

# Basic UNIX/LINUX commands

Dateien und Verzeichnisse (Ordner)		Umgebung		Skriptsprachen und Compiler	
				awk	Textmuster
~ Homeverzeichnis . Arbeitsverzeichnis (derzeitiges)		~/ .bash* Definition der Umgebung		bc	Rechner
		alias	Abkürzungen erzeugen/anzeigen		
cd	Verzeichnis wechseln	env	Umgebungsvariablen anzeigen	perl	Datenbank
pwd	Arbeitsverzeichnis ausgeben	set	Umgebungsvariablen definieren	make	Dateien
ls	Verzeichnisinhalt ausgeben	export	Shellvariablen exportieren	gcc	GNU C-Compiler
mkdir	Verzeichnis erstellen	whereis	Befehlsstrukturen suchen	gfortran	GNU FORTRAN-Compiler
rmdir	Verzeichnis löschen	which	Programmpfad ausgeben	gdb	GNU Debugger
rm	Datei und/oder Verzeichnis löschen	function	Funktion definieren		
chmod	Datei-/Verzeichnisstatus ändern			<b>Ausgabe</b>	
chown	Datei-/Verzeichnisbesitzer ändern	<b>Prozesse</b> CTRL - c Vordergrundprozess abbrechen CTRL - z Vordergrundprozess		echo	unformatierte Ausgabe
ln	Verweis erzeugen			printf	formatierte Ausgabe

cp	Datei/Verzeichnis kopieren	anhalten ... & Prozess im Hintergrund starten		cut	Ausschneiden von Zeichen
mv	Datei/Verzeichnis verschieben				
find	Datei/Verzeichnis suchen	top	Prozesse anzeigen	Hilfe	
locate	Datei/Verzeichnis suchen	ps	Prozesse anzeigen	man	Hilfe für Befehle/Programme
du	Datei/Verzeichnisgröße ausgeben	kill	Prozesse beenden	help	Hilfe für Kernelbefehle
df	Festplattenspeicherbenutzung	fg	Prozess in den Vordergrund holen	info	
touch	Zeitstempel erneuern	bg	Prozess in den Hintergrund senden		
				Fast jeder Befehl hat ein oder mehrere Argumente und kann durch Wahl von Optionen spezifiziert werden. Syntax:	
<b>Inhalte von Dateien</b>		<b>Prozessausgabe und -verknüpfung</b>		<p>command [options] [arguments]</p> <p>Optionen beginnen in der Regel mit einem - (manchmal mit einem +). Beispiel:</p> <p><i>Ls -a -l ordnername</i></p> <p>Manche Befehle brauchen kein Argument, bzw. haben ein Default-Argument, z.B.</p> <p><i>Ls -a -l</i></p> <p>bezieht sich auf das derzeitige Arbeitsverzeichnis. Auch Optionen können Argumente haben, z.B. die ignore-Option:</p>	
cat	Inhalt ausgeben	Weiterleitung der Standardausgabe			
more	Inhalt seitenweise ausgeben	> Umleitung der Standardausgabe in Datei			
less	Inhalt seitenweise ausgeben	>> Anhängen der Standardausgabe an Datei			
sed	Inhalt zeilenweise manipulieren	>& Umleitung des Standardfehlers in Datei			
vi, vim	Inhalt editieren	<< ... END Umleitung von Text zum Befehl			
wc	Inhalt zählen	tee	Duplizieren der Standardausgabe		

grep	Inhalt durchsuchen			<i>ls -a -l -I*.o</i>
diff	Inhalt vergleichen	<b>Archive</b>		Optionen können in der Regel zusammengefasst werden:
sort	Inhalt sortieren	tar	komprimierte Archive (ent)packen	<i>ls -aLI*.o</i>
head	Anfang ausgeben	gzip	komprimierte Archive (ent)packen	Manche Befehle benötigen zwei oder mehr Argumente: <i>cp -rp name_quelle name_ziel</i>
tail	Ende ausgeben	bzip2	komprimierte Archive (ent)packen	
tac	Inhalt rückwärts ausgeben			

Anmerkung: Die meisten Befehle haben die Form von ausführbaren Dateien und sind nicht fest im Betriebssystem integriert. So kann es je nach Installation passieren, dass manche Befehle/Programme nicht vorhanden sind. Insbesondere die GNU-Compiler sind auf einfachen Systemen oft nicht installiert.