

# 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?	<b>who</b>
Den eigenen Usernamen abfragen:	<b>who am i</b> <b>whoami</b> <b>who is this</b> <b>id</b>
User-Info	<b>finger &lt;username&gt;</b> oder <b>finger @&lt;rechnername&gt;</b>

## 3 Hilfe

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

andere Hilfe- und Informationsmöglichkeiten per WWW

## 4 Account

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

## 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> ( <b>/</b> ) beginnt, bis zum gewünschten File/Directory. Die einzelnen Teilpfade werden durch <b>/</b> getrennt.
relativ (relative):	Liste von Directories, die im aktuellen Directory startet. Bis zum gewünschten File/Directory durch <b>/</b> getrennt.

### 5.2 Directory-Wechsel

Aktuelles Directory anzeigen (Print Working Directory):	<b>pwd</b>
Wechseln ins Home-Directory (Change Directory):	<b>cd</b>
Wechseln in ein (beliebiges) Directory:	<b>cd &lt;dirname&gt;</b>

### 5.3 Abkürzungen

Home-Directory:	<b>~</b>
Home-Directory von <username>:	<b>~&lt;username&gt;</b>
Aktuelles Directory:	<b>.</b>
Vorfahre des aktuellen Directories:	<b>..</b>

## 6 Files auflisten

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

## 7 Files anschauen

File komplett anschauen:	<b>cat &lt;filename&gt;</b>
File seitenweise anschauen:	<b>more</b> <b>less</b> <b>page</b> <b>pg</b> } <b>&lt;filename&gt;</b>
die ersten 20 Zeilen ansehen:	<b>head -20 &lt;filename&gt;</b>
die letzten 20 Zeilen ansehen:	<b>tail -20 &lt;filename&gt;</b>

## 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
<b>-rwx rwx rwx</b>	<b>d rwx rwx rwx</b>
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:

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

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

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

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

## 22 Dateiararten und Viewer

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