

Kurzreferenz Unix

Kommando Syntax	Kleinbuchstaben benutzen. Jedes Kommando mit "Return" beenden. <i>Text</i> durch die eigenen Angaben ersetzen.
<i>command options files</i>	<i>options</i> , soweit vorhanden, gehen in der Regel <i>files</i> voran. Es können ein oder mehrere <i>files</i> angegeben werden. Nähere Details in den man pages.
<i>command < file</i>	<i>command</i> benutzt als Standardeingabe <i>file</i> anstelle der Tastatur (Umleitung der Standardeingabe)
<i>command > file</i>	<i>command</i> sendet seine Standardausgabe an <i>file</i> anstelle des Terminals. Wenn <i>file</i> existiert, wird es überschrieben, sofern set clobber gesetzt ist.
<i>command >> file</i>	<i>command</i> hängt seine Standardausgabe an das existierende <i>file</i> an. Falls <i>file</i> noch nicht existiert, wird es angelegt.
<i>command1 command2 ...</i>	Die Standardausgabe von <i>command1</i> wird zur Standardeingabe von <i>command2</i> , usw.
<i>nice -10 command</i>	<i>command</i> wird mit einer niedrigeren Priorität gestartet. Dies sollte unbedingt benutzt werden, wenn bekannt ist, daß <i>command</i> die Systemressourcen stark beansprucht.
Online Hilfe	* (Nicht auf jedem UNIX Rechner vorhanden)
help	Tips für Einsteiger *
man topic	Zeigt das Manual zu <i>topic</i> (z.B. Kommandooptionen, Funktionsparameter); siehe man man
man topic lpr	Druckt das Manual zu <i>topic</i> auf dem Standarddrucker.
man -k topic	Listet alle Kommandos des OnlineManuals auf, die sich auf <i>topic</i> beziehen.
info	Startet die interaktive Oberfläche für die installierten GNU-Manuals. *
info topic	Startet die interaktive Oberfläche für die GNU-Manuals und zeigt die Manuals zu <i>topic</i> an. *
Grundlegende Kommandos	Die AccountPaßwörter sollten regelmäßig (möglichst monatlich) aus Sicherheitsgründen geändert werden.
passwd	Ändert das Passwort auf allen Unix Station.
logout	Beendet die Terminal Sitzung auf dieser Maschine.
finger user	Zeigt Informationen zu <i>user</i> . Wird <i>user</i> weggelassen, werden die aktuellen Nutzer angezeigt.
clear	Löscht das Terminalfenster.
history	Zeigt eine Liste der zuletzt benutzten Kommandos (incl. frühere Sitzungen)
Dateien und Verzeichnisse	Pfade sind mit "/" getrennte Sequenzen von Verzeichnissen, die mit einer Datei oder einem Verzeichnis enden. Standardpfade sind relativ zum aktuellen Verzeichnis. Absolute Pfade beginnen mit einem "/" und sind relativ zur Wurzel des Dateisystems.
ls -al directory	Zeigt alle Files (incl. Files, deren Namen mit einem Punkt beginnen) im angegebenen <i>directory</i> im Langformat an. Wird <i>directory</i> weggelassen, wird das aktuelle Verzeichnis gelistet.
more file	Zeigt den Inhalt von <i>file</i> seitenweise. more kann mit q beendet werden.
pwd	Zeigt den absoluten Pfad des aktuellen Verzeichnisses.
cd directory	Wechselt zu <i>directory</i> . Wenn kein <i>directory</i> angegeben ist, wird zum HomeVerzeichnis gewechselt.
cd ..	Geht eine Ebene im aktuellen Verzeichnisses Pfad zurück.

Kurzreferenz Unix

<code>mkdir directory</code>	Erzeugt ein Verzeichnis namens <i>directory</i> .
<code>rmdir directory</code>	Entfernt das Verzeichnis <i>directory</i> . <i>directory</i> muß dafür leer sein.
<code>cp file newfile</code>	Kopiert <i>file</i> nach <i>newfile</i> .
<code>cp f1 f2 f3 ... directory</code>	Kopiert alle <i>f1 f2 f3 ...</i> in das Verzeichnis <i>directory</i> . Die Originaldateien verbleiben ungeändert.
<code>cp -r f1 f2 d1 d2 ... directory</code>	Kopiert alle <i>f1 f2 d1 d2 ...</i> nach <i>directory</i> . Verzeichnisse werden mit der gesamten Datei und Unterverzeichnisstruktur kopiert.
<code>mv file newfile</code>	Benennt <i>file</i> in <i>newfile</i> um.
<code>mv f1 f2 d1 d2 ... directory</code>	Bewegt alle <i>f1 f2 d1 d2 ...</i> nach <i>directory</i> . Verzeichnisse werden mit der gesamten Datei und Unterverzeichnisstruktur bewegt.
<code>rm -i f1 f2 f3 ...</code>	Löscht alle angegebenen Dateien. Ohne i wird nicht nachgefragt und <i>f1 f2 f3 ...</i> verschwinden für immer (ein Undelete ist unter UNIX nicht möglich).
<code>rm -ir f1 f2 d1 d2 ...</code>	Löscht alle angegebenen Dateien und Verzeichnisse (incl. der in diesen Verzeichnissen befindlichen Files und Unterverzeichnisse).
Nützliche Kürzel	Wildcards werden vor Ausführung des Kommandos durch die Shell aufgelöst. Die folgenden Kürzel sind Beispiele.
<code>ls ..</code>	Zeigt den Inhalt des übergeordneten Verzeichnisses. Der Name des aktuellen Verzeichnisses ist mit in der Liste enthalten.
<code>cp file ~</code>	Kopiert <i>file</i> in Dein Home Verzeichnis, unabhängig vom aktuellen Verzeichnis.
<code>cp file /tmp; chmod a+r /tmp/file</code>	Kopiert <i>file</i> in das System Behelfsverzeichnis und erlaubt den Lesezugriff für alle Nutzer.
Hängengeblieben?	Wenn eine Sitzung hängen geblieben ist, kann man sich von einem anderen Terminal aus einloggen und die hängende Sitzung wie folgt beenden (sofern es sich um die eigene handelt):
<code>tty</code>	Zeigt den Terminal Namen der laufenden Sitzung.
<code>ps -uxwww</code>	Listet alle Prozesse der aktuellen Sitzung. PID und TT (Terminal names) beachten.
<code>kill -9 pid</code>	Beendet den Prozeß mit der PID (Process ID number) <i>pid</i> .
Besondere Tasten	
Ctrl-C (Strg-C)	Beendet das aktuelle Programm und kehrt zum Kommandozeileninterpreter (shell) zurück.
Ctrl-Z (Strg-Z)	Unterbricht die Ausführung des aktuellen Programmes und kehrt zur shell zurück. Siehe Jobkontrolle.
Ctrl-D (Strg-D)	Signalisiert Eingabeende an das wartende Programm. Funktioniert auch mit dem login-prompt.
Dateischutz	Dateien und Verzeichnisse gehören einem Nutzer und einer Gruppe und haben unabhängige read , write und execute -Rechte für user , group und others . Für ein Verzeichnis sind execute -Rechte notwendig, um es in einem Pfad angeben zu können. Das Erzeugen/Löschen eines Files erfordert write- und execute -Rechte am Verzeichnis. ls an einem Verzeichnis erfordert read- und execute -Rechte.
<code>groups</code>	Listet alle Gruppenmitgliedschaften.
<code>chmod u+x file</code>	Macht <i>file</i> für den Eigentümer ausführbar.
<code>chmod o+r file</code>	Macht <i>file</i> für alle Nutzer lesbar. Eigentümer und Gruppenrechte werden nicht beeinflusst.
<code>chmod g+r file</code>	Macht <i>file</i> für die Gruppenmitglieder lesbar.

Kurzreferenz Unix

<code>chmod 640 test.txt</code>	Setzt die Rechte auf read und write für den Eigentümer, auf read für die Gruppe und sperrt alle Zugriffe für die Welt. Mit <code>ls -la test.txt</code> wird dann <pre>-rw-r----- 1 ss17 ite 122 Jan 24 2001 test.txt</pre> angezeigt.
<code>chmod 751 hello</code>	Setzt die Rechte auf read , write und execute für den Eigentümer, auf read und execute für die Gruppe und nur execute für die Welt. Mit <code>ls -la hello</code> wird dann <pre>-rwxr-x--x 1 ss17 ite 127 Apr 21 09:59 hello</pre> angezeigt.
<code>chgrp group file</code>	Datei <i>file</i> einer anderen Gruppe zuordnen, die mit groups angezeigt wird.
<code>chown owner file</code>	Ändern des Eigentümers der Datei <i>file</i> . (<i>Es sind höhere Rechte notwendig</i>)
Jobkontrolle	Jobs entsprechen von der Shell gestarteten Kommandos. Jobs können bel. viele Prozesse erstellen.
<code>jobs</code>	Listet alle Jobs (z.B. mit Ctrl-Z angehaltene). Job-Nummern beachten.
<code>fg %jobnumber</code>	Setzt angehaltenen Job im Vordergrund fort. Wird <i>jobnumber</i> weggelassen, wird der + Job fortgesetzt.
<code>bg %jobnumber</code>	Setzt angehaltenen Job im Hintergrund fort. Bei einer versuchter Terminalausgabe hält der Job an, wenn er nicht in einem eigenen Fenster läuft.
<code>kill %jobnumber</code>	Beendet den Job <i>jobnumber</i> und dessen verbundene Prozesse.
Beliebte Software	
<code>xemacs datei &</code>	Editor mit vielen Features. Es wird eine Datei mit dem Namen „ <i>datei</i> “ bearbeitet. Wenn diese noch nicht existiert, wird sie angelegt. Das „&“ am Ende der Zeile sorgt dafür, das der „xemacs“ als hintergrund Prozess gestartet wird. Dadurch ist das Terminalfenster danach sofort wieder bereit zur Eingabe. Dies ist möglich, da der „xemacs“ ein eigenes Fenster beim Start öffnet.
<code>gcc -o outfile file.c</code>	Übersetzt das C-Programm <i>file.c</i> . Das ausführbare Programm heißt <i>outfile</i> .
<code>ssh -X user@host</code>	Einloggen auf einer remote-Maschine. z.B.: <code>ssh -X ss17@serv9.inf.tu-dresden.de</code> <code>ssh -X ss17@irz751.inf.tu-dresden.de</code> <code>ssh -X sschoene@rcs52.urz.tu-dresden.de</code> Der Schalter <code>-X</code> ist nur unter einer X-Window Oberfläche möglich. Nach dem Einloggen in die remote-Maschine können dann auch auf dem entfernten Rechner Programme wie „xemacs“ gestartet werden. Dabei wird das „xemacs“ Fenster auf dem gleichen rechner geöffnet, wo auch die „ssh“ shell gestartet ist.
<code>scp user@host: datei .</code>	Kopiert Dateien und/oder Verzeichnisse von einem entfernten Rechner auf den lokalen Rechner oder umgekehrt. Dazu muß auf dem entfernten Rechner auch ein entsprechender „scp_Client“ laufen. Im Beispiel wird <i>datei</i> vom entfernten Rechner unter dem gleichen Namen im aktuellen Verzeichnis auf dem lokalen Rechner kopiert. Es gelten alle Schalter und Einstellungen des Kommandos <code>cp</code> .

Beispiele zu scp:

```
serv9 SS17:~/c_prog/c_tests> scp sschoene@rcs52.urz.tu-dresden.de:~/public_html/C/refcard_1.pdf .
sschoene@rcs52.urz.tu-dresden.de's password:
refcard_1.pdf  100% |*****| 103 KB  00:00
```

```
serv9 SS17:~/c_prog/c_tests> scp sschoene@rcs12.urz.tu-dresden.de:~/public_html/C/refcard_3.pdf .
The authenticity of host 'rcs12.urz.tu-dresden.de' can't be established.
DSA key fingerprint in md5 is: 95:09:b2:c6:4b:8d:d4:4c:a5:94:c0:34:8a:2e:7b:32
Are you sure you want to continue connecting(yes/no)?yes
Warning: Permanently added 'rcs12.urz.tu-dresden.de,141.30.60.22' (DSA) to the list of known hosts.
sschoene@rcs12.urz.tu-dresden.de's password:
scp: warning: Executing scp1.
select: Bad file number
```

Kurzreferenz Unix

```
serv9 SS17:~/c_prog/c_tests> scp sschoene@rcs52.urz.tu-dresden.de:~/public_html/C/refcard_3.pdf .
sschoene@rcs52.urz.tu-dresden.de's password:
refcard_3.pdf 100% |*****| 136 KB 00:00

serv9 SS17:~/c_prog/c_tests> ls -la refcard*
-rw-r--r-- 1 ss17 ite 106036 Apr 21 08:16 refcard_1.pdf
-rw-r--r-- 1 ss17 ite 140149 Apr 21 08:19 refcard_3.pdf
serv9 SS17:~/c_prog/c_tests>
```

Für Windows kann man ein ssh und scp von:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

<http://putty.cryptofr.net/>

<http://www.uni-giessen.de/hrz/software/putty/#tunnel>

<http://winscp.sourceforge.net>

herunterladen.

Alias für den Start des BlueJ einrichten

Um den BlueJ nicht immer mit der langen Zeile:

```
/usr/local/bluej/bluej &
```

starten zu müssen, kann man in die Datei „.cshrc“ einen Alias einbauen. Dazu wird der Texteditor „xemacs“ verwendet. Der Start des Texteditor erfolgt dabei aus dem privaten „HOME“ Verzeichnis wie folgt:

```
xemacs .cshrc &
```

Im Editorfenster ist dann hinter der letzten Zeile in der Datei „.cshrc“ der ALIAS einzufügen. Diese Zeile sieht wie folgt aus.

```
alias bluej „/usr/local/bluej/bluej &“
```

Nach dem Speichern der geänderten Datei und dem Verlassen des „EMACS“, muß nun noch das Konsolenfenster beendet und neu gestartet werden. Nun muß der neue Alias wirksam sein. Es werden alle aktiven Alias angezeigt, wenn man am Prompt das Kommando:

```
alias
```

eingibt. Es werden alle aktiven Alias angezeigt.

Der **BlueJ** kann jetzt mit dem Kommando:

```
bluej
```

gestartet werden.