

Einführung in UNIX

Es gibt nur einige wenige Dinge, die Du wissen musst, um mit UNIX loszulegen. Alles andere ist eine Frage der Gewohnheit. Diese kurze Übersicht soll Dir helfen in den Umgang mit der Shell, typischen UNIX-Programmen etc. zu erlernen.

Die Shell

Die Shell ist ein textbasierter **Befehlsinterpreter**. Sie erlaubt es Dir, Befehle einzutippen und sorgt dafür, dass sie ausgeführt werden. Es gibt verschiedene Shells unter UNIX. Wir benutzen die **tcsh**. Sie ist eine Erweiterung der **Berkley-UNIX-C-Shell** (→**man tcsh**). Wenn Du eine Shell startest, siehst Du als erstes den **prompt**. Die Shell zeigt Dir damit an, dass sie bereit ist, ein neues Kommando entgegenzunehmen.

Kommandos haben in UNIX meistens folgendes Format:

```
command [-options] [parameters] [control]
```

command ist dabei der Befehl oder das Programm, das ausgeführt werden soll. **options** sind die Optionen, die man dem Programm mitgeben kann. **parameters** sind weitere Argumente, die das Programm benötigt. Welche Optionen ein Programm hat und welche Parameter es erwartet, kannst Du jeweils im **Online Manual** nachlesen. Dieses wird mit **man** und dem entsprechenden Befehl gestartet. Für den Befehl **ls** findest Du die Informationen also z.B. unter **man ls**. Falls der Name des gesuchten Befehls nicht genau bekannt ist, kannst Du mit **apropos stichwort** alle Manpages auflisten, die **stichwort** enthalten.

Normalerweise kommt der nächste **prompt** erst, wenn der Befehl oder das Programm beendet ist. Bei Programmen, wie z.B. Netscape kann es aber sehr lästig sein, wenn man die Shell nicht mehr weiterverwenden kann. Deshalb kann man Programme mit einem **&** am Ende im Hintergrund starten, d.h. der **prompt** kommt, obwohl das Programm noch läuft.

Copy-Paste unter UNIX

Die meisten Programme bieten ein ähnliches **Copy/Paste** wie Windows an. Zusätzlich gibt es aber auch das klassische Unix **Copy/Paste**, dass wie folgt funktioniert. Zuerst markierst Du in einem beliebigen Fenster, den zu kopierenden Text (z.B. in der Shell). Dann klickst Du mit der **mittleren Maustaste** dorthin, wo der Text hinkopiert werden soll. Und das war's auch schon.

Die wichtigsten Kommandos

```
pwd  
cd dir  
cd ..  
cd  
ls  
ls -l  
ls -a  
cat file  
more file  
cp file1 file2  
cp file dir  
mv file1 file2  
mv file dir  
mkdir dir  
rmdir dir  
rm file
```

aktuelles Verzeichnis der Shell
ins Verzeichnis **dir** wechseln
ins übergeordnete Verzeichnis wechseln
ins Home-Verzeichnis wechseln
Inhalt des aktuellen Verzeichnisses anzeigen
ausführlicheres Inhaltsverzeichnis
versteckte Dateien anzeigen
Datei **file** anzeigen
Datei **file** seitenweise anzeigen
eine Kopie von **file1** mit dem Namen **file2** machen
Datei **file** ins Verzeichnis **dir** kopieren
Datei **file1** in **file2** umbenennen
Datei **file** ins Verzeichnis **dir** kopieren
Verzeichnis **dir** erstellen
Verzeichnis **dir** löschen
Datei **file** löschen

Du kannst auch Optionen kombinieren z.Bsp **ls -al** oder **ls -a -l**

Umleitungen und Pipes

```
cmd > file  
cmd < file  
cmd1 | cmd2
```

Ausgabe des Programms **cmd** wird in der Datei **file** gespeichert
Das Programm **cmd** bekommt seine Eingabe aus der Datei **file**
Die Ausgabe des Programms **cmd1** wird als Eingabe für das Programm **cmd2** verwendet

Einige weniger wichtige Kommandos

```
head  
tail  
ps  
kill pid  
kill -9 pid  
chmod  
grep  
diff  
sort  
wc
```

Die ersten 10 Zeilen (einstellbar) einer Datei anzeigen
die letzten 10 Zeilen (einstellbar) einer Datei anzeigen
Alle Prozesse anzeigen, die von dieser Shell gestartet wurden
Prozess mit der Nummer **pid** beenden (**sanft**)
Prozess mit der Nummer **pid** beenden (**brutal**)
Rechte einer Datei oder eines Verzeichnisses ändern. Bsp: **chmod o+w test**
Zeilen einer Datei anzeigen, in der ein bestimmter String vorkommt. Bsp: **grep "alias" .alias**
Dateien vergleichen
Zeilen sortieren
Zeilen, Wörter, Zeichen in einer Datei zählen.
Bsp: **grep "alias" .alias | wc -l**
(zählt Anzahl definierter Aliases)

Programme

Programme unter UNIX können wie ganz normale Kommandos gestartet werden. Normalerweise startet man sie aber im Hintergrund, d.h. z.B. **netscape &**.

```
emacs      empfohlener GNU Text-Editor (ohne Graphik)
xemacs    andere Emacs-Implementation (mit Graphik)
nedit     Windows-ähnlicher Editor
textedit  Xview Editor
netscape Netscape Communicator (WWW- und
           Email-Programm)
soffice   StarOffice (ähnlich wie Microsoft Office), vor
           dem Starten module add staroffice
           ausführen
pine      textbasiertes Email-Programm
dtmail    Email-Programm von Solaris (sollte bereits
           konfiguriert sein)
gimp      Grafik-Programm
xv        Picture Viewer
gv        Viewer für Postscript Dateien
acroread  Acrobat Reader (für pdf-Dateien)
```

spezielle Files und Directories

```
/home/LOGIN      Home-Verzeichnis, für deine Dateien und Directories
/home/LOGIN/.tcshrc  Konfigurations-Datei für die tcsh
/home/LOGIN/.alias  Definition von alias-Befehlen
```

/home/LOGIN/.tcshrc

Bevor Du an diesem File rumspielst, solltest Du eine Sicherheitskopie davon machen. Die meisten Sachen sind eigentlich bereits default-mässig konfiguriert. Hier nur noch ein paar zusätzliche Sachen, die das ganze noch ein bisschen schöner machen.

Bei den **module add ...** kannst Du noch **module add staroffice** hinzufügen, damit Du das nicht immer eingeben musst, wenn Du **soffice** starten willst.

```
set prompt ... kannst Du abändern in set prompt = "<%B%m %T %~%b>,"
damit der Prompt ein bisschen schöner aussieht.
```

Falls gegen Ende des Files nicht **if (-e ~/.alias) ...** oder ähnliches steht, solltest Du unbedingt noch diese Zeile einfügen: **if (-e ~/.alias) source ~/.alias**.

Achte sowohl bei **.tcshrc** als auch bei **.alias** darauf, dass am Ende des Files eine Leerzeile steht.

Andere Einstellungsmöglichkeiten findest Du wie schon gesagt mit **man tcsh**.

/home/LOGIN/.alias

alias-Befehle kannst Du ganz nach deinen eigenen Wünschen definieren. Die Syntax ist **alias newcommand 'command with options and parameters'**. Hier ein paar Beispiele:

```
alias ll 'ls -l'
alias la 'ls -la'
alias dir 'ls -l'
alias nmail 'netcape -mail'
alias xgv 'gv -center -magstep -2 -a4 -watch'
alias load 'xload -update 1 &; perfmeter -a -g -s 1 -S -V &'
```

XEmacs

Für Anfänger ist nedit wahrscheinlich der beste Texteditor. Wenn Du ein bisschen mehr willst, solltest Du Dir aber unbedingt auch emacs oder xemacs anschauen. Die Stärke von Emacs ist, dass er vor allem über sogenannte Shortcuts (Tastenkombinationen) bedient wird. Am Anfang ist das sehr umständlich, aber wenn man einmal ein paar Shortcuts kennt, ist das ganze sehr effizient und bequem.

Um Dir den Einstieg zu erleichtern, hier eine Auswahl der wichtigsten Shortcuts. taste1-taste2 steht für das Drücken von taste1 und dann taste2, während man taste1 gedrückt hält. C-... steht für die CTRL-Taste oder STRG-Taste und M-... für die META-Taste (neben der ALT-Taste).

C-x C-c	Emacs verlassen
C-x C-f	Datei öffnen
C-x 5 C-f	Datei in einem neuen Fenster öffnen
C-x 0	dieses Fenster schliessen (bei mehreren Fenstern)
C-x C-s	Datei speichern
C-x C-w	Datei unter einem anderen Namen speichern
C-s	String suchen
C-r	String rückwärtig suchen
C-k	Zeile löschen
C-w	markierten Bereich löschen
C-h i	Emacs-Hilfe

Urs Hardegger <uhardegg@iic.ethz.ch>
Corsin Decurtins <cdecurti@iic.ethz.ch>