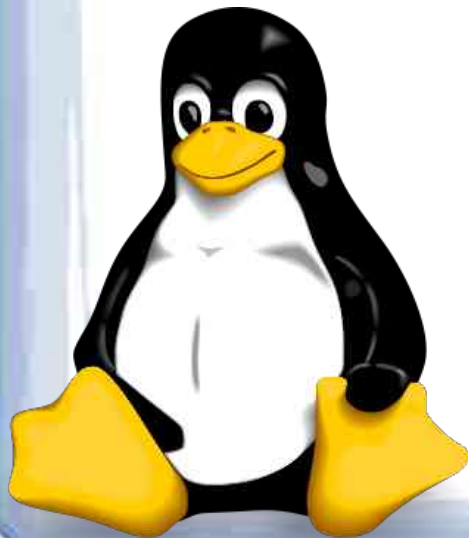


Herzlich Willkommen

Beim SLT 2011 der Linux User Schwabach e.V (LUSC)

„Einführung in die Kommandozeile“

Von und mit Peter Botschafter / „sudo“



Gibt es als PDF-Dokument zum Download unter:

<https://lusc.de/dokuwiki/Events.start>

Worum geht's denn eigentlich?

- Warum Linux?
- Was ist die Kommandozeile?
- Die wichtigsten Befehle zum Einstieg
- Killerkommandos – Finger weg!!
- Websites zu Linux und der Shell

Linux ist „Open-Source“ Software

- GNU/Linux ist Open Source Software d.h. der Quellcode ist frei verfügbar
- Die Entwicklergemeinde ist über die ganze Welt verteilt
- Fehler werden sehr schnell behoben
- Jeder kann mithelfen (Foren, Übersetzungen, Programmieren, etc..)
- Linux ist unendlich flexibel (Distributionen, Tools, Entwickler ...)

Unterschiede Microsoft vs Funktion (Linux)

- Linux arbeitet auf Textbasis. Die Grafik „liegt“ nur „oben drauf“.
Man kann – genug Fachwissen vorausgesetzt von der Konsole aus die Grafik wieder reparieren (sollte diese ihren Dienst quittieren).
- Microsoft Systeme hingegen setzen rein auf die Grafik. Wenn diese versagt, ist eine Neuinstallation unumgänglich.
- Was steht nochmal im Quellcode von MS-Windows oder MS-Office?

Leider geht nix ohne ein bisschen Theorie

Die Kommandozeile

- wird auch als Konsole, Terminal oder „Shell“ bezeichnet
- kann Befehle entgegennehmen und ausführen
- ist bei vielen Aktionen deutlich schneller als die Grafik
- kann sehr nützlich sein (was tun, wenn die Grafik versagen sollte?)

Die verschiedenen Shell's

Es gibt unter Linux verschiedene Arten von Shell's.

Hier die geläufigsten:

- Bourne-Shell ([sh](#)) Die Urshell aus Unix von AT&T
- Bourne again Shell ([bash](#)) Verbesserung der sh ,z.Zt. Standard
- C-Shell ([csh](#)) BSD Entwicklung an C-Syntax angelehnt
- Tenex-C-Shell ([tcsh](#)) benutzerfreundliche Erweiterung der C-Shell
- Z-Shell ([zsh](#)) Vereint viele Features von bash, csh, tcsh

Grafische und „echte Terminals“

- Grafische Terminals öffnet man auf der grafischen Oberfläche
- Mit „strg“ „alt“ und „F1 -F6“ öffnet man sogenannte tty-Terminals
- tty bedeutet „Teletypewriter“ also Daten Ein- und Ausgabe
- Mit „strg“ „alt“ und „F7“ schaltet man wieder auf die Grafik um
- Man kann mehrere Terminal gleichzeitig geöffnet haben

Was geht denn alles in der Shell? - Fast Alles

- Fast alles machbar wie auf der grafischen Oberfläche wie z.B.
 - im Internet surfen ([lynx](#), [links](#), [elinks](#), [w3m](#))
 - Texte schreiben ([vim](#)) / editieren (weiß jeder was das ist?)
 - Musik hören ([moc](#))
 - Bilder anschauen ([fbi](#))
 - Videos anschauen ([mplayer](#)) und vieles mehr

- Was nicht geht sind z.B. Flash Videos im Browser – noch nicht!

Die „dunkle Seite der Macht“

Man hat die Macht alles zu machen – aber auch die Macht, alles kaputt zu machen !!!

Achtung: Es wird zwischen **Groß und Kleinschreibung unterschieden** !! D.h. das Verzeichnis /Home existiert nicht !

Das Verzeichnis /home dagegen schon

Anmerkung: **Immer ganz wichtig: Erst denken – dann tippen !!**

Ohne Netz und doppelten Boden – eben echt

- Unter der Linux Shell gibt es keinen „Mülleimer“
- d.h. Befehle, die Linux ausführen kann werden auch ausgeführt
- Unter Unix bzw. Linux ist ALLES eine DATEI !!!
- d.h. z.B. DVD-Laufwerke, Festplatten, etc. sind unter Linux Dateien
.... daran muss man sich am Anfang erstmal gewöhnen.

Der Linux Verzeichnisbaum

Verzeichnis	kurze Beschreibung
/	Wurzel/Basisverzeichnis
/sbin/	Programme für den Systemverwalter
/bin/	allgemeine Kommandozeilen-Befehle (GNU-Programme)
/boot/	Verzeichnis für Startdateien und den Kernel
/dev/	Geräte-dateien (für Tastatur, Maus, Modem, Festplatte, CDROM, ...)
/home/	Benutzerverzeichnisse
/root/	Verzeichnis des Systemverwalters
/tmp/	temporäre Dateien
/usr/	Dateien/Programme für die Benutzer (Unix-System-Ressources)
/usr/bin/	Anwendungsprogramme für die Benutzer
/usr/src/	Quellcodes (z. B. vom Linux-Kernel)
/var/	Verzeichnis für sich ändernde Daten (Mailverzeichnis, Druckerspooler, ...)
/var/log/	Protokoll-Dateien
/proc/	Systeminformationen; Schnittstelle zum Kernel
/opt/	optionale Programme
/mnt/	Mount-Point für Disketten, CD-ROMs, etc...
/media	Mount-Point für USB-Sticks, USB-Platten, etc...
/etc/	systemweite Konfigurationsdateien
/etc/init.d/	Start-/Stopskripte
/etc/X11/	Konfigurationsdateien von XFree86
/lib/	allgemeine Programm-Bibliotheken (Programm-lader, Hauptbibliothek glibc)
/usr/lib/	Programm-Bibliotheken für Anwendungsprogramme

Hilfeseiten - manpages

- `man man` = Manpage zu den Manpages d.h. eine Hilfeseite zu Hilfeseiten => verlassen mit „q“ (wie quit)
- Bei manchen Prog. gibt es statt `man` auch `--help` oder `--info`
- `pwd` = print working directory (wo stehe ich gerade?)
- Achtung: **Linux unterscheidet**: Verzeichnisse sind „Ordner“, aber alles ist eine Datei. **Dies kann am Anfang ein wenig verwirrend sein.**

Der „list“ Befehl

`ls` = list (Auflistung aller Dateien und Verzeichnisse im aktuellen Verzeichnis. Verzeichnisse sind blau, ausführbare Dateien grün, und Bilder pink dargestellt).

`ls -l` = list long (Auslistung ausführlicher incl. Rechten)

`ls -la` = list long all (Auflistung sehr ausführlich incl. aller „versteckten“ Dateien) => siehe auch man ls für weitere Optionen

TIP: „Pfeil nach oben Taste“ zeigt den letzten Befehl an (kann auch mehrmals gedrückt werden – funktioniert natürlich auch umgekehrt)

Wer, wo, was, wie?

whoami (wer bin ich? zeigt den User Namen an)

whereis (wo ist eine bestimmte Datei/Verzeichnis) z.B. whereis
firefox

last (zeigt an wer, wann, wo angemeldet war – vor allem bei
mehreren Benutzern sehr interessant)

who (zeigt an, wer zur Zeit wo angemeldet ist)

Wieviel Platz ist auf meiner Platte noch frei?

`df` = disk free (zeigt den freien Festplatten Speicherplatz an)

`df -h` = (wie oben nur „menschlich lesbar“ in Gigabyte)

`clear` = macht den Bildschirm wieder „sauber“

Anmerkung: Wie so oft hilft auch hier ein Blick in die manpage weiter ;-)

Wieviel Platz habe ich schon verbraucht?

`du` = disk usage (zeigt an wieviel Speicherplatz benutzt ist)

`du -h` = disk usage (menschlich lesbar in Gigabyte)

`du -hs` = disk usage human readable summary

(hört sich recht wichtig an – zeigt aber nur den belegten Speicherplatz der Festplatte in Gigabyte an, je nachdem, wie man die Platte partitioniert hat)

Wie kann ich das Verzeichnis wechseln?

`cd` = change directory (Verzeichnis wechseln)

`cd /usr/bin`

Anmerkung: Nur z.B. `cd /us` dann Tabulator Taste drücken dann `bin` dann Tabulator Taste drücken. Die Linux Shell hat eine sogenannte „bash-completion“ - **Autovervollständigung – oft sehr sinnvoll !!**

`cd „Enter“` befördert dich wieder direkt in dein Homeverzeichnis

`cd ..` => geht eine Ebene tiefer (im o.g. Fall z.B. von `/usr/bin` zu `/usr`)

Datum, Zeit und Kalender ...

`date` (zeigt Datum und Uhrzeit an) `date +%x` (Tag/Monat/Jahr)

`cal` (zeigt den Kalender des aktuellen Monats an)

`cal 2010` = Kalender für 2010 (`cal 2025` macht dasselbe für 2025)

`cal 04 2010` = Kalender für April 2010 usw.

Neue Verzeichnisse oder Dateien anlegen

`mkdir` = make directory (erstellt ein neues Verzeichnis)

z.B. `mkdir test-1 test2` (legt die l.g. Verzeichnisse an)

`cd test-1` (in das Verzeichnis test-1 wechseln)

`touch testdatei` (legt eine neue leere Datei an)

Dateien verschieben und umbenennen

Das `mv` Kommando hat zwei Funktionen!!

`mv` = move (Dateien verschieben)

z.B. `mv /home/deinhome/test-2/foo1.txt /home/deinhome/test-1`

`mv` (Dateien umbenennen und verschieben)

z.B. `mv /home/deinhome/test1/foo1.txt /home/deinhome/foobar.txt`

Anmerkung: Auch hier macht die Autovervollständigung vieles leichter und schneller ;-)

Dateien kopieren

`cp` = copy (kopiert Dateien)

```
cp /home/deinhome/test-1/foo1.txt /home/deinhome/test-2
```

etwas komplexer ;-)

```
mkdir neue_musik
```

```
cp /home/deinhome/musik/*.mp3 /home/deinhome/neue_musik
```

Kopieren mit dd („disk-dump“)

Zur Erklärung:

disk dump=input file von wo? output file=wohin?

Befehlsbeispiel:

`dd if=/dev/sdb1 of=/dev/sda7` oder `dd if= /dev/sdb1 of=/dev/sdb2`

kopiert z.B. Daten von einem USB Stick auf unsere /Home Partition
(je nachdem, wie diese angelegt wurde.....)

..... oder von einem USB-Stick auf einen anderen.

mount / umount – einbinden und ausbinden

`mount` (alle Laufwerke unter Linux werden „gemountet“ sprich ins Dateisystem **eingebunden**)

`mount` (zeigt an, was alles gemountet ist)

Achtung: Einen USB Stick **nie einfach abziehen!** **Immer** aushängen

Unter Ubuntu bzw. Debian werden Laufwerke normalerweise automatisch gemountet

Ausbinden mit `umount` (**nicht verwechseln mit unmount**) z.B.
`umount /media/stick`

Systemdaten auslesen oder Daten umleiten

`cat` (schreiben und auslesen von Dateien) z.B. `cat /proc/cpuinfo` oder
`cat /etc/issue`

`sudo cat /etc/apt/sources.list`

`cat >> Dateiname` (leitet das Geschriebene in die angegebene Datei um, der vorhandene Text der Datei bleibt erhalten)

z.B. `cd test-1`

`cat >> foo1.txt =>` anschl. „Enter“ =>Text eingeben

beenden mit „strg“ und „d“

Systemdaten auslesen – Große Dateien

`cat/proc/partitions` (zeigt die Partitionen an)

`cat/proc/meminfo` (zeigt Informationen über den RAM an)

`cd /etc/init.d` (wechseln in das Verzeichnis /etc/int.d)

`cat udev` (die Datei „udev“ anzeigen lassen)

`less udev` (wo ist der Unterschied? `less` zeigt uns Dateien seitenweise an und man kann vor – und zurück blättern. Dies ist bei großen Dateien äußerst sinnvoll)

=> beenden mit „q“ wie quit

Word Count – Wassn dass ???

Sehr nützlich, um Zeilen, Wörter und Dateigrößen auszulesen

`wc -l /etc/apt/sources.list` (listet die enthaltenen Zeilen auf)

`wc -w /etc/apt/sources.list` (listet die enthaltenen Wörter auf)

`wc -c /etc/apt/sources.list` (listet die Datengröße in Bytes auf)

Prozesse ausgeben lassen und „killen“

Jede unter Linux gestartete Anwendung oder Eingabe ist ein Prozess

`ps axf` = Prozessausgabe

`ps axfu` = Prozessausgabe für den User

`kill pid Nr` = Prozess mit der folgenden Nummer beenden

`top` (diverse System, Speicher und Prozessorauslastungen) gibt es auch als `htop` (etwas schöner aufbereitet)

„Alles was Recht ist“

Unter Unix / Linux gibt es drei Rechte:

r wie **r**ead (Leserecht)

w wie **w**rite (Schreibrecht)

x wie **e**xecute (Ausführungsrecht)

Rechte und Gruppen

Es gibt auch drei unterschiedliche Gruppen:

u wie user

g wie group

o wie others

Rechte Beispiele

```
drwx----- 2 peter peter 4096 20. Sep 09:16 aktuelles
drwx----- 2 peter peter 4096 6. Mär 2011 bilder
drwx----- 3 peter peter 4096 15. Jan 2011 bilder-neu
-rw-r--r-- 1 peter peter 2129 9. Sep 12:49 csu_pocht_auf_vorratsdatenspeicherung.txt
-rw-r--r-- 1 peter peter 1190 9. Sep 12:55 debatte_um_vorratsdatenspeicherung.txt
drwxr-xr-x 2 peter peter 4096 24. Sep 19:14 Desktop
-rw-r--r-- 1 peter peter 3305 9. Sep 13:00 diskussion_ueber_vorratsdatenspeicherung.txt
drwxr-xr-x 7 peter peter 12288 24. Sep 17:50 downloads
drwx----- 5 peter peter 4096 20. Sep 11:26 dwhelper
drwx----- 2 peter peter 4096 5. Sep 14:32 f
drwxr-xr-x 3 peter peter 4096 20. Sep 14:20 HardRadio.com
-rw-r--r-- 1 peter peter 340 20. Sep 13:32 install_streamtuner2_squeeze.txt
drwxr-xr-x 2 peter peter 4096 23. Sep 16:53 iso-images
```

Rechte Erläuterung

Ein „d“ am Anfang steht für directory – wie Verzeichnis
Dateien haben ein „-“ Zeichen am Anfang stehen

```
drwxr-xr-x 2 peter peter 4096 24. Sep 19:14 Desktop
```

Dies bedeutet also:

rwX

user

darf alles

r-X

group

lesen und ausf.

r-X

others

lesen und ausf.

Rechte hinzufügen oder entfernen

Rechte können in zwei unterschiedlichen Modi geändert werden:

Symbolischer Modus:

Beispiel: -rw-r--r--

mit „+“ od. „-“ Zeichen und dem jeweiligen Benutzer

chmod u+x Dateiname

chmod go-x Dateiname

chmod = change modified

Oktaler Modus:

Beispiel: -rw-r—r--

r = 4

w = 2

x = 1

chmod 755 Dateiname

Runlevel – Die verschiedenen Optionen

Die wichtigsten Runlevel sind 0 / 2 und 6

runlevel

0 = halt (System herunterfahren)

1 = single user mode (Einzelbenutzersystem)

2 = multi user mode + network + X-server
(Mehrbenutzersystem)

3 -5 = nicht genutzt

6 = reboot (System neustarten)

- Runlevel wechseln : `sudo init runlevel Nr.` (z.B. `sudo init 6`)
- Runlevel anzeigen : `sudo runlevel`

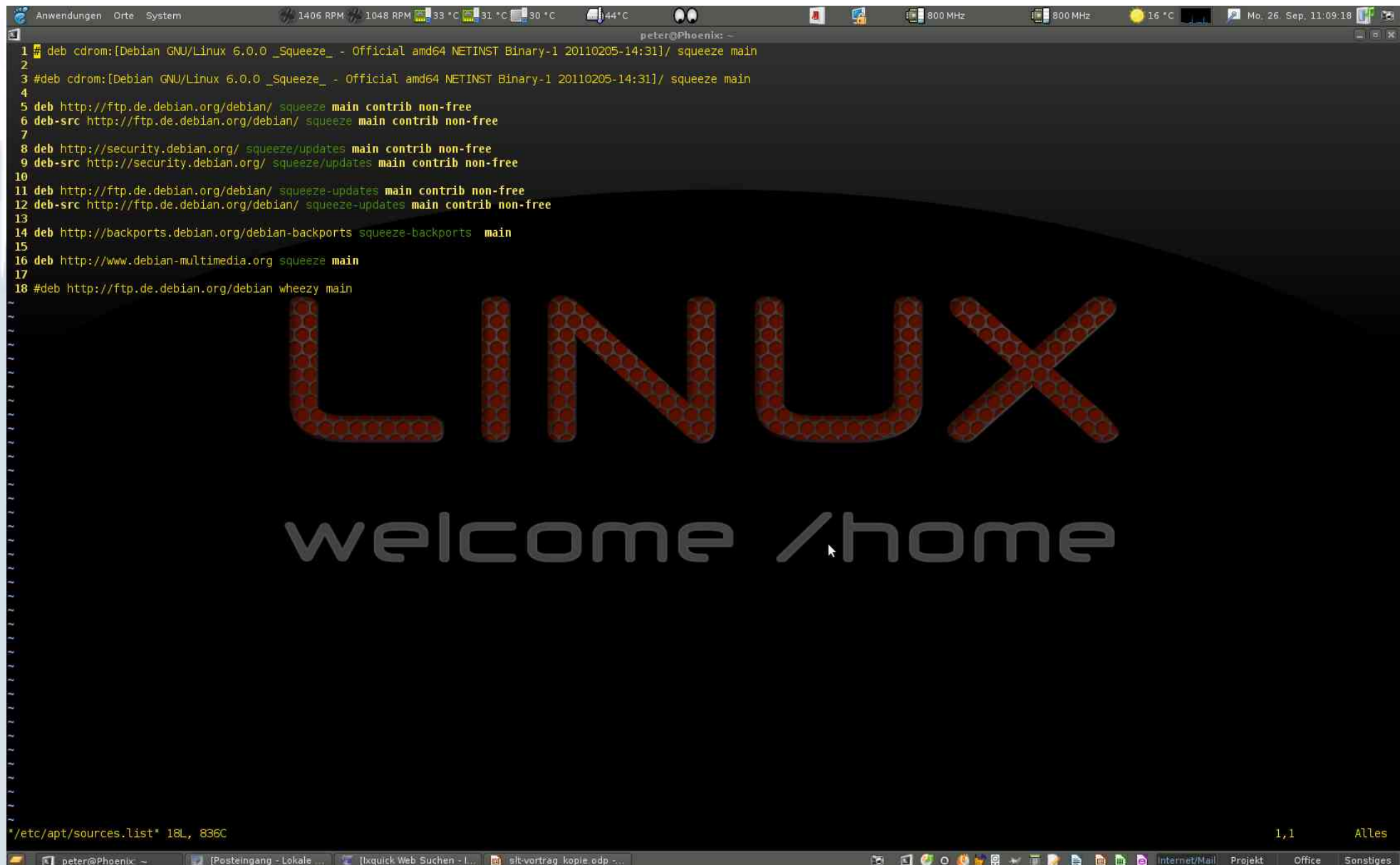
Die „mindestenszweimalüberleg“ Befehle !!

ACHTUNG:

Hiermit kann man sich schlimmstenfalls alles zerstören

- `# rm` = remove (löscht Dateien) hierzu bitte man rm lesen
- `# rm -r` (löscht Verzeichnisse incl. Dateien) bitte ebenfalls man rm
- `# rmdir` = remove directory (löscht Verzeichnisse) manpage !!
- `# passwd` (hiermit kann man sein Passwort auf ein neues ändern)

Konsoleneditor „vim“ - weil's Spass macht ;-)



```
1 deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official amd64 NETINST Binary-1 20110205-14:31]/ squeeze main
2
3 #deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official amd64 NETINST Binary-1 20110205-14:31]/ squeeze main
4
5 deb http://ftp.de.debian.org/debian/ squeeze main contrib non-free
6 deb-src http://ftp.de.debian.org/debian/ squeeze main contrib non-free
7
8 deb http://security.debian.org/ squeeze/updates main contrib non-free
9 deb-src http://security.debian.org/ squeeze/updates main contrib non-free
10
11 deb http://ftp.de.debian.org/debian/ squeeze-updates main contrib non-free
12 deb-src http://ftp.de.debian.org/debian/ squeeze-updates main contrib non-free
13
14 deb http://backports.debian.org/debian-backports squeeze-backports main
15
16 deb http://www.debian-multimedia.org squeeze main
17
18 #deb http://ftp.de.debian.org/debian wheezy main
```

LINUX
welcome /home

"/etc/apt/sources.list" 18L, 836C 1,1 Alles

Datei erstellen im „vim“

vim besitzt einen Kommandomodus und einen Eingabemodus

`vim test.txt` (erstellt eine neue Datei test.txt im aktuellen Verzeichnis)

`i` (wie insert, um in den Eingabemodus zu kommen)

`esc-Taste` (schaltet wieder in den Kommandomodus um)

`esc /` Suchbegriff um in einer Datei nach etwas zu suchen

speichern im „vim“

`esc:q` (quit) um eine Datei zu schließen

`esc:q!` wie oben, allerdings um das Schließen zu erzwingen

`esc:wq` (write quite) um zu speichern und zu schließen

man `vim` für alles was man mit vim machen kann – und er kann echt viel ;-)

`vimtutor` (Lernprogramm für vim)

Meine Kommandozeilen Tool Highlights

- `mc` (Midnight Commander)
- `vim` (Kommandozeilen-Editor)
- `moc` (music on console)
- `ffmpeg` (konvertiert z.B. flv nach mp3)
- `screen` (Fenstermanager)
- `mplayer` (videos auf der Konsole anschauen)
- `hwinfo` (Hardwareinfo)
- `fbi` (Bildbetrachter Konsole)
- `lshw` (listet Hardware auf)

Killerkommandos – Nicht zur Nachahmung empfohlen

`sudo rm -rf /*` „Enter“ („Read mail really fast“ – scherzhaft!!)

lösche rekursiv alle Verzeichnisse und Dateien im Rootverzeichnis ohne Rückfragen

`sudo dd if=/dev/urandom of=/dev/sda`

schreibe Zufallszahlen aus der Gerätedatei `/dev/urandom` auf die Festplatte

Weitere Möglichkeiten des „Datennirvana“

```
sudo cp -R /dev/urandom /dev/sda
```

kopiert eine Zufallszahlen-Datei auf die Festplatte

```
sudo mv -R ~/* /dev/null
```

verschiebt alle Verzeichnisse und Dateien aus dem Homeverzeichnis nach /dev/null (einen Speicherort, den es nicht gibt)

Wie man sich die Rechte nehmen kann!

```
cd /
```

```
sudo chmod 000 -R *
```

Entzieht dem „root“ Benutzer alle Rechte

```
sudo rm -R /etc/passwd
```

löscht sämtliche Dateien mit Passwörtern der Benutzer

Tips und Tricks

- Terminal öffnen => **vimtutor** eingeben = Übungsprogramm für vim
- Versuchen so viel wie möglich in der Konsole zu erledigen
- Regelmäßig die „Linux-User“ lesen (auch für Einsteiger geeignet)
- Regelmäßig Foren / Treffen besuchen, Mailinglisten abonnieren
- Manpages lesen, ixquick befragen, und **NIEMALS AUFGEBEN !!!**

„Vorsicht ist die Mutter der Porzellanbox“

- Das A und O ist also immer wieder vorher überlegen, was man tut!
- Speziell bei Befehlen, bei denen man sich unsicher ist, lieber einmal mehr in Foren stöbern.
- Besondere Vorsicht ist geboten, wenn **„root“** bzw. **„sudo“** Rechte benötigt werden. Damit kann man am Meisten kaputtmachen
- Und immer daran denken - **„Übung macht den Meister“**

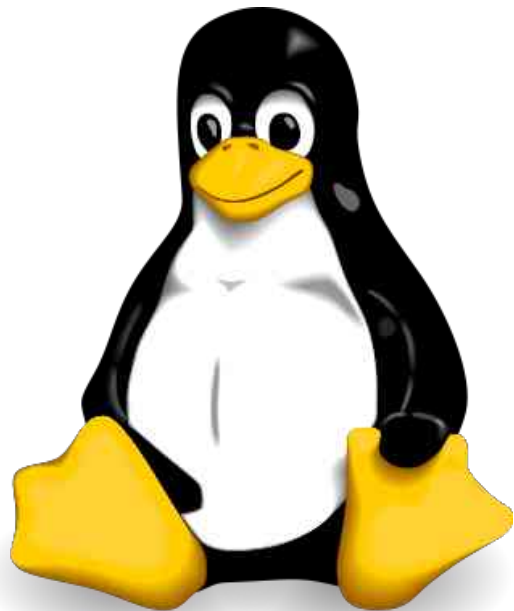
Websites rund um Linux

- debian.de
- linux-forum.de
- debianforum.de
- bin-bash.de
- linux-community.de/
- distrowatch.com
- ubuntuusers.de
- aptosid.de (Super Handbuch!!)

...noch ein paar websites

- selflinux.org/selflinux
- galileocomputing.de/katalog/openbook
- freiesmagazin.de
- felix-schwarz.name/files/opensource/atricles/Linux_ist_nicht_Windows/
- wikipedia.de
- tuxfutter.de
- wikimedia.de

Vielen Dank! Ich habe fertig!



*Ich bedanke mich für die
Aufmerksamkeit*

*und wünsche allen viel Spaß
und Erfolg auf der Shell*

sudo

Linux User Schwabach e.V

LUSC

Lizenzbedingungen

Diese Präsentation steht unter der Creative Commons 3.0 Unported Lizenz. Eine kommerzielle Nutzung ist erlaubt! Die Weitergabe und Erweiterung dieser Dokumentation ist erlaubt, sofern die Namensnennung und Herkunft erfolgt. Die Weitergabe sowie etwaige Änderungen müssen unter der gleichen Lizenz erfolgen.

<http://creativecommons.org/licenses/by-sa/3.0/de/>

Für Rückfragen bin ich per E-Mail unter **sudo-1@web.de** zu erreichen

Das Linux-Maskottchen TUX darf frei benutzt werden, solange die Urheberrechtsinhaber Larry Ewing, Simon Budig und Anja Gerwinski mit der Grafik genannt werden.

<http://creativecommons.org/licences/by-sa/2.0/de/>