Linux Advanced

Das Terminal

Josef, Jan

01.07.2017

Free and Open Source Software AG

Arbeiten mit Ordnern, Dateien und Archiven

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

`chmod` ändert die Zugriffsrechte von Dateien und Ordnern

- Syntax: `chmod [mode] [path]`
- Beispiel (`/home/[username]/neuer-ordner/`):
  - aktuelle Rechte: `u=rwx g=r-x o=r-x`
  - `chmod g=rwx ~/neuer-ordner` (wir geben der Gruppe Schreib-Recht)
  - `chmod o-r ~/neuer-ordner` (wir nehmen allen anderen das Lese-Recht)
  - neue Rechte: `u=rwx g=rwx o=--x`
  - es gibt auch `a` als Argument, was für all steht
  - Bsp. Schreib-Recht von `group` und `others` nehmen: chmod g-x,o-x [path]

change owner

Ändern des Besitzers von Ordern 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

head & tail

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 (Default: -q):
  - `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

more or less

- kein Editor
- Navigation
- less ist mehr als more

vim

- default Editor
- `:q` - Beenden
- `ESC` - Modus beenden
- `i` - insert

Sortieren

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

- `sort [option] [File]` (liest `File` und sortiert Inhalt)
  - `-g` ist Default. generic sort
  - `-r` Ausgabe umdrehen
  - `-c` Check ob Inhalt von Datei sortiert ist
  - Bsp.: `sort -r output.txt`

Nutzer

switch user

- Wechsel des Benutzers innerhalb der Konsole
- Ohne Parameter wird versucht sich als `root` einzuloggen
- Ansonsten `su [user]` einloggen als `[user]`

Prozesse

nice

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

- Default: p = 0; negative Werte nur als Root
- Syntax: `nice [option] [command] [arg] ...`
- Parameter (Default: -n 10):
  - `-n (--adjustment=N)` aufaddieren von Integer N auf aktuellen Nice-Wert (Default: N=10)
- Bsp. `ping`:
  - Syntax: `ping [option] [addr]`
  - Parameter: `-c [num]` (setze Anzahl der Pings), `-W [time]` (setze Timeout)
  - `nice -n 10 ping foss-ag.de`

renice und nohup

`renice` ermöglicht nachträgliches Einstellen des Nice-Wertes eines 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)

Zusatz

ip

Meist wurde `ifconfig` benutzt um sich seine lokalen IP-Adressen des Rechners anzeigen zu lassen. Dieser Befehl ist allerdings ein wenig obsolet und wurde ersetzt durch `ip addr`.

nmon

`nmon` dient zur Überwachung der Auslastung der PC Komponenten

- Shortcuts:
  - `c`, `m`, `d` zeigen Auslastung von CPU-Kerne, Arbeitsspeicher, 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

Reguläre Ausdrücke sind Muster, die eine Grundstruktur von gesuchten Begriffen angeben. Um Fehlinterpretationen zu vermeiden, sollte der 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

- Syntax - 2:
  - `{n}` Zeichen genau `n` mal
  - `[...]` Zeichen kommt in den Klammern vor
  - `[...-...]` Zeichen aus Zeichenklasse
  - `[^...]` und `[^...-...]` verneinte Zeichenklasse
  - `^` Zeilenanfang
  - `$` Zeilenende
  - `<` Wortanfang
  - `>` Wortende
  - `a1|a2` alternative Ausdrücke; a1 oder a2
  - `(...)` Gruppe von Ausdrücken
  - `\` Sonderbedeutung von nächstem Symbol ignorieren. Bsp.: `|`

Regular Expression

- Bsp.: `.*p+ython[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
- Achtung!! bei find muss RegEx dem Pfad entsprechen

Alias

Im System kann für einen Befehl ein Alias definiert werden. Die Datei `.bashrc` wird beim Einloggen des Users geladen. Vorsicht beim Bearbeiten!!! Bsp.: ll anstatt von 6ls -l

- editiere die Datei /.bashrc
- füge die Zeile `alias ll='ls -l'` hinzu
- lade Datei neu: `source /.bashrc` bzw. neu einloggen
- Tadaa!!!!