Linux-Einführung WS 2010/2011

1 Hilfequellen

- studihelp@coli.uni-sb.de
- Systemgruppe: http://www.coli.uni-sb.de/sg/
- Fachschaft: http://www.coli.uni-sb.de/fs-coli/ –> Skripte
- Linuxkurs-Webseite: http://www.ruthless.zathras.de/facts/linux/
- Kommiliton_innen fragen
- Im Internet suchen!
- Bibliothek

Arbeitsumgebung (Desktop Environment)

Vor dem Einloggen kann man sich für eine der installierten Arbeitsumgebungen entscheiden. (links unten `Options -> Select Session`). Für den Anfänger empfiehlt es sich, GNOME oder KDE auszuwählen. Wenn man das als Standard/Default wählt, wird es in Zukunft automatisch beim Starten geladen. Die Arbeitsumgebung stellt die graphische Benutzerschnittstelle zum Rechner her. Sie ermöglicht es, auf Daten und Einstellungen zuzugreifen und diese zu ändern und Programme zu starten. Sie stellt u.a. Fenster, Icons, Menus, Werkzeugleisten zur Verfügung. Allerdings gibt es Dinge, die man mit ihr nicht oder nur umständlich machen kann.

2 Kommandozeile und Kommandos

Die Kommandozeile, (auch als *Konsole* oder *Terminal* oder *Shell* bezeichnet), ist der Eingabebereich für die Steuerung einer Software oder eines Betriebssystems, typischerweise im Textmodus. Man gibt Befehle ein und führt damit Programme aus. Man kann sich auch Informationen über den Rechner, auf dem man sich befindet, zeigen lassen.

`Tab`	Autovervollständigen des Befehls/Dateinamen
Pfeiltasten ↑, ↓	durch bereits ausgeführte Befehle blättern
`Strg + C`	Aktuellen Befehl/Eingabe abbrechen
`history`	Liste aller bisherigen Befehle ausgeben
`Strg + R`	Bisherige Befehle durchsuchen
`Strg + A`	Cursor am Zeilenanfang positionieren
`Strg + E`	Cursor am Zeilenende positionieren
`Strg + K`	Vom Cursor bis zum Zeilenende alle Zeichen löschen
`clear`	Bildschirm löschen

Der Aufbau eines Kommandos

Kommando	(z.B. `ls`)
Kommando [Lücke] Argument	(z.B. `ls VERZEICHNIS`)
Kommando [Lücke] Minuszeichen Option [Lücke] Argument	(z.B. `ls -s DATEI`)

Wie finde ich mehr über einen Befehl heraus?

- Manual Pages: *man Kommando* (z.B. `man psnup`)
  - weiterblättern: Pfeil-Tasten und Leertaste
  - Handbuch verlassen: Buchstabe `q` tippen
  - auf Deutsch umstellen: vorher `LANG=de_DE` eintippen
- `--help` hinter den Befehl schreiben (`psnup --help`)
- Im Internet suchen

Programme starten und beenden

- Wenn ein Programm im Terminal aufgerufen wird, blockiert es das Terminal, bis es beendet wird. Bei graphischen Programmen kann man dieses Blockieren verhindern, indem man `&` dahinterschreibt
- Falls man das `&` beim Starten vergessen hat, kann man nachträglich doch noch Zugriff auf das Terminal erhalten, indem man `Strg+Z` und dann `bg` eingibt.
- Im Notfall können Programme meist mit `Strg-C` oder `Strg-D` abgebrochen werden
- Mit `Strg-S` kann man die Bildschirmausgabe anhalten, um von dort fortzusetzen, muss man `Strg-Q` eingeben
- Ein Programm beenden kann man sonst auch mit dem Befehl `top`, der alle laufenden Programme anzeigt. Bedienung:
  - ? Hilfe
  - q Beenden
  - u nur Programme eines bestimmten Users anzeigen
  - k Programme abschießen (signal: 9)

3 Dateien und Verzeichnisse

`pwd`	Anzeigen, wo man in der Datei-Hierarchie ist
`ls`	Inhalt vom Verzeichnis auflisten
`ls -s DATEI`	Grösse anzeigen
`ls -l`	Detail-Information anzeigen (z.B. Größe, Lese/Schreibrechte)
`ls -a`	Auch Dateien mit `.` am Anfang anzeigen (Konfigurationsdateien)
`rm DATEI`	Datei löschen
`rm -f DATEI`	Datei ohne Nachfragen löschen
`mkdir VERZ`	Verzeichnis hinzufügen
`rmdir VERZ`	Verzeichnis löschen
`rmdir -r VERZ`	Verzeichnis rekursiv löschen
`cd VERZ`	in ein Verzeichnis wechseln
`cd ..`	eine Ebene höher wechseln
`cd`	zu seinem eigenen Home-Verzeichnis zurückwechseln
`cp DATEI1 DATEI2`	Datei kopieren
`mv DATEI VERZ`	Datei verschieben
`mv DATEI1 DATEI2`	Datei umbenennen
`mv *.txt VERZ`	Alle Dateien mit der Endung `.txt` verschieben

`pushd`	speichert das derzeitige Verzeichnis, damit man später leichter zurück kann
`pushd VERZ`	wechselt zusätzlich in das Verzeichnis VERZ
`popd`	wechselt in das mit pushd gespeicherte Verzeichnis zurück
`du`	anzeigen, wieviel Platz das Verzeichnis belegt
`du DATEI`	anzeigen, wieviel Platz die Datei belegt
`du -h`	belegten Platz anzeigen in menschenlesbarem Format
`du -s`	belegten Platz anzeigen als Summe
`more DATEI`	(Text-) Datei anzeigen und vorwärts blättern (beenden mit `q`)
`less DATEI`	(Text-) Datei seitenweise anzeigen und hin und her blättern (beenden mit `q`)

Blättern mit `<SPACE>` (seitenweise) und `<RETURN>` (zeilenweise), Wort suchen mit "/suchwort`<RETURN>`"

Anmerkung: In Dateinamen sollte man keine Umlaute, Sonderzeichen, Leerzeichen verwenden. Suffixe muss man selber hinzufügen. Groß- und Kleinschreibung wird unterschieden.

Zusammenfassung der Abkürzungen:

`~`	bedeutet `HOME`
`.`	bedeutet *hier*
`..`	bedeutet *eine Ebene drüber*
`*`	bedeutet *alle Dateien*
`*.txt`	bedeutet *alle Dateien, die in `.txt` enden*

4 Zugriffsrechte

- Es gibt 3 Arten von Benutzern:

user	`u`	der Besitzer der Datei (du selbst)
gruppe	`g`	meist `stud` oder `users` (Mitarbeiter)
others	`o`	alle, die einen Account haben

  Außerdem: `a` steht für *alle*
- Rechte anzeigen: mit `ls -l`
  d$\underbrace{\texttt{rwx}}_{user}$ $\underbrace{\texttt{rwx}}_{gruppe}$ $\underbrace{\texttt{rwx}}_{others}$ (normalerweise: gruppe=`stud`)
- Erstes Zeichen: Bei Dateien `-`, bei Verzeichnissen `d`
- Drei mögliche Rechte: Lesen (`r`), schreiben (`w`), ausführen (`x`)

Zugriffsrechte ändern

Allgemein:
`chmod {augo}{+=-}{rwx} DATEI` oder `VERZ`

Konkrete Beispiele:

`chmod go-w DATEI`	kein anderer darf hier was ändern...
`chmod u+rw DATEI`	...aber ich schon!
`chmod a+rx VERZ`	alle dürfen in dieses Verzeichnis wechseln
`chmod a+r VERZ/*`	...und seinen Inhalt lesen
`chmod u-w wichtig.txt`	nicht einmal ich selbst darf das hier ändern

Anmerkung: Wenn ein Verzeichnis nicht lesbar ist, kann man Dateien weder löschen noch erstellen!

5 Nützliche Programme

Datei-Typ	Kommando	Programm im Menu
pdf	`acroread DATEI, xpdf DATEI`	Graphics –> KPDF
ppt,doc,xls	`ooffice FILE`	Office –> OpenOffice
txt,keine,etc	`emacs DATEI, kate DATEI`	Accessoires –> Emacs, Kate, Texteditor
txt,keine,etc	`pico DATEI`	?
ps	`gv DATEI`	?

Dateien suchen

```
whereis DATEI
find DATEI
locate DATEI
```

In Dateien suchen

Eine sehr flexible Möglichkeit, eine Zeichenkette zu finden, ist Get Regular Expression Pattern, `grep`. Es gibt einem alle Zeilen aus, in denen das gesuchte Wort/der gesuchte Ausdruck vorkommt. Dieser Befehl wird oft und gerne an allen möglichen Befehlen als Filter verwendet. Erst der Suchbegriff, dann der Suchort. Groß- und Kleinschreibung beachten!

```
grep verschollenesWort GrosseDatei.txt
```

```
grep verschollenesWort *.txt
```
Sucht verschollenes Wort in allen Dateien, die mit .txt enden.

```
grep -r verschollenesWort *.txt
```
Sucht verschollenes Wort in allen Dateien im aktuellen und allen Unterverzeichnissen, die mit .txt enden.

Text bearbeiten

Linux stellt eine Reihe von Text-Editoren zur Verfügung. Ein sehr gebräuchlicher ist Emacs, der sich mit `emacs` oder `emacs DATEI` aus der Kommandozeile öffnen lässt.

wichtige Tastenkombinationen für Emacs

`Strg+x Strg+f`	Datei öffnen
`Strg+x Strg+s`	speichern
`Strg+s`	suchen
`Strg+_`	rückgängig machen
`Strg+x 2`	Fenster teilen
`Strg+x 1`	nur ein Fenster

Datei-Archive entpacken

Wenn Endung	dann auspacken mit
`.gz`	`gunzip DATEI.gz`
`.zip`	`unzip DATEI.zip`
`.tar`	`tar xvf DATEI.tar`

Sich auf andere Server einloggen

Neben dem Rechner im CompLap, auf dem man sich lokal anmeldet, gibt es noch viele andere Rechner bzw. Server (auch Host genannt), auf die man sich über die Konsole einloggen kann. Das ist zum Beispiel nötig, wenn man Zugriff auf Dateien haben will, die nur auf einem bestimmten Server liegen oder wenn man Programme nutzen will, die nur dort installiert sind. Auch wenn man sich von außerhalb der Uni einloggen will, um auf seine Dateien zuzugreifen, ist das praktisch. Das Programm, mit dem man das tun kann, heißt `ssh` (secure shell).

`ssh Benutzer@Servername`	allgemeine Form
`ssh Servername`	wenn Benutzer identisch ist

`ssh Benutzer@login.coli.uni-saarland.de`
- einlogggen auf den zentralen Server namens *login*
`ssh Benutzer@httpd.coli.uni-saarland.de`
- sich auf den Webserver *httpd* einloggen, auf dem man Dateien für eigene Homepage speichern kann

Vorsicht: Man wird nach dem Passwort gefragt, dessen Eingabe man mit Enter abschließen muss. Aus Sicherheitsgründen wird die Eingabe nicht am Bildschirm angezeigt!

Mit `exit` oder `logout` kann man den Server wieder verlassen.

Dateien zwischen Servern verschieben

Manchmal muss man Dateien oder Verzeichnisse zwischen verschiedenen Servern bewegen, bzw. hoch- und runterladen, (z.B. Dateien der eigenen Homepage auf den Webserver).

Datei zum Host kopieren:
```
scp Quelldatei.bsp Benutzer@Host:Verzeichnis/Zieldatei.bsp
```
Datei vom Host kopieren:
```
scp Benutzer@Host:Verzeichnis/Quelldatei.bsp Zieldatei.bsp
```

6 Verkettung von Linuxkommandos

Pipes

Mit dem Pipe genannten Symbol "lassen sich Linux-Befehle verketten. Dadurch vervielfältigen sich die möglichen Funktionen! Häufig werden z.B. `grep` und die Pipe zusammen als Filter eingesetzt.

Möchte man z.B. herausfinden, ob man selbst Verzeichnisse herumfliegen hat, bei denen die eigene "group"write-Permission hat, leitet man den 'alles detailiert auflisten'-Befehl ("ls -la") an das Filter-Kommando 'grep' weiter, der das Ergebnis auf das gewünschte Permission-Muster "d...rw"hin filtert, und dann erst das Ergebnis ausgibt:

```
ls -al | grep "d...rw"
```

Wenn man sich nicht mehr genau an einen kürzlich eingegebenen Befehl und die zugehörigen Optionen erinnern kann, kann man danach suchen:

```
history | grep BEFEHL
```

Ausgabeumleitungen

Statt dass ein Befehl sein Ergebnis standardmäßig ausgibt (ins Shell-Fenster), kann er ihn auch mit Hilfe der Ausgabeumleitung `>` in eine Datei schreiben, wenn man das Ergebnis speichern will. Existiert der angegebene Dateiname noch nicht, wird eine Datei mit diesem Namen angelegt. Existiert der angegebene Dateiname dagegen schon, wird die schon bestehende Datei mit dem neuen Inhalt überschrieben! Auch ist darauf zu achten, dass die Quell- und die Zieldatei verschiedene Namen haben müssen!

Im Beispiel wird die ausführliche Liste aller Dateien in diesem Verzeichnis als Datei gespeichert:

```
ls -al > meineDateien.txt
```

Um die Ausgabedaten an eine schon existierende Datei anzuhängen, ohne diese zu überschreiben, nimmt man $>>$ statt `>`:

```
ls -al >> Liste.txt
```