

Grundpraktikum Netz- und Datensicherheit

Linux-Befehlsreferenz

Lehrstuhl für Netz- und Datensicherheit
Ruhr-Universität Bochum



Zusammengestellt von: Cornelia Menzel, Endres Puschner
Stand: 15. September 2014
Version: 2.0

Inhaltsverzeichnis

1	Einleitung	2
2	Referenz	2
2.1	Beenden einer Shell	2
2.2	einfache Dateioperationen	2
2.3	Verzeichnis-Operationen	3
2.4	Dateien anzeigen	3
2.5	Dateien erzeugen und editieren	3
2.6	Dateieigenschaften	4
2.7	Dateien suchen	4
2.8	Textdateien	4
2.9	Datenkomprimierung	4
2.10	Festplatten und Dateisysteme	5
2.11	Prozesse	5
2.12	Hosts im Netz	5
2.13	Bildschirmausgaben	6
3	Literatur	6

1 Einleitung

Dies ist eine unvollständige Sammlung von hilfreichen Befehlen. Sie soll den Umgang mit der Shell unter Linux für das Grundlagenpraktikum zur IT-Sicherheit vereinfachen.

Weitere Informationen zu den Befehlen erhält man in den man-pages, die man durch Eingabe von

```
$ man befehl
```

erreicht, oder im Internet.

Hierbei wäre unter anderem <http://www.selflinux.org/selflinux/> eine gute Anlaufstelle.

Anmerkung: Das Dollarzeichen \$ ist nicht mitzuschreiben, sondern verdeutlicht nur, dass es sich hierbei um eine Linux-Shell handelt.

Korrekturen, Verbesserungsvorschläge oder Ergänzungen können an den Praktikumsbetreuer gerichtet werden.

2 Referenz

2.1 Beenden einer Shell

`exit` oder mit der Tastenkombination Strg+D

Bsp.: \$ exit

2.2 einfache Dateioperationen

`ls` – listet die Dateien in einem Verzeichnis auf

Bsp.1: \$ ls

Bsp.2: \$ ls dir1 dir2 – Dateien von bestimmten Verzeichnissen auflisten

`cp` – kopiert Dateien

Bsp.1: \$ cp datei datei2 – kopiert eine Datei in eine andere mit angegebenem Namen

Bsp.2: \$ cp datei datei2 datei3 verz – kopiert mehrere Dateien in ein Verzeichnis

`mv` – Umbenennen / Verschieben von Dateien

Bsp.1: \$ mv datei1 datei2 – datei1 in datei2 umbenennen

Bsp.2: \$ mv datei1 datei2 datei3 zielverzeichnis – Dateien (und Verzeichnisse) in Zielverzeichnis verschieben

`rm` – löscht Dateien

Bsp.1: \$ rm datei1 datei2 – Dateien löschen

Bsp.2: \$ rm -r verz1 verz2 – ganze Verzeichnisse löschen

2.3 Verzeichnis-Operationen

`cd` – in ein angegebenes Verzeichnis wechseln

Bsp.1: `$ cd /drei/vier/fuenf` – in Verzeichnis `/drei/vier/fuenf` wechseln

Bsp.2: `$ cd e` – in Unterverzeichnis `e` gehen

Bsp.3: `$ cd ../verz` – ein Verzeichnis nach oben gehen und dann ins Verzeichnis `verz`

`pwd` – aktuelles Verzeichnis ausgeben

Bsp.: `$ pwd`

`/home/student/verzeichnis`

`mkdir` – Verzeichnisse anlegen

Bsp.1: `$ mkdir verz` – Verzeichnis `verz` anlegen

Bsp.2: `$ mkdir -p /home/hans/Desktop/wichtig` – Alle Verzeichnisse bis hin zu `/home/hans/Desktop/wichtig` erzeugen. Nach diesem Aufruf existiert der Pfad garantiert.

`rmdir` – ein leeres Verzeichnis löschen

Bsp.: `$ rmdir verz`

`rm -r` – ein Verzeichnis mit Inhalt löschen; mehr siehe **einfache Dateioperationen**

2.4 Dateien anzeigen

`cat` – zeigt eine Datei vollständig an

`less` – zeigt eine Datei Seite für Seite an (bis `q` gedrückt wurde)

`head` – zeigt die ersten zehn Zeilen einer Datei an

Bsp.: `$ head datei`

`tail` – zeigt die letzten zehn Zeilen einer Datei an

`nl` – zeigt Dateien mit nummerierten Zeilen an

2.5 Dateien erzeugen und editieren

`nano` – Texteditor zum Editieren von Dateien; wenn die Datei noch nicht vorhanden ist, wird sie erzeugt

Bsp.: `$ nano datei`

Hinweis: Dies funktioniert auch mit jedem anderen Texteditor, wie z.B. `vim` oder `emacs`.

`touch` – eine leere Datei erzeugen (hat auch noch weitere Funktionen)

`echo` – die Ausgabe eines Textes auf der Shell (kann mit `>` auch zur Umleitung in eine Datei genutzt werden)

Bsp.: `$ echo text > datei`

2.6 Dateieigenschaften

`wc` – zählt Zeilen, Wörter und Zeichen (in dieser Reihenfolge) einer Datei

`file` – Typ einer Datei ermitteln

2.7 Dateien suchen

`find` – sucht Dateien in einer Verzeichnishierarchie

Bsp.1: `$ find / -type f -name datei` – eine Datei (`-type f`) im Wurzelverzeichnis (`/`) mit dem Namen `datei` suchen und den Pfad der gefundenen Dateien ausgeben

Bsp.2: `$ find / -type d` – alle Verzeichnisse (`-type d`) ausgeben

`which` – sucht ausführbare Dateien im Suchpfad

Bsp.: `$ which which`
`/usr/bin/which`

`type` – funktioniert wie `which`, ist allerdings bei der Shell standardmäßig dabei

`whereis` – sucht ausführbare Dateien, Dokumentationen und Quelldateien

2.8 Textdateien

`grep` – sucht Zeilen in einer Datei, die einem gegebenen Muster entsprechen

Bsp.: die Datei `datei1` enthält den Text:

Komm ich heute nicht, komm ich morgen.

Viele Köche verderben den Brei.

Ei, mei, einerlei.

`$ grep Brei datei1` – wir suchen die Zeile mit „Brei“

Viele Köche verderben den Brei.

`tee` – kopiert eine Datei und gibt sie gleichzeitig aus

2.9 Datenkomprimierung

Achtung: Wenn eine Datei dekomprimiert wird, werden gleichnamige Dateien im Verzeichnis überschrieben.

`gzip` – komprimiert Dateien im `gzip`-Format

Bsp.: `$ gzip datei` – komprimiert `datei` und erzeugt `datei.gz`; Die Ursprungsdatei `datei` wird dabei gelöscht

`gunzip` – dekomprimiert Dateien im `gzip`-Format

Bsp.: `$ gunzip datei.gz` – dekomprimiert `datei.gz` und erzeugt `datei`; Die Ursprungsdatei `datei.gz` wird dabei gelöscht

`bzip2` – komprimiert Dateien im `BZip`-Format

`bunzip2` – dekomprimiert Dateien im `BZip`-Format

zip – komprimiert Dateien im ZIP-Format

Bsp.: \$ zip datei.zip datei1 datei2 datei3 – packen zu datei.zip

unzip – dekomprimiert Dateien im ZIP-Format

Bsp.1: \$ unzip -l datei.zip – Inhalt des Archivs anzeigen

Bsp.2: \$ unzip datei.zip – entpacken

tar – komprimiert und dekomprimiert Dateien im tar-Format

Bsp.1: \$ tar tvzf paket.tar.gz | less – Inhalt des Pakets ausgeben für tar.gz-Dateien

Bsp.2: \$ tar tvjf paket.tar.bz2 | less – Inhalt des Pakets ausgeben für tar.bz2-Dateien

Bsp.3: \$ tar xvzf paket.tar.gz – Inhalt der tar.gz-Datei entpacken

Bsp.4: \$ tar xvjf paket.tar.bz2 – Inhalt der tar.bz2-Datei entpacken

Bsp.5: \$ tar cvzf paket.tar.gz datei1 datei2 – Dateien in ein .tar.gz-Archiv packen

Bsp.6: \$ tar cvjf paket.tar.bz2 datei1 datei2 – Dateien in ein .tar.bz2-Archiv packen

2.10 Festplatten und Dateisysteme

cd – wechselt das aktuelle Verzeichnis; mehr s. **Verzeichnis-Operationen**

mount – ermöglicht den Zugriff auf eine andere Partition oder ein anderes Gerät

Bsp.1: \$ mount /dev/cdrom /mnt/cdrom – CD-Laufwerk einbinden

Bsp.2: \$ mount /dev/sdb1 /mnt/usb – USB-Stick einbinden

umount – hängt eine Partition oder ein Gerät aus

Bsp.: \$ umount /media/usb

Hinweis: Wenn das Gerät noch verwendet wird, weil z.B. ein Ordner auf dem Gerät noch geöffnet ist, kann das Gerät nicht ausgehängen werden.

2.11 Prozesse

kill – beendet einen Prozess

Bsp.1: \$ kill 13243 – sendet Befehl zum Beenden an PID 13243

Bsp.2: \$ kill -KILL 13243 – erzwingt das Beenden

ps – zeigt laufende Prozesse an

uptime – zeigt die Systemauslastung an, und wie lange das System schon hochgefahren ist

2.12 Hosts im Netz

host – ermittelt Hostnamen, IP-Adressen und DNS-Informationen

ping – prüft, ob ein entfernter Host erreichbar ist

traceroute – zeigt den Netzwerkpfad zu einem entfernten Host an

2.13 Bildschirmausgaben

echo – siehe **Dateien erzeugen und editieren**

printf – gibt formatierten Text aus

```
Bsp.: $ printf "Der Benutzer %s ist %d Jahre alt.\n" Teresa 23
      Der Benutzer Teresa ist 23 Jahre alt.
```

clear – leert das Fenster

Verlaufsspeicher des Terminals (bei GNOME Terminal 2.30.2) aktivieren:

In dem Terminal im Menü Bearbeiten → Profileinstellungen → Rollbalken → unbegrenzt den Haken setzen.

Verlauf des Terminals in einer Textdatei abspeichern:

Unter Bearbeiten → Alles markieren auswählen und kopieren (Strg+Shift+C). Dann in einem Editor der Wahl einfügen (Strg+V, im Terminal mit Shift drücken) und abspeichern.

3 Literatur

Die Befehlsreferenz basiert auf folgendem Buch:

Barrett, Daniel J., *Linux – Kurz & gut*, deutsche Übersetzung von Torsten Wilhelm. O'Reilly Verlag. 2004.