

bash – Shellbefehle

(nach Aufgaben gruppiert)

Webseiten zur weiteren Information über Shellbefehle:

- <https://wiki.ubuntuusers.de/Shell/Befehls%C3%BCbersicht/>
- <http://www.shellbefehle.de>
- <http://linuxwiki.de/LinuxKommandos>
- https://en.wikipedia.org/wiki/List_of_Unix_commands

Syntax in der folgenden Befehlsbeschreibung:

- Großbuchstaben + Unterstreich: Platzhalter für einen beliebigen String (Beispiel: NAME)
- Eckige Klammern: In den Klammern steht eine optionale Ergänzung, die weggelassen werden kann (Beispiel: [-r])

Basis-Befehle:

Befehl	Auswirkung
apropos <u>STRING</u>	Sucht den STRING in allen Hilfetexten und gibt die gefundenen Befehle aus (stdout)
man <u>COMMAND</u>	Manual-Eintrag (Hilfetext) für den Befehl COMMAND
cat <u>FILE</u>	Inhalt der Textdatei FILE komplett ausgeben (stdout)
more <u>FILE</u>	Inhalt der Textdatei FILE seitenweise ausgeben (stdout) Leertaste: Seite weiter, Enter: Zeile weiter, q: Ende
head [-n] <u>FILE</u>	Die ersten n Zeilen der Textdatei FILE ausgeben (stdout)
sort [<u>FILE</u>] ..	Die Zeilen aller Textdateien sortiert ausgeben (stdout)
echo <u>STRING</u>	Zeichenkette STRING ausgeben (stdout)
date	Aktuelles Datum und Uhrzeit ausgeben (stdout)
alias <u>NEW=OLD</u>	Zeichenersetzung von String OLD durch NEW definieren
> <u>FILE</u>	Standardausgabe (stdout) auf FILE umlenken, FILE ggf. neu erzeugen oder überschreiben
sudo <u>COMMAND</u>	Den Befehl COMMAND als Superuser ausführen
exit Strg-d	Shell beenden / stdin schließen

Prozesse:

Befehl	Auswirkung
<code>ps [-ef]</code>	Informationen über alle Prozesse ausgeben
<code>pstree [-ch]</code>	Prozess-Informationen als Baumstruktur ausgeben (Eltern-/Kindprozesse)
<code>top [-d <u>SECS</u>]</code>	Informationen über alle Prozesse ausgeben und nach SECS Sekunden aktualisieren (Default-Wert: 3) Nimmt interaktiv Befehle an (h: Hilfe, q: Ende)
<code>kill [-9] <u>PID</u></code>	Prozess mit der Prozess-ID PID abbrechen (beenden)

Start von ausführbaren Programmdateien:

Befehl	Auswirkung
<code>chmod a+x <u>FILE</u></code>	Datei FILE als ausführbares Programm deklarieren (für alle Benutzer Datei-Attribut „x“ <i>executable</i> setzen)
<code><u>PROG</u></code>	Ausführbares Programm PROG starten (in \$PATH suchen)
<code><u>PROG1</u> <u>PROG2</u></code>	Befehlsverkettung (Pipe): Ausgabe PROG1 = Eingabe PROG2
<code><u>PROG</u> &</code>	Programm PROG direkt im Hintergrund starten (ohne Benutzereingaben)
<code>Strg-c</code>	Laufendes Programm abbrechen

Dateien und Verzeichnisse (1):

Befehl	Auswirkung
<code>ls [-la] [<u>SPEC</u>]</code>	Inhalt des aktuellen Arbeitsverzeichnisses als Liste von Dateinamen ausgeben. Übergebene Infos SPEC (Dateiname oder Verzeichnis) werden verwendet
<code>pwd</code>	Namen des aktuellen Arbeitsverzeichnisses ausgeben
<code>cd [<u>DIR</u>]</code>	Aktuelles Arbeitsverzeichnis zu DIR wechseln
<code>df [-h]</code>	Informationen über Dateisysteme anzeigen
<code>du [-h -d <u>LEVEL</u>] [<u>DIR</u>]</code>	Platzverbrauch für Verzeichnisse anzeigen, beginnend bei DIR oder im aktuellen Verzeichnis
<code>mkdir <u>DIR</u></code>	Neues Verzeichnis im aktuellen Arbeitsverzeichnis erstellen
<code>rmdir <u>DIR</u></code>	Verzeichnis DIR löschen (muss leer sein bis auf . und ..)
<code>rm [-ir] <u>FILE</u></code>	Datei FILE löschen, ggf. rekursiv (über mehrere Baumebenen inkl. Verzeichnissen)

Dateien und Verzeichnisse (2): Suchen und Finden

Befehl	Auswirkung
<code>find <u>DIR</u> -name <u>FILE</u> -print</code>	Suche in allen Verzeichnissen und Unterverzeichnissen (beginnend im Verzeichnis DIR) eine Datei namens FILE und gib den Dateipfad aus
<code>locate <u>STRING</u></code>	Suche in allen Verzeichnissen alle Dateien, in deren Namen die Zeichenkette STRING vorkommt und gib den Dateipfad aus (Achtung: Suche in einer Datenbank! Aktualisierung mit <code>updatedb</code>)
<code>updatedb</code>	Aktualisiere die Verzeichnisinformationen für den <code>locate</code> -Befehl
<code>grep [-r] <u>STRING</u> [<u>FILE</u>]</code>	Suche <u>in der Datei</u> FILE (oder <code>stdin</code>) nach der Zeichenkette STRING, ggf. rekursiv in allen Unterverzeichnissen

Dateien und Verzeichnisse (3): Kopieren und Verschieben

Befehl	Auswirkung
<code>cp [-i] <u>SOURCE</u> <u>DEST</u></code>	Kopiere die Datei SOURCE. Die neue Dateikopie heißt DEST (im selben Verzeichnis) oder liegt im Verzeichnis DEST unter dem Namen SOURCE (wenn DEST ein Verzeichnis ist)
<code>mv <u>SOURCE</u> <u>DEST</u></code>	Datei SOURCE zu DEST umbenennen oder ins Verzeichnis DEST verschieben
<code>ln [-s] <u>DEST</u> <u>LINK</u></code>	[symbolischen] Verweis („Link“) LINK → DEST erzeugen
<code>mount -t <u>TYPE</u> <u>DEVICE</u> <u>DIR</u></code>	„Einhängen“ eines anderen Dateisystems vom Typ TYPE, das sich auf dem Gerät DEVICE befindet, in das Verzeichnis DIR