ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

# Informatik I - D-ITET

Wintersemester 2003/2004

*Prof. Markus Gross, Bruno Heidelberger, Richard Keiser, Nicky Kern, Edouard Lamboray, Christoph Niederberger, Tim Weyrich, Felix Eberhard, Manuel Graber, Nathalie Kellenberger, Marcel Kessler, Lior Wehrli*

# Uebung 0 - Unix und die GNU Programmierumgebung

| | |
|---|---|
| Ausgabe: | 27. Oktober 2003 |
| Abgabe: | keine |
| Autor: | Edouard Lamboray |

## 1. Anmelden, E-Mail und WWW

Als erstes musst du dich an einem Tardis-Rechner anmelden. Bei dem **`login:`** Feld musst du deinen Benutzernamen angeben und die **`Return`** Taste drücken. Das System verlangt dann nach deinem Passwort. Falls alles korrekt eingegeben wurde, erscheint die graphische Benutzeroberfläche und du bist im System angemeldet. Zuerst musst du eine Shell öffnen, damit du die folgenden Befehle als Kommandos eingeben kannst.

Starte einen Web-Browser, z.B. Netscape, und besuche die Informatik I Homepage unter **http://graphics.ethz.ch/info1/**. Was ist der Unterschied zwischen[1]

```
> netscape
```

und

```
> netscape &
```

Die Vorlesungsunterlagen kannst du im Laufe des Semesters unter dem Stichwort **`Course Notes`** respektive **`Homework`** beziehen. Alle Dokumente werden unter dem PDF-Format bereitgestellt. Eine PDF-Datei kannst du mit dem Befehl

```
> acroread filename.pdf &
```

anschauen, respektive ausdrucken. Die Unterlagen der Vorlesung, "Einführung in Unix", werden dir helfen, die folgenden Aufgaben zu bewältigen. Auf der Informatik I Homepage kannst du dich auch in einer Uebungsgruppe anmelden.

Falls du dein ursprüngliches Passwort noch nicht geändert hast, solltest du das jetzt mit dem Kommando **`passwd`** tun. Nützliche Hinweise zur Wahl eines korrekten Passwortes findest du unter **http://www.graphics.ethz.ch/teaching/info1/PasswortTips.html**.

Benutze nun den Newsreader deiner Wahl, z.B. Netscape, und schau dir die vorlesungsbezogene Newsgruppe **ethz.amiv.course.ee.informatik-1** auf dem Server **news.ethz.ch** an. Sie enthält alle Nachrichten die letztes Semester unter den Studenten und Assistenten ausgetauscht wurden.

Im Laufe des Semesters werden wir die Programmiersprache C++ erlernen. Versuche im Internet herauszufinden, wer die Programmiersprache C++ erfunden hat. Benutze hierzu eine Suchmaschine, wie z.B. **http://www.google.com**. Versuche auch weitere Informationen über die Geschichte von C++ herauszufinden (Wann?, Wo?, Warum?).

1. Das Zeichen **>** bedeutet, dass man die nachfolgenden Kommandos entsprechend in einer Shell eingeben kann.

Verlasse Netscape jetzt noch nicht, du brauchst es noch für Aufgabe 2.

## 2. Unix Kommandos

### Prozesse

Schau dir mit **`top`** die derzeit laufenden Prozesse auf deiner Workstation an. Du wirst feststellen, dass neben deinen eigenen Prozessen, auch eine Vielzahl von Systemprozessen aktiv sind. Ist zurzeit auch noch ein anderer Benutzer auf deinem Rechner eingeloggt?

**Zusatzaufgabe.** Versuche nun mit dem Kommando **`ps`** deinen eigenen Netscape Prozess zu finden.

### Dateien

Damit du mit wachsender Anzahl von Dateien in deinem Homedirectory nicht die Uebersicht verlierst, schlagen wir vor, dass du dir für jede Vorlesung ein Verzeichnis anlegst, also z.B. **`info1`** für die Informatik I Vorlesung.

Lade nun von der Vorlesungshomepage den Quellcode, welcher für die nächsten Aufgaben benötigt wird. Ueberzeuge dich, dass die Datei im **`info1`** Verzeichnis liegt. Wenn nicht, verschiebe sie dorthin.

Bei der **`u0.tar.gz`** Datei handelt es sich um eine archivierte und komprimierte Datei. Das heisst, sie enthält eigentlich mehrere Dateien, die zusammengefasst wurden. Schau dir die Grösse von **`u0.tar.gz`** an.

Mit **`gzip`** und **`gunzip`** können Dateien komprimiert, respektive dekomprimiert werden. Dekomprimiere **`u0.tar.gz`** und schau dir die neue Grösse an. Was stellst du fest?

Die einzelnen Dateien in der Archivdatei können mit dem Befehl

```
> tar -xvf u0.tar
```

herausgelöst werden. Wende ihn an und schaue dir auf ein neues dein Verzeichnis an. Was ist passiert?

## 3. Der GNU Compiler

Bewege dich nun in das Verzeichnis **`./u0/simple`** und kompiliere und linke die Datei **`u0_simple.cpp`**. Nach erfolgreichem Kompilieren und Linken kannst du **`u0_simple`** aufrufen. Was passiert? Wie kannst du das Problem beheben? (Tipp: Denke an die **`PATH`** Umgebungsvariable.)

**Zusatzaufgabe.** Bewege dich nun in das Verzeichnis **`../source`**. Gebe dann folgendes Kommando ein:

```
> gmake
```

Am Output erkennst du, dass einige der heruntergeladenen Dateien kompiliert und zu einer statischen Bibliothek[1] zusammengelinkt werden. In einem **`Makefile`** kann man festlegen, nach welchen Regeln ein Programm, welches aus mehreren Dateien besteht,

1. Eine statische Bibliothek erkennt man an der **`.a`** Extension des Dateinamens.

kompiliert werden soll. Wie man selbst ein solches **Makefile** schreibt, lernen wir allerdings erst später im Semester.

Rufe noch einmal **gmake** auf. Stellst du einen Unterschied fest?

Versuche nun mittels **man** herauszufinden, was das Kommando **touch** verursacht. Wende es auf die **u0_math.h** Datei an. Wende wieder **gmake** an. Kannst du dir erklären, was passiert?

Bewege dich nun in das Verzeichnis **../advanced**. Dort findest du eine Quelldatei **u0_advanced.cpp**, welche Funktionen aus der **u0_math** Bibliothek benutzt. Kompiliere die **u0_advanced.cpp** Datei. Beachte, dass du das **../source/** Verzeichnis als Includedirectory angeben musst. Was passiert, wenn du dies nicht tust?

Versuche **u0_advanced** einmal direkt mit der Objektdatei **../source/u0_math.o**, als auch einmal mit der Bibliothek **../lib/libu0_math.a** zu linken. Was passiert, wenn du weder **u0_math.o** noch **-lu0_math** angibst?

Nach erfolgreichem Kompilieren und Linken kannst du **u0_advanced** aufrufen.

**Bemerkung.** Der Umfang des **u0_advanced** Programms erfordert selbstverständlich nicht das Anlegen einer besonderen **u0_math** Bibliothek. Auch für das Berechnen der Quadratwurzel kann direkt **sqrt()** aus der normalen Mathematik-Bibliothek verwendet werden. Es handelt sich hierbei also um ein rein didaktisches Beispiel um die Benutzung von Bibliotheken kennenzulernen.

## 4. Editieren und Debuggen

Starte nun einen Editor deiner Wahl, am Besten natürlich **xemacs**, mit der Quelldatei **u0_simple.cpp**. Versuche zuerst, absichtlich einen Fehler hinzuzufügen. (Vorschlag: Lösche einen Strichpunkt). Versuche nun die Datei neu zu kompilieren. Was beobachtest du?

Mache nun deine Aenderung wieder rückgängig. Schau dir die Quelldatei genau an. Neben der bereits benutzten Funktion **QuadratWurzel()** findest du auch noch eine andere Funktion. Ersetze nun in **u0_simple.cpp** **QuadratWurzel()** durch die zweite Funktion. Dann kompiliere und linke **u0_simple** auf ein Neues und führe es aus. Hat sich etwas verändert?

**Zusatzaufgabe .** Schau dir dein Programm mit dem Debugger an, starte also:
**> ddd u0_simple &**

Versuche durch das Programm durchzusteppen und die Variablenwerte anzuschauen.