

Linux/Unix Bash Terminal Cheat Sheet

(Computergrundlagen WS 09/10)

- Aufbau eines Kommandos: `command [options] [arguments]`
- Hilfe zum Befehl "command" mit: `man command` oder kürzer `command --help` (Bewegung in den man-files mit Pfeiltasten, q für "quit")
- "Escapen" von Steuerungszeichen mit Anführungszeichen oder dem Backslash \
- Mit der Tab-Taste werden Kommandos und Dateinamen vervollständigt.
- Die Datei `~/ .bashrc` wird am Anfang jeder `bash` ausgeführt.

Dateioperationen

<code>ls</code>		Auflistung der Dateien im Verzeichnis auf die Standardausgabe
	<code>-l</code>	mehr Information (Größe, Rechte, ...)
	<code>-A</code>	auch versteckte Dateien
	<code>-h</code>	Dateigröße "human readable"
<code>cd dir</code>		in Verzeichnis "dir" wechseln
<code>pwd</code>		aktuelles Verzeichnis ausgeben
<code>mkdir dir</code>		Verzeichnis "dir" erstellen
<code>rm file</code>		Datei "file" löschen
	<code>-r</code>	rekursiv (für ganze Verzeichnisse)
	<code>-f</code>	ohne Nachfragen
<code>cp file1 file2</code>		file1 nach file2 kopieren
	<code>-r</code>	rekursiv (ganzes Verzeichnis)
<code>more file</code> oder <code>less file</code>		Datei anzeigen (mit q beenden)
<code>nano file</code>		Sehr einfacher Texteditor
<code>ln -s file linkname</code>		symbolischen Link erstellen

Besonderheiten und Wildcards

<code>cd</code>		ohne Argument: Ins Home wechseln
<code>~/</code>		Home-Verzeichnis
<code>./</code>		aktuelles Verzeichnis
<code>../</code>		ein Verzeichnis höher im Baum
<code>/</code>		Wurzelverzeichnis
<code>*</code>		beliebig lange Folge beliebiger Zeichen
<code>?</code>		ein einzelnes beliebiges Zeichen

Standardausgabe

<code>echo string</code>		"string" auf die Standardausgabe ausgeben
<code>cat file</code>		Inhalt von "file" auf die Standardausgabe ausgeben
<code>command > file</code>		Standardausgabe von "command" in "file" weiterleiten, "file" überschreiben
<code>command >> file</code>		Standardausgabe an "file" anhängen
<code>command < file</code>		Inhalt von "file" als Eingabe in Befehl leiten
<code>command1 command2</code>		Standardausgabe von "command1" an "command2" übergeben

Mehrere Befehle

<code>command &</code>		Kommando im Hintergrund ausführen
<code>command1 & command2</code>		command1 und command2 werden gleichzeitig ausgeführt
<code>command1 ; command2</code>		command2 wird nach command1 ausgeführt
<code>command1 && command2</code>		command2 wird nach command1 ausgeführt, falls dieses keinen Fehler meldet
<code>command1 command2</code>		command2 wird nach command1 ausgeführt, falls dieses einen Fehler meldet

Suchen

<pre>find dir -name "pattern" locate file grep pattern files ls * grep "gesucht"</pre>		<p>Suche "pattern" im Verzeichnis "dir" Schnelles Finden einer Datei, die "file" im Namen hat. Nicht auf jedem System möglich!</p> <p>Suchen nach Zeilen in "files", die "pattern" enthalten grep funktioniert auch mit Standardeingabe</p>
---	--	---

Kompression

<pre>tar [f tar-filename] files tar cvzf file.tar.gz dir/</pre>	<p>f c x z j v</p>	<p>Dateien oder Verzeichnisse "files" in tarball (mit Namen "tar-filename") packen notwendig, wenn Dateiname angegeben wird! Sollte immer am Ende der Optionen stehen!</p> <p>tarball erstellen (create) tarball auslesen (extract) tarball zusätzlich mit "gzip" packen oder entpacken tarball zusätzlich mit "bzip2" packen oder entpacken mehr Ausgabe, was gemacht wird, zur Kontrolle Beispiel: Verzeichnis "dir" wird in einen tarball gesteckt und dieser mit gzip gepackt. Das ganze kommt dann in die Datei "file.tar.gz"</p>
--	--	---

Prozessmanagement

<pre>top ps killall process kill procnum</pre>	<p>-o -a -9</p>	<p>zeigt die aktuell laufenden Prozesse an Ordnet die Prozesse nach Verbrauch (Beispiel: -o cpu Prozessorleistung)</p> <p>gibt alle laufenden Prozesse mit Prozessnummer auf die Standardausgabe aus auch die Prozesse anderer Benutzer schießt alle Befehle mit Namen "process" ab! schießt den Prozess mit der Nummer "procnum" ab schießt den Prozess ab, auch wenn er nicht mehr antwortet! Vorsicht!</p>
--	-------------------------	--

Tastenkürzel

<pre>Strg+c Strg+z fg bg Strg+u ↑ Strg+r Strg+a Strg+e</pre>		<p>bricht das aktuell laufende Kommando ab pausiert das aktuell laufende Kommando letztes pausiertes Kommando im Vordergrund weiter laufen lassen pausiertes Kommando im Hintergrund weiter laufen lassen löscht die aktuelle Zeile komplett nach und nach die letzten Kommandos anzeigen Tippen um nach letztem ähnlichen Kommando zu suchen Springt zum Anfang der Zeile Springt zum Ende der Zeile</p>
---	--	---

Bash-Variablen anhand eines Beispiels

<pre>export VARNAME=file.txt echo \$VARNAME echo \${VARNAME%.txt} echo \${VARNAME#file} echo \${VARNAME/txt/dat}</pre>		<p>Variable setzen. Variablen sind in Großbuchstaben. Gesetzte Variablen können mit \$ referenziert werden In geschweiften Klammern können die Werte verändert werden. Hier: % schneidet die nachfolgende Zeichenfolge hinten ab. # schneidet die Zeichenfolge vorne ab ersetzt "txt" mit "dat"</p>
---	--	---

Netzwerk-Tools

<pre>ssh user@host scp user@host:file.dat ./ rsync user@host:file ./</pre>		<p>von extern auf einem Computer einloggen Datei "file.dat" von anderem Computer ins aktuelle Verzeichnis kopieren Das gleiche wie scp, nur mit updates, falls manche Dateien schon existieren</p>
---	--	--