GNU/Linux Befehle

18. Februar 2012

Zusammenfassung

Dieses Dokument beschäftigt sich mit einigen GNU/Linux Befehlen die mir über den Weg gelaufen sind ;). Hoffentlich ist diese bei weitem nicht komplette Liste von Befehlen auch noch für jemanden anderen nützlich. Bitte nicht wundern, es gibt Befehle zu den verschiedensten Themenbereichen. Wann immer ich lange mit einem Befehl gearbeitet habe, oder ein besonders tolles unix werkzeug mir die arbeit erleichter hat, wird es kurzerhand einfach hier niedergeschrieben...

Inhaltsverzeichnis

1 Grundlegendes

Hier gibts die typischen Unix Befehle.

1.1 Eine leere Datei fixer Größe erstellen

Das erstellt eine 10 KB große Datei.

```
cat /dev/zero | dd bs=1024 count=10 > ${datei}
```

1.2 Standard Ausgang in Datei(en) kopieren

```
tee ${file1} ${file2} ...
```

1.3 Dateien auflisten die regulären Ausdruck beinhalten

```
find ${directory} -exec grep -l -E "${regex}" {} \;
```

2 Mounten

Hier die wichtigsten Befehle um Datein und Partitionen ins System einzuhängen.

2.1 Ein Iso-Image einhängen

Um iso-images zu mounten muss man folgenden Befehl ausführen.

```
mount -o loop,ro ${iso_path} ${mountpoint}
```

2.2 Einen Ssh-Account einhängen

Benötigt wird:

- fuse
- fuse-sshfs

Damit man als User fusermounts machen kann, muss man in der Gruppe *fuse* sein (Benutzerverwaltung).
Um nun einen ssh-Account zu laden, muss man folgende Befehle ausführen:

```
sshfs "${user}@${host}:${path}" ${mountpoint}
```

Um den ssh Account wieder freizugeben muss man folgendes ausführen:

```
fusermount -u ${mountpoint}
```

2.3 Ein Swapfile einhängen

Zuerst braucht man eine Datei mit der benötigten Größe (Dies kann z.B. mit dd erreicht werden). Der folgende Befehl erstellt eine Swap-Partition in der Datei *swapfile*:

```
mkswap ${swapfile}
```

Nun wird dem System gesagt, dass es die Datei verwenden soll:

```
sudo swapon ${swapfile}
```

Mit *swapoff* kann man die datei danach wieder aushängen.

3 Die Packeteverwaltung

Apt ist eine Packeteverwaltung für Ubuntu und sollte zum installieren benutzt werden. Weiters kann man auch mit *gdebi* Packete installieren und mit *dpkg*.

3.1 Programme installieren

Dieser Code installiert das Programm mit Namen *name*

```
sudo apt-get install ${name}
```

Ein lokales Packet[1] installieren:

```
sudo gdebi ${packet}
```

3.2 Nur die Source-Codes herunterladen

Dieser Code läd alle Datein für das Packet *name* herunter. Das Archive ist dann im gerade aktiven Verzeichnis zu finden.

```
apt-get source ${name}
```

3.3 Nur das *.deb Packet herunterladen

Dieser Code läd alle Datein für das Packet *name* herunter. Das Archive ist dann unter `/var/cache/apt/archives` zu finden.

```
sudo apt-get -d install ${name}
```

3.4 Andere Version von einem Program verwenden

Um die zu verwendente Version[2] auszuwählen z.b. für Java:

```
sudo update-alternatives --config java
```

[1] eine *.deb Datei
[2] wurde bereits installiert

3.5 Liste aller installierten Version

Um eine Liste aller installierter Versionen zu bekommen kann folgender Befehl[3] verwendet werden:

```
update-alternatives --list java
```

4 Netzwerk

Netzwerkbefehle.

4.1 Lokale Rechner

Lokale Rechner werden unter `${rechnername}.local` angesprochen.

4.2 X-Server tunneln

Das *C* bei den Flags steht für Kompression und sollte auch verwendet werden. Um das X11 Protokoll zu tunneln muss man das `X` oder `Y` Flag setzten

```
ssh -XC "${user}@${rechnerdomain}"
ssh -YC "${user}@${rechnerdomain}"
```

4.3 Netzwerk neustarten

```
sudo /etc/init.d/networking restart
```

5 Programmieren

Hier gehts um Programmiertools und alles was dazugehört.

5.1 Dfu-Programmer

Brenner für z.B. den *at90usb162*. Man muss das Programm als root aufrufen (mit sudo) oder chmod u+s setzten. Um ein neues Programm in den At90 zu laden muss man ihn zuerst löschen und danach das Programm brennen:

```
(sudo) dfu-programmer at90usb162 erase
(sudo) dfu-programmer at90usb162 flash ${hexfile}
```

[3] Am Beispiel java

5.2 Avrdude

Tool um diverse ISP Brenner zu benutzen. Verwendet wird hier ein STK500v2 kompatibles Gerät.

- Avrdude Terminal Mode: Hier kann man diverse Befehle eingeben und interaktiv den avr auslesen:

```
avrdude -p ${device} -c ${protocol} -P ${port} -t
avrdude -p m8 -c stk500v2 -P /dev/ttyACM0 -t
```

6 Git Versionsverwaltungssystem

Schon seit längerem benutze ich regelmäßig dieses äußerst Praktische Werkzeug. Es gibt sehr gute Einführungen in dieses Thema, hier werde ich nur Befehle abdecken, die mir irgendwann nützlich waren und wo ich vielleicht schon x-mal nachschlagen musste...

6.1 'diff' nachdem man 'add' aufgerufen hat

Das kennen vielleicht viele: Grad hat man alle Dateien geadded die sich verändert habe und man ist kurz davor die commit-message zu schreiben, da fällt einem nichtmehr ein was man eigentlich alles geändert hat. Hier hilft:

```
git diff --cached
```

am besten in einem neuen Terminal damit man das ganze ein wenig durchgehen kann und während man so die groben Änderungen beschreibt...

6.2 'merge' ohne die gesamte history

Wenn man z.b. einen Test-Zweig hat und dort mehrere Commits macht, im Endeffekt aber dann nur einen endgültigen sinnvollen Stand hat möchte man vielleicht die Zwischenstände nicht in den Hauptzweig übernehmen. Das kann man mit der `--squash` Option erreichen:

```
git merge --squash ${some_branch}
```

7 VirtualBox

Nützliche Dinge über oder für VirtualBox.

7.1 Virtualbox Kernelmodul installieren

Man muss vor der Installation das Packet:

```
virtualbox-ose-modules-2.6.24-19-generic
```

... natürlich für die richtige Kernel Version installiert haben.

7.2 COM Ports emulieren

Um COM Ports zu emulieren kann die Einstellung 'Host-Pipe' (erzeuge Host Pipe angehackt) gewählt werden. Als Dateiname kann einfach irgendein pfad zu einer Datei die möglichst nicht existiert (da sie sonst wohl überschrieben wird) eingegeben werden. Z.B.: nur com1 erzeugt bei mir eine Datei com1 im Home-Verzeichnis des gerade aktiven Benutzers. Soweit ich das verstanden habe erzeugt das wohl einen unix socket. Sehr nützlich ist hier das Tool *socat* da es sehr viele Protokolle auf andere umsetzten kann. Mit folgendem Befehl

```
socat UNIX-CONNECT:${path_to_com_pipe} TCP-LISTEN:${port}
socat UNIX-CONNECT:${path_to_com_pipe} TCP-LISTEN:8040
```

kann der unix-socket umgesetzt werden auf tcp. Danach kann man bequem mit

```
telnet localhost 8040 -e ^X
```

darauf zugeriffen werden.

8 Linux Specifisches

Hier kommt Linux Specifisches herein (also alles, das wirklich den Kernel betrifft).

8.1 Kernel Modul kompilieren

Um kernel Module zu installieren, sollten die beiden Packete *linux-headers* und *build-essential* (auf Ubuntu basierenden Systemen) mittels folgender Zeile installieret werden:

```
sudo apt-get install linux-headers-‘uname -r‘ build-essential
```

Dies kann oft nötig sein, wenn zusätzliche Kernel Module wie zum Beispiel die Virtual Box Guest Additions installiert werden sollen.