

Cheat Sheet Linux – Linux Essentials – Teil 2

Mit Texten arbeiten

```
cat    schreibt Datei(en) auf stdout
head   gibt die ersten 10 Zeilen aus
  -n <zahl>    gibt <zahl> Zeichen aus
tail   gibt die letzten 10 Zeilen aus
  -n <zahl>    gibt <zahl> Zeichen aus
  -f          follow-Modus
sort    sortiert eine Datei leerzeichengetrennt
  -k <zahl>    sortiert nach Spalte <zahl>
  -t <Trenner> sortiert nach <trenner>
  -g          sortiert korrekt nach Nummern
cut     ausgewählte Teile einer Datei ausgeben
  -d <Trenner> nimmt <Trenner> als Begrenzer
  -f <Spalte>  bestimmt die Spalte
wc      zählt Zeilen, Wörter und Zeichen
  -l          zeigt nur die Anzahl Zeilen
  -w          zeigt nur die Anzahl Wörter
  -m          zeigt nur die Anzahl Zeichen
grep    sucht in einer Datei nach einem Muster
  |          Pipe-Symbol: stdout wird an weiteres
            Programm weitergeleitet
```

IO Redirection

```
cmd < datei    stdin kommt von datei
cmd > datei     stdout wird in datei geschrieben
cmd >> datei    stdout wird an datei angehängt
cmd 2> datei    stderr wird in datei geschrieben
cmd 2>&1        stderr wird an die gleiche
               Stelle wie stdout geschrieben
```

Beispiel:

```
find /etc -iname hosts 1> ausgabe.txt 2>&1
```

Reguläre Ausdrücke mit grep

```
grep <muster> <datei>
  sucht nach <muster> in der <datei>
grep -c <muster> <datei>
  gibt die Anzahl der Treffer zurück
grep -n <muster> <datei>
  gibt zusätzlich die Zeilennummer ausführbare
grep -i <muster> <datei>
  sucht nicht case-sensitive nach <muster>
grep -E <muster> <datei>
  verwendet erweiterte Regurläre Ausdrücke
grep ^<muster> <datei>
  sucht nach <muster> am Zeilenanfang
grep <muster>$ <datei>
  sucht nach <muster> am Zeilenende
grep ^[<muster>] <datei>
  <muster> darf nicht am Zeilenanfang stehen
```

Alltagsbeispiele mit Regulären Ausdrücken

```
grep ^[#] /etc/sources-list
  Ausgabe der Sources-List ohne die Kommentare
dmesg |grep -Ei "warn|error|fail"
  Durchsucht Kernelringpuffer nach Problemen
```

Shell-Skripte

```
Ziel: Mehrere Befehle in Skript zusammenfassen
Executebit setzen    chmod u+x <Skriptname>.sh
Shebang oben einf.   #!/bin/bash
Aufruf               ./<Skriptname>.sh
Ausgabe von Text      echo
Ausgabe o. Umbruch    echo -n
Benutzereingabe       read -p "<Text>" <var>
Zeitverzögerung       sleep <Sekunden>
Bedingunge prüfen     if/then/else/fi
For-Schleife           for/do/done
While-Schleife         while/do/done
```

Variablen

Shellvariable setzen	<name>=<wert>
Umgebungsvar. setzen	export <name>=<wert>
Variable ausgeben	echo \$<name>
alle Umgebungsvars ausg.	env
Variable löschen	unset <name>

Besondere Shellsript-Variablen

\$0	Name des Skriptes
\$1-\$9	Kommandozeilenargumente
\$?	Rückgabewert 0=kein Fehler 1-255=Fehler

Computerhardware auslesen

lscpu	Infos zum Prozessor
free	Infos zur Nutzung des RAMS
lspci	Infos zu PCI und verbundenen Geräten
lsusb	Infos zu USB und verbundenen Geräten
lsmod	Infos zu geladenen Kernelmodulen
fdisk	Bearbeiten von MBR-Plattenpartitionen
gdisk	Bearbeiten von gpt-Plattenpartitionen
/proc/cpuinfo	Infos zum Prozessor
/proc/meminfo	Infos zum Speicher
top	Nutzung v. Linux-Ressourcen in Echtzeit

Prozesse sehen und manipulieren

/proc	Pseudoverzeichnis mit Prozessen
ps	listet Prozesse auf
ps -ef	listet Prozesse ausführlicher auf
pstree	zeigt Prozess in Baumstruktur
top	zeigt Prozesse in Echtzeit
STRG+Z	stoppt Prozess
jobs	zeigt aktive jobs
<cmd> &	startet Befehl im Hintergrund
fg %<jobid>	holt Prozess in den Vordergrund
bg %<jobid>	setzt Prozess in den Hintergrund

wichtige Prozesssignale

1	SIGHUP → auflegen und Konsole schließen
2	SIGINT → Keyboard-Interrupt (STRG+C)
9	SIGKILL → Prozess wird zwangsbeendet
15	SIGTERM → Prozess soll sich selbst beenden

Prozesssteuerung mit kill

kill 100	beendet Prozess 100 mit Sig 15
kill -15 100	beendet Prozess 100 mit Sig 15
kill -TERM 100	beendet Prozess 100 mit Sig 15
kill -9 100	beendet Prozess 100 mit Sig 9
kill -KILL 100	beendet Prozess 100 mit Sig 9
killall <cmd>	beendet alle Prozesse v. <cmd>

Log-Dateien und Befehle

dmesg	zeigt Infos zum Kernel-Ring-Buffer
/var/log	Standardverzeichnis für Log-Files
/var/log/boot.log	Infos zum Start
/var/log/messages	Infos ab Start von syslogd
/var/logsyslog	Alternative zu messages

Netzwerk – Wichtige Dateien und Befehle

ip	Grundbefehl Netzwerk-Konfig. Z.B.:
ip address show	
ip route show	
ifconfig	Konfig. von NICs (alt)
route	Konfig. von Routing (alt)
ping <host>	Prüfung ob Ziel erreichbar
netstat	Infos zu Netzwerk-Konfig
dig <domain>	findet IP-Adresse zu Domain
dig -x <IP>	findet Domain zu IP-Adresse
/etc/hosts	lokale Namensauflösung
/etc/resolv.conf	DNS-Server-Einträge
/etc/nsswitch.conf	Reihenfolge der Beachtung

Cheat Sheet Linux – Linux Essentials – Teil 2

Benutzerinformationen erhalten

id Ausgabe der eigenen UID,
 primäre GID und Gruppen
id <user> Ausgabe der Infos für <user>
getent alle Benutzer eines Systems
/etc/passwd alle Benutzer eines Systems
/etc/shadow dazugehörige Passwörter
/etc/group Gruppen eines Systems
who aktuell angemeldete Nutzer
w wie who nur detaillierter
last wertet /var/log/wtmp aus
 (alle bisher angemeldeten User)

Benutzer-Wechsel

su <user> startet neue Shell als User <user>
su startet neue Shell als root
su -
su -l
su --login startet auch Umgebung als root
sudo <cmd> führt Programm als superuser aus

Benutzer und Gruppen verwalten

groupadd <gr> legt eine neue Gruppe <gr> an
 -r erzeugt Systemgruppe GID < 999
groupmod Gruppe verändern
 -n ändert Gruppenname
 -g ändert GID (VORSICHT)
groupdel Gruppe löschen
useradd legt einen neuen User an
 -D Standardwerte anzeigen/setzen
 -u UID manuell setzen
 -m Homeverzeichnis erzeugen
usermod verändert Benutzerkonfiguration
 Bsp: usermod -aG <grp> <user>
 <user> an Gruppe <grp> anhängen
userdel User löschen
 -r auch Homeverzeichnis löschen
passwd ändert das eigene Passwort
passwd <user> ändert das Passwort von <user>

Datei-Berechtigungen

ls -l zeigt u. a. die Berechtigungen
stat <datei> ausführliche Ansicht der Rechte
für: Alle(a), User(u), Gruppe(g), Andere(o)
Zeichen: lesen(r), schreiben(w), ausführen(x)
Oktal: lesen(4), schreiben(2), ausführen(1)

Rechte setzen

chmod a+x <dateiname>
 setzt für alle zusätzlich das Execute-Recht
chmod o-x <dateiname>
 entfernt das Execute-Recht für Others
chmod -R u=rwx <ordnername>
 setzt Ordner rekursiv für Nutzer auf lesen,
 schreiben, ausführen
chmod 750 <dateiname>
 setzt alle Rechte für Nutzer, lesen und
 ausführen für Gruppe, alle andere haben
 keine Rechte
chown <user>:<group> <dateiname>
 wechselt den Dateieigner und die Gruppe
chown -R <user>:<group> <ordnername>
 wechselt rekursiv Dateieigner und Gruppe
chgrp <group> <dateiname>
 wechselt nur die Gruppe

Spezielle Berechtigungen

setuid Programm wird als Programmeigner
 ausgeführt
 rwsr-xr-x = 4755 (kl. s: x-Recht)
 rwSr--r-- = 4644 (gr. S: kein x)
setgid Programm wird als hinterlegte
 Gruppe ausgeführt
 rwxr-sr-x = 2755 (kl. s: x-Recht)
 rw-r-Sr-- = 2644 (gr. S: kein x)
sticky bit Ordner mit Schreibrechte für alle
 sorgt dafür dass darin erstellte
 Dateien nur vom Eigentümer
 manipuliert/gelöscht werden kann
 rwxrwxrwt = 1777 (kl. t: x-Recht)
 rwxrwx--T = 1770 (gr. T: kein x)

Links

ln <datei> <linkname>
 Setzt einen Hardlink auf datei
ln -s <datei> <linkname>
 Setzt einen Softlink auf datei

Tipps und Tricks (good to know)

sudo -i root-Rechte über sudo
find / -nogroup verwaiste Dateien finden
uptime letzter Start und Auslastung
history -d <id> löscht Eintrag aus History
 mit Nummer <id>