

UNIX–Kurzzusammenfassung der wichtigsten Befehle

CTRL-D bedeutet: gleichzeitiges Betätigen der **Control** und **D** Taste.

1 Die Sitzung

1.1 Einloggen

Username und Password auf Prompt eingeben.

1.2 Ausloggen

logout oder CTRL-D eingeben.

In vielen Shells funktioniert auch **exit**. Wurde versehentlich statt CTRL-D CTRL-S eingegeben, so „steht“ der Bildschirm. Abhilfe mit CTRL-Q.

1.3 Syntax von Unix-Kommandos

kommando -optionen arg1 ... argn <RETURN>
(siehe auch **man intro**)

2 Users

Wer ist eingeloggt?	who
Den eigenen Usernamen abfragen:	who am i whoami who is this id
User-Info	finger <username> oder finger @<rechnername>

3 Hilfe

Hilfe zu einem Kommando anzeigen: **man <kommando>**
fensterbasiert mit: **xman**, **tkman**

andere Hilfe- und Informationsmöglichkeiten per WWW

4 Account

Gruppenzugehörigkeit:	groups
Plattenplatzbelegung und Limits für den User <username>:	quota -v <username>
Diskusage für aktuelles Directory:	du
für Directory <directory>:	du <directory>
Plattenplatzbelegung auf allen gemounteten Filesystemen:	df

5 Pfade

5.1 Pfadnamen

einfach (simple):	Ein Name (File oder Directory), um Zugriff auf ein in dem aktuellen Directory vorhandenes File/Directory zu haben.
absolut (absolute):	Liste von Directories, die mit dem <i>Root Directory</i> (/) beginnt, bis zum gewünschten File/Directory. Die einzelnen Teilpfade werden durch / getrennt.
relativ (relative):	Liste von Directories, die im aktuellen Directory startet. Bis zum gewünschten File/Directory durch / getrennt.

5.2 Directory-Wechsel

Aktuelles Directory anzeigen (Print Working Directory):	pwd
Wechseln ins Home-Directory (Change Directory):	cd
Wechseln in ein (beliebiges) Directory:	cd <dirname>

5.3 Abkürzungen

Home-Directory:	~
Home-Directory von <username>:	~<username>
Aktuelles Directory:	.
Vorfahre des aktuellen Directories:	..

6 Files auflisten

im aktuellen Directory:	ls
im Directory <dirname>:	ls <dirname>
alle Files mit sämtlichen Informationen auflisten:	ls -al
nur Directory-Namen anzeigen, nicht dessen Inhalt:	ls -d

7 Files anschauen

File komplett anschauen:	cat <filename>
File seitenweise anschauen:	more less page pg } <filename>
die ersten 20 Zeilen ansehen:	head -20 <filename>
die letzten 20 Zeilen ansehen:	tail -20 <filename>

8 Sicherheit

8.1 Passwörter

Gute Passwörter enthalten Klein- und Großbuchstaben, Ziffern und Sonderzeichen. Sie sollten mindestens 5 Zeichen enthalten.

Schlechte Passwörter sind Namen, Worte wie in Wörterbüchern, Telefonnummern, Autokennzeichen etc.

Wechseln des Passworts: **passwd** oder **yppasswd**

8.2 Zugriffsrechte

File	Directory
-rwx rwx rwx	d rwx rwx rwx
user group others	user group others

mit r = read, w = write, x = execute

Ändern der Zugriffsrechte durch:

chmod w?y <filename>
w = u, g, o, a für user/group/other/all
? = +, - für Rechte addieren/entfernen
y = r, w, x für read/write/execute

Alternative Möglichkeit:

chmod abc <filename>
abc wird wie folgt spezifiziert:

	a: user	b: group	c: other
read	4	4	4
write	2	2	2
execute	1	1	1

Für die gewünschten Rechte werden die Spalten aufsummiert.
Beispiel: Ein File soll die Rechte rwxrw— erhalten.

chmod 760 <filename>

UNIX–Kurzzusammenfassung der wichtigsten Befehle

8.3 Zugriffsrechte generell festlegen

umask im `.cshrc` (für `csh` und `tcsh`) legt die Rechte für alle neu angelegten Files fest.

umask Wert	Datei	Directory
022	-rw-r--	drwxr-xr-x
077	-rw-----	drwx-----
027	-rw-r--	drwxr-x--

Ausnahme z.B. bei ftp !!!

9 File anlegen

per Editor, Shell-Umlenkungskommando oder `touch`.

`echo "Hallo Welt !" > <neufilename>` legt ein neues File `<neufilename>` mit dem Inhalt „Hallo Welt!“ an

`touch <neufilename>` legt ein leeres File `<neufilename>` an

10 Directory anlegen

`mkdir <dirname>` Legt ein neues Directory `<dirname>` an (make directory).

11 Files kopieren (duplizieren)

(`cp` copy)

`cp <altfile> <neufilename>` kopiert `<altfile>` nach `<neufilename>`

`cp <altfile> <dirname>` kopiert `<altfile>` in das Directory `<dirname>`

`cp -r <altdir> <neudir>` kopiert das komplette Directory `<altdir>` in das Directory `<neudir>` (es wird in `<neudir>` das Unter-Verzeichnis `<altdir>` angelegt).

12 Files umbenennen (verschieben)

(`mv` move)

`mv <altfile> <neufilename>` umbenennen von `<altfile>` in `<neufilename>`

`mv <altfile> <dirname>` schiebe `<altfile>` in das Directory `<dirname>`

`mv <altdir> <neudir>` umbenennen von `<altdir>` in `<neudir>`

`mv <altdir> <dirname>` schiebe `<altdir>` in das Directory `<dirname>`

13 Files/Directories löschen

Vorsicht: Einmal gelöschte Files/Directories sind nicht mehr restaurierbar!

(`rm` remove)

`rm <filename>` lösche `<filename>`

`rm -i <filename>` fragt vorher, ob das File `<filename>` gelöscht werden soll (interaktiv)

`rmdir <dirname>` lösche leeres Directory `<dirname>`

`rm -r <dirname>` lösche Directory `<dirname>` samt Inhalt

`rm -ri <dirname>` fragt vorher, ob das Directory `<dirname>` gelöscht werden soll (interaktiv)

14 Files drucken

`lpr -P<pname> <filename>` druckt das File `<filename>` auf dem Drucker `<pname>` aus

`lpq -P<pname>` zeigt die Printer-Queue für Drucker `<pname>` an

`lprm -P<pname> <jobnum>` löscht den Job mit der Nummer `<jobnum>` aus der Printer-Queue von Drucker `<pname>`

15 Files formatieren

Mit `pr` werden Files zum Drucken aufbereitet.

`a2ps <filename>` (ASCII to PostScript) bereitet `<filename>` zur Ausgabe auf Laserdrucker vor. **Achtung**, dies ist kein Standard UNIX-Befehl!

UNIX–Kurzzusammenfassung der wichtigsten Befehle

16 History

Die *cs*h und *tc*sh bieten einen History-Mechanismus an. Die History-Einträge können mit folgenden Kommandos manipuliert werden:

history	zeige die letzten Kommandos
!!	wiederhole letztes Kommando
!<i>n</i>	wiederhole Kommando Nr. <i>n</i>
!<i><his></i>	wiederhole letztes Kommando, das mit <i><his></i> beginnt
!<i>?<his></i>	wiederhole letztes Kommando, das <i><his></i> enthält
~<i>a</i>~<i>b</i>	wiederhole letztes Kommando, aber ersetze <i>erstes</i> Auftreten von <i>a</i> durch <i>b</i>
!<i>gs/a/b</i>	wiederhole letztes Kommando, aber ersetze <i>jedes</i> Auftreten von <i>a</i> durch <i>b</i>

17 Kommandos editieren

Achtung: Diese Funktionen bietet nur die *tc*sh!

CTRL-P	↑	vorheriges Kommando
CTRL-N	↓	nächstes Kommando
CTRL-F	→	Cursor ein Zeichen nach rechts
CTRL-B	←	Cursor ein Zeichen nach links
<ESC-F>		Cursor ein Wort nach rechts
<ESC-B>		Cursor ein Wort nach links
CTRL-A		Cursor zum Zeilenanfang
CTRL-E		Cursor zum Zeilenende
CTRL-U		gesamte Zeile löschen
CTRL-K		von aktueller Position aus bis zum Zeilenende löschen

18 Umlenken von Ein- & Ausgabe

Achtung: Es werden die Umlenkungskommandos für *cs*h oder *tc*sh beschrieben!

kommando >	<file>	Umlenken der Standard-Ausgabe in Datei <file>
kommando >&	<file>	Umlenken von Standard-Ausgabe und Standard-Error in Datei <file>
kommando >>	<file>	Anhängen der Ausgabe an die Datei <file>
kommando <	<file>	Alle Eingaben werden aus der Datei <file> gelesen
kommando1 kommando2		(Pipe) Ausgabe von kommando1 wird als Eingabe für kommando2 verwendet

Mit folgender Konstruktion können Standard-Ausgabe und Standard-Error in *unterschiedliche* Dateien geschrieben werden:

```
(kommando > <stdout>) >& <stderr>
```

Wird an das Umlensymbol ein „!“ angehängt, so wird auch eine bereits bestehende Datei ohne Warnung überschrieben:

```
kommando >! <file> oder kommando >&! <file>
```

19 Prozesse

ps	Anzeigen der eigenen Prozesse (mit <i>PID</i>)
ps -ax	Anzeigen aller Prozesse (BSD)
ps -ef	Anzeigen aller Prozesse (SYSV)
jobs	Anzeigen aller im Hintergrund gestarteten Prozesse (nur <i>cs</i> h und <i>tc</i> sh)
kill <PID>	beende Prozess mit Id <PID>. Achtung: es können nur <i>eigene</i> Prozesse abgebrochen werden!
kommando &	führe kommando im Hintergrund aus
CTRL-Z	den aktuellen (Vordergrund-) Job unterbrechen
bg	zuletzt unterbrochenen Job in den Hintergrund schicken
bg % <i>jobnr</i>	unterbrochenen Job Nr. <i>jobnr</i> in den Hintergrund schicken
fg	hole aktuellen Job in den Vordergrund
fg % <i>jobnr</i>	Job Nr. <i>jobnr</i> in den Vordergrund holen
nice + <i>n</i> <kommando>	erniedrigt die Priorität eines Kommandos

UNIX–Kurzzusammenfassung der wichtigsten Befehle

20 (T)C-Shell

20.1 Startup-Files

.login .cshrc

20.2 Wildcards

Platzhalter	Bedeutung
~	Home-Directory
~<username>	Home-Directory von <username>
?	ein beliebiges Zeichen
*	beliebig viele (auch kein) Zeichen a oder e
[ae]	ein Buchstabe zwischen a bis z
[a-z]	jeder der angegebenen Strings
{ab,cd,eg}	

20.3 Variablen

Umgebungsvariablen	Shellvariablen
gelten global	gelten für aktuelle shell
HOME	home
PATH	path
TERM	term
USER	user
DISPLAY	history
PRINTER	prompt
...	...
setenv VAR value	set var=value
unsetenv VAR	(un)set var
echo \$VAR	echo \$var
env	set

20.4 Erweiterung des Suchpfades

```
setenv PATH ${PATH}:%HOME/bin
set path=(%path %home/bin)
```

20.5 Update des Suchpfades

rehash

20.6 Aliases

aliases anzeigen	alias
alias von <name>	alias <name>
alias definieren	alias <name><command>
alias löschen	unalias <name>

20.7 Bekanntgabe von Änderungen im .cshrc

source ~/.cshrc

20.8 Maskierung von Sonderzeichen

durch " " ' ' \

20.9 Wechsel der Shell

durch Eingabe von **sh**, **csh**, **ksh**, **bash**, **tcsh**

21 Nützliche Unix-Kommandos

file	was für ein File
which	welches Kommando
where	wo ist Kommando (nur <i>(t)csh</i>)
(f,e)grep	Suchen von Ausdrücken in Textdateien
find	Suchen von Dateien im Dateibaum
sort	Sortieren in Textdateien
diff	Vergleichen von Dateien
wc	Zählen von Zeilen, Worten oder Zeichen in Dateien
spell	(englische) Rechtschreibprüfung
split	Datei in kleinere aufteilen
g(un)zip,zcat	(De)Komprimieren von Dateien
tar	Archivieren von Dateibäumen
uuencode	Verschicken/Empfangen
uudecode	von Binärfiles per mail

22 Dateiararten und Viewer

Dateiart	Endung	Viewer
PostScript	.ps	gv oder ghostview <file.ps>
DVI	.dvi	xdvi <file.dvi>
PDF	.pdf	acroread <file.pdf>
MS Word/Excel	.doc/.xls	pcfileviewer <file.doc> oder per StarOffice

23 DOS- und UNIX-Befehle

cd	pwd	Pfad des aktuellen Verzeichnisses
cd <dirname>	cd <dirname>	in's Verzeichnis <dirname> wechseln
copy	cp	eine Datei kopieren
del	rm	eine Datei löschen
deltree	rm -rf	ein Verzeichnis löschen
dir	ls -l	den Inhalt eines Verzeichnisses auflisten
md	mkdir	ein Verzeichnis anlegen
move	mv	Dateien oder Verzeichnisse umbenennen oder verschieben
rd	rmdir	ein leeres Verzeichnis löschen
xcopy	cp -r	ein Verzeichnis komplett kopieren
<kommando> /?	man <kommando>	Hilfe zu <kommando>