

UNIX/Linux-Minikurs

Ralf Seidler, Daniel Walther, Torsten Bosse

Institut für Informatik
Friedrich-Schiller-Universität Jena

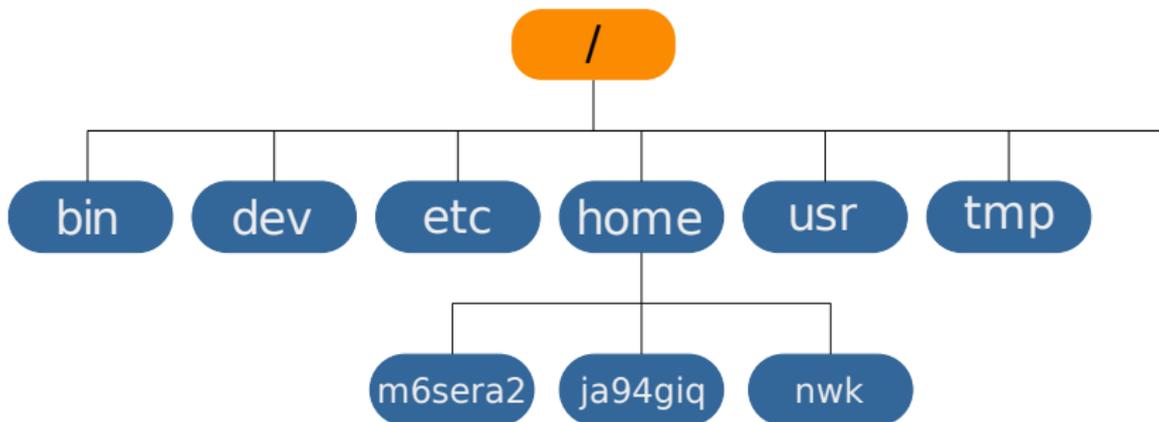


seit 1558

Anmeldung

- ▶ Melden Sie sich mit username und password an!
- ▶ Starten Sie ein Terminal!

Arbeiten mit Verzeichnissen



- ▶ Lassen Sie sich das aktuelle Verzeichnis und seinen Inhalt anzeigen!

\$

\$

- ▶ Lassen Sie sich das aktuelle Verzeichnis und seinen Inhalt anzeigen!

```
$ pwd
```

```
$ ls
```

- ▶ Führen Sie folgende Befehle aus

```
$ cd /bin
```

```
$ cd /opt
```

```
$ cd -
```

```
$ cd
```

```
$ cd ~
```



NAME

killall - kill processes by name

SYNOPSIS

```
killall [-Z,--context pattern] [-e,--exact] [-g,--process-group]
[-i,--interactive] [-q,--quiet] [-r,--regexp] [-s,--signal signal]
[-u,--user user] [-v,--verbose] [-w,--wait] [-l,--ignore-case]
[-V,--version] [--] name ...
killall -l
killall -V,--version
```

DESCRIPTION

killall sends a signal to all processes running any of the specified commands. If no signal name is specified, SIGTERM is sent

```
m6sera2@ppc624:~$ killall --help
```

```
Aufruf: killall - [OPTION]... [--]NAME...
```

```
killall -l, --list
```

```
killall -V, --version
```

-e,--exact	verlange genaue Uebereinstimmung fuer sehr lange Namen
-I,--ignore-case	Groß- und Kleinschreibung nicht beachten
-g,--process-group	breche Prozessgruppe statt Einzelprozess ab
-y,--younger-than	Prozesse juenger als ZEIT abbrechen
-o,--older-than	Prozesse aelter als ZEIT abbrechen
-i,--interactive	verlange vor Abbruch Bestaetigung des Benutzers
-l,--list	liste alle bekannten Signalnamen auf
-q,--quiet	keine Warnungen und Fehler ausgeben
-r,--regex	NAME ist ein erweiterter regulaerer Ausdruck
-s,--signal SIGNAL	sende benutzerdefiniertes Signal anstatt SIGTERM
-u,--user USER	nur Prozesse von angegebenem Benutzer abbrechen
-v,--verbose	berichte, falls das Signal erfolgreich gesendet wurde
-V,--version	zeige Version
-w,--wait	warte auf das Ende der Prozesse

\$ Kommando [Optionen] Argument(e)

- ▶ Finden Sie mit Hilfe der man-page heraus, wie Sie sich detaillierte Informationen zu jeder Datei und versteckte Dateien anzeigen lassen können!

\$

\$

\$

\$

- ▶ Wodurch wird eine versteckte Datei / ein verstecktes Verzeichnis gekennzeichnet?

- ▶ Finden Sie mit Hilfe der man-page heraus, wie Sie sich detaillierte Informationen zu jeder Datei und versteckte Dateien anzeigen lassen können!

\$ man ls

\$ ls -l

\$ ls -a

\$ ls -la

- ▶ Wodurch wird eine versteckte Datei / ein verstecktes Verzeichnis gekennzeichnet?
Durch den führenden Punkt im Dateinamen!



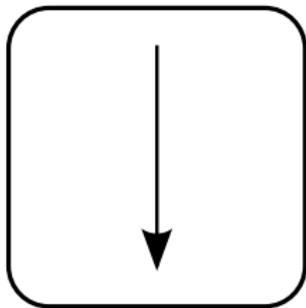
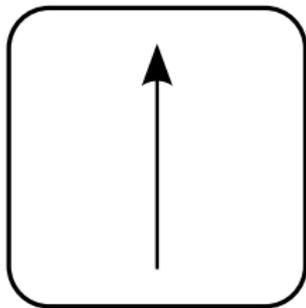
drwxrwxrwx m6sera2 ira



Shift  + Pg 



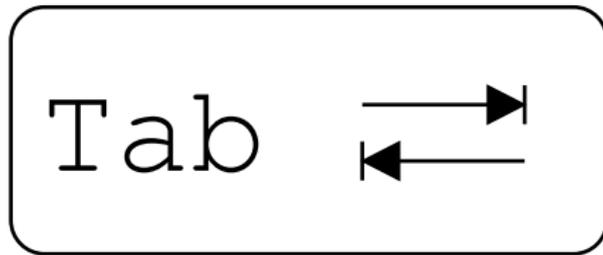
- ▶ Testen Sie Mausrad und Shift+PgUp!



String[^] + R

- ▶ Testen Sie die folgende Tasten(kombinationen):
 - \$ Cursor-Up/Down
 - \$ Strg+r
 - \$ opt
 - \$

- ▶ Testen Sie die folgende Tasten(kombinationen):
 - \$ Cursor-Up/Down
 - \$ Strg+r
 - \$ opt
 - \$ cd /opt



- ▶ Begrüßen Sie den Tabulator:

```
$ cd /opt
```

```
$ cd g<Tab>
```

```
$ <Tab>
```

```
$ <Tab><Tab>
```

```
$
```

- ▶ Begrüßen Sie den Tabulator:

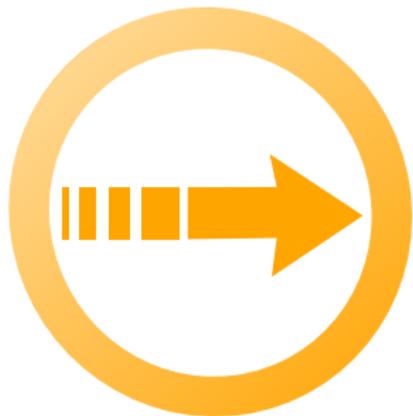
```
$ cd /opt
```

```
$ cd g<Tab>
```

```
$ <Tab>
```

```
$ <Tab><Tab>
```

```
$ cd google/chrome/
```



```
mkdir path
```

```
cp src [[src]] dst
```

```
mv src [[src]] dst
```

```
rm [-rf] victim
```

```
rw [-rf] victim
```

```
wa src [[src]] dst
```

```
cb src [[src]] dst
```

- ▶ Erzeugen Sie in Ihrem Home-Verzeichnis ein Unterverzeichnis *tmp*, wechseln Sie in dieses Verzeichnis und überzeugen Sie sich, daß Sie sich im richtigen Verzeichnis befinden!

\$

\$

\$

\$

- ▶ Erzeugen Sie in Ihrem Home-Verzeichnis ein Unterverzeichnis *tmp*, wechseln Sie in dieses Verzeichnis und überzeugen Sie sich, daß Sie sich im richtigen Verzeichnis befinden!

```
$ cd
```

```
$ mkdir tmp
```

```
$ cd tmp
```

```
$ pwd
```

/

./

..

- ▶ Lassen Sie sich den Inhalt des aktuellen und des übergeordneten Verzeichnisses ausgeben!

\$ ls

\$ ls .

\$ ls ..

\$

\$

- ▶ Lassen Sie sich den Inhalt des aktuellen und des übergeordneten Verzeichnisses ausgeben!

```
$ ls
```

```
$ ls .
```

```
$ ls ..
```

```
$ ls ~/tmp
```

```
$ ls /tmp
```

- ▶ Erzeugen Sie ein tiefes Unterverzeichnis
\$ mkdir -p ich/bin/ein/subdir
- ▶ Lassen Sie sich das Unterverzeichnis `subdir` anzeigen
\$
\$

- ▶ Erzeugen Sie ein tiefes Unterverzeichnis
\$ mkdir -p ich/bin/ein/subdir
- ▶ Lassen Sie sich das Unterverzeichnis `subdir` anzeigen
\$ ls i<Tab><Tab><Tab><Tab>
\$ ls ich/bin/ein/subdir

- ▶ Löschen Sie alle Unterverzeichnisse in `~/tmp/` mit einem einzigen Befehl!

```
$ rm i*
```

```
$
```

- ▶ Löschen Sie alle Unterverzeichnisse in `~/tmp/` mit einem einzigen Befehl!

```
$ rm i*
```

```
$ rm -rf i*
```

Dateien und Ausgabeumleitung



Umleitung

Kommando > Datei

Kommando >> Datei

Kommando | Kommando

```
Kommando | Kommando
Kommando >> Datei
Kommando > Datei
```

- ▶ Leiten Sie das aktuelle Datum *date* in die Datei `/tmp/whatever` um. Lassen Sie sich diese Datei mit *cat* anzeigen.

\$

\$

\$

- ▶ Leiten Sie das aktuelle Datum *date* in die Datei */tmp/whatever* um. Lassen Sie sich diese Datei mit *cat* anzeigen.

```
$ date > /tmp/whatever
```

```
$ cat <esc>.
```

```
$ cat /tmp/whatever
```

- ▶ Leiten Sie das aktuelle Datum *date* in das Programm *wc* um.

\$

\$

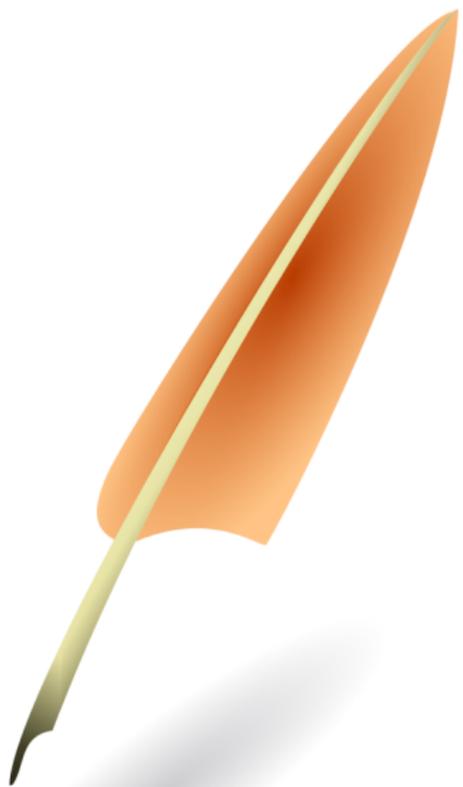
- ▶ Leiten Sie das aktuelle Datum *date* in das Programm *wc* um.

```
$ date | wc
```

```
1  6 30
```

```
$ rm wc
```

Editoren, Scripte, Permissions



vi

vim

gvim

emacs

xemacs

kwrite

kate

nano

joe

jed

- ▶ Starten Sie den Editor *gedit* und erzeugen Sie die Datei *script*
\$ gedit script
- ▶ mit folgendem Inhalt:
#!/bin/bash
ICH=\$(whoami)
ZEIT=5
echo "\${ICH} hat in den letzten \${ZEIT}
Minuten folgende Dateien erzeugt:"
find ~/ -cmin -\${ZEIT}

- ▶ Lassen Sie sich die Zugriffsrechte von *script* anzeigen!

```
$
```

- ▶ Machen Sie *script* ausführbar und überprüfen Sie die Zugriffsrechte.

```
$ chmod +x script
```

```
$
```

- ▶ Lassen Sie sich die Zugriffsrechte von *script* anzeigen!

```
$ ls -la script
```

- ▶ Machen Sie *script* ausführbar und überprüfen Sie die Zugriffsrechte.

```
$ chmod +x script
```

```
$ ls -la script
```

- ▶ Starten Sie das Script!
\$./script

SCP



```
scp [-r] Quelle[n][user@]host:[Ziel]
```

```
scp [-r] Quelle[n][user@]host:[Ziel]
```

- ▶ Erstellen Sie eine Prüfsumme Ihres Scripts!

```
$ sha1sum script
```

- ▶ Kopieren Sie *script* per scp!

```
$ scp script friedolin-login@login.rz.uni-jena.de:
```

SSH



```
ssh [user@]host [Kommando]
```

```
ssh [user@]host [Kommando]
```

- ▶ Loggen Sie sich auf *login.rz.uni-jena.de* ein:
\$ ssh **friedolin-login**@login.rz.uni-jena.de

- ▶ Geben Sie den Hostname der Maschine aus:
\$ hostname

- ▶ Erstellen und vergleichen Sie die Prüfsumme Ihres Scripts:
\$

- ▶ Loggen Sie sich auf *login.rz.uni-jena.de* ein:
\$ ssh **friedolin-login**@login.rz.uni-jena.de

- ▶ Geben Sie den Hostname der Maschine aus:
\$ hostname
login.rz.uni-jena.de

- ▶ Erstellen und vergleichen Sie die Prüfsumme Ihres Scripts:
\$ sha1sum script

- ▶ Starten Sie das Script!
\$./script



putty & winscp

Aufräumen und Ausloggen

- ▶ Löschen Sie das script und loggen Sie sich aus dem URZ aus:
\$ rm script
\$ exit
- ▶ Wechseln Sie in Ihr Home-Verzeichnis!
\$
- ▶ Löschen Sie Ihr temporäres Verzeichnis!
\$
- ▶ Beenden Sie das Terminal!
\$
- ▶ Loggen Sie sich aus!

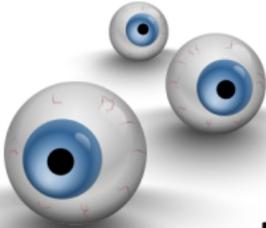
- ▶ Löschen Sie das script und loggen Sie sich aus dem URZ aus:
\$ rm script
\$ exit
- ▶ Wechseln Sie in Ihr Home-Verzeichnis!
\$ cd
- ▶ Löschen Sie Ihr temporäres Verzeichnis!
\$ rm -rf tmp
- ▶ Beenden Sie das Terminal!
\$ exit
- ▶ Loggen Sie sich aus!

Hausaufgaben

- ▶ Unix hat noch 5000 andere tolle Befehle.
Informieren Sie sich über
 - \$ man grep
 - \$ man lpr
 - \$ man file
 - \$ man screen
 - \$ man apropos
 - \$ man which
 - \$ man tar
 - \$ man less
 - \$ man make
 - \$... kill, killall, pgrep, top, ps, w,...

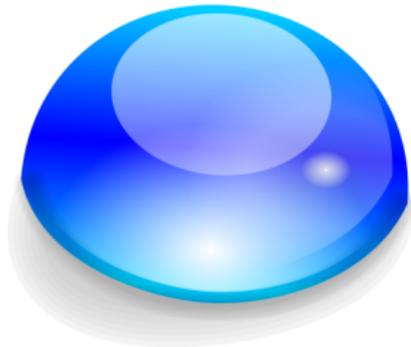
top - 15:14:03 up 85 days, 4:43, 1 user, load average: 1.00, 1.08, 1.08
 Tasks: 140 total, 2 running, 138 sleeping, 0 stopped, 0 zombie
 Cpu(s): 0.4%us, 0.2%sy, 16.7%ni, 82.3%id, 0.4%wa, 0.0%hi, 0.1%si, 0.0%st
 Mem: 24907336k total, 24441656k used, 465680k free, 727216k buffers
 Swap: 0k total, 0k used, 0k free, 21506424k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11761	kobo	39	19	1144m	8112	32	R	100	0.0	22765:33	java
27084	gridsphe	20	0	695m	129m	13m	S	2	0.5	275:47.32	java
25348	adi	20	0	3112	1600	1264	R	1	0.0	0:00.11	top
8444	postgres	20	0	46056	5696	4792	S	0	0.0	2:19.59	postgres
1	root	20	0	2544	792	664	S	0	0.0	0:23.06	init
2	root	15	-5	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0	0.0	0:09.72	migration/3
4	root	15	-5	0	0	0	S	0	0.0	0:03.99	ksoftirqd/3



ps - report a snapshot of the current processes.
top - display Linux tasks
pgrep - look up processes based on name

printenv - print all or part of environment



```
export VARIABLE=WERT  
unset VARIABLE  
echo $VARIABLE
```

```
echo $VARIABLE  
unset VARIABLE  
export VARIABLE=WERT
```

```
SHELL=/bin/bash
EDITOR=vim
PWD=/home/m6sera2
USERNAME=m6sera2
HOME=/home/m6sera2
LANG=C
DISPLAY=:0.0
```



```
PATH=/usr/local/bin:/usr/bin:/bin  
LD_LIBRARY_PATH=/my/own/libdir
```

pax07e3~\$ **make** love

Make: Don't know how to make love.



lpr filename

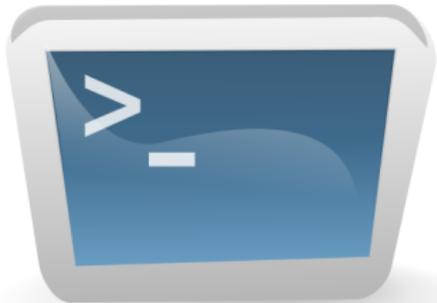
lpq

lprm

lprw

lpd

lpr lpq lprm



```
vi ~/.bashrc
```

```
cd -
```

```
cd -
```

```
cd ~/.bashrc
```