (Teil 2)

In diesem zweiten Teil des Berichtes über die Diplomandenumfrage geht es vor allem um zwei Dinge: Was zeichnet die Informatikabteilung aus, was sollte verändert werden? Wie praxisorientiert ist unsere Ausbildung, und wie stellen sich die Absolventen ihr zukünftiges Berufsleben vor?

Das Studium - Positive Aspekte

Was gefiel den Informatikern an ihrem Studium? Zum einen wurde die Vermittlung des breiten Basiswissens gelobt. Allgemein sind die Diplomanden dankbar, ein Grundlagenwissen auf hohem Niveau erhalten zu haben. Die Strukturierung des gesamten Studiums mit seinen zahlreichen Komponenten (Grundstudium, Kernfächer, Vertiefungsfächer, Nebenfach, Semester- und Diplomarbeit, Praktikum, usw.) wurde auch positiv hervorgehoben. Einerseits kann man Schwerpunkte selbst setzen, andererseits ist aber ein breites Basiswissen garantiert. Wie schon in der letzten Ausgabe angesprochen wird das Nebenfach als sehr sinnvoll erachtet. Es trägt zusammen mit dem grossen Angebot an Vertiefungsfächern (wurde mehrfach erwähnt) zur Vielseitigkeit des "technischen" Studiums Informatik bei. Das im allgemeinen hohe Niveau dieser Vertiefungsfächer und deren Aktualität macht das Fachstudium attraktiv. Besonders interessant fanden viele Diplomanden ihre Semesterarbeiten und die abschliessende Diplomarbeit. Das Praktikum, das Einblick in die Industrie geben soll, wurde auch mehrfach positiv hervorgehoben. Ebenso Lob erntete der "Betrieb ETH", der eine kurze Studiendauer ermöglicht, die im Vergleich zu den anderen deutschsprachigen technischen Hochschulen einmalig ist. Die Hochschule und insbesondere auch das Departement Informatik geniessen gerade in Technikerkreisen hohes Ansehen, nicht zuletzt wegen ihrer oft international bekannten Professoren. Ein Diplomand erwähnte hierzu noch die gute Ausstattung, etwa im Bereich Computerhardware. Dass ein Kandidat den VIS als d a s Beste am Informatikstudium bezeichnet, darf natürlich nicht unerwähnt bleiben.

Das Studium - Negative Aspekte

Selbstverständlich gibt es auch Punkte, welche die Diplomanden an ihrem Studium bemängeln. Der eine Punkt sind die hohen Durchfallsquoten in den Vordiplomen, besonders beim "gefürchteten" zweiten Vordiplom. Fairerweise

muss man sagen, dass diese in letzter Zeit doch wieder etwas gesunken sind, aber dennoch sicherlich hoch sind. Ein anderer Punkt betrifft konkrete Unterrichtsfächer. Das Thema Computergrafik im allgemeinen, Graphical User Interfaces im speziellen, wird in keiner obligatorischen Vorlesung angesprochen, geschweige denn eingehender behandelt. Ebenso hört der "Durchschnittsstudent" nie etwas über Fuzzy-Logik, neuronale Netzwerke, genetische Algorithmen und Jackson Diagramme. Diese Punkte wurden zumindest explizit erwähnt. Das Thema Berechnungstheorie, Rekursivität und Komplexität fanden einige Studenten in der jetzigen Form als zu ausführlich und zu abstrakt behandelt. Ein weiterer Kritikpunkt ist die Auswahl von gelehrten Betriebssystemen und Programmiersprachen. Das "System" Oberon, Ceres, NS 32000 steht (zum Teil auch verständlicherweise) im Vordergrund. In der restlichen Zeit versucht man aber, zu viele andere Betriebssysteme und Programmiersprachen zu behandeln. Unix & C werden niemals wirklich systematisch erarbeitet.
Es trat auch der Wunsch nach mehr persönlicher Beratung und Betreuung durch den Professor auf, vor allem was die Wahl der Vertiefungsfächer betrifft. Aber auch so etwas wie "Berufsberatung" sollte es an der Hochschule geben. Erfreulich ist schliesslich, dass 85 Prozent der Diplomanden das Informatikstudium an der ETH weiterempfehlen würden.

Praxisorientierte Hochschule?
Jetzt zu einem anderen Thema. Obwohl nicht explizit gefragt wurde wie praxisorientiert die Diplomanden die Ausbildung an unserer Abteilung eingeschätzen, gab es hierzu zahlreiche Stellungnahmen. Ich sehe ganz klar, dass der Punkt Praxisbezug einen grossen Stellenwert bei der Beurteilung des Studiums einnimmt. Positiv wurde, und hier wiederhole ich mich, die Breite der Ausbildung inklusive Nebenfach, Semesterarbeiten und Praktikum erwähnt. Trotzdem ist deutlich zu spüren, dass sich die Mehrheit der Diplomanden ein etwas mehr auf die Industrie ausgerichtetes Studium wünschen und zwar in zwei Richtungen. Die eine betrifft die bessere Auswahl von Informatikmitteln. Darunter verstehe ich den vermehrten Einsatz von praxisnahen Programmiersprachen, Betriebssystemen und Softwaretools.
Die zweite Richtung ist die organisatorische Komponente. Hierbei wurde auf das Fach Informatik-Projektentwicklung aufmerksam gemacht, das zwar das "wichtigste Fach überhaupt" sei, aber momen-

tan viel zu oberflächlich behandelt werde. Es sollte an der ETH Pflicht sein, so ein Diplomand, eine systematische Methode zur Projektentwicklung kennenzulernen. Es wurde beispielsweise auf die Booch-Methode (dient zur Analyse und zum Design von objektorientierten Systemen) verwiesen. Kritisiert wurde auch, dass zu wenig Bezug auf in der Praxis verwendete Methoden genommen werden. Schlagwort sei hier "CASE-Tools". Auf Software-Engineering sollte also mehr Wert gelegt werden. Ausserdem wurde eine Verstärkung der Kontakte zur Industrie gewünscht. Dies entspricht interessanterweise auch den Kritikpunkten der Departementevalutation, von denen die Diplomanden zum Zeitpunkt dieser Umfrage noch nichts wissen konnten.

Die berufliche Zukunft

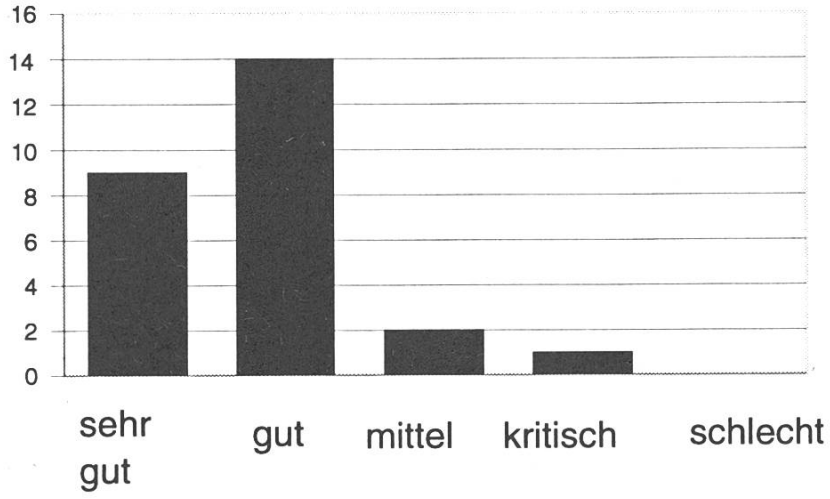
Berücksichtigt man das gerade Genannte, erstaunt es auch nicht, dass ein Grossteil der Befragten nach dem Studium am liebsten einer organisatorischen Tätigkeit nachgehen würde. Darunter verstand man besonders eine Beratungs- oder Managementfunktion. Was das Fachliche anbelangt, stand der Wunsch, in der Entwicklung von Systemsoftware zu arbeiten, knapp vor dem Erstellen von Datenbankapplikationen. Im Bereich Hardware fühlt sich kaum jemand wohl (siehe Statistik). Konkret wollen die meisten Diplomanden gleich nach dem Studium in die (Privat-) Wirtschaft gehen. Dienstleistungsbetriebe scheinen geringfügig attraktiver als Industriebetriebe zu sein. Eine Abneigung gegen Grossfirmen gibt es nicht. Im Gegenteil: Doppelt so viele Absolventen ziehen einen Grossbetrieb einer Kleinfirma vor. Immerhin haben rund ein Viertel die Erwerbung einer Zusatzausbildung, wie Nachdiplomstudium, Studium an der HSG oder Erwerbung eines MBA's ins Auge gefasst. Ebenso viele wollen vorerst an der ETH bleiben. Immerhin würden gut 25 Prozent der Diplomanden gerne gleich ausserhalb der Schweiz arbeiten. Die Chancen, am derzeitigen Arbeitsmarkt eine ädiquate Stelle zu finden, werden als ziemlich gut eingestuft. Dies entspricht eigentlich auch der kürzlich im "ETH-Intern" veröffentlichten Statistik (siehe Ausgabe 10/95).

Zusammenfassend kann man sagen, dass die Absolventen entgegen der in der Öffentlichkeit oft gängigen Meinung hohen Wert auf Interdisziplinarität und sozialen Umgang mit Menschen legen. Sie sehen sich vielmehr als "Systemdenker" (dieser Ausdruck stammt von einem Diplomanden) denn als Wissenschaftler oder Forscher.

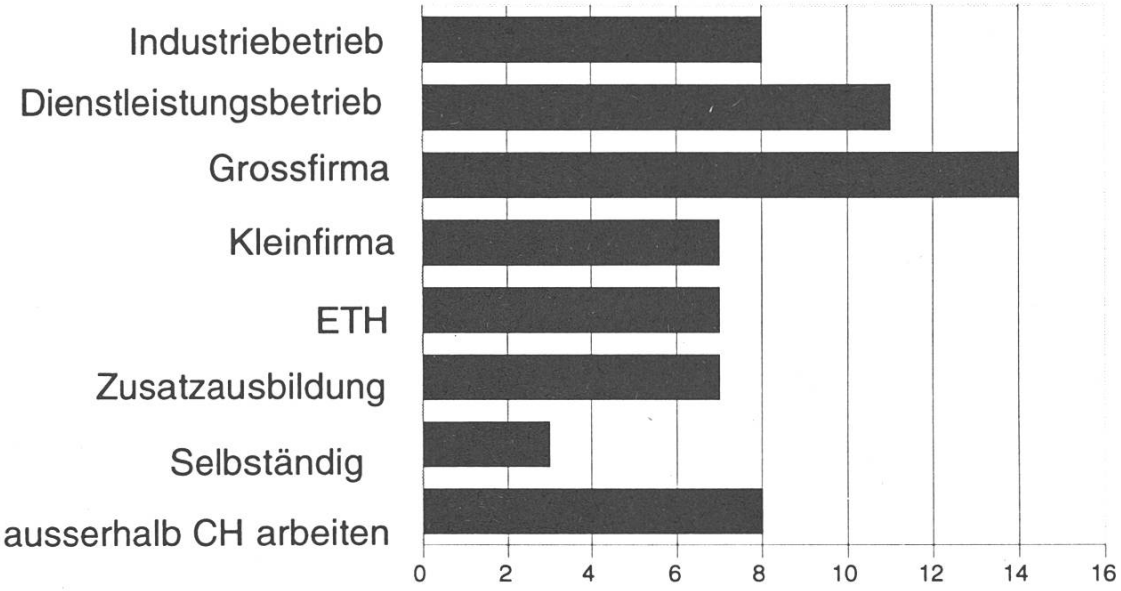
Somit möchte ich das Kapitel "Diplomandenumfrage" abschliessen, aber doch noch folgendes anregen: Wäre es nicht angebracht, eine Vereinigung ehemaliger ETH-Informatiker zu gründen, die mit dem VIS liiert sein könnte, aber nicht müsste? Eine solche Organisation wäre meiner Meinung nach eine sinnvolle Ergänzung zur bereits sehr lange existierenden Gesellschaft ehemaliger Studierender an der ETH, kurz GEP. Es würde mich freuen, wenn sich ein paar Absolventen aufraffen würden, in diese Richtung etwas zu unternehmen.

Michael Bischof
Ressort Unterricht

Wie schätzt Du Deine Chancen auf dem Arbeitsmarkt ein?

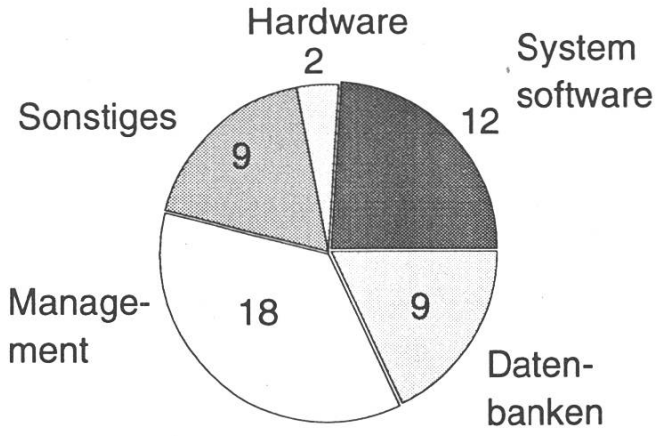


Was willst Du nach dem Studium machen?



(Mehrfachnennungen möglich)

Wie stellst Du Dir Deine Tätigkeit vor?



(Mehrfachnennungen möglich)

CALANDA

CALANDA ist nicht nur der Name eines Biers, sondern auch der Name eines Berges. Und nach diesem Berg wurde das Projekt benannt, an welchem ich während meiner Praktikumszeitam Ubilab mitprogrammierte. Dabei handelt es sich um ein Verwaltungssystem für Zeitreihen. Darunter kann man sich zum Beispiel Börsenkurse vorstellen, die gespeichert werden und bei Bedarf jederzeit wieder in ihrer chronologischen Reihenfolge abrufbar sind. Entwickelt wurde dieses System unter UNIX, mit der Absicht, es später auch auf andere Umgebungen zu portieren.

Die Grundaufgabe meines Praktikums war das Auftrennen einer Applikation in zwei separate Prozesse, die mittels Interprozesskommunikation wieder als eine Einheit agieren sollten, um später anderen Programmen den Zugriff auf die Datenbank zu vereinfachen. Es sind dies ein Kernel, der einem objektorientierten Datenbankmanagementsystem (ObjectStore) aufgesetzt ist, und eine grafische Benutzungsschnittstelle (GUI), die während der Entwicklung des Systems aus Zeitdruck direkt an den Kernel programmiert wurde. Doch bevor ich diese Aufgabe in Angriff nehmen konnte, musste ich mich an UNIX gewöhnen, mich mit der Entwicklungsumgebung SNIFF vertraut machen und die Klassenbibliothek ET++ unter die Lupe nehmen, da ich noch nie mit etwas ähnlichem gearbeitet habe. ET++ zu verstehen war anfangs nicht sehr einfach, da fast keine Dokumentation dazu existiert. Statt dessen waren der ganze Quellcode der Klassenbibliothek und einige Beispielprogramme vorhanden, die mir das Einarbeiten erleichterten. Ich lernte also, indem ich die Beispiele studierte und hunderte von Codezeilen las, was mir wenig Mühe bereitete, da ich glücklicherweise die Programmiersprache C++ zu diesem Zeitpunkt gut beherrschte und auch schon einige Erfahrung mit objektorientiertem Programmieren hatte.

Als erstes Programm entstand eine kleine Applikation, die verschiedenste grafische Objekte verwaltete, die auf einer Oberfläche mit Hilfe der Maus frei bewegt werden konnten. Es war erstaunlich, wie man mit sehr wenig Zeilen Code ein vollwertiges Window mit Menubalken und einer scrollbaren Oberfläche auf dem Bildschirm erscheinen lassen konnte, da der grösste Teil in ET++ schon vorbereitet ist und man nur seine eigenen Klassen von den vorhandenen ableiten muss. Und dank der Ent-

wicklungsumgebung war auch das Suchen von Zusammenhängen und Fehlern im Programm ein leichtes Spiel. SNIFF zeichnet sich besonders dadurch aus, dass der Programmierer mit wenigen Handgriffen nach Instanzvariablen, Klassen, Methoden und vielem mehr suchen kann, während er gerade am Editieren oder Debuggen seines Projektes ist. Als Vorbereitung auf meine eigentliche Arbeit studierte ich Klassen, die es in ET++ erlauben, Nachrichten zwischen zwei Prozessen zu verschicken und abzuarbeiten. Ich machte einige Gehversuche mit kleineren Testprogrammen, da ich auf diesem Gebiet noch keinerlei Erfahrung hatte. Nach etwa vier Wochen wagte ich mich dann zum ersten mal an das GUI. Auch hier hiess es, erst einmal Code lesen und mich mit der Architektur von CALANDA vertraut machen. Da in den letzten Monaten der Kernel etwas umstrukturiert und optimiert wurde, war es eine ideale Übung, das etwas zurückgebliebene GUI an die neuste Version des Kernels anzupassen. Ich musste Prozeduraufrufe angleichen und einige Methoden umschreiben, da Parameter hinzugefügt oder umbenennt wurden. Dabei korrigierte ich auch gleich Memoryleaks, die das GUI hinterliess und verfeinerte einige Programmteile.

Mit jedem Tag lernte ich das System besser kennen, sodass ich mich nach zwei weiteren Wochen auf mein eigentliches Ziel, die Interprozess-kommunikation konzentrieren konnte. Um später Meldungen zwischen den Prozessen mit einem maschinenunabhängigen Format zu versenden, implementierte ich als erstes diverse Methoden, die die angeforderten Daten in Strings verwandeln, denn so kann man sicher sein, dass auf jeder Maschine die Daten richtig interpretiert werden. Um auch sicher zu sein, dass die neuen Methoden richtig funktionierten, schrieb ich dazu ebenfalls kleine Testprogramme.

Ein anderes Problem waren die vielen Pointer im GUI, die auf Kernelobjekten zeigten. Diese direkte Verbindung zum Kernel musste entfernt werden, da das Ziel zwei Prozesse sind, die nur über eine Interprozessschnittstelle miteinander Daten austauschen dürfen. Das war schwierig, denn das GUI bezog die meisten Informationen von der Datenbank direkt über diese Pointer. Nach einer eingehenden Diskussion einigten wir uns, Identitätsnummern einzuführen und eine Proxyschicht zu implementieren. Diese Schicht sollte dann auch gleich die Schnittstelle zwischen den beiden Prozessen werden. Es

gab also zwei grundlegende Dinge zu tun: Zum einen musste die Proxyschicht programmiert werden, da noch nichts entsprechendes vorhanden war, und zum anderen mussten die Requests vom GUI an den Kernel über diese neue Schicht gefüührt werden. Dahinter steckte mehr Arbeit, als erwartet, denn es waren nicht nur ziemlich viele Pointer, die durch Ids ersetzt werden mussten, sondern auch die Komplexität von Aufrufen an die Datenbank stellte mich oft vor grössere Probleme. Obwohl ich mit jeder Woche grosse Fortschritte verzeichnen konnte, reichte meine Zeit nicht aus, das vorgegebene Ziel, zwei unabhängige Prozesse, zu erreichen.

Nach fünf Monaten Praktikum am UBILAB habe ich nicht nur die meisten der 40000 Programmzeilen des Systems abgearbeitet, sondern ich habe auch gelernt, was es heisst, im Team an einem grossen Projekt zu arbeiten. Das gute Klima unter den Mitarbeitern und das hochstehende Niveau, mit welchem am UBILAB gearbeitet wird, motivierten mich jeden Tag aufs neue, an meine SUN zu sitzen und produktiv zu sein.

Roberto Italia
IIIc/5

Aus der Vorlesung

Prof. Wirth in Fallstudie Oberon: "Es gibt nur wenige Systeme deren Quellcode man studieren kann, und noch weniger solche, deren Quellcode man auch studieren möchte."

Prof. Birolini in Elekrotechnik 1: " Wenn Sie das nicht sehen , dannn wie gesagt, helfen ein paar Brillen."

Prof. Waldvogel in Algebra 1: "Diese Wandtafel hat einen Vorzeichenfehler. Wenn ich hier nach unten drücke, fährt die Tafel nach oben. Ich habe mehrere Semester gebraucht, um das heraus zu finden."

Fächertest: Architektur und Realisierung von Datenbanksystemen II

Typ: Vertiefung
Umfang: 2V, 1U, 6 Kreditpunkte
Dozent: Prof. H.-J. Schek
Besuchte Vorlesung: WS 94/95

Inhalt
Die Vorlesung befasst sich mit der Architektur von Datenbanken (im Gegensatz zur Vorlesung I, die sich um die Transaktionen kümmert). Themen, die behandelt werden, sind: wie können Queries optimiert werden, welche Indizes sollen über Relationen angelegt und benützt werden, welche Vorschläge gibt es, direkt auf Relationen zuzugreifen, wie können Daten effizient gespeichert bzw approximiert werden, usw. Dabei werden verschiedene Ansäze besprochen und die Kosten berechnet.

Komplexität und Anforderungen
Die Vorlesung setzt Grundkenntnisse im Datenbankbereich voraus, die durch den Besuch des Kernfaches Informationssysteme gegeben sind. Sie vertieft diejenigen Gebiete dahingehend, dass man einen Blick hinter die Kulissen eines Datenbanksystemes enthält.

Übungen und Assistenten
Es gibt theoretische und praktische Übungen, wobei der Aufwand bei beiden etwa gleich gross ist. Das Problem bei den praktischen Übungen ist, dass nicht jeder Datenbankadministrator spielen und somit an den Oracleparametern herumschrauben kann. Die Assistenten haben mit grossem Aufwand diese Probleme gelöst und stehen immer als kompetente Anlaufstelle zur Verfügung.

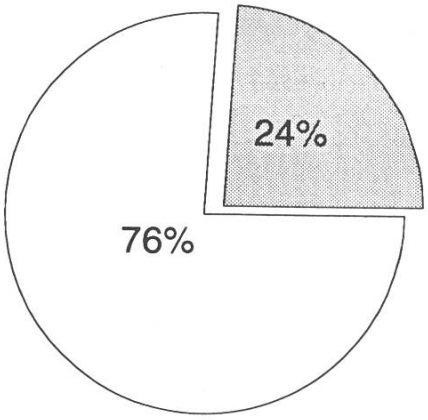
Unterlagen
Es wird ein Skript ausgegeben, das mit der Foliensammlung übereinstimmt. Dieses ist recht ausführlich.

Dozent
Wie soll man Prof. Scheks schwäbischen Vorlesungsstil beschreiben? Ich glaube, man muss ihn selber erlebt haben. Auf alle Fälle sind die Stunden alles andere als trocken, was nicht heisst, dass der Informationsgehalt darunter leidet.

Zusammenfassung
Die Vorlesung ist sehr interessant und ich frage mich, ob sie nicht in die Reihe der Pflichtvorlesungen gehört. Dazu kommt, dass Datenbanken je länger je grössere Bedeutung erhalten.

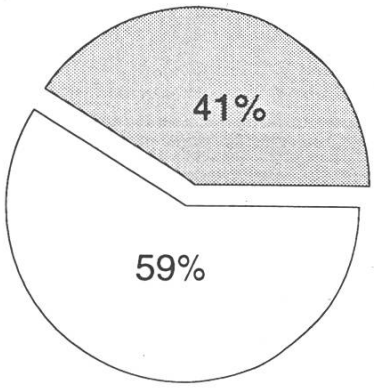
emptypage

1. Vordiplom, Frühling 1995



Schnitt: 4.35

2. Vordiplom, Frühling 1995



Schnitt: 4.02

durchgefallen
durchgekommen

Statistik: Kernfächer, Frühling 1995:

	Durchfallquote	Schnitt
Informationssysteme:	2.5%	5.0
Wiss.Rechnen III	13.6%	4.5
System-Software	14.6%	4.7
Digitaltech. & Rechnerstr.	0%	5.2 *)
Rekursiv. & Komplexität	0%	4.8 *)

*) wegen sehr geringer Anzahl Kandidaten kaum aussagekräftig

Ausführliche Berichte folgen in der nächsten Ausgabe.

VISKAS 95™

findet am

Do. 15.Juni ab Nachmittag

wie immer am

Katzensee

statt

Wenn Du Lust hast bei der Organisation oder auch nur am Fest selbst mitzuhelfen, dann schreib einfach ne E-Mail an vis@vis.inf.ethz.ch oder komm direkt im VISBüro vorbei.

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**
Aus dem Ressort Unterricht und Prüfungen **S. 6**
Fächertest: CAD: Geometrie und Datenstrukturen **S. 7**
Kommentar zur Prüfung in VLSI **S. 9**
USENET und tin **S. 11**
Zen oder die Kunst, ein Netzwerk zu verwalten **S. 19**
Diplomandenumfrage (Teil 2) **S. 21**
Praktikumsbericht: Ubilab **S. 26**
Fächertest: A + R von Datenbanksystemen II **S. 29**