Inhaltsverzeichnis

1. Linux anwenden

1.1. Ziele

☑ Linux anwenden können
- ☐ User
- ☐ Files, Verzeichnisse
- ☐ Prozesse
- ☐ Administration

1.2. User und Userverwaltung

Zunächst in aller Kürze.

☑ Der Systemadministration heisst **root.**
☑ Jeder User hat ein **Home**verzeichnis
☑ Linux kennt User und Gruppen.
☑ User können verschiedenen Gruppen angehören.
☑ User haben eine eindeutige UID.

☑ Usern ist nach dem Einloggen eine shell (meist die **bash**) zugeordnet.
☑ Die entsprechenden Systemkonfigurationsdateien lauten: **/etc/passwd**, **/etc/shadow**, **/etc/group**

1.2.1. Anlegen und Löschen von Benutzern

Das Kommando useradd dient zum Anlegen neuer Benutzer. Dieser Befehl ist root vorbehalten.

Ein Beispiel:

```
useradd -u 1005 -g users -d /home/emil -s /bin/bash -m emil
```

legt den Benutzer emil mit Home-Verzeichnis /home/emil, Benutzer-ID 1005 an. emil gehört zur Gruppe users und nach dem Einloggen steht ihm die bash-shell zur Verfügung.

Mit

```
passwd emil
```

vergibt root ein Passwort für den Anwender emil.

Bei der Neuanlage eines Benutzers werden alle Dateien aus **/etc/skel** in das Home-Verzeichnis des neuen Users kopiert, so dass automatisch eine gewisse minimale systemweite Vorkonfigurierung aller Benutzer erfolgen kann. Selbstverständlich kann jeder Benutzer des Systems diese Dateien nach seinen eigenen Vorstellungen anpassen. Für das komfortable Anlegen und Löschen von Benutzereinträgen ist jedoch die Benutzerverwaltung (OpenSuse:YaST, Ubuntu:Systemmenü, ...) zu empfehlen.

1.2.2. Aufgaben Userverwaltung

☑ Legen Sie einen neuen User namens schueler an.
☑ Sein Passwort soll comein lauten.
☑ Melden Sie sich als User schueler an und ändern Sie ihr Passwort.

☑ Frage: In welcher Datei finden Sie die Angaben für alle eingetragenen User?
☑ Antwort: _____
Studieren Sie den Aufbau dieser Datei.

☑ Frage: Gibt es auf ihrem System den user nobody? Wofür wird dieser eingesetzt?
☑ Antwort: _____

1.3. Verzeichnisse und Dateien

1.3.1. Kommandosyntax für Dateien und Verzeichnisse

Das Verzeichnis-Trennzeichen ist unter Unix der slash "/". Ein Pfad ist demnach eine Zeichenkette, in der die Verzeichnisnamen durch slashes getrennt sind. Ein einzelner slash bezeichnet dabei das **Wurzelverzeichnis**.

Unix unterscheidet **Gross- und Kleinschreibung.**

Eine angenehme Erleichterung bei der Eingabe von Datei- bzw. Verzeichnisnamen ist die Funktion der **TAB Taste**. Geben Sie die ersten Buchstaben der gewünschten Datei ein und drücken Sie TAB . Die Shell ergänzt nun den kompletten Dateinamen (wenn er durch den/die ersten Buchstaben eindeutig bestimmt ist). Zweimaliges Drücken der Tabulatortaste zeigt bei Mehrdeutigkeiten alle Möglichkeiten.

Der Punkt "." entspricht dem **aktuellen** Verzeichnis und ".." entspricht dem **übergeordneten** Verzeichnis.

1.3.2. Arbeiten mit Verzeichnissen

Nach dem Einloggen ist das user-eigene home-Verzeichnis als aktuelles eingestellt. Der Name dieses Verzeichnisses kann mit dem Kommando **pwd** (print working directory) ausgegeben werden:

```
pwd
/home/schueler
```

Um in ein anderes Verzeichnis zu wechseln, dient der Befehl **cd** (change directory).

```
cd /usr/sbin

cd
cd texte
cd ../bilder
```

Wird cd ohne ein Argument aufgerufen, so wird in das jeweilige home-Verzeichnis gewechselt. Das home-Verzeichnis kann auch mit Hilfe der Tilde (~) bezeichnet werden. Die Eingabe von **cd ~/texte** wechselt in das Verzeichnis texte unter dem jeweiligen home-Verzeichnis.

Neue Verzeichnisse werden mit dem Kommando **mkdir** (make directory) angelegt.

```
mkdir texte
mkdir /home/schueler/bilder
```

Leere Verzeichnisse können mit dem Befehl **rmdir** (remove directory) gelöscht werden.

```
rmdir texte
```

Neben dem Wurzelverzeichnis gibt es noch weitere wichtige und unverzichtbare Verzeichnisse in einem UNIX-Filesystem.

Jedes der dargestellten Verzeichnisse hat eine feste Funktion im UNIX-System und darf auch nicht anders benannt oder verwendet werden.

Verzeichnis	Funktion
/bin	Dienst- und Systemprogramme des LINUX-Systems, ausführbare „Binärdateien“
/dev	Gerätedateien („special files“) für alle Geräte - z.B. fd0, fd1, hda1, hda2, ...
/etc	Konfigurationsdateien, z.B. fstab, hosts, passwd, profile, ...
/usr	Anwendersoftware, Programmpakete zur Nutzung unter UNIX
/proc	Prozessinformation (virtuelles Verzeichnis, das HS-Bereiche abbildet)
/sbin	Programmdateien für die Systemadministration
/var	Log-Dateien.
/home	Anwenderverzeichnisse
/home/root	Verzeichnis des Administrators
/usr/bin	Programme, ausführbare Binärdateien der Anwendersoftware
/usr/lib	Bibliotheken, die von der Anwendersoftware benötigt werden
/usr/src	Quellprogramme der Anwendersoftware
/usr/man	Manpages für Anwenderprogramme (Hilfetexte)

1.4. Arbeiten mit Dateien

1.4.1. Informationen über Dateien

Der Befehl **ls** (list) zeigt den Inhalt des aktuellen Verzeichnisses an. Ausgegeben wird eine Liste aller Verzeichnisse und Verzeichnisnamen. Es kann auch durch Angabe eines Arguments der Inhalt eines anderen Verzeichnisses angezeigt werden:

```
ls
ls /usr/bin
```

Eine nützliche Option von ls ist **-l**. Dadurch werden zusätzliche Informationen wie Zugriffsrechte, Eigentümer, Gruppenzugehörigkeit und Grösse ausgegeben:

```
ls -l
drwxr-xr-x 6 schueler     users      1024     Mar 21 12:39 ./
drwxr-xr-x 4 schueler     users      1024     Mar 21 17:13 ../
drwxr-xr-x 2 schueler     users      1024     Nov  6 16:19 bin/
-rwxr-xr-x 1 schueler     lxbuch     4160     Mar 21 12:38 check*
drwxr-xr-x 2 schueler     users      1024     Nov  6 16:23 etc/
drwxr-xr-x 2 schueler     users      1024     Nov  6 16:19 sbin/
drwxr-xr-x 9 schueler     users      1024     Nov  6 18:20 usr/
```

```
-rw-r--r-- 1 schueler     users    185050     Mar 15 12:33 xvi.tgz
-rw-r--r-- 1 schueler     users     98444     Mar 14 12:30 xvnews.tgz
```

Die Bedeutung der einzelnen Felder:

Rechte:	Das erste Zeichen dieses Feldes bezeichnet den Dateityp. Hierbei steht d für Verzeichnis, l für Link und - für eine normale Datei. Die folgenden 9 Zeichen geben die Zugriffrechte für den Besitzer, die Gruppe und alle anderen Benutzer an (jeweils drei Zeichen). Hierbei steht r für lesen, w für schreiben, und x für ausführen. **-rw-r--r--** bezeichnet demnach eine Datei, die vom Eigentümer, den Mitgliedern der Gruppe, und allen anderen gelesen werden kann, aber nur vom Eigentümer verändert werden kann.
Besitzer:	Der Eigentümer der Datei.
Gruppe:	Die Gruppenzugehörigkeit der Datei
Größe:	Die Größe der Datei in Byte
Letzte Änderung:	Das Datum der letzten Änderung der Datei. Bei Dateien, deren letzte Änderung über ein Jahr zurückliegt, wird anstatt der Uhrzeit das Jahr angegeben.
Name:	Der Name der Datei.

☑Frage:
ls -a
zeigt auch ??????????
☑Antwort:

_____

1.5. Inhalt von Dateien

cat /etc/passwd

zeigt den Inhalt der Datei /etc/passwd an

```
cat eins zwei  > einsundzwei
```
zeigt den Inhalt der Dateien eins und zwei nicht auf dem Bildschirm an, sondern schreibt das Ergebnis in die Datei einsundzwei.

```
less sehrGrosseDatei.txt
```
Der Inhalt einer Datei kann mit dem Befehl less seitenweise angezeigt werden. Mit der Leertaste kann eine Seite nach vorne, mit b eine Seite zurück geblättert werden.

1.5.1. Kopieren, Umbenennen und Löschen von Dateien

Der Befehl, um Dateien zu kopieren, lautet

```
cp quelldatei zieldatei
```

```
cp  /etc/passwd  ~/
```
Um die Datei passwd aus dem Verzeichnis /etc in das eigene home-Verzeichnis zu kopieren.

```
cd
cp  /etc/passwd  .
```
macht das gleiche wie oben. Beachte den Punkt.

```
cp prog/*.c backup/schule
```
Kopiert ?????????

```
cp -r prog/*.c backup/schule
```
Kopiert ?????????

Dateien können mit dem Befehl **rm** (remove) gelöscht werden. Eine nützliche Option ist **-r** (recursive), wodurch auch alle Unterverzeichnisse und die darin enthaltenen Dateien gelöscht werden (vergleichbar mit dem unter DOS verfügbaren Kommando deltree). Die Eingabe von

```
rm -r bin
```
löscht z.B. das Verzeichnis bin und alle sich darin befindlichen Dateien und Verzeichnisse. Diese Option ist mit äusserster **Vorsicht** anzuwenden, da keine Möglichkeit besteht, versehentlich gelöschte Dateien wiederherzustellen!

```
rm -ri bin
```
?????????

```
rm -rf /
```
?????????

Der Befehl **mv** (move) verschiebt Dateien oder Verzeichnisse. Die Syntax ist identisch mit cp. So wird durch Eingabe des Kommandos

```
mv  passwd  passwd.original
```

die Datei passwd auf die Datei passwd.original "verschoben", was einem einfachen Umbenennen gleichkommt.

Interessanter wird es erst, wenn ganze Verzeichnisse verschoben werden:

```
mv bin ~/texte
```

verschiebt das Verzeichnis bin (sofern ein solches im aktuellen Verzeichnis existiert) nach ~/texte. Alle Verzeichnisse und Dateien die vorher unter bin zu finden waren, befinden sich jetzt unter ~/texte/bin. Auch mit diesem Befehl sollte **vorsichtig** umgegangen werden, da schnell ganze Verzeichnisbäume an später nur schwer wiederauffindbare Stellen verschoben werden können. Das Verschieben eines kompletten Verzeichnisbaumes ist nur innerhalb eines Dateisystems (also einer Partition) möglich.

1.5.2. Suchen und Durchsuchen von Dateien

Ein sehr nützlicher Befehl: find. Um in allen Unterverzeichnissen des aktuellen Verzeichnisses nach der Datei emil zu suchen, sollte folgendes eingegeben werden:

```
find  .  -name  "emil"
```

Das erste Argument bezeichnet dabei das Verzeichnis, ab dem die Suche gestartet werden soll. Die Option name verlangt einen zu suchenden String, in dem auch wildcards erlaubt sind.

Um also nach allen Dateien zu suchen, die die Zeichenkette emil im Namen enthalten, müsste der String folgendermaßen geändert werden:

```
find  .  -name  "*emil*"
```

Wenn nicht nach einem bestimmten Dateinamen, sondern nach einer **Zeichenkette in einer Datei gesucht** werden soll, kann dazu das Kommando **grep** verwendet werden. Die folgende Eingabe sucht in der Datei emil nach der Zeichenkette detektive:

```
grep  "detektive"  emil
```

Auf diese Weise lassen sich grosse Textmengen schnell nach bestimmten Zeichenketten durchsuchen. Es können beliebig viele Dateinamen angegeben werden. Auch eine Suche mit wildcards und regulären Ausdrücken wird unterstützt. Als Ergebnis der Suche wird jede Zeile ausgegeben, in der die zu suchende Zeichenkette enthalten ist.

```
ls /usr/lib | grep  "libc"
```

??????????

```
cat /etc/passwd | grep -i "schueler"
```

????????

1.5.3. Symbolische Links

Durch die Verwendung von symbolischen Links kann einer Datei quasi ein **zusätzlicher Name** gegeben werden. Dieser Name "zeigt" dann auf diese Datei. Es kann z.B. vorkommen, dass verschiedene Versionen eines Programmes aufgehoben werden sollen, dass aber die jeweils benutzte Version immer unter dem gleichen Namen verfügbar sein soll.

```
ln  -s  check.2.4.1  check
```

Erzeugt den symbolischen link check, der auf die Datei check.2.4.1 zeigt. Im Verzeichnis sieht dies folgendermaßen aus:

```
lrwxrwxrwx  1  mz  users   1024  Mar 21 17:13 check  ->  check.2.4.1*
```

Links können genau wie Dateien mit rm entfernt werden. Hierbei wird der Link und nicht die referenzierte Datei entfernt!

☑Frage: Befinden sich im Verzeichnis /usr/lib symbolische Links?
☑Anwtort:

1.5.4. Daten archivieren und sichern

Zum Erzeugen und Auspacken von Archiven dient das tar-Kommando (tape archive). Üblicherweise gibt man komprimierten Archiven die Endung **.tgz** oder .**tar.gz**, unkomprimierten **.tar**. Die wichtigsten Anwendungsfälle des **tar**-Befehls sind:

1. Auspacken von Archiven:

```
tar xvfz Archiv.tgz
```

tar entpackt nun das komprimierte Archiv und legt dabei selbständig evtl. Unterverzeichnisse an.

2. Erzeugen von Archiven:

```
tar cvfz Archiv.tgz Datei1 Verz1
```

tar erzeugt das komprimierte Archiv Archiv.tgz, in dem die Datei Datei1 und alle Dateien von Verz1 einschließlich seiner Unterverzeichnisse enthalten sind.

3. Ansehen des Archivinhalts:

```
tar tvfz Archiv.tgz
```

tar gibt ein Inhaltsverzeichnis von Archiv.tgz aus.

Das **flag z** gibt an, dass komprimierte Dateien erzeugt/ausgepackt werden sollen.

☑**Frage**: Was bewirkt folgender Befehl:
tar xvfz lampp.tgz -C /opt

☑**Anwort:**?????????

1.5.5. Aufgaben Arbeiten mit Dateien

☑ Weshalb liefern die Befehle cat emil und cat EMIL unter Umständen unterschiedliche Ergebnisse?

☑ Antwort:

☑ Wofür steht die Angabe Tilde (~) in UNIX-Kommandos?

☑ Antwort:

☑ Nennen Sie alle Unterverzeichnisse, die direkt unterhalb des Hauptverzeichnisses standardmäßig angeordnet sind und beschreiben Sie deren Aufgaben!

☑ Antwort:

☑ Was ist das Kennzeichen einer versteckten Datei? Mit welchem UNIX-Kommando werden versteckte Dateien im Verzeichnis erkennbar?

☑ Antwort:

☑ Mit welchem Befehl können aus einem Verzeichnis alle Dateien und Unterverzeichnisse gelöscht werden? Können die so gelöschten Dateien und Unterverzeichnisse wiederhergestellt werden?

☑ Antwort:

☑ Welche Form der wildcards unterstützt UNIX?

☑ Antwort:

☑ Was ist der Zweck des tar-Kommando? Nennen Sie Beispiele!

☑ Antwort:

☑ Welche zusätzlichen Angabe zeigt das Kommandol ls -l gegenüber dem einfachen ls-Kommando?

☑ Antwort:

☑ Welches UNIX-Kommando bewirkt eine seitenweise Dateiausgabe?

☑ Antwort:

☑ Beschreiben Sie die Wirkungen der folgenden UNIX-Kommandos:
a) shutdown b) init 6 c) tar d) passwd e) pwd f) mkdir g) grep
h) find i) cd j) useradd k) man l) rmdir m) ls n) cat
o) cp p) less q) ln r) mv

☑ Richten Sie in Ihrem Arbeitsverzeichnis folgende Verzeichnisstruktur ein:

home - schueler - schule - bs
- db
- prog
- privat - mutti
- vati
- download
- sonst

☑ Ändern Sie die Verzeichnisstruktur der vorherigen Aufgabe in folgende Struktur ab:

home - schueler - school - bs
- db
- private - karl
- heinz
- sonst
- download

☑ Kopieren Sie die Dateien fstab, passwd und hosts aus /etc in Ihr Homeverzeichnis und untersuchen Sie den Erfolg der Kopie! Verschieben Sie anschließend diese 3 Dateien in das Unterverzeichnis „sonst“! Danach kopieren Sie diese Dateien in alle Unterverzeichnisse Ihres Arbeitsverzeichnisses!

☑ Antwort:

☑ Lassen Sie sich aus dem Inhalt des Verzeichnisses /usr/bin folgende Dateiangaben anzeigen:
- alle Dateien, die die Zeichenfolge ss irgendwo im Namen enthalten.
- alle Dateien, die mit a, b oder c beginnen
- alle Dateien, die als zweites Zeichen im Namen eine Ziffer führen

☑ Antwort:

☑ Suchen Sie die Datei profile und lassen Sie sich diese Datei mit less und cat anzeigen!

☑ Antwort:

☑ Lassen Sie sich alle tgz-Dateien anzeigen, die im System gespeichert sind!

☑ Antwort:

☑ Archivieren Sie mit dem tar-Kommando den gesamten Inhalt Ihres Arbeitsverzeichnisses!

☑ Antwort:

☑ Installieren Sie lampp und auch die Entwicklungsmodule.

☑ Antwort:

1.6. Zugriffsrecht

Nur der super user (root) hat als Systemverwalter uneingeschränkte Zugriffsrechte auf alle

Dateien. Die Verteilung der Zugriffsrechte auf eine Datei ist folgendermaßen gegliedert:

1.6.1. Gliederung der Zugriffsrechte

Eigentümer	Gruppenmitglieder	Andere Systembenutzer

Jede dieser drei Kategorien wird in einem Verzeichnis durch drei Zeichen angezeigt. Zusammen mit dem ersten Zeichen (dem Dateityp: d, l, oder -) ergeben sich die 10 Flags vor jeder Datei. Jedes Flag wird durch ein Zeichen repräsentiert. Die möglichen Flags sind für alle drei Kategorien gleich: **r** für **(read)**, **w** für **(write)** und **x** für **(execute)**. Ist ein Flag nicht gesetzt, so wird dies durch - gekennzeichnet. Betrachten wir die Datei linux.info:

-rw-r-xr-- 1 emil suse 29524 Jun 29 13:11 linux.info

Über diese Datei könnte man sagen: „Der Eigentümer (emil) darf sie ändern und lesen, die Mitglieder der Gruppe *suse* dürfen sie nur lesen (r) und ausführen (execute), während alle anderen Systembenutzer linux.info nur lesen dürfen.“. Ganz ähnlich verhält es sich mit Verzeichnissen. Dann steht nämlich vor den 9 Zeichen, die Rechte zuordnen, noch ein "d" (directory) und könnte so aussehen:

drwxr-xr-x 3 emil suse 1024 Jun 29 13:11 info/

"x" bedeutet in diesem Fall, dass man in dieses Verzeichnis wechseln kann.

1.6.2. Ändern der Besitzrechte

Mit Hilfe des Kommandos

```
chown  Besitzer  Datei
```

kann der Eigentümer einer Datei auf einen anderen Benutzer gewechselt werden. Es handelt sich um eine Tätigkeit, die mit Administratorrechten durchgeführt wird.

1.6.3. Ändern von Zugriffsrechten

Die Änderung von Zugriffsrechten geschieht mit dem Befehl **chmod**

Folgende Eingabe setzt z.B. die Rechte der Datei linux.info für Gruppenmitglieder auf lesbar, veränderbar und ausführbar:

```
chmod  g+rwx  linux.info
```

Wenn Rechte für alle drei Kategorien gesetzt werden sollen, genügt die Angabe der zu ändernden Rechte. Folgende Eingabe setzt die Rechte für die Datei linux.info so, dass niemand Schreiberlaubnis besitzt:

```
chmod  -w  linux.info
```

Die Rechte für Lesen und Ausführen werden davon nicht betroffen.

Eine zweite Möglichkeit der Anwendung des chmod-Befehls ergibt sich durch die Codierung der Maske der Zugriffsbits in oktaler Form. Für jede Zugriffskategorie (user, group, others) wird eine Ziffer (0,... ,7) angegeben.

```
chmod 777 linux.info
```

```
-rwxrwxrwx

chmod 755 linux.info
-?????????

chmod -R 644 linux.info
-?????????
```

Der switch -R beudeutet **rekursiv**

In diesem Zusammenhang interessante Kommandos sind chown (change owner) und chgrp (change group).

Mit dem Kommandu umask können sie die Defaulteinstellung der Zugriffsrechte einer Datei festlgen

```
umask 022
```

bewirkt, dass beim Erzeugen einer Datei diese die Zugriffsrechte 755 (rwxr-xr-x) erhält. Den Befehl umask schreibt man in eine startup-file der shell (.bashrc).

1.6.4. Informationen über den Systemzustand

Häufig ist es wichtig, Auskunft über den Zustand des Systems zu erhalten. Hierbei helfen die Kommandos df, free, ps, top.

```
df -h
```

Gibt Auskunft Über den verfügbaren und benutzten Plattenplatz. Die Ausgabe erfolgt in der Form:

Filesystem	1024-blocks	Used	Available	Capacity	Mounted on
/dev/sda4	699392	659258	5165	99%	/
/dev/sda1	102384	23955	73310	25%	/tmp
/dev/sdb1	2097136	2070485	26651	99%	/archiv_1
/dev/sda3	126976	106908	20068	84%	/hilfe/gnu
Fermi:/cdrom	613934	613934	0	100%	/cd_Fermi

```
free
```

Informiert über die Auslastung des RAM und der swap-Partition:

```
w
```

Anzeige aller momentan angemeldeten Benutzer: Dieses Kommando liefert eine ganze Reihe nützlicher Informationen: Neben der Anzahl der angemeldeten Benutzer erfahren Sie, wie lange das System bereits läuft,wie es momentan belastet ist und was die einzelnen Benutzer so tun.

```
du
```

(disk usage) Gibt Auskunft über den von Unterverzeichnissen und einzelnen Dateien belegten Speicherplatz.

1.6.5. +Mount und Umount von Dateisystemen

Mit dem Befehl mount, der nur von root ausgeführt werden kann, wird ein Datenträger in das

Linux-Dateisystem eingebunden. mount benötigt hierzu zwei Argumente: den Namen des Datenträgers (entspricht der Device-Bezeichnung, z.B. /dev/hda3) und ein Verzeichnis, unter dem der Datenträger eingebunden werden soll. Die Option -t filesystem gibt das Dateisystem an.

```
mount -t msdos /dev/hda2 /dosa
```

stellt die Dos-Partition hda2 unter dem Verzeichnis /dosa zur Verfügung. Mögliche Dateisysteme: ext2, minix, msdos, umsdos, vfat, iso9660, nfs, ncpfs, xiafs, hpfs, proc. Durch die Option -r wird ein Datenträger read-only gemountet. Schreiben ist dann auf diesem Datenträger nicht erlaubt. **mount** schreibt die gemounteten Dateisysteme in die Datei **/etc/mtab**. Wird mount ohne Argumente aufgerufen, so wird der Inhalt dieser Datei ausgegeben.

Durch umount wird ein Datenträger aus dem Linux-Dateisystem entfernt2. Als Argument kann entweder der Name der Device-Bezeichnung oder der Name des Verzeichnisses, in welches der Datenträger eingebunden ist, angegeben werden. Um also z.B. /dev/hda2, eingebunden unter /dosa, zu entfernen, kann

```
umount /dosa
od.
umount /dev/hda2
```

eingegeben werden.

```
mkdir /mnt/win

sudo mount –t smbfs //fs-i/scripten /mnt/win –o username=USERNAME,
workgroup=WORKGROUPNAME
```

Dieser Befehl zeigt das mounten eines Windows-Netzlaufwerkes.

Frage:
vmware bzw. Virtualbox unterstützt die Verwendung sogenannter Shared Folders. Beschreiben Sie hier, was zu tun ist, um dieses Feature nutzen zu können.

Tipp: s. 4ME/HOWTO-VIDEO

Antwort:
??????????????

1.6.6. Manual-Seiten

über Kommandos, Konfigurationsdateien und C-Bibliothekfunktionen geben Ihnen die Manual-Seiten Auskunft:

```
man -1 <Stichwort>          Basch Befehle.
man -2 <Stichwort>          Systemaufrufe der versch. Bibliotheken
man -3 <Stichwort>          Die C-Bibliotheksfunktionen.
man -4 <Stichwort>          Beschreibung v. Konfigurationsdateien
man -5 <Stichwort>          Syntax/Aufbau wichtiger Dateien
...
```

```
man -8 <Stichwort>          Kommandos des Systemverwalters
man -9 <Stichwort>          Beschreibung der Linux-Kernelroutinen
```

Eventuell finden Sie unter /usr/doc/ mehr Information, z. B. unter /usr/doc/howto.

1.7. Prozesssteuerung im LINUX

1.7.1. Die Kommandos ps und kill

Senden von Signalen an laufende Prozesse. Erfordert die Angabe der "**PID**" (ProzessID),die mit **ps** ermittelt werden kann.

```
kill 1233
```
Schickt dem Prozess mit der ID 1233 das TERM-Signal.

```
kill -9 1233
```
Kann der Prozess darauf nicht mehr reagieren, so kann er mit dem optionalen Parameter "-9" dennoch beendet werden.

Den Status von Prozessen anzeigen:

```
ps
```
Zeigt die vom user gestarteten Prozesse an. Weitere Infos liefert man ps.

Mit
```
ps -a
```
werden Ihnen auch die laufenden Prozesse der anderen user angezeigt.

Weitere Optionen:
```
ps -aux
oder
ps aux
oder
pstree
```

```
top
```
Anzeige aller momentan laufenden Prozesse, der Systemauslastung u. v. m.; die Anzeige wird in zeitlichen Abständen aktualisiert. Beenden der Anzeige erfolgt mit q .

1.7.2. Start eines Hintergrundprozesses mit &

```
gedit &
```

```
fg
```

```
bg
```

```
jobs
```

1.8. +Anhang: Die Editoren

1.8.1. Der Editor vi

Die Bedienung des vi ist etwas gewöhnungsbedürftig. Er wird an dieser Stelle anderen Editoren vorgezogen, weil er zum einen auf jedem Unix-ähnlichen Betriebssystem zur Verfügung steht und bei Linux zum standardmäßigen Installationsumfang gehört; zum anderen, weil seine Bedienung eindeutig ist und dadurch i.a. keine Mißverständnisse auftreten. Diese Kurzanleitung sollte Sie in die Lage versetzen, mit Hilfe des vi z.B. diverse Konfigurationsdateien zu editieren. Konzept:

Der vi kennt **3 Betriebsarten (Modi)**

command mode	Jeder Tastendruck wird als Teil eines Befehlsinterpretiert.
input mode	Tastendrücke werden als Texteingaben interpretiert.
last line mode	Für komplexere Kommandos, die in der letzten Zeile editiert werden.

Die wichtigsten Befehle des **command mode** sind:

i wechselt in den input mode (Zeichen werden an der aktuellen Cursorposition eingegeben)

a wechselt in den input mode (Zeichen werden nach__der aktuellen Cursorposition eingegeben)

A wechselt in den input mode (Zeichen werden am Ende der Zeile angehängt)

R wechselt in den input mode (überschreibt den alten Text)

r wechselt zum Überschreiben eines Zeichens in den input mode

s wechselt in den input mode (das Zeichen, auf dem der Cursor steht, wird durch die Eingabe überschrieben)

C wechselt in den input mode (der Rest der Zeile wird durch den neuen Text ersetzt)

o wechselt in den input mode (nach der aktuellen Zeile wird eine neue Zeile eingefügt)

O wechselt in den input mode (vor_ der aktuellen Zeile wird eine neue Zeile eingefügt)

x löscht das aktuelle Zeichen

dd löscht die aktuelle Zeile

dw löscht bis zum Ende des aktuellen Worts

cw wechselt in den input mode (der Rest des aktuellen Worts wird durch die Eingabe überschrieben)

u nimmt das letzte Kommando zurück

j hängt die folgende Zeile an die aktuelle an

. wiederholt das letzte Kommando

: wechselt in den last line mode

Allen Kommandos kann eine Zahl vorangestellt werden, die angibt, auf wieviele Objekte sich der folgende Befehl beziehen soll. So können durch Eingabe von 3dw drei Wörter auf einmal

gelöscht werden. Durch Eingabe von 10x erreicht man das Löschen von zehn Zeichen ab der Cursorposition, 20dd löscht 20 Zeilen.

Die wichtigsten Befehle des **last line** mode:

:q! verlässt den vi, ohne Änderungen zu speichern

:w [Datei] speichert unter [Datei]

:x speichert die geänderte Datei und verlässt den Editor

:e [Datei] editiert [Datei]

:u nimmt das letzte Editierkommando zurück

Eingabe von **ESC** im input mode wechselt in den command mode.