

Linux Terminal Basic

BLABLABLA

Jan-Marius Lenk, Christoph Parnitzke, Josef Schneider,
Yannick Bungers

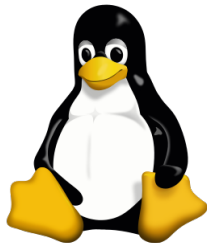
Free and Open Source Software AG
Fakultät für Informatik

1. November 2016

Inhaltsverzeichnis

- ▶ Theorie
- ▶ Arbeiten mit Ordnern, Dateien und Archiven
- ▶ Prozesse
- ▶ Zusatz

Linux Philosophie



- ▶ Philosophie besteht aus drei Punkten:
 - ▶ 1. Schreibe Programme, die nur eine Sache tun und dies erfolgreich (DOTADIW)
 - ▶ 2. Schreibe Programme um zusammen zu Arbeiten
 - ▶ 3. Schreibe Programme, die mit Text arbeiten, denn dies ist universell
- ▶ Ist auch heute noch Kernaussage

Terminal vs. Shell

- ▶ Terminal ist zeilenweise Eingabe von Befehlen
 - 1) TTY (teletype) sind Terminals, erreichbar via STRG+ALT+F[1-7]
 - 2) VTerm: virtueller Terminal in grafischer Oberfläche
 - 3) Shell: interpretiert Eingabe des Benutzers

switch user

- ▶ wechsel des Benutzers innerhalb der Konsole
- ▶ ohne Parameter wird versucht sich als 'root' einzuloggen
- ▶ ansonsten 'su [user]' einloggen als 'user'

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn
Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:
 - ▶ '-b' (background) führt Befehl im Hintergrund aus
[kommt später]

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:
 - ▶ '-b' (background) führt Befehl im Hintergrund aus [kommt später]
 - ▶ '-e' (edit) öffnet Datei zum editieren, erstellt temp. Backup

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus 'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:
 - ▶ '-b' (background) führt Befehl im Hintergrund aus [kommt später]
 - ▶ '-e' (edit) öffnet Datei zum editieren, erstellt temp. Backup
 - ▶ '-s' (shell) öffnet eine Shell

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:
 - ▶ '-b' (background) führt Befehl im Hintergrund aus [kommt später]
 - ▶ '-e' (edit) öffnet Datei zum editieren, erstellt temp. Backup
 - ▶ '-s' (shell) öffnet eine Shell
 - ▶ '-u [user]' führt Befehl als Benutzer aus

super user do

Mit großer Macht kommt große Verantwortung

'sudo' führt einen Befehl mit administrativer Berechtigung aus
'sudo [command]'. [Achtung!!] 'sudo' funktioniert nur, wenn Benutzer in root-Gruppe ist.

- ▶ führt man Befehle als Root aus, sollte man vorsichtig sein
- ▶ Parameter:
 - ▶ '-b' (background) führt Befehl im Hintergrund aus [kommt später]
 - ▶ '-e' (edit) öffnet Datei zum editieren, erstellt temp. Backup
 - ▶ '-s' (shell) öffnet eine Shell
 - ▶ '-u [user]' führt Befehl als Benutzer aus
- ▶ wird '[nur Root]' angegeben, muss Befehl mit sudo gestartet werden

apt-get vs. apt vs. aptitude

[nur Root]

- ▶ Paket Management: apt Kurzform für apt-get und apt-cache
- ▶ wichtigsten Befehle:
 1. 'apt install [pkg...]'
 2. 'apt update'
 3. 'apt [upgrade — dist-upgrade — full-upgrade]'
 4. 'apt remove [pkg...]'
- ▶ 'aptitude' ist etwas grafischer und hat Zusatzfunktionen

man page

Wenn man mal nicht weiter weiß

- ▶ Benutzeranleitung für das System
- ▶ Anleitung zu installierten Programmen/Tools
- ▶ Sektionen, die besonders interessant sind:
 - ▶ NAME, SYNOPSIS, DESCRIPTION, OPTIONS (EXAMPLE und BUGS)
- ▶ Wie werden Einträge gesucht?
 1. ausführbare Programme und Shell Kommandos
 2. Systemaufrufe (vom Kernel bereit gestellt)
 3. Bibliotheksaufrufe (innerhalb von Programm-Bibliotheken)
 4. spezielle Dateien (in /dev)
 5. Dateiformate und Konventionen (/etc/passwd)
 6. Spiele
 7. Sonstiges (Macropakete und Konventionen)
 8. System administrations Kommandos (nur Root)
 9. Kernel Routinen (kein Standard)

list

Ein kleines Licht in der Dunkelheit

- ▶ listet alle Ordner und Dateien in Verzeichnis auf
- ▶ aktuelles Verzeichnis ist dabei Default
- ▶ mögliche Argumente:
 - ▶ '-a' (all) inkl. versteckter Verzeichnisse

list

Ein kleines Licht in der Dunkelheit

- ▶ listet alle Ordner und Dateien in Verzeichnis auf
- ▶ aktuelles Verzeichnis ist dabei Default
- ▶ mögliche Argumente:
 - ▶ '-a' (all) inkl. versteckter Verzeichnisse
 - ▶ '-l' (long listing format) inkl. Rechte, Besitzer, Größe, etc.

list

Ein kleines Licht in der Dunkelheit

- ▶ listet alle Ordner und Dateien in Verzeichnis auf
- ▶ aktuelles Verzeichnis ist dabei Default
- ▶ mögliche Argumente:
 - ▶ '-a' (all) inkl. versteckter Verzeichnisse
 - ▶ '-l' (long listing format) inkl. Rechte, Besitzer, Größe, etc.
 - ▶ '-h' (human readable) Größen der Dateien leserlicher

list

Ein kleines Licht in der Dunkelheit

- ▶ listet alle Ordner und Dateien in Verzeichnis auf
- ▶ aktuelles Verzeichnis ist dabei Default
- ▶ mögliche Argumente:
 - ▶ '-a' (all) inkl. versteckter Verzeichnisse
 - ▶ '-l' (long listing format) inkl. Rechte, Besitzer, Größe, etc.
 - ▶ '-h' (human readable) Größen der Dateien leserlicher
 - ▶ '-r' (recursive) umgekehrte Reihenfolge

list

Ein kleines Licht in der Dunkelheit

- ▶ listet alle Ordner und Dateien in Verzeichnis auf
- ▶ aktuelles Verzeichnis ist dabei Default
- ▶ mögliche Argumente:
 - ▶ '-a' (all) inkl. versteckter Verzeichnisse
 - ▶ '-l' (long listing format) inkl. Rechte, Besitzer, Größe, etc.
 - ▶ '-h' (human readable) Größen der Dateien leserlicher
 - ▶ '-r' (recursive) umgekehrte Reihenfolge
 - ▶ '-m' Namen durch Kommata getrennt

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'
 - ▶ Wechsel in Home-Directory mit 'cd' (ohne Pfad)

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'
 - ▶ Wechsel in Home-Directory mit 'cd' (ohne Pfad)
 - ▶ Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'
 - ▶ Wechsel in Home-Directory mit 'cd' (ohne Pfad)
 - ▶ Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))
 - ▶ 'cd /home/[username]/Dokumente'

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

► Notationen:

- Wechsel in Parent-Directory mit '../'
- Wechsel in Home-Directory mit 'cd' (ohne Pfad)
- Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))
 - 'cd /home/[username]/Dokumente'
 - 'cd ~/Dokumente' (Abkürzen von Home-Directory mit ~)

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

► Notationen:

- Wechsel in Parent-Directory mit '../'
- Wechsel in Home-Directory mit 'cd' (ohne Pfad)
- Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))
 - 'cd /home/[username]/Dokumente'
 - 'cd ~/Dokumente' (Abkürzen von Home-Directory mit ~)
 - 'cd Dokumente' (relativ vom aktuellen Verzeichnis)

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'
 - ▶ Wechsel in Home-Directory mit 'cd' (ohne Pfad)
 - ▶ Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))
 - ▶ 'cd /home/[username]/Dokumente'
 - ▶ 'cd ~/Dokumente' (Abkürzen von Home-Directory mit ~)
 - ▶ 'cd Dokumente' (relativ vom aktuellen Verzeichnis)
- ▶ Funfact: 'cd .' (wechselt in das aktuelle Verzeichnis)

change directory

Die kleine Form der Teleportation

'cd' bietet das Navigieren durch das Dateisystem

- ▶ Notationen:
 - ▶ Wechsel in Parent-Directory mit '../'
 - ▶ Wechsel in Home-Directory mit 'cd' (ohne Pfad)
 - ▶ Wechsel in Verzeichnis (Bsp.: Dokumente (bzw. Documents))
 - ▶ 'cd /home/[username]/Dokumente'
 - ▶ 'cd ~/Dokumente' (Abkürzen von Home-Directory mit ~)
 - ▶ 'cd Dokumente' (relativ vom aktuellen Verzeichnis)
- ▶ Funfact: 'cd .' (wechselt in das aktuelle Verzeichnis)
 - ▶ es tut nichts, aber dafür sehr gut

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])
- ▶ Beispiel:

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])
- ▶ Beispiel:
 - ▶ 'mkdir neuer-ordner/' (relativ zum aktuellen Pfad)

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])
- ▶ Beispiel:
 - ▶ 'mkdir neuer-ordner/' (relativ zum aktuellen Pfad)
 - ▶ 'mkdir /tmp/neuer-ordner/tmp1/tmp2/'

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])
- ▶ Beispiel:
 - ▶ 'mkdir neuer-ordner/' (relativ zum aktuellen Pfad)
 - ▶ 'mkdir /tmp/neuer-ordner/tmp1/tmp2/'
- ▶ '-p' erstellt Parent-Directory, falls es nicht existiert

make directory

Verzeichnisse erschaffen

'mkdir' erstellt Ordner auf dem Rechner

- ▶ Syntax: 'mkdir [path]' (erzeugt Ordner mit Pfad [path])
- ▶ Beispiel:
 - ▶ 'mkdir neuer-ordner/' (relativ zum aktuellen Pfad)
 - ▶ 'mkdir /tmp/neuer-ordner/tmp1/tmp2/'
- ▶ '-p' erstellt Parent-Directory, falls es nicht existiert
- ▶ '-mode=[mode]' Modus mit dem der Ordner erstellt wird (siehe chmod)

Gruppenverwaltung

- ▶ 'groups [username]' (zeigt Gruppen von Benutzer an)

Gruppenverwaltung

- ▶ 'groups [username]' (zeigt Gruppen von Benutzer an)
- ▶ 'groupadd [groupname]' (legt Gruppe an) [nur Root]

Gruppenverwaltung

- ▶ `'groups [username]'` (zeigt Gruppen von Benutzer an)
- ▶ `'groupadd [groupname]'` (legt Gruppe an) [nur Root]
- ▶ `'groupdel [groupname]'` (löscht Gruppe) [nur Root]

Gruppenverwaltung

- ▶ 'groups [username]' (zeigt Gruppen von Benutzer an)
- ▶ 'groupadd [groupname]' (legt Gruppe an) [nur Root]
- ▶ 'groupdel [groupname]' (löscht Gruppe) [nur Root]
- ▶ 'groupmod [option] [groupname]' (Bsp.: -new-name [Name]) [nur Root]

Gruppenverwaltung

- ▶ 'groups [username]' (zeigt Gruppen von Benutzer an)
- ▶ 'groupadd [groupname]' (legt Gruppe an) [nur Root]
- ▶ 'groupdel [groupname]' (löscht Gruppe) [nur Root]
- ▶ 'groupmod [option] [groupname]' (Bsp.: -new-name [Name]) [nur Root]
- ▶ 'usermod -aG [username] [groupname]' (Benutzer zu Gruppe hinzufügen) [nur Root]

Gruppenverwaltung

- ▶ 'groups [username]' (zeigt Gruppen von Benutzer an)
- ▶ 'groupadd [groupname]' (legt Gruppe an) [nur Root]
- ▶ 'groupdel [groupname]' (löscht Gruppe) [nur Root]
- ▶ 'groupmod [option] [groupname]' (Bsp.: -new-name [Name]) [nur Root]
- ▶ 'usermod -aG [username] [groupname]' (Benutzer zu Gruppe hinzufügen) [nur Root]
- ▶ 'deluser [username] [groupname]' (Benutzer aus Gruppe entfernen) [nur Root]

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'
- ▶ Beispiel (/home/[username]/neuer-ordner/):

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'
- ▶ Beispiel (/home/[username]/neuer-ordner/):
 - ▶ aktuelle Rechte: u=rwx g=r-x a=r-x

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'
- ▶ Beispiel (/home/[username]/neuer-ordner/):
 - ▶ aktuelle Rechte: u=rwx g=r-x a=r-x
 - ▶ 'chmod g=rwx ~/neuer-ordner' (wir geben der Gruppe Schreib-Recht)

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'
- ▶ Beispiel (/home/[username]/neuer-ordner/):
 - ▶ aktuelle Rechte: u=rwx g=r-x a=r-x
 - ▶ 'chmod g=rwx ~/neuer-ordner' (wir geben der Gruppe Schreib-Recht)
 - ▶ 'chmod a-r ~/neuer-ordner' (wir nehmen allen anderen das Lese-Recht)

change mode

Mein Ordner gehört mir

'chmod' ändert die Zugriffsrechte von Dateien und Ordnern

- ▶ Syntax: 'chmod [mode] [path]'
- ▶ Beispiel (/home/[username]/neuer-ordner/):
 - ▶ aktuelle Rechte: u=rwx g=r-x a=r-x
 - ▶ 'chmod g=rwx ~/neuer-ordner' (wir geben der Gruppe Schreib-Recht)
 - ▶ 'chmod a-r ~/neuer-ordner' (wir nehmen allen anderen das Lese-Recht)
 - ▶ neue Rechte: u=rwx g=rwx a=-x

Ändern des Besitzers von Ordnern und Dateien.

- ▶ zu jeder Datei/Ordner gehören 2 Besitzer:
 - ▶ ein Nutzer
- ▶ 'chown [user] [file]' [file] gehört jetzt [user]
- ▶ 'chown [user]:[group] [file]' [file] gehört jetzt zu [group] und [user]
- ▶ Parameter -R ändert die Zugehörigkeit rekursiv

Ändern des Besitzers von Ordnern und Dateien.

- ▶ zu jeder Datei/Ordner gehören 2 Besitzer:
 - ▶ ein Nutzer
 - ▶ eine Gruppe
- ▶ 'chown [user] [file]' [file] gehört jetzt [user]
- ▶ 'chown [user]:[group] [file]' [file] gehört jetzt zu [group] und [user]
- ▶ Parameter -R ändert die Zugehörigkeit rekursiv

Löschen von leeren Ordnern.

- ▶ Befehl ist weniger mächtig als man erwartet
- ▶ Vorsicht ist geboten, auch leere Verzeichnisse können wichtig sein!!
- ▶ leere Ordner in dem leeren Ordner werden auch gelöscht

Input, Output und Error

- ▶ jedes Programm hat drei Datenströme:
 - 1) stdin (Standard Input [number: 0])
 - 2) stdout (Standard Output [number: 1])
 - 3) stderr (Standard Error Output [number: 2])
- ▶ Bsp.: 'echo'
 - ▶ 'echo' gibt einen eingegebenen Text zurück

Input, Output und Error

- ▶ jedes Programm hat drei Datenströme:
 - 1) stdin (Standard Input [number: 0])
 - 2) stdout (Standard Output [number: 1])
 - 3) stderr (Standard Error Output [number: 2])
- ▶ Bsp.: 'echo'
 - ▶ 'echo' gibt einen eingegebenen Text zurück
 - ▶ Syntax: 'echo [text]'

Input, Output und Error

- ▶ jedes Programm hat drei Datenströme:
 - 1) stdin (Standard Input [number: 0])
 - 2) stdout (Standard Output [number: 1])
 - 3) stderr (Standard Error Output [number: 2])
 - ▶ Bsp.: 'echo'
 - ▶ 'echo' gibt einen eingegebenen Text zurück
 - ▶ Syntax: 'echo [text]'
- text ist stdin und die Konsole ist stdout und stderr

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'
- ▶ fancy stuff:

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'
- ▶ fancy stuff:
 - ▶ Umleiten von stderr: 2> statt >

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'
- ▶ fancy stuff:
 - ▶ Umleiten von stderr: 2> statt >
 - ▶ also durch Angabe von Stream Nummer

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'
- ▶ fancy stuff:
 - ▶ Umleiten von stderr: 2> statt >
 - ▶ also durch Angabe von Stream Nummer
 - ▶ Bsp.: Umleiten von stderr auf stdout

Pipe, < und >

Pipes helfen die Ein- und Ausgabe umzuleiten

- ▶ '<' und '>' laden und schreiben Dateien
 - ▶ '>' Schreiben in eine Datei
 - ▶ Bsp.: 'cal > test.txt'
 - ▶ '<' Laden aus einer Datei
 - ▶ Bsp.: 'less < test.txt'
- ▶ Pipe '|'
 - ▶ Ausgabe von Befehl → Eingabe anderer Befehl
 - ▶ Bsp.: 'date | less'
- ▶ fancy stuff:
 - ▶ Umleiten von stderr: 2> statt >
 - ▶ also durch Angabe von Stream Nummer
 - ▶ Bsp.: Umleiten von stderr auf stdout
 - ▶ ls -r > output.txt 2> &1

more

Less than everything

'more' ist ein Filereader

- ▶ Angucken von Datei ohne Editieren
- ▶ Arbeitet auf Standardausgabe
- ▶ Beherrscht alle dem System bekannten Codierungen
- ▶ Navigierung ist allerdings ein wenig hinderlich
- ▶ Grundlage zu less

less

Even more than 'more

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauf Tasten

less

Even more than 'more'

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauftasten
 - ▶ '/pattern' Durchsucht das Dokument nach angegebenen Muster

less

Even more than 'more'

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauftasten
 - ▶ '/pattern' Durchsucht das Dokument nach angegebenen Muster
 - ▶ '?pattern' ist wie /pattern nur rückwärts

less

Even more than 'more'

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauftasten
 - ▶ '/pattern' Durchsucht das Dokument nach angegebenen Muster
 - ▶ '?pattern' ist wie /pattern nur rückwärts
 - ▶ '&pattern' zeigt nur Zeilen, mit Muster

less

Even more than 'more'

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauftasten
 - ▶ '/pattern' Durchsucht das Dokument nach angegebenen Muster
 - ▶ '?pattern' ist wie /pattern nur rückwärts
 - ▶ '&pattern' zeigt nur Zeilen, mit Muster
 - ▶ wird pattern benutzt, dann zeigt 'n' das nächste Vorkommen

less

Even more than 'more'

'less' ist intelligenter Nachfolger von more

- ▶ unterstützt Scrolling
- ▶ Navigation:
 - ▶ Pfeil- und Bildlauftasten
 - ▶ '/pattern' Durchsucht das Dokument nach angegebenen Muster
 - ▶ '?pattern' ist wie /pattern nur rückwärts
 - ▶ '&pattern' zeigt nur Zeilen, mit Muster
 - ▶ wird pattern benutzt, dann zeigt 'n' das nächste Vorkommen
 - ▶ ':n' ruft das nächste Dokument auf (wenn less mehrere bekommen hat)

concatenate

Was eine Katze mit Linux zu tun hat

'cat' konkateniert und zeigt die Inhalte von Dateien

- ▶ an sich nicht mächtig, sehr einfacher Befehl

concatenate

Was eine Katze mit Linux zu tun hat

'cat' konkateniert und zeigt die Inhalte von Dateien

- ▶ an sich nicht mächtig, sehr einfacher Befehl
- ▶ Parameter

concatenate

Was eine Katze mit Linux zu tun hat

'cat' konkateniert und zeigt die Inhalte von Dateien

- ▶ an sich nicht mächtig, sehr einfacher Befehl
- ▶ Parameter
 - ▶ '-n' Nummeriert die ausgegebenen Zeilen

concatenate

Was eine Katze mit Linux zu tun hat

'cat' konkateniert und zeigt die Inhalte von Dateien

- ▶ an sich nicht mächtig, sehr einfacher Befehl
- ▶ Parameter
 - ▶ '-n' Nummeriert die ausgegebenen Zeilen
 - ▶ '-s' Unterdrückt Ausgabe von sich wiederholenden Leerzeichen

concatenate

Was eine Katze mit Linux zu tun hat

'cat' konkateniert und zeigt die Inhalte von Dateien

- ▶ an sich nicht mächtig, sehr einfacher Befehl
- ▶ Parameter
 - ▶ '-n' Nummeriert die ausgegebenen Zeilen
 - ▶ '-s' Unterdrückt Ausgabe von sich wiederholenden Leerzeichen
- ▶ Bsp.: 'cat -m output.txt'

head & tail

Dateien von Kopf bis Fuß

Ähnlich zu Cat, sind 'head' und 'tail' einfach Textdatei Betrachter. Allerdings beschränken sich diese auf bestimmte Teile der Datei.

- ▶ 'head' gibt ersten 10 Zeilen der angegebenen Datei aus
- ▶ 'tail' gibt letzten 10 Zeilen der angegebenen Datei aus
- ▶ Parameter:
 - ▶ head:
 - ▶ '-n [val]' gibt ersten [val] Zeilen aus
 - ▶ tail:

head & tail

Dateien von Kopf bis Fuß

Ähnlich zu Cat, sind 'head' und 'tail' einfach Textdatei Betrachter. Allerdings beschränken sich diese auf bestimmte Teile der Datei.

- ▶ 'head' gibt ersten 10 Zeilen der angegebenen Datei aus
- ▶ 'tail' gibt letzten 10 Zeilen der angegebenen Datei aus
- ▶ Parameter:
 - ▶ head:
 - ▶ '-n [val]' gibt ersten [val] Zeilen aus
 - ▶ tail:
 - ▶ '-n [val]' analog zu head

head & tail

Dateien von Kopf bis Fuß

Ähnlich zu Cat, sind 'head' und 'tail' einfach Textdatei Betrachter. Allerdings beschränken sich diese auf bestimmte Teile der Datei.

- ▶ 'head' gibt ersten 10 Zeilen der angegebenen Datei aus
- ▶ 'tail' gibt letzten 10 Zeilen der angegebenen Datei aus
- ▶ Parameter:
 - ▶ head:
 - ▶ '-n [val]' gibt ersten [val] Zeilen aus
 - ▶ tail:
 - ▶ '-n [val]' analog zu head
 - ▶ '-f' fügt weitere Zeilen zur Ausgabe hinzu, falls Inhalt weiter anwächst

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)
 - ▶ '-r' Ausgabe umdrehen

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)
 - ▶ '-r' Ausgabe umdrehen
 - ▶ '-c' Check ob Inhalt von Datei sortiert ist

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ `'sort [option] [File]'` (liest die Datei File und sortiert den Inhalt)
 - ▶ `'-r'` Ausgabe umdrehen
 - ▶ `'-c'` Check ob Inhalt von Datei sortiert ist
 - ▶ Bsp.: `'sort -r output.txt'`

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)
 - ▶ '-r' Ausgabe umdrehen
 - ▶ '-c' Check ob Inhalt von Datei sortiert ist
 - ▶ Bsp.: 'sort -r output.txt'
- ▶ 'find [path] [pattern]' (suchen nach Datei im System via Muster)

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)
 - ▶ '-r' Ausgabe umdrehen
 - ▶ '-c' Check ob Inhalt von Datei sortiert ist
 - ▶ Bsp.: 'sort -r output.txt'
- ▶ 'find [path] [pattern]' (suchen nach Datei im System via Muster)

Achtung!! Suche kann lange dauern

Sortieren und Suchen

Auch wenn man beim Sortieren erst einmal an das Sortieren von Dateien denkt, bezieht sich das Sortieren auf den Inhalt von Dateien. Das Suchen dagegen auf das Finden von Dateien.

- ▶ 'sort [option] [File]' (liest die Datei File und sortiert den Inhalt)
 - ▶ '-r' Ausgabe umdrehen
 - ▶ '-c' Check ob Inhalt von Datei sortiert ist
 - ▶ Bsp.: 'sort -r output.txt'
- ▶ 'find [path] [pattern]' (suchen nach Datei im System via Muster)

Achtung!! Suche kann lange dauern

- ▶ Bsp.: 'find -name output.txt' (suche im Home-Folder 'output.txt')

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
 - ▶ '-n' (no-clobber) überschreibt **niemals**
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
 - ▶ '-n' (no-clobber) überschreibt **niemals**
 - ▶ '-backup' erstellt Backup vor Überschreiben
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
 - ▶ '-n' (no-clobber) überschreibt **niemals**
 - ▶ '-backup' erstellt Backup vor Überschreiben
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ selben wie 'mv'

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
 - ▶ '-n' (no-clobber) überschreibt **niemals**
 - ▶ '-backup' erstellt Backup vor Überschreiben
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ selben wie 'mv'
 - ▶ '-r' (recursive) ermöglicht verschieben von Ordnern

copy & move

Die Kunst des Klonens und Umbenennens

'cp' und 'mv' sind neben 'ls' und 'cd' wohl die wichtigsten Befehle in Linux.

- ▶ Dateiname muss bei beiden Befehlen nicht gleich bleiben
- ▶ 'mv [path1] [path2]'
 - ▶ verschiebt Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ '-i' (interactive) fragt vor Überschreiben
 - ▶ '-f' (force) überschreibt **immer**
 - ▶ '-n' (no-clobber) überschreibt **niemals**
 - ▶ '-backup' erstellt Backup vor Überschreiben
- ▶ 'cp [path1] [path2]'
 - ▶ kopiert Datei von [path1] zu [path2]
 - ▶ Parameter:
 - ▶ selben wie 'mv'
 - ▶ '-r' (recursive) ermöglicht verschieben von Ordnern
 - ▶ '-s' (symbolic-link) erstellt symbolische Links, statt Dateien

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht
 - ▶ '-i' Nachfragen bei jeder Löschung

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht
 - ▶ '-i' Nachfragen bei jeder Löschung
 - ▶ '-r' Löschung rekursiv → (Ordner und Unterordner)

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht
 - ▶ '-i' Nachfragen bei jeder Löschung
 - ▶ '-r' Löschung rekursiv → (Ordner und Unterordner)
 - ▶ '-d' leere Verzeichnisse löschen

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht
 - ▶ '-i' Nachfragen bei jeder Löschung
 - ▶ '-r' Löschung rekursiv → (Ordner und Unterordner)
 - ▶ '-d' leere Verzeichnisse löschen
- ▶ 'rm' löscht nicht entgültig, Dateien sind nicht unleserlich

remove

Flutsch! Und weg!

Immer wieder im Leben kommt eine Zeit in der man etwas Ballast abwerfen möchte. 'rm' hilft dabei

- ▶ Syntax: 'rm [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) es wird nicht nachgefragt und einfach alles gelöscht
 - ▶ '-i' Nachfragen bei jeder Löschung
 - ▶ '-r' Löschung rekursiv → (Ordner und Unterordner)
 - ▶ '-d' leere Verzeichnisse löschen
- ▶ 'rm' löscht nicht entgültig, Dateien sind nicht unleserlich
- ▶ soll entgültig gelöscht werden, dann benutze 'shred' (später)

shred

RRRRRRRRRRRRRTTZZZZZZZZZZZZ!

'shred' überschreibt Dateien und löscht sie damit endgültig

- ▶ Syntax: 'shred [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) Versuch Schreibrechte zu bekommen, falls benötigt

- ▶ Bsp.: 'shred -u test.txt'

Achtung!! kein Löschen von Backups, evtl. Wiederherstellung möglich

shred

RRRRRRRRRRRRRTTZZZZZZZZZZZZ!

'shred' überschreibt Dateien und löscht sie damit endgültig

- ▶ Syntax: 'shred [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) Versuch Schreibrechte zu bekommen, falls benötigt
 - ▶ '-n [num]' Anzahl [num] der Schreibvorgänge pro Datei
- ▶ Bsp.: 'shred -u test.txt'

Achtung!! kein Löschen von Backups, evtl. Wiederherstellung möglich

shred

RRRRRRRRRRRRRTTZZZZZZZZZZZ!

'shred' überschreibt Dateien und löscht sie damit endgültig

- ▶ Syntax: 'shred [option] [file]'
- ▶ Parameter:
 - ▶ '-f' (force) Versuch Schreibrechte zu bekommen, falls benötigt
 - ▶ '-n [num]' Anzahl [num] der Schreibvorgänge pro Datei
 - ▶ '-u' Löschen nach dem Überschreiben
- ▶ Bsp.: 'shred -u test.txt'

Achtung!! kein Löschen von Backups, evtl. Wiederherstellung möglich

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv
 - ▶ '-x' (extract) extrahieren einer Datei

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv
 - ▶ '-x' (extract) extrahieren einer Datei
 - ▶ '-v' (verbose) Fortschritt auflisten

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv
 - ▶ '-x' (extract) extrahieren einer Datei
 - ▶ '-v' (verbose) Fortschritt auflisten
 - ▶ '-z' (gzip format) komprimieren als .gz, etc.

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv
 - ▶ '-x' (extract) extrahieren einer Datei
 - ▶ '-v' (verbose) Fortschritt auflisten
 - ▶ '-z' (gzip format) komprimieren als .gz, etc.
 - ▶ '-f' erzeugt beim Entpacken einen Ordner mit Namen von Archiv

tape archiver

Archivieren

Dateien mit .tar, .tar.gz, .zip, etc. können natürlich auch via der Oberfläche entpackt werden, allerdings kann es auch sein, dass man dies über die Konsole vornehmen muss.

- ▶ Syntax entpacken: 'tar [option] [path]'
- ▶ Syntax verpacken: 'tar [option] [path-archiv] [path-files]'
- ▶ Parameter:
 - ▶ '-c' (create) erzeugt neues Archiv
 - ▶ '-x' (extract) extrahieren einer Datei
 - ▶ '-v' (verbose) Fortschritt auflisten
 - ▶ '-z' (gzip format) komprimieren als .gz, etc.
 - ▶ '-f' erzeugt beim Entpacken einen Ordner mit Namen von Archiv
- ▶ Bsp.: 'tar -xvf test.tar'

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
- ▶ 'htop'
- ▶ Ansicht:

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'

- ▶ Ansicht:

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
- ▶ Ansicht:

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers
 - ▶ Anteil vom Swap und bisherige Laufzeit des Systems

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers
 - ▶ Anteil vom Swap und bisherige Laufzeit des Systems
 - ▶ PID (Prozess ID), User, PRI (Priorität), NI (Nice-Wert)

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers
 - ▶ Anteil vom Swap und bisherige Laufzeit des Systems
 - ▶ PID (Prozess ID), User, PRI (Priorität), NI (Nice-Wert)
 - ▶ VIRT und RES (benutzer virtueller und physischer Speicher)

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers
 - ▶ Anteil vom Swap und bisherige Laufzeit des Systems
 - ▶ PID (Prozess ID), User, PRI (Priorität), NI (Nice-Wert)
 - ▶ VIRT und RES (benutzer virtueller und physischer Speicher)
 - ▶ S (State; $S \in \{ S[\text{leeping}], R[\text{unning}], D[\text{isk sleep}], Z[\text{ombie}], T[\text{raced/suspended}], W[\text{Paging}] \}$)

htop - interaktiver Prozessmanager

- ▶ 'top'
 - ▶ ist bereits installiert
 - ▶ '-h' um Befehle anzeigen zu lassen
- ▶ 'htop'
 - ▶ muss noch installiert werden
 - ▶ leichtere Bedienung und somit verwaltung von Prozessen
- ▶ Ansicht:
 - ▶ Auslastung der einzelnen CPU-Kerne und des Arbeitsspeichers
 - ▶ Anteil vom Swap und bisherige Laufzeit des Systems
 - ▶ PID (Prozess ID), User, PRI (Priorität), NI (Nice-Wert)
 - ▶ VIRT und RES (benutzer virtueller und physischer Speicher)
 - ▶ S (State; $S \in \{ S[\text{leeping}], R[\text{unning}], D[\text{isk sleep}], Z[\text{ombie}], T[\text{raced/suspended}], W[\text{Paging}] \}$)
 - ▶ CPU% und MEM% (genutzter CPU und Arbeitsspeicher Anteil) und TIME (bisherige Laufzeit)

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:
 - ▶ '-n (-adjustment=N)' aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:
 - ▶ '-n (-adjustment=N)' aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)
- ▶ Bsp. 'ping':

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:
 - ▶ '-n (-adjustment=N)' aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)
- ▶ Bsp. 'ping':
 - ▶ Syntax: 'ping [option] [addr]'

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:
 - ▶ '-n (-adjustment=N)' aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)
- ▶ Bsp. 'ping':
 - ▶ Syntax: 'ping [option] [addr]'
 - ▶ Parameter: '-c [num]' (Anzahl der pings setzen), '-W [time]' (Timeout setzen)

nice - nette Programme

Ausführen von Programmen mit modifizierter scheduling Priorität p , mit $p \in [-20,19]$. Hierbei ist -20 die höchste Priorität

- ▶ normalerweise werden Programme mit Priorität 0 gestartet
- ▶ Syntax: 'nice [option] [command] [arg] ...'
- ▶ Parameter:
 - ▶ '-n (-adjustment=N)' aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)
- ▶ Bsp. 'ping':
 - ▶ Syntax: 'ping [option] [addr]'
 - ▶ Parameter: '-c [num]' (Anzahl der pings setzen), '-W [time]' (Timeout setzen)
 - ▶ 'nice -n 20 ping www.foss-ag.de'

renice und nohup

'renice' ermöglicht nachträgliches Einstellen des Nice-Wertes einen bereits laufenden Prozesses. 'nohup' ermöglicht es ein Programm weiterlaufen zu lassen, auch wenn der Benutzer sich ausloggt.

- ▶ 'renice'
 - ▶ Syntax: 'renice [-n] priority [-g — -p — -u] identifier'
- ▶ 'nohup'
 - ▶ Syntax: 'nohup [commando] [args]'
 - ▶ Ausgabe wird dabei in eine nohup.txt Datei im Home-Directory geschrieben (umlenken möglich)

renice und nohup

'renice' ermöglicht nachträgliches Einstellen des Nice-Wertes einen bereits laufenden Prozesses. 'nohup' ermöglicht es ein Programm weiterlaufen zu lassen, auch wenn der Benutzer sich ausloggt.

- ▶ 'renice'
 - ▶ Syntax: 'renice [-n] priority [-g — -p — -u] identifier'
 - ▶ '-n [num]' ist neuer Nice-Wert
- ▶ 'nohup'
 - ▶ Syntax: 'nohup [commando] [args]'
 - ▶ Ausgabe wird dabei in eine nohup.txt Datei im Home-Directory geschrieben (umlenken möglich)

renice und nohup

'renice' ermöglicht nachträgliches Einstellen des Nice-Wertes einen bereits laufenden Prozesses. 'nohup' ermöglicht es ein Programm weiterlaufen zu lassen, auch wenn der Benutzer sich ausloggt.

- ▶ 'renice'
 - ▶ Syntax: 'renice [-n] priority [-g — -p — -u] identifier'
 - ▶ '-n [num]' ist neuer Nice-Wert
 - ▶ '-g [group-id]', '-p [process-id]', '-u [user-id]'
- ▶ 'nohup'
 - ▶ Syntax: 'nohup [commando] [args]'
 - ▶ Ausgabe wird dabei in eine nohup.txt Datei im Home-Directory geschrieben (umlenken möglich)

& - Backstage

bei Angabe dieses Operators am Ende eines Prozess-Aufrufs, wird der Prozess in den Hintergrund verschoben

- ▶ Prozess blockiert Shell nicht, sodass weiteres Arbeiten möglich

[Achtung!!] `&` \equiv Prozess in Hintergrund; `&&` \equiv Prozesse hintereinander ausführen (`p1 && p2` : `p2` wird ausgeführt, wenn `p1` erfolgreich)

& - Backstage

bei Angabe dieses Operators am Ende eines Prozess-Aufrufs, wird der Prozess in den Hintergrund verschoben

- ▶ Prozess blockiert Shell nicht, sodass weiteres Arbeiten möglich
- ▶ Ausgabe wird aber weiterhin auf Shell geschrieben

[Achtung!!] `&` \equiv Prozess in Hintergrund; `&&` \equiv Prozesse hintereinander ausführen (`p1 && p2` : `p2` wird ausgeführt, wenn `p1` erfolgreich)

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
- ▶ 'nano' (release 1999):
- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
 - ▶ reine Textverarbeitung, mehrere Dokumente möglich, Syntaxhighlighting

- ▶ 'nano' (release 1999):

- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
 - ▶ reine Textverarbeitung, mehrere Dokumente möglich, Syntaxhighlighting
 - ▶ kein Highlighting von einem Wort, kein Text-Splitscreen, keine RegEx suche
 - ▶ 'nano' (release 1999):
-
- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
 - ▶ reine Textverarbeitung, mehrere Dokumente möglich, Syntaxhighlighting
 - ▶ kein Highlighting von einem Wort, kein Text-Splitscreen, keine RegEx suche
 - ▶ 'nano' (release 1999):
 - ▶ ebenfalls sehr minimalistisch
-
- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
 - ▶ reine Textverarbeitung, mehrere Dokumente möglich, Syntaxhighlighting
 - ▶ kein Highlighting von einem Wort, kein Text-Splitscreen, keine RegEx suche
- ▶ 'nano' (release 1999):
 - ▶ ebenfalls sehr minimalistisch
 - ▶ Nachfolger von pico
- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 1: ed, nano, pico

You would (not) love Ed!!

Das Bearbeiten von Textdateien ist in der Konsole schwierig.
Ein Beispiel ist 'ed' (release 1970; Grundlage für 'grep').

- ▶ 'pico' (release 1992):
 - ▶ sehr minimalistischer Editor
 - ▶ reine Textverarbeitung, mehrere Dokumente möglich, Syntaxhighlighting
 - ▶ kein Highlighting von einem Wort, kein Text-Splitscreen, keine RegEx suche
- ▶ 'nano' (release 1999):
 - ▶ ebenfalls sehr minimalistisch
 - ▶ Nachfolger von pico
 - ▶ bietet mehr Funktionsumfang, z.B.: Text-Splitscreen, RegEx Suche
- ▶ 'nano' und 'pico' sind sehr minimalistisch, aber nicht zu unterschätzen

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
- ▶ 'jed' (release 1992):
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
 - ▶ verbesserte Suchfunktionen, mächtiger als nano und pico
- ▶ 'jed' (release 1992):
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
 - ▶ verbesserte Suchfunktionen, mächtiger als nano und pico
 - ▶ Standard bei 'git'
 - ▶ 'jed' (release 1992):
-
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
 - ▶ verbesserte Suchfunktionen, mächtiger als nano und pico
 - ▶ Standard bei 'git'
 - ▶ 'jed' (release 1992):
 - ▶ der Editor für Programmierer
-
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
 - ▶ verbesserte Suchfunktionen, mächtiger als nano und pico
 - ▶ Standard bei 'git'
- ▶ 'jed' (release 1992):
 - ▶ der Editor für Programmierer
 - ▶ liefert Templates für verschiedene Anwendungsfälle
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

Texteditoren - 2: vim, jed

Good thing Ed is not around anymore

Wir haben grad die minimalistischen Editoren betrachtet, nun die etwas mächtigeren.

- ▶ 'vim' (release 1991):
 - ▶ Großmufti unter den Editoren
 - ▶ verbesserte Suchfunktionen, mächtiger als nano und pico
 - ▶ Standard bei 'git'
- ▶ 'jed' (release 1992):
 - ▶ der Editor für Programmierer
 - ▶ liefert Templates für verschiedene Anwendungsfälle
 - ▶ weniger mächtig als vim, aber dennoch viele Features
- ▶ Bei der Wahl des Editors ist es wie immer im Leben, Geschmäcker sind verschieden

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:
 - ▶ '-P' (poweroff)

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:
 - ▶ '-P' (poweroff)
 - ▶ '-r' (reboot)

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:
 - ▶ '-P' (poweroff)
 - ▶ '-r' (reboot)
 - ▶ '-h' (halt) kein Timer möglich

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:
 - ▶ '-P' (poweroff)
 - ▶ '-r' (reboot)
 - ▶ '-h' (halt) kein Timer möglich
- ▶ 'sudo reboot' anstelle von 'sudo shutdown -r'

shutdown

Wenn man mal das real life genießen will

ermöglicht Ausschalten oder Neustart des Systems [nur Root]

- ▶ Syntax: 'shutdown [options] [time]'
- ▶ Parameter:
 - ▶ '-P' (poweroff)
 - ▶ '-r' (reboot)
 - ▶ '-h' (halt) kein Timer möglich
- ▶ 'sudo reboot' anstelle von 'sudo shutdown -r'
- ▶ bei 'reboot' kein Timer möglich

ip addr

Meist wurde 'ifconfig' benutzt um seine lokalen IP-Adressen des Rechners anzuzeigen. Dieser Befehl ist allerdings ein wenig obsolet und wurde ersetzt durch 'ip addr'.

- ▶ Um lokale IP-Adresse zu erfahren, ohne über Router zu gehen
- ▶ Wird benötigt, wenn man sich via 'ssh' einloggen will
- ▶ leider etwas unübersichtlich
- ▶ Parameter:
 - ▶ '-4' Anzeigen der IPv4-Adressen
 - ▶ '-6' Anzeigen der IPv6-Adressen

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph
 - ▶ 'n' Auslastung der Netzwerk Schnittstelle

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph
 - ▶ 'n' Auslastung der Netzwerk Schnittstelle
 - ▶ 'j' Auslastung des Dateisystems (Platz und Aufbau)

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph
 - ▶ 'n' Auslastung der Netzwerk Schnittstelle
 - ▶ 'j' Auslastung des Dateisystems (Platz und Aufbau)
 - ▶ 't' wie top, Auflistung aller laufenden Prozesse

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph
 - ▶ 'n' Auslastung der Netzwerk Schnittstelle
 - ▶ 'j' Auslastung des Dateisystems (Platz und Aufbau)
 - ▶ 't' wie top, Auflistung aller laufenden Prozesse
 - ▶ '+' bzw. '-' Refresh erhöhen bzw. verringern

nmon

'nmon' dient zur Überwachung der Auslastung der PC Komponenten

- ▶ Elemente können der Anzeige bei Bedarf hinzugefügt und entfernt werden
- ▶ Shortcuts:
 - ▶ 'c' zeigt die Auslastung aller CPU Kerne an
 - ▶ 'm' zeigt die Auslastung des Arbeitsspeichers an (auch mit SWAP)
 - ▶ 'd' zeigt die Auslastung der Festplatte an
 - ▶ 'k' zeigt Kernel Statistiken an (Queue, Forks, Interrupts)
 - ▶ 'r' Details zum System und Prozessor
 - ▶ 'l' CPU Auslastung als Graph
 - ▶ 'n' Auslastung der Netzwerk Schnittstelle
 - ▶ 'j' Auslastung des Dateisystems (Platz und Aufbau)
 - ▶ 't' wie top, Auflistung aller laufenden Prozesse
 - ▶ '+' bzw. '-' Refresh erhöhen bzw. verringern
 - ▶ 'H' für Hilfe (Shortcuts auflisten)

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ 'c' ein konstantes Zeichen c

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ 'c' ein konstantes Zeichen c
 - ▶ '.' genau **ein** beliebiges Zeichen

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ 'c' ein konstantes Zeichen c
 - ▶ '.' genau **ein** beliebiges Zeichen
 - ▶ '.*' Folge von beliebigen Zeichen (auch keines)

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ `'c'` ein konstantes Zeichen `c`
 - ▶ `'.'` genau **ein** beliebiges Zeichen
 - ▶ `'*'` Folge von beliebigen Zeichen (auch keines)
 - ▶ `'*'` Zeichen vor `*` ist beliebig oft (auch gar nicht)

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ `'c'` ein konstantes Zeichen `c`
 - ▶ `'.'` genau **ein** beliebiges Zeichen
 - ▶ `'.*'` Folge von beliebigen Zeichen (auch keines)
 - ▶ `'*'` Zeichen vor `*` ist beliebig oft (auch gar nicht)
 - ▶ `'?'` Zeichen vor `?` ist **genau** einmal oder gar nicht

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

- ▶ Syntax - 1:
 - ▶ 'c' ein konstantes Zeichen c
 - ▶ '.' genau **ein** beliebiges Zeichen
 - ▶ '.*' Folge von beliebigen Zeichen (auch keines)
 - ▶ '*' Zeichen vor * ist beliebig oft (auch gar nicht)
 - ▶ '?' Zeichen vor ? ist **genau** einmal oder gar nicht
 - ▶ '+' Zeichen vor + ist min. einmal

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

► Syntax - 1:

- 'c' ein konstantes Zeichen c
- '.' genau **ein** beliebiges Zeichen
- '.*' Folge von beliebigen Zeichen (auch keines)
- '*' Zeichen vor * ist beliebig oft (auch gar nicht)
- '?' Zeichen vor ? ist **genau** einmal oder gar nicht
- '+' Zeichen vor + ist min. einmal
- '{n,m}' Zeichen wird n bis m mal wiederholt

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

► Syntax - 1:

- 'c' ein konstantes Zeichen c
- '.' genau **ein** beliebiges Zeichen
- '*' Folge von beliebigen Zeichen (auch keines)
- '*' Zeichen vor * ist beliebig oft (auch gar nicht)
- '?' Zeichen vor ? ist **genau** einmal oder gar nicht
- '+' Zeichen vor + ist min. einmal
- '{n,m}' Zeichen wird n bis m mal wiederholt
- '{n}' Zeichen genau n mal

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

► Syntax - 1:

- 'c' ein konstantes Zeichen c
- '.' genau **ein** beliebiges Zeichen
- '*' Folge von beliebigen Zeichen (auch keines)
- '*' Zeichen vor * ist beliebig oft (auch gar nicht)
- '?' Zeichen vor ? ist **genau** einmal oder gar nicht
- '+' Zeichen vor + ist min. einmal
- '{n,m}' Zeichen wird n bis m mal wiederholt
- '{n}' Zeichen genau n mal
- '[...]' Zeichen kommt in den Klammern vor

Regular Expression - 1

The art of black magic

Reguläre Ausdrücke sind Muster, eine Grundstruktur von gesuchten Begriffen angeben. Um eine Fehlinterpretation der Shell zu vermeiden, sollte ein RegEx immer in Anführungszeichen angegeben werden.

► Syntax - 1:

- 'c' ein konstantes Zeichen c
- '.' genau **ein** beliebiges Zeichen
- '*' Folge von beliebigen Zeichen (auch keines)
- '*' Zeichen vor * ist beliebig oft (auch gar nicht)
- '?' Zeichen vor ? ist **genau** einmal oder gar nicht
- '+' Zeichen vor + ist min. einmal
- '{n,m}' Zeichen wird n bis m mal wiederholt
- '{n}' Zeichen genau n mal
- '['...]' Zeichen kommt in den Klammern vor
- '['...-...]' Zeichen aus Zeichenklasse

Regular Expression - 2

The art of black magic

- ▶ Syntax - 2:
 - ▶ '['^...]' und '['^...-...]' verneinte Zeichenklasse
 - ▶ '^' Zeilenanfang
 - ▶ '\$' Zeilenende
 - ▶ '<' Wortanfang
 - ▶ '>' Wortende
 - ▶ 'a1|a2' alternative Ausdrücke; a1 oder a2
 - ▶ '(...)' Gruppe von Ausdrücken
 - ▶ '\' Sonderbedeutung von nächstem Symbol ignorieren.

Bsp.: —

- ▶ Bsp.: `'.*python[1-99]*.*\ |.*\.pyc?'`
- ▶ es werden alle Wörter/Dateien gesucht, die:
 - ▶ 'python' im Namen haben oder mit .py bzw. .pyc aufhören
 - ▶ dürfen mit Zahlen zwischen 1 und 99 enden
 - ▶ min. ein p am Anfang haben

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen
 - ▶ '-x' gibt nur Zeilen aus, die ganze dem RegEx entsprechen

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen
 - ▶ '-x' gibt nur Zeilen aus, die ganze dem RegEx entsprechen
- ▶ Ausgabe:

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen
 - ▶ '-x' gibt nur Zeilen aus, die ganze dem RegEx entsprechen
- ▶ Ausgabe:
 - ▶ '-c' gibt Anzahl der passenden Zeilen aus

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen
 - ▶ '-x' gibt nur Zeilen aus, die ganze dem RegEx entsprechen
- ▶ Ausgabe:
 - ▶ '-c' gibt Anzahl der passenden Zeilen aus
 - ▶ '-o' gibt nur gematchten Teile einer Zeile aus

grep

Gonna catch em' Strings

Durchsucht eine Eingabe nach einem RegEx.

- ▶ Selektoren:
 - ▶ '-E' Interpretiert Suchwort als erweiterten RegEx
 - ▶ '-F' Interpretiert Suchwort als fixed String
 - ▶ '-G' Interpretiert Suchwort als basic RegEx
- ▶ Kontrolle:
 - ▶ '-f [file]' liest RegEx aus [file]
 - ▶ '-v' invertiert das Matchmaking (alle Zeilen ohne RegEx)
 - ▶ '-w' gibt nur Zeilen aus, in denen ganze Worte dem RegEx entsprechen
 - ▶ '-x' gibt nur Zeilen aus, die ganze dem RegEx entsprechen
- ▶ Ausgabe:
 - ▶ '-c' gibt Anzahl der passenden Zeilen aus
 - ▶ '-o' gibt nur gematchten Teile einer Zeile aus
 - ▶ '-q' keine Ausgabe, gibt Exitcode 0 aus, wenn erfolgreich