

Linux Befehlsübersicht

Schnellreferenz

V 0.8 Mar-2003 Jörg Hartmann © megalearn www.megalearn.de

Hilfe	
man <i>Befehl</i>	die Handbuchseite von <i>Befehl</i> anzeigen
man -k <i>string</i>	in den Kurzbeschreibungen der Handbuchseiten nach <i>String</i> suchen
whatis <i>Befehl</i>	Kurzbeschreibung zu <i>Befehl</i> anzeigen
www.tldp.org	Linux Documentation Project
www.linuxhaven.de/dlhp/HOWTO/	Deutsches Linux Howto Projekt

Verzeichnisse	
pwd	aktuelles Verzeichnis anzeigen
ls [<i>option</i>] [<i>Dir</i>]	den Inhalt des Verzeichnisses <i>Dir</i> anzeigen, fehlt <i>Dir</i> , wird der Inhalt des aktuellen Verzeichnisses ausgegeben
-l	mehr Informationen anzeigen
-a	auch 'versteckte' Dateien .* anzeigen
-d	nur das Verzeichnis, nicht den Inhalt anzeigen
cd [<i>Dir</i>]	in das Verzeichnis [<i>Dir</i>] wechseln
mkdir <i>Dir</i>	das Verzeichnis <i>Dir</i> anlegen
rmdir <i>Dir</i>	das leere Verzeichnis <i>Dir</i> löschen
du [<i>option</i>] [<i>Dir</i>]	belegten Plattenplatz von [<i>Dir</i>] anzeigen
-s	nur eine Gesamtsumme anzeigen
-h	Ausgabe in Kilo-, Mega- bzw. Gigabyte
-a	Ausgabe für jede Datei, nicht nur Verzeichnisse

Dateien	
cp [<i>opt</i>] <i>Quelle</i> <i>Ziel</i>	Kopieren von <i>Quelle</i> nach <i>Ziel</i>
-R	rekursiv in Verzeichnisse absteigen
-v	Ablauf auf der Konsole protokollieren
-p	Datei-Attribute möglichst nicht ändern
-f	existierende Zieldateien überschreiben
rm [<i>option</i>] <i>Objekt</i>	Datei bzw. Verzeichnis <i>Objekt</i> löschen
-R	rekursiv in Verzeichnisse absteigen
-f	niemals nachfragen
-i	immer nachfragen
mv <i>Quelle</i> <i>Ziel</i>	Datei oder Verzeichnis <i>Objekt</i> umbenennen bzw. verschieben
file <i>Datei</i>	Anzeige des Types von <i>Datei</i>
ln [<i>opt</i>] <i>Quelle</i> <i>Ziel</i>	einen Link namens <i>Ziel</i> anlegen, der auf <i>Quelle</i> verweist
-s	einen Softlink anlegen

Speicherplatz	
df [<i>option</i>] [<i>Gerät</i>]	freien Speicherplatz ermitteln
-h	Angaben in Kilo-, Mega- bzw. Gigabyte
quota	Anzeige des Plattenplatzlimits
free	Anzeige der Hauptspeichernutzung

Suchen nach und in Dateien	
find <i>Pfad</i> <i>Objekt</i>	Objekt <i>Objekt</i> innerhalb von <i>Pfad</i> suchen
-name <i>foo</i>	Objekt heißt <i>foo</i>
-mmin <i>n</i>	letzte Änderung vor <i>n</i> Minuten (modification time)
-mtime <i>n</i>	letzte Änderung vor <i>n</i> * 24h
-group <i>foo</i>	Objekt gehört der Gruppe <i>foo</i>
-user <i>bar</i>	Objekt gehört dem Benutzer <i>bar</i>
-size <i>n</i> [<i>ck</i>]	Datei belegt <i>n</i> Blöcke, Zeichen (c) bzw. Kilobyte (k)
-type <i>x</i>	Objekt ist vom Typ Verzeichnis (d), Datei (f), symb. Link (l), Socket (s), Gerät (b c) oder Pipeline (p)
-exec	führt mit gefundenem Objekt ein Kommando aus, der Name wird als {} übergeben
[+] <i>n</i>	exakt <i>n</i> , bzw. größer (+) oder kleiner (-) als <i>n</i>
find /var -name "*.log" -exec rm -f {} \;	
grep [<i>option</i>] <i>Muster</i> <i>Datei</i>	<i>Muster</i> (reg. Ausdrücke) in <i>Datei</i> suchen
-C	nur die Anzahl der gefundenen Zeilen ausgeben
-f <i>Datei</i>	Suchmuster aus <i>Datei</i> lesen
-i	beim Suchen Groß-/Kleinschreibung mißachten
-n	jeder Ausgabezeile ihre Zeilennummer voranstellen
-r	rekursiv in Verzeichnissen suchen
-v	(invers) nur Zeilen ohne <i>Muster</i> ausgeben
-w	nur ganze Wörter finden (keine Teilzeichenketten)

Muster (reguläre Ausdrücke)	
[...]	Liste zutreffender Zeichen (Bsp: [abc])
[^...]	Liste nicht zutreffender Zeichen (Negation)
^, \$	bezeichnet den Zeilenanfang bzw. das Zeilenende
\<, \>	bezeichnet den Wortanfang bzw. das Wortende
Wiederholungs-Operatoren	
?	der vorangehende Eintrag null oder einmal
*	der vorangehende Eintrag null oder mehrmahlig
+	der vorangehende Eintrag ein- oder mehrmals
{ <i>n</i> }	der vorangehende Eintrag genau <i>n</i> mal
{ <i>n</i> ,}	der vorangehende Eintrag <i>n</i> oder mehrmals
{ <i>n</i> , <i>m</i> }	der vorangehende Eintrag mind. <i>n</i> , höchstens <i>m</i> mal

In gewöhnlichen regulären Ausdrücken besitzen die Metazeichen ?, +, {, }, |, (und) keine besondere Bedeutung. Statt dessen können die mit einem Rückstrich geschützten Versionen \?, \+ etc. benutzt werden.

```
grep -i foobar Datei
grep [Mm][ae][iy]e\?r Datei
```

Rechteverwaltung																															
chown [<i>option</i>] <i>Besitzer</i> [<i>Gruppe</i>] <i>Datei</i>	<i>Besitzer</i> von Dateien bzw. Verzeichnissen ändern																														
-R	rekursiv in Verzeichnisse absteigen																														
chmod [<i>option</i>] <i>Modus</i> <i>Datei</i>	Rechte an Dateien oder Verzeichnissen ändern																														
-R	rekursiv in Verzeichnisse absteigen																														
Modus	[<i>ugo</i>]+ - =[<i>rw</i> x] oder oktale Angabe: 0664																														
u	Rechte des Besitzers des Objektes (user)																														
g	Gruppenrechte (group)																														
o	der 'Rest der Welt' (other)																														
a	Alle, Zusammenfassung von 'ugo'																														
+, -, =	Recht hinzufügen, löschen bzw. exakt diese Rechte setzen																														
rwX	Leserecht, Schreibrecht, Ausführungsrecht bei Dateien bzw. das Recht in Verzeichnisse zu wechseln																														
	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th></th> <th colspan="3">user</th> <th colspan="3">group</th> <th colspan="3">other</th> </tr> </thead> <tbody> <tr> <td>oktal</td> <td>4</td><td>2</td><td>1</td> <td>4</td><td>2</td><td>1</td> <td>4</td><td>2</td><td>1</td> </tr> <tr> <td>symbolisch</td> <td>r</td><td>w</td><td>x</td> <td>r</td><td>w</td><td>x</td> <td>r</td><td>w</td><td>x</td> </tr> </tbody> </table>		user			group			other			oktal	4	2	1	4	2	1	4	2	1	symbolisch	r	w	x	r	w	x	r	w	x
	user			group			other																								
oktal	4	2	1	4	2	1	4	2	1																						
symbolisch	r	w	x	r	w	x	r	w	x																						
chmod 644 <i>Datei</i>	Benutzer erhält Lese- und Schreibrecht, die Gruppe und alle anderen haben nur Lese-recht																														
chmod g+rw <i>Datei</i>	der Gruppe wird Lese- und Schreibrecht gewährt																														

Benutzer	
passwd [<i>User</i>]	eigenes Paßwort, oder von <i>User</i> ändern (root)
last	Liste zuletzt angemeldeter Benutzer anzeigen
who	Liste gerade angemeldeter Benutzer anzeigen
su {- <i>User</i> }	Identität von root (-) oder <i>User</i> annehmen

Ausgabe

cat <i>Datei</i>	Datei auf stdout ausgeben
echo <i>String</i>	gibt <i>String</i> auf stdout aus
tail [<i>option</i>] <i>Datei</i>	die letzten Zeilen von <i>Datei</i> ausgeben
-n	<i>n</i> Zeilen ausgeben (default: 10)
-f	fortlaufend, neu hinzukommende Zeilen ausgeben
head [<i>option</i>] <i>Datei</i>	die ersten Zeilen von <i>Datei</i> ausgeben
-n	<i>n</i> Zeilen ausgeben
more <i>Datei</i>	Datei seitenweise ausgeben

Steuerkommandos

Leertaste	eine Seite vorwärts
b	eine Seite rückwärts
=	aktuelle Zeilennummer anzeigen
h	Hilfe anzeigen
/foobar	String foobar im Text suchen
n	nächstes Auftreten des Suchstrings finden
q	Quit - Programm beenden

Prozesse

ps [<i>option</i>]	Informationen über Prozesse anzeigen
a	alle Prozesse mit einem Terminal anzeigen
x	Prozesse ohne kontrollierendes Terminal
-C <i>Befehl</i>	Auswahl der anzuzeigenden Prozesse über den Befehlsnamen
U <i>User</i>	Prozesse des Benutzers <i>User</i> anzeigen
-H	Anzeige der Prozesshierarchie
die GNU/Linux Version von ps unterstützt Unix- und BSD-Optionen Unix-Optionen werden durch '-' eingeleitet, BSD-Optionen nicht die Options-Arten dürfen beliebig gemischt werden	
pstree	Prozessbaum anzeigen
top	Prozesse fortlaufend anzeigen
h	Hilfe anzeigen
q	Beenden von top
kill [- <i>signal</i>] <i>PID</i>	Signal an Prozess <i>PID</i> senden
15	SIGTERM : Terminate (Prozess beenden)
1	SIGHUP : Hang up (Verbindung beenden)
9	SIGKILL : Kill (Prozess abbrechen)
-l	Listet die verfügbaren Signale auf
Standradwert für <i>signal</i> ist SIGTERM (15)	
killall [<i>option</i>] <i>name</i>	Prozess <i>name</i> beenden
-i	interaktiv

Filter

sort [<i>option</i>] <i>file</i>	Datei sortieren und auf stdout ausgeben
-b	führende Leerzeichen ignorieren
-f	Groß- / Kleinschreibung ignorieren
-n	anhand des Zahlenwertes sortieren
-r	Sortierrichtung umkehren (reverse)
-t < <i>sep</i> >	Feldtrenner < <i>sep</i> > benutzen
cut [<i>option</i>] <i>file</i>	Ausgabe ausgewählter Spalten (Felder) von <i>Datei</i>
-d< <i>Zeichen</i> >	Spalten sind durch <i>Zeichen</i> begrenzt
-f< <i>Liste</i> >	<i>Liste</i> der auszugebenden Felder
paste <i>file1 file2</i>	<i>file1</i> und <i>file2</i> spaltenweise zusammenfügen
uniq [<i>option</i>] <i>file1 file2</i>	doppelte Zeilen aus einer sortierten Datei entfernen
-i	Groß- / Kleinschreibung mißachten
-d	nur die doppelten Zeilen ausgeben
-c	beginne alle Zeilen mit der Anzahl der Vorkommen
wc [<i>option</i>]	Zählen der Eingabewerte (word count)
-c	Buchstaben zählen
-w	Worte zählen
-l	Zeilen zählen

Bash Kommandozeile

ctrl-c	Kommandozeile verwerfen
ctrl-u	Kommandozeile von Cursor bis Zeilenanfang löschen
ctrl-z	interaktiven Prozess anhalten (suspend)
ctrl-l	Bildschirm löschen (clear) (kleines 'el')
ctrl-s	Bildschirmanzeige anhalten
ctrl-q	Bildschirmanzeige nach ^S wieder fortsetzen
ctrl-p	letzten Befehl wiederholen (auch !!)
jobs	eigene im Hintergrund laufende Jobs anzeigen
bg [%job]	angehaltenen (suspended) Job %job in den Hintergrund schieben
fg [%job]	Hintergrundprozess %job interaktiv machen, d.h. in den Vordergrund bringen
cmd &	Befehl <i>cmd</i> im Hintergrund ausführen
cmd1 ; cmd2	durch ; getrennte Liste von Befehlen, die nacheinander abzuarbeiten sind
cmd1 && cmd2	cmd2 wird nur dann ausgeführt, wenn cmd1 ohne Fehler lief (UND)
cmd1 cmd2	cmd2 wird nur dann ausgeführt, wenn cmd1 mit Fehler abbrach (ODER)

Ein-/Ausgabeumlenkung

0	Standard-Eingabe (stdin)
1	Standard-Ausgabe (stdout)
2	Standard-Fehlerkanal (stderr)
cmd1 cmd2	die Ausgaben (stdout) von cmd1 werden cmd2 als Eingaben (stdin) übergeben
cmd > file	cmd schreibt seine Ausgaben (stdout) in file
cmd >> file	cmd hängt seine Ausgaben (stdout) an file an
cmd < file	cmd liest seine Eingaben (stdin) aus file
cmd < file1 > file2	cmd liest seine Eingaben (stdin) aus file1 und schreibt seine Ausgaben (stderr) nach file2
cmd 2> file	cmd schreibt seine Fehlermeldungen (stderr) in die Datei file
cmd > file 2>&1	Fehlermeldungen werden auf stdout umgeleitet und alle Ausgaben (stdout) werden in die Datei file geschrieben

Bash Shell-Variablen

FOO=bar	der Variablen FOO den Wert bar zuweisen
\$FOO	den Wert der Variablen benutzen (Referenz)
export FOO	FOO exportieren, d.h. anderen Prozessen zugänglich machen
FOO=\$(Befehl)	die Ausgabe des Befehls wird zum Wert der Variable FOO
HOME	Pfad zum Home-Dir des Users
PATH	Liste von Verzeichnissen, in denen nach Programmen gesucht wird
SHELL	Angabe des Kommandointerpreters
UID	numerische User-ID
PS1	der primäre Prompt
DISPLAY	Ausgabegerät des X Window System (Host/Display)
USER	der aktuelle Benutzername
TERM	der eingestellte Terminaltyp
unset FOO	Variable FOO löschen

Metacharacters und Dateinamenserweiterung

?	steht für ein beliebiges Zeichen
*	steht für null oder mehr Zeichen
[...]	Liste, steht für eines der eingeschlossenen Zeichen
{foo,bar}	Liste, steht für jeden der eingeschlossenen Strings
Quoting	nimmt den Metacharacters ihre Sonderbedeutung
\	nimmt dem folgenden Metachar. seine Sonderbedeutung, er wird als normales Zeichen behandelt
'string'	Metachar. werden nur als Zeichen behandelt
"string"	nur der Metacharacter \$ behält der Sonderbedeutung