

Computational Physics - Eine Einführung

Kurzreferenz

Helge Todt & Thorsten Tepper García

Institut für Physik und Astronomie



Version vom 22. Oktober 2012

Inhaltsverzeichnis

I. UNIX/Linux	3
1. Der Computerpool	3
2. Shell	4
2.1. Dateibefehle	4
2.2. Login und Datenaustausch	6
2.3. Shell und Prozesse	7
2.4. Informationen	9
2.5. Drucken	10
3. Linux-Anwendungsprogramme	10
4. Editoren	10
4.1. vi	10
4.2. nedit	11
4.3. emacs	11
 II. Index	 16
Index	16

Bedeutung der Schrifttypen:

xvzf	wörtlich einzugebender Text
<i>argument</i>	Platzhalter für selbst zu ersetzenden Text
[...]	optionaler Teil eines Befehls
befehl	UNIX/Linux-Befehl
STRG	Taste, die gedrückt werden soll

Teil I.

UNIX/Linux

1. Der Computerpool

Der Lehrcomputerpool befindet sich im Raum 0.087 im Haus 28 (Institut für Physik und Astronomie), der Studentencomputerpool im Raum 1.100 im selben Haus. Derzeit gibt es folgende Rechner (Hostnames):

Raum 0.087		Raum 1.100
NFS-Server	nachts aus	SSH-Login möglich
born	bohr	bizet
boyle	berry	brahms
grieg	biot	chopin
hertz	bloch	liszt
joule	hahn	plato
volta	haydn	ravel
weber	mozart	vivaldi
	niels	wagner
	nobel	
	tesla	

`.stud.physik.uni-potsdam.de`

Im Raum 0.087 gibt es den PostScript-Drucker **strauss** und im Raum 1.100 den PostScript-Drucker **mach**, welcher ein Seitenlimit von 500 Seiten pro Nutzer und Semester eingestellt hat.

Für Nutzer ohne eigenen Account gibt es auf jedem Rechner einen lokalen (!) User-Account, dessen Logindaten auf dem linken Whiteboard im Raum 0.087 stehen.

2. Shell

Jede Eingabe in der Shell und jede Zeile in einem Shellsript beginnt mit einem Befehl. (Ausnahmen: Kommentarzeichen #, Zuweisungen mit = in der bash)

2.1. Dateibefehle

Bei den Befehle **chmod**, **chown**, **cp**, **ln**, **mv**, **rm** und **scp** haben folgende Optionen dieselbe Bedeutung:

- f force: keine Nachfrage (übertrumpft -i)
- i interaktiv: Nachfragen
- r, -R rekursiv: beziehe Verzeichnisse mit ein

cat *datei*

gibt *datei* auf STDOUT aus.

Beispiel: Konkatenieren von Dateien

```
cat datei2 >> datei1
```

cd [*verzeichnis*]

Verzeichniswechsel. Ohne Argument Wechsel ins Homeverzeichnis.

cd - Wechsel ins vorige Verzeichnis.

chmod [-Rf] *modus datei*

Änderung der Zugriffsrechte auf *datei*. Es gibt zwei verschiedene Modi. Am einfachsten zu merken ist der Modus, der Buchstaben verwendet, Setzen der Rechte mit - Entziehen der Rechte mittels +

u User (Eigentümer der Datei)	r readable: lesbar
g Gruppe	w writeable: beschreibbar
o others: alle anderen	x executable: ausführbar (Programme, Skripte, Verzeichnisse)
a alle: sowohl u als auch g als auch o	

Beispiel:

```
chmod -R go-rwx ~/.ssh
```

entzieht Gruppe und anderen alle Rechte für Verzeichnis `~/.ssh` und dessen Inhalt (rekursiv **-R**). Die Rechte können alternativ auch direkt mittels einer dreistelligen Oktalzahl gesetzt werden. Dabei steht die erste Stelle für den User, die zweite für die Gruppe und dritte für alle anderen. Die einzelnen Stellen sind jeweils die Summe der vergebenen Zugriffsrechte:

0 keinerlei Zugriffsrechte
1 executable: Ausführungsrechte
2 writeable: Schreibrechte
4 readable: Leserechte

Beispiel:

```
chmod 750 Dokumente
```

Setzt alle Rechte für den User, Lese- und Ausführungsrecht für die Gruppe und keinerlei Rechte für alle anderen.

chown [-Rf] [*user*][:*gruppe*] *datei*

Setze User- und Gruppeneigentümer für *datei*.

Beispiel:

```
chown -R tepper:atp Downloads
```

cp [-frpL] *quelle ziel*

Kopieren von Dateien/Verzeichnissen.

Optionen:

- p preserve: Behalte Dateirechte und Zeiten bei.
- L Link dereferenzieren (statt Link kopieren)

Beispiel:

```
cp -r ~/mail/ private/.
```

df [-h]

Zeigt den verfügbaren und belegten Speicherplatz auf den Festplatten an. Ausgabe in GByte usw. mittels **-h**.

du [-hs]

Ermittle Speicherplatzbedarf des aktuellen Verzeichnisses und seiner Unterverzeichnisse/Dateien. Ausgabe in GByte usw. mittels **-h**, mit **-s** (short) nur Speicherplatz des gesamten Verzeichnisses.

file *datei*

Zeige Dateityp von *datei* an.

find *verzeichnis kriterium optionen*

Sucht in *verzeichnis* und dessen Unterverzeichnissen Dateien, die *kriterium* entsprechen und führt dann Aktion *optionen* aus. **Kriterien:**

- name** *name* Dateien mit Namen *name*, reguläre Ausdrücke müssen mit " " geschützt werden
- mtime** *n* Dateien, die in den letzten $n \times 24$ h geändert wurden
- iname** *name* wie **-name**, aber ignoriert Groß-/Kleinschreibung

grep [-ABCilmnrv] *muster* [*datei*[*en*]]

sucht nach *muster* in *datei*[*en*] oder in STDOUT, falls hinter einer Pipe.

- A** *n* zeigt *n* Zeilen nach Fundort an
- B** *n* zeigt *n* Zeilen vor Fundort an
- C** *n* wie **-A** und **-B** zusammen
- i** ignore case: unterscheide nicht zw. Groß- u. Kleinschreibung
- l** zeige nur Namen der Dateien, in denen *muster* gefunden wurde
- m** *n* stoppe Lesen nach *n*tem Fund in Datei
- n** zeige auch Zeilennummer des Fundorts
- r** rekursiv: auch Dateien in Unterverzeichnissen
- v** invert: zeige nur Zeilen, die nicht *muster* enthalten

less *datei*

Zeigt den Inhalt einer Datei seitenweise an (sog. Pager), daher auch recht nützlich wenn rechts von einer Pipe, z.B.

```
ls -l | less
```

Blättern in *less* kann man mit der Leertaste (seitenweise) aber auch mit den Cursortasten (zeilenweise).

ln [-sf] *datei linkname*

Anlegen eines Links. Symbolische Links (**-s**) können über Verzeichnisse hinweg angelegt werden.

ls [-aclrt] [*pfad*]

Listet den Inhalt eines Verzeichnisses auf. Häufigste Optionen:

- a** all: alle, auch versteckte Dateien (mit . am Anfang) anzeigen
- c** zusammen mit **-l**: Zeit der letzten Veränderung
- h** human readable: Größe in besser lesbaren Einheiten (MByte, GByte, usw.)
- l** long format: Zugriffsrechte, Anzahl der Links, Besitzer, Gruppe, Größe, Zeit
- r** reverse: kehre Sortierreihenfolge um
- t** time: nach Zeit sortieren
- S** size: nach Größe sortieren
- u** zusammen mit **-l**: Zeit des letzten Zugriffs

Beispiel:

```
ls -altr zeigt alle Dateien nach Zeit umgekehrt sortiert an
```

2.2. Login und Datenaustausch

mkdir [-p] *verzeichnis*

Anlegen eines neuen Verzeichnisses. Die Option **-p** erzeugt bei Angabe eines Pfades auch die entsprechenden Oberverzeichnisse (parental directories), bzw. gibt keine Fehlermeldung, falls das Zielverzeichnis schon existiert.

more *datei*

Zeigt den Inhalt einer Datei seitenweise an (sog. Pager), ähnlich wie `→ less`, aber nur seitenweises Blättern.

mv *quelle ziel*

Verschieben oder Umbenennen von Dateien oder Verzeichnissen.

pwd

Gibt den absoluten Pfad zum aktuellen Verzeichnis zurück. Vgl. auch Shell-Variable `PWD`.

rm [-ifr] *datei*

Löschen (remove) von Dateien bzw. Verzeichnissen (mit Option **-r** für rekursiv). Es können auch mehrere Dateien/Verzeichnisse angegeben werden.

rmdir

Löschen von leeren Verzeichnissen.

tar *aktion archiv [datei]*

Tape archiver zum Zusammenpacken von Dateien und Verzeichnissen. Aktionen:

- c** erzeuge *archiv* (create) aus Datei/Verzeichnis *datei*
- f** Archiv ist ein file (default: Bandlaufwerk)
- t** zeige Inhalt von Archiv
- v** zeige ausgeführte Aktionen (verbose)
- x** extrahiere Archiv
- z** Archiv ist gegzippt (**.tar.gz** bzw. **.tgz**)

Beispiel:

tar xvzf paket.tar.gz

Entpackt das Archiv `paket.tar.gz`.

touch *datei*

Anlegen einer leeren bzw. ändern des Zeitstempels einer bestehenden Datei.

2.2. Login und Datenaustausch

hostname [-d]

Ausgabe des Rechnernamens bzw. der Domäne (Option **-d**).

rsync [-actvz] [*quelle*] [*ziel*]

Remote-Copy über verschlüsselte Verbindung (ssh, scp), mit der Möglichkeit nur geänderte Dateien zu übertragen.

- | | |
|--|--|
| -a Archiv-Modus= -rlptgoD | -o owner: Beibehaltung des Besitzers (nur als root) |
| -c checksum: geänderte Dateien an Checksumme erkennen (statt Datum und Größe) | -p permissions: Beibehaltung von Zugriffsrechten |
| -D Beibehaltung von speziellen Dateien und Geräten | -r rekursiv: auch Unterverzeichnisse |
| -g group: Beibehaltung der Gruppe | -t time: behalte Modifikationszeiten bei |
| -l links: Kopiere symbolische Links (keine Dereferenzierung) | -v verbose: Ausgabe weiterer Informationen |
| | -z komprimiere Dateien für Übertragung |

Beispiel:

```
rsync -tvzc * joule:lehre/.
```

Übertrage nur die geänderten (regulären) Dateien des aktuellen Verzeichnisses zum Host **joule** nach **~/lehre/**

```
scp [-Cpr] [quelle] [user@][host]:[ziel]
```

```
scp [-Cpr] [user@][host]:[quelle] [ziel]
```

Remote-Copy über verschlüsselte Verbindung.

-C compress: komprimierte Datenübertragung

-p preserve: behalte Dateiattribute (Zeit) bei

-r recursive: kopiere Verzeichnisse

```
ssh [user@] host [-Xvvv] [befehl]
```

Verschlüsselte Verbindung (secure shell - SSH) zu entferntem Rechner *host*. Falls möglich, wird durch **-X** eine Weiterleitung von grafischen Fenstern (X11) zum lokalen Rechner erlaubt. Debug-Ausgabe durch **-v**, **-vv** oder **-vvv** (volle Debug-Ausgabe). Statt eines interaktiven Logins kann auch *befehl* ausgeführt werden.

ssh-keygen

generiert ein SSH-Schlüsselpaar

```
su [-] [user]
```

Substituiere User. Anmeldung als anderer *user*, mit **-** wird dessen Login-Skript ausgeführt, oder (ohne Argument) als **root**.

```
sudo befehl
```

Führe *befehl* als **root** aus, **root**-Passwort erforderlich.

```
w [-f]
```

Gibt Liste der zur Zeit auf dem Rechner eingeloggten User aus. Mit **-f** (from) auch angezeigt, von wo aus sich diese eingeloggt haben.

whoami

Gibt den Namen (ID) des aktuellen Users aus.

2.3. Shell und Prozesse

```
befehl1 | befehl2
```

Pipesymbol, verbindet STDOUT des Befehls links vom Pipesymbol (*befehl1*) mit STDIN des Befehls rechts vom Pipesymbol (*befehl2*).

Beispiel:

```
ls -l | grep "^d"
```

Zeigt alle Verzeichnisse im aktuellen Verzeichnis an.

```
befehl > datei
```

Umleitungsoperator, lenkt STDOUT von *befehl* in Datei *datei* um. Existiert die Datei bereits, so wird sie überschrieben. Soll statt dessen angehängt werden, muss **>>** verwendet werden, z.B.

```
date >> anmeldung.log
```

Soll STDERR umgeleitet werden (bash): **befehl 2> datei**

Soll STDOUT und STDERR umgeleitet werden (bash): **befehl 2> datei 2>&1**

```
&
```

Ausführen eines Jobs im Hintergrund, wird an den entsprechenden Befehl angehängt.

2.3. Shell und Prozesse

STRG + **c**

Sendet an den im Terminal laufenden Prozess das Signal **TERM**, um diesen zu beenden.

STRG + **d**

Shell beenden (**exit**).

STRG + **q**

Beendet das Pausieren der Shell, falls **STRG+s** vorher gedrückt wurde.

STRG + **s**

Shell suspenden (pausieren).

alias [name[=wert]] (bash)

alias [name[wert]] (tcsh)

Ausgabe aller Aliase, des Aliases mit Namen *name* oder Setzen des Aliases mit Namen *name* auf *wert*.

bg [job]

Führt *job* oder (ohne Argument) den zuletzt gestarteten Job im Hintergrund der Shell fort, insbesondere nach <**STRG**>+z (Prozess pausieren), Jobnummer wird in eckigen Klammern in der ersten Spalte angezeigt.

echo [-n] [text]

Gibt *text* auf **STDOUT** aus. Mit **-n** unterbleibt der abschließende Zeilenumbruch.

exit

Beendet die laufende Shell-Sitzung.

export [VARIABLE=wert] (bash)

Setzt die Umgebungsvariable (Vererbung auf alle Tochtershells) *Variable* und initialisiert sie mit dem Wert *wert*. Ohne Argumente werden alle gesetzten Umgebungsvariablen angezeigt.

fg [job]

Bringt *job* wieder in den Vordergrund, bzw. (ohne Argument) den zuletzt gestarteten Job. Siehe **bg** .

history [-c]

Zeigt die zuletzt eingegebenen Befehle an. Löschen der History mit **-c**.

kill [-signal] PID

Sendet *signal* an Prozess mit der Prozess-ID *PID* (kann mit **top** oder **ps** angezeigt werden). Die beiden wichtigsten Signale sind **TERM** bzw. **15**, welches den Prozess regulär beendet und **KILL** bzw. **9**, welches den Prozess sofort beendet (kein Abspeichern o.ä.).

nice [-n inc] programm

Starte *programm* mit herabgesetzter Priorität. Falls kein Inkrement *inc* gegeben, wird Priorität um +10 geändert. Superuser darf auch negatives Inkrement angeben. Prozessprioritäten im Bereich -20 (höchste Priorität) bis +20 (niedrigste Priorität). Rechenjobs sollten stets mit niedrigster Priorität gestartet werden (vor allem auf älteren UNIX-Systemen), um den Rechner nicht zu blockieren.

ps [axu]

Anzeige laufender Prozesse. Ohne Argumente werden nur die Prozesse des aktuellen Terminals angezeigt. Die Optionen **axu** lässt **ps** alle Prozesse auf dem Host zusammen mit den Namen der User anzeigen.

set [variable=wert] (tcsh)

variable=wert (bash)

Setzen der lokalen Variablen *variable* und Zuweisung von *wert*. Ohne Argumente werden alle gesetzten lokalen Variablen angezeigt.

setenv [*VARIABLE wert*] (tcsh)

Setzt die Umgebungsvariable (Vererbung auf alle Tochtershells) *Variable* und initialisiert sie mit dem Wert *wert*. Ohne Argumente werden alle gesetzten Umgebungsvariablen angezeigt.

unset *variable*

unsetenv *VARIABLE* (tcsh)

Löschen der Variablen *variable*. Man beachte, dass auf der **tcsh** der Befehl **unset** nur lokale, aber keine Umgebungsvariablen löscht.

top

Interaktive Anzeige laufender Prozesse, folgende Eingaben sind nach dem Start möglich:

- l** Zeige Load für jeden Prozessorkern einzeln an
- i** idle: Zeige nur laufende, keine, schlafenden Prozesse
- k** *PID* kill - Beende Prozess *PID*
- q** quit - Beenden
- u** *user* Zeige nur Prozesse von *user*

Eine verbesserte Version von **top** ist das Programm **htop**.

which [*befehl*]

Zeigt den Pfad zum Programm *befehl* an, bzw. ob es sich um einen shellinternen Befehl oder um einen Alias (einige Shells, z.B. tcsh) handelt.

xterm [*-ls*]

Öffnet ein neues Xterm. Mit **-ls** wird die Shell darin zu einer Login-Shell.

2.4. Informationen

apropos *keyword*

Listet Befehle auf, in deren Beschreibung *keyword* auftaucht

id [*user*]

Ausgabe der User-ID und Gruppenzugehörigkeit.

info *befehl*

Gibt Informationen zu *befehl* aus, wie **man**

man *befehl*

Gibt Informationen zu *befehl* aus, wie **info**

/proc, z.B. **cat /proc/cpuinfo**

Virtuelles Verzeichnis, in welchem Informationen über Prozesse und Hardware abgelegt werden.

uname [*-a*]

Informationen über das Betriebssystem (Kernelversion). Mit **-a** werden alle Informationen ausgegeben. Auf SuSE-Linux-Systemen findet man die Distributionsversion in der Datei `/etc/SuSE-release`.

2.5. Drucken

a2ps [-o *output*] *datei*

Wandelt ASCII-Datei *datei* nach PostScript um, sendet die Ausgabe an den Standarddrucker, es sei denn mittels **-o** ist eine Ausgabedatei spezifiziert.

lpr [-P*drucker*] *datei*

Druckt *datei* auf *drucker*

lpq [-p*drucker*]

Gibt Informationen zu Druckjobs für *drucker* aus.

lpstat [-t1] [-p*drucker*]

Gibt Informationen zum Status von *drucker* aus, mit **-t1** erfolgt eine detaillierte Ausgabe.

3. Linux-Anwendungsprogramme

acroread [*datei.pdf*]

Startet den Acrobat Reader zum Anzeigen und Ausdrucken von pdf-Dateien.

cal [*monat*] [*jahr*]

Ausgabe einer Kalenderansicht für den aktuellen Monat, einen angegebenen Monat oder ein ganzes Jahr.

firefox [*datei.htm[l]*]

Der Standard-Webbrowser.

gv [*datei.ps*]

Anzeigeprogramm für PostScript-Dateien (**.ps**).

konqueror [*datei.htm[l]*]

Datei- und Webbrowser des KDE.

nautilus

Dateibrowser des GNOME-Desktops.

okular [*datei.pdf* | *datei.ps*]

Anzeigeprogramm für pdf- und ps-Dateien.

xeyes

ein aufmerksames X11-Augenpaar.

4. Editoren

4.1. vi

vi [*datei*]

Der **vi** unterscheidet zwischen Einfüge- und Befehlsmodus. Er läuft in jedem Terminal und darf daher nicht im Hintergrund (mit **&**) gestartet werden. Meist läuft statt des **vi** der **vim**, bei dem man auch im Einfügemodus die Cursortasten benutzen kann.

i Wechseln in den Einfügemodus

a Wechseln in den Einfügemodus (Anhängen)

- ESC** Wechseln in den Befehlsmodus
- :w [datei]** Schreiben (in Datei) - im Befehlsmodus
- :q** Verlassen - im Befehlsmodus
- :q!** Verlassen (ohne Nachfrage) - im Befehlsmodus
- x** Ein Zeichen Löschen - im Befehlsmodus
- dd** Eine ganze Zeile löschen

4.2. nedit

nedit [datei]

Windowsartiger Editor, der spaltenweise mit der Maus markieren kann und auf vielen Systemen verfügbar ist.

- STRG** + **Linksklick** Spaltenweises Markieren mit der Maus.
- STRG** + **c** Markierten Bereich kopieren
- STRG** + **v** Kopierten Bereich einfügen
- STRG** + **s** Datei speichern
- STRG** + **q** Beenden

4.3. emacs

emacs [-nw] [datei]

emacs ist auf fast jedem Unix-System verfügbar und kann unter MacOS nachinstalliert werden (DarwinPorts), wobei es auch Implementierungen für die jeweiligen Grafischen Oberflächen gibt (**xemacs** für X11, **aquamacs** für MacOS).

Die Option **-nw** (no window) lässt den **emacs** im Terminal laufen (ähnlich dem **vi**). Die u.g. Tasten sind ihrer Beschriftung (deutsches Layout) entsprechend durch **STRG** usw. bezeichnet. Sollen zwei Tasten gleichzeitig gedrückt werden, so ist dies durch ein + angezeigt, ansonsten sind die Tasten nacheinander zu betätigen.

- STRG** + **g** letzten Befehl abbrechen
- STRG** + **x** **STRG** + **s** Speichern des aktuellen Buffers (Fensterinhalt)
- STRG** + **x** **STRG** + **c** Schließen (close) des Editorfensters
- STRG** + **k** Ausschneiden ab Cursor bis Zeilenende (kill)
- STRG** + **y** Einfügen des kopierten/ausgeschnittenen Bereichs (yank)
- STRG** + **x** **r** **k** Spaltenweises Ausschneiden des markierten Bereichs (rectangular kill)

4.3. emacs

STRG + **x** **r** **y** Spaltenweises Einfügen des markierten Bereichs (rectangular yank)

STRG + **SPACE** Bereich Markieren Anfang. Cursor mit Pfeiltasten (Cursortasten) bis zum Ende des Bereichs bewegen.

STRG + **w** Ausschneiden des markierten Bereichs

ESC **w** Kopieren des markierten Bereichs

STRG + **s** Suchen eines Begriffs im Dokument

ESC **g** Gehe zur Zeile (Zeilennummer)

STRG + **x** **i** Eine Datei einfügen

Teil II.

Index

Index

/proc, 9
>, 7
&, 7

a2ps, 10
acoread, 10
alias, 8
apropos, 9

bg, 8

cal, 10
cat, 4
cd, 4
chmod, 4
chown, 4
cp, 4

df, 5
Drucker, 3
du, 5

echo, 8
emacs, 11
exit, 8
export, 8

fg, 8
file, 5
find, 5
firefox, 10

grep, 5

gv, 10

history, 8
hostname, 6
htop, 9

id, 9
info, 9

kill, 8
konqueror, 10

less, 5
Link, 5
ln, 5
lpq, 10
lpr, 10
lpstat, 10
ls, 5

man, 9
mkdir, 6
more, 6
mv, 6

nautilus, 10
nedit, 11
nice, 8

okular, 10

Pipe, 5, 7
proc, 9
ps, 8

PWD, 6
pwd, 6

rm, 6
rmdir, 6
rsync, 6

scp, 7
set, 8
setenv, 9
ssh, 7
ssh-keygen, 7
su, 7
sudo, 7
SuSE-release, 9

tar, 6
top, 9
touch, 6

Umleitungsoperator, 7
uname, 9
unset, 9
unsetenv, 9

vi, 10

w, 7
which, 9
whoami, 7

xeyes, 10
xterm, 9