Orientierung und Produktivität auf der Kommandozeile

Silke Meyer
Chemnitzer Linuxtage 2022
silke@silkemeyer.net
@freiefunken @freiefunken@mastodon.social

Basics

Shells	Anm.
/bin/sh	Bourne Shell, Urgestein
/bin/bash	.bashrc, .bash_history (oft Standard)
/usr/bin/zsh	.zshrc, .zsh_history
uvm.	

```
$ echo $0 # aktuell ausgeführte Shell
$ cat /etc/shells # installierte Shells zeigen
$ chsh -s /bin/bash # eigene Standard-Shell ändern
# usermod -s /bin/zsh username # Shell eines
  ↪ anderen Accounts ändern
$ echo $PATH # Verz. mit direkt ausführbaren
  ↪ Dateien
$ <= unprivilegierter Account
# <= root
```

- Autovervollständigung von Befehlen/Dateinamen mit Tabulator-Taste
- Hochscrollen konfigurierbar
- Terminal Emulator für den Desktop: Terminator
- Markieren mit Maus, Kopieren/Einfügen: Strg **Shift** c/v

```
$ cat ~/.bashrc
 HISTSIZE=5000
 HISTTIMEFORMAT="%F %T "
 alias docs="cd /home/user/my-repos/my-
  ↪ documentation/docs && python3 -m venv py3-
  ↪ sphinx && source py3-sphinx/bin/activate"
 $ docs # siehe Alias
```

```
$ history # Bash-Historie ausgeben
$ Strg r # Suche in der History
$ # Pfeiltasten hoch/runter: History navigieren
$ !25     # 25. Befehl nochmal ausführen
$ !!      # letzten Bef. nochmal ausführen
$ sudo !! # letzten Bef. mit Root-Rechten ausführen
$ ESC . # letzten Teil des vorigen Befehls einfügen
```

Orientierung

Hilfestellung finden: uneinheitlich	
`man <tool>`	Linux manual pages: ausführlichere Dokumentation
`tool -help`	Kurzhilfe
`tool --help`	Kurzhilfe
`tool -h`	Kurzhilfe oder anderer Schalter, z.B. "human-readable"
`apropos <tool>`	Übersicht über die man pages, die <tool> erwähnen

```
$ pwd # Pfad zum aktuellen Ordner zeigen
$ tree # Dateien/Ordner in/unterhalb des aktuellen
  ↪ Ordners zeigen
$ cd # ins Homeverzeichnis wechseln (cd ~)
$ cd - # ins letzte Verz. zurückkehren
$ cd ../folder/ # relativer Pfad
```

Suche nach Dateien

```
$ find ~ -type f -name "*ello*" # sucht Dateien im
  ↪ Homeverz., deren Name "ello" enthält
$ find -name "*.txt" # sucht alles im aktuellen
  ↪ Verz., dessen Name auf ".txt" endet
$ find -name "*.txt" -exec head {} \; # wie oben +
  ↪ Ausgabe d. ersten 10 Zeilen (head) aller
  ↪ Treffer
$ ls -rtl # ausführl. Listing, neueste Datei unten
$ ls -rSlh # größte Datei unten
```

Suche in Dateien

```
$ grep -i string datei # Groß-/Kleinschr.
  ↪ ignorieren
$ grep -v string datei # alles außer string zeigen
$ rgrep (-i) -e error -e fatal /var/log # Verz.
  ↪ rekursiv nach mehreren Begriffen durchsuchen
$ zgrep string datei.3.gz # Suche in .gz-
  ↪ komprimierten Dateien
$ grep -c string datei # nur Trefferanzahl zeigen
$ find -name "*.txt" -exec grep -Hni string {} \; #
  ↪  Dateinamen u. Zeilen ausgeben, Groß-/
  ↪ Kleinschr. ignorieren
```

Erste Regular Expressions

```
^ Zeilenanfang
$ Zeilenende
. ein beliebiges Zeichen
.* ein beliebiges Zeichen beliebig oft
```

Komplexere Aktionen

Mit Pipes (|) kann man die Ausgabe eines Befehls als Eingabe für einen anderen Befehl nutzen:

```
$ grep string datei | less # Ergebnis in Pager ö
  ↪ ffnen
$ rgrep string datei | tee output.txt # tee:
  ↪ Ergebnis nach STDOUT und in Datei schreiben
$ rgrep "^string" datei # String muss am
  ↪ Zeilenanfang stehen
$ grep string datei | cut -d " " -f 1 # nur
  ↪ bestimmte Felder ausgeben
$ grep string datei | cut -d " " -f 1 | sort | uniq
  ↪  -c # zeigen, wie oft welcher Wert im 1. Feld
  ↪ der Datei steht
$ for distro in "Debian" "Ubuntu"; do echo "OS:
  ↪ $distro"; done # Schleife
```

Dateien ansehen

```
$ cat datei # Datei auf Bildschirm ausgeben
$ zcat datei.3.gz / bzcat datei.bz2 # komprimierte
  ↪ Dateien auf Bildschirm ausgeben
$ less datei # zum Lesen/Durchblättern öffnen,
  ↪ verlassen mit 'q'
$ zless README.Debian.gz # gzip-komprimierte Datei
  ↪ zum Lesen öffnen (analog: bzless)
$ head # Dateianfang zeigen (Standard: 10 Zeilen)
$ tail # Dateiende zeigen (Standard: 10 Zeilen)
```

Dateien bearbeiten

(siehe unten zum Editor vim)

```
# Umleitung der Ausgabe in Datei:
$ echo string > datei # String in Datei schreiben
$ echo " string" >> datei # ans Ende anhaengen
$ cp -a ordner/unterordner neuer_ordner/ # Ordner
  ↪ inkl. Inhalt kopieren
$ mkdir -p ordner/unterordner/unterunterordner #
  ↪ rekursiv mehrere Verzeichnisse anlegen
$ mv ../datei.txt . # Datei verschieben mit
  ↪ relativer Pfadangabe
```

Prozesse

```
$ ps aux # laufende Prozesse zeigen
$ pstree # Prozesse als Baum darstellen
$ pgrep apache2 # pids von Apache-Prozessen zeigen
$ pidof chromium # auch: pid eines Prozesses
$ killall chromium # Prozess(e) beenden
$ kill (+ Prozess-ID) # Prozess beenden
$ kill -9 $(pidof chromium) # Beenden erzwingen
$ top / htop (iotop / apachetop / mytop) # Details
  ↪ über Resourcenauslastung zeigen
```

Dateitransfer

```
$ wget https://example.org/datei.txt -O ~/Downloads/datei.txt # Datei über HTTP herunterladen, optional mit Zielort
$ scp example.org:~/datei.txt . # Datei über ssh holen
$ scp ~/.bashrc example.org: # lokale Datei über ssh hochladen
# Ordner-Synchronisation: rsync, man page!
$ rsync -av example.org:~/ordner . # entfernten Ordner auf lokalen Rechner syncen
$ rsync -av example.org:~/ordner/ ordner/ # Nur Inhalt des entfernten Ordners in existierenden lokalen Ordner syncen
```

Netzwerk

```
$ ip a # eigene IP-Adressen zeigen
$ ip r bzw. ip -6 r # eigene Routingtabelle zeigen
$ host example.org # DNS-Abfrage
$ host -t AAAA example.org # DNS-Abfrage ipv6
# siehe auch dig und nslookup
$ mtr example.org # traceroute + ping
$ telnet example.org 443 # Verbindung zu Port testen
# Verlassen mit Strg + Alt Gr + 8, dann 'quit'
$ netstat -tulpen # offene Netzwerksockets m. Prozess-ID, erw. Infos, ohne Namensauflösung
```

Nette Tools

- **screen**, siehe "Mehrere Terminals in einem mit GNU Screen" (Axel Beckert, CLT 2015)
- **tmux**, siehe "tmux – Terminal Multiplexer" (Sven Guckes, CLT 2021)
- **mosh**, für Arbeiten über ssh trotz instabiler Verbindung
- **clusterssh**, parallel ssh-Verbindungen zu mehreren Servern öffnen, um überall dasselbe Kommando auszuführen (schnell in Kombination mit bash-Aliasen)
- midnight commander (mc) / mcedit

Weiterführende Themen

- bash-Kanäle und Umleitungen
- awk
- sed
- mehr Regular Expressions
- bash-Skripte
- mit Anderen Tipps austauschen

vim

```
$ update-alternatives --config editor # Standardeditor setzen
$ vimtutor # mitgeliefertes Tutorial
$ cat ~/.vimrc # Konfiguration
```

vim-Shortcut	Aktion
:q / :q! / ZZ	Datei verlassen
:w	Datei speichern
:wq / :x / :wq!	Datei speichern und verlassen
Öwq	= :wq bei engl. Tastaturbelegung
/String ?String	Vorwärts- bzw. Rückwärtssuche
u / Strg r	undo / redo
gg / Shift g	an Dateianfang/-ende springen
:5 / 5 Shift g	zu Zeile 5 springen
vim datei.txt +:5	beim Öffnen zu Zeile 5 springen
dd / 5dd	eine / fünf Zeilen löschen
dw / 5dw	ein / fünf Wörter löschen
yy / 5yy	eine / fünf Zeilen kopieren
p	einfügen
cw / 5cw	nächstes Wort löschen + in Insert-Modus wechseln
zz / zt / zb	Cursorzeile in Bildschirmmitte bzw. oben/unten platzieren
i	Insert-Modus (zum Schreiben)
ESC	zurück in "normalen"Modus
vim-Einstellungen (dauerhaft: in ~/.vimrc)	
:se nu / :se nonu	Zeilennummerierung an/aus
:set paste	im Insert-Modus keine Formatierung vornehmen
:se nu!	Verhalten mit demselbem Befehl an-/ausschalten
Abfolge: ganzen Absatz auskommentieren	
1. Cursor platzieren	
2. Strg v	visueller Modus
3. Pfeil runter	Zeilenanfänge markieren
4. Shift i	in Insert-Modus wechseln
5. #	gewünschtes Kommentarzeichen
5. ESC	Insert-Modus verlassen
Suchen / Ersetzen (Beispiele)	
:%s/foo/bar/	
:%s/foo/bar/gi	in der ganzen Datei, ohne Berücksichtigung von Groß- / Kleinschr.
:%s/foo/bar/gc	bei jedem Treffer fragen

Debugging

- Don't panic! ;-)
- Fehler beschreiben
- Fehler ggf. reproduzieren
- relevante Logdateien nach Fehlermeldungen durchsuchen
- ggf. Loglevel auf Debug erhöhen

```
$ tail -f /var/log/apache/*.log /var/log/syslog # fortlaufend Logs beobachten
$ journalctl -u slapd -f # journal einer systemd unit beoachten
```

- systematisch eingrenzen:
  - Welche Schritte werden gemacht?
  - Bei welchem Schritt tritt der Fehler auf?
  - Mit welchen Befehlen kann man die Schritte einzeln untersuchen?
- Exemplarische Sammlung von Befehlen für Troubleshooting mit Webserver und Zertifikaten

```
$ host example.org # DNS-Abfrage
$ mtr example.org # Route zum Ziel inkl. Paketverluste zeigen
$ telnet example.org 80 # bzw. Port 443, Erreichbarkeit des Ports prüfen
$ curl https://example.org # Website-Aufruf
$ curl -I https://example.org # nur Header holen (z.B. Status Code)
$ apache2ctl -t # gezielter Konfigtest-Befehl für Apache Webserver
$ openssl s_client -connect example.org:443 # siehe entspr. Doku!
$ openssl x509 -in cert.pem -noout -subject -issuer -issuer_hash # Prüfung der Zertifikatskette: Aussteller-Hash von Serverzertifikat abgleichen mit...
$ openssl x509 -in cert.pem -noout -subject -issuer -hash # Prüfung der Zertifikatskette: ... Hash des Intermediate bzw. CA-Zertifikates
```