

GNU/Linux – Grundlagen

Kommandozeile

Jakob Franzel, 3AHITS, 03/10/2022

Inhalt

1.Vorbereitung	3
2.Die Kommandozeile	3
2.1.Anzeigen des Verzeichnisinhalts	3
2.2.Der Verzeichnisbaum	4
2.3.Wechseln in ein Verzeichnis	4
2.4.Verzeichniskürzel	4
2.5.Absolute und relative Pfade	5
3.Textdateien	6
3.1.Der Texteditor „nano“	6
3.2.Den Inhalt einer Datei ausgeben	6
4.Posix Rechte und Ausgabe von „ls -l“	7
4.1.Ausgabe von „ls -l“	7
4.2.Besitzer und Gruppe ändern	7
4.3.Posix Rechte	8
5.Händisch einen User anlegen.....	9
5.1.Die Datei „/etc/passwd“	9
5.2.Die Datei „/etc/group“	10
5.3.Ein Passwort setzen.....	10
5.4.Erstellen des Homeverzeichnisses	10
5.5.Den neuen Benutzer verwenden	11

1. Vorbereitung

- Logge dich in deiner virtuellen Maschine per SSH als Benutzer „root“ ein!

2. Die Kommandozeile

2.1. Anzeigen des Verzeichnisinhalts

Das Programm „**ls**“ zeigt den Inhalt des aktuellen Verzeichnisses an.



Führe das Programm „**ls**“ aus!

```
root@debian-client-jakob:~# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

Das Programm „**ls**“ [*list*] hat viele Optionen, die beim Programmaufruf einfach hinter dem Befehl „**ls**“ angegeben werden können. Hier die wichtigsten:

- „**-l**“: [*list*] Die Ausgabe erfolgt in Listenform. So werden wesentlich mehr Informationen über vorhandene Dateien ausgegeben, als in der normalen Ausgabe.
- „**-a**“: [*all*] Zeigt auch versteckte Dateien an.
- „**-h**“: [*human readable*] Gibt Dateigrößen nicht in Bytes, sondern mit den üblichen Multiplikatoren (Kilo, Mega, Giga,...) an.

Diese Optionen lassen sich auch kombinieren. So gibt zum Beispiel der Befehl „**ls -lh**“ den Inhalt des aktuellen Verzeichnisses in Listenform mit den Dateigrößen in menschenlesbarer Form aus.



Führe das Programm „**ls**“ nochmals aus, diesmal aber mit den Optionen „**-l**“, „**-a**“ und „**-h**“!

```
root@debian-client-jakob:~# ls -lah
total 116K
drwx----- 15 root root 4.0K Oct  3 08:58 .
drwxr-xr-x 18 root root 4.0K Sep 19 10:57 ..
-rw-----  1 root root  26K Oct  3 08:29 .bash_history
```

```
-rw-r--r--  1 root root  571 Apr 10  2021 .bashrc
drwx-----  4 root root 4.0K Oct  3 08:26 .cache
drwxr-xr-x  5 root root 4.0K Oct  3 08:26 .config
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Desktop
-rw-r--r--  1 root root   35 Oct  3 08:26 .dmrc
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Documents
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Downloads
drwx-----  3 root root 4.0K Oct  3 08:26 .gnupg
-rw-----  1 root root    0 Oct  3 08:26 .ICEauthority
drwxr-xr-x  3 root root 4.0K Sep 26 08:51 .local
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Music
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Pictures
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Public
drwx-----  2 root root 4.0K Sep 26 09:10 .ssh
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Templates
drwxr-xr-x  2 root root 4.0K Oct  3 08:26 Videos
-rw-----  1 root root   64 Oct  3 08:58 .Xauthority
-rw-----  1 root root 2.7K Oct  3 08:58 .xsession-errors
-rw-----  1 root root 4.2K Oct  3 08:42 .xsession-errors.old
```

Später wirst du noch herausfinden, was diese Ausgabe genau bedeutet.

2.2. Der Verzeichnisbaum

Bei allen GNU/Linux Varianten sieht der Verzeichnisbaum annähernd gleich aus. Er beginnt immer mit „/“. Die wichtigsten Verzeichnisse in der ersten Ebene darunter sind:

- **„etc“**: [*et cetera*] Hier werden Konfigurationsdateien für installierte Dienste gespeichert. Anders als bei Windows werden das System und Dienste nicht über eine Registry gesteuert, sondern meist über Textdateien – sogenannte Konfigurationsdateien
- **„home“**: In diesem Verzeichnis hat jeder Benutzer ein eigenes Verzeichnis für Dateien, die ihm gehören. Man nennt das ein Homeverzeichnis.
- **„var“**: [*variable data*] Hier werden Dateien gespeichert, die sich häufig verändern. Zum Beispiel werden hier Log-Dateien abgelegt, aber auch HTML-Dateien, die von einem Webserver verwendet werden.
- **„dev“**: [*devices*] Hardwaregeräte sind in GNU/Linux als Datei ansprechbar. Diese Dateien liegen im Verzeichnis „dev“. So kann man zum Beispiel einen Ton ausgeben, indem man einfach etwas in die entsprechende Datei in „/dev/snd/“ schreibt.
- **„bin“**: [*binaries*] Hier sind Programme, also ausführbare Dateien abgelegt.
- **„sbin“**: [*system binaries*] Hier sind Systemprogramme gespeichert.

Diese Liste ist nicht vollständig, aber in Kürze wirst du mehr herausfinden!

2.3. Wechseln in ein Verzeichnis

Das Verzeichnis wechselt man mit dem Befehl **„cd“** [*change directory*]. Hinter dem Befehl wird ganz einfach das Verzeichnis angegeben, in das man springen möchte.



Wechsle in das Verzeichnis „/“ und lass dir den Inhalt in Listenform anzeigen!

```
root@debian-client-jakob:/# ls -l

total 60

lrwxrwxrwx   1 root root    7 Sep 19 10:47 bin -> usr/bin
drwxr-xr-x   4 root root 4096 Sep 26 10:49 boot
drwxr-xr-x  17 root root 3220 Oct  3 08:26 dev
```

```
drwxr-xr-x 92 root root 4096 Oct  3 08:34 etc
drwxr-xr-x  2 root root 4096 Mar 19  2022 home
lrwxrwxrwx  1 root root    31 Sep 19 10:57 initrd.img -> boot/initrd.img-5.10.0-18-amd64
lrwxrwxrwx  1 root root    31 Sep 19 10:48 initrd.img.old -> boot/initrd.img-5.10.0-13-amd64
lrwxrwxrwx  1 root root    7 Sep 19 10:47 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Sep 19 10:47 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Sep 19 10:47 lib64 -> usr/lib64
lrwxrwxrwx  1 root root   10 Sep 19 10:47 libx32 -> usr/libx32
drwx----- 2 root root 16384 Sep 19 10:47 lost+found
drwxr-xr-x  3 root root 4096 Sep 19 10:47 media
drwxr-xr-x  2 root root 4096 Sep 19 10:47 mnt
drwxr-xr-x  2 root root 4096 Sep 19 10:47 opt
dr-xr-xr-x 195 root root    0 Oct  3 08:26 proc
drwx----- 15 root root 4096 Oct  3 08:58 root
drwxr-xr-x 21 root root  620 Oct  3 08:57 run
lrwxrwxrwx  1 root root    8 Sep 19 10:47 sbin -> usr/sbin
drwxr-xr-x  2 root root 4096 Sep 19 10:47 srv
dr-xr-xr-x 13 root root    0 Oct  3 08:26 sys
drwxrwxrwt 12 root root 4096 Oct  3 08:58 tmp
drwxr-xr-x 14 root root 4096 Sep 19 10:47 usr
drwxr-xr-x 11 root root 4096 Sep 19 10:47 var
```

```
lrwxrwxrwx  1 root root    28 Sep 19 10:57 vmlinuz -> boot/vmlinuz-5.10.0-18-amd64

lrwxrwxrwx  1 root root    28 Sep 19 10:48 vmlinuz.old -> boot/vmlinuz-5.10.0-13-amd64
```

Hier hast du also die vollständige Liste aus Punkt 1.2!



Wechsle in das Verzeichnis „**var**“ und lass dir den Inhalt anzeigen!

```
root@debian-client-jakob:/var# ls

backups  cache  lib    local  lock   log    mail   opt    run    spool  tmp
```

2.4. Verzeichniskürzel

- „~“: Die Tilde ist ein Kürzel für das Homeverzeichnis des aktuellen Benutzers. Wenn mein Benutzername z.B. „**user1**“ lautet, dann steht die Tilde in meinem Fall für „**/home/user1**“.
- „.“: Der Punkt steht für das Verzeichnis, in dem man gerade steht.
- „..“: „Punkt Punkt“ steht für das Verzeichnis, das eine Ebene über dem aktuellen Verzeichnis im Verzeichnisbaum ist. Steht man also gerade in „**/home/user1**“, dann steht „..“ für „**/home**“.

2.5. Absolute und relative Pfade

- Ein absoluter Pfad beginnt in GNU/Linux immer mit einem „/“. Er gibt nämlich den Weg vom Wurzelverzeichnis bis zur Datei hin an – und natürlich den Dateinamen selbst auch. Wenn also der Benutzer „**user1**“ in seinem Homeverzeichnis eine Datei „**test1**“ liegen hat, dann lautet der absolute Pfad zu dieser Datei „**/home/user1/test1**“.
- Ein relativer Pfad beginnt nie mit einem „/“. Er gibt den Weg zur gewünschten Datei ausgehend von dem Verzeichnis an, in dem man sich gerade befindet. Steht man also gerade zum Beispiel im Verzeichnis „**/home**“, dann lautet der relative Pfad zur Datei aus dem obigen Beispiel „**user1/test1**“.

Jetzt wäre doch auch ein Befehl interessant, der ausgibt, in welchem Verzeichnis man sich gerade befindet. Diesen Befehl gibt es natürlich, er lautet „**pwd**“ [*print working directory*].



Du solltest dich gerade im Verzeichnis „/var“ befinden. Überprüfe das mit dem Befehl „pwd“!

```
root@debian-client-jakob:/var# pwd  
  
/var
```



Mit welchem Befehl kannst du nun unter Verwendung des Verzeichniskürzels für dein Homeverzeichnis in dein Homeverzeichnis wechseln?

```
root@debian-client-jakob:/var# cd ~  
  
root@debian-client-jakob:~#
```



Mit welchem Befehl kannst du jetzt unter Verwendung eines relativen Pfades in das Verzeichnis „/etc“ wechseln?

```
root@debian-client-jakob:~# cd ../etc  
  
root@debian-client-jakob:/etc#
```



Mit welchem Befehl kannst du jetzt wieder in dein Homeverzeichnis wechseln? Verwende diesmal einen absoluten Pfad – und kein Verzeichniskürzel!

```
root@debian-client-jakob:/etc# cd /home  
  
root@debian-client-jakob:/home#
```


3. Textdateien

3.1. Der Texteditor „nano“

Um eine Textdatei zu bearbeiten benötigst du einen Texteditor. Ein beliebiger Texteditor auf der Kommandozeile ist das Programm „**nano**“. Wenn du einfach nur das Programm „**nano**“ aufrufst, öffnet der Texteditor eine neue leere Datei. Wenn du hinter „**nano**“ einen Dateinamen angibst, also „**nano** [dateiname]“, dann öffnet nano diese Datei.



Öffne mit dem Texteditor „**nano**“ die Datei „**neueDatei**“. Welchen Befehl verwendest du dazu?

```
root@debian-client-jakob:/home# nano neueDatei
```

Der Texteditor ist jetzt geöffnet. Du kannst jetzt in die Datei Text hineinschreiben. Verwende die Pfeiltasten, um dich im Text zu bewegen. Am unteren Bildschirmrand sind die Tastenkombinationen angeführt, mit denen „**nano**“ gesteuert wird. Die wichtigsten drei sind „**Strg**“ + „**o**“ [*write out*] zum Speichern, „**Strg**“ + „**x**“ [*exit*] zum Beenden und „**Strg**“ + „**w**“ [*where is*] zum Suchen.



Schreibe Text in die soeben geöffnete Datei, speichere sie und beende dann den Texteditor.

3.2. Den Inhalt einer Datei ausgeben

Um den Inhalt einer Datei auf der Konsole auszugeben gibt es mehrere Möglichkeiten. Die einfachste ist der Befehl „**cat** [dateiname]“. Dabei wird einfach der gesamte Inhalt der Textdatei auf die Konsole geschrieben.



Gib den Inhalt der Datei „**neueDatei**“ auf der Konsole aus!

```
root@debian-client-jakob:/home# cat neueDatei
```

```
Hallo?
```

Man kann bei dieser Art der Dateiausgabe auf einem rein textbasierendem System allerdings nicht scrollen, also sich nicht nach oben oder nach unten im Text bewegen. Das ist nicht immer praktisch. Deswegen gibt es auch das Programm „`less [dateiname]`“. Mit „`less`“ kannst du in der Textdatei mit den Pfeiltasten oder mit den Bild-Tasten nach oben und nach unten scrollen. Beendet wird „`less`“ mit der Taste „`q`“.



Öffne die Datei „`/var/log/syslog`“ mit dem Programm „`less`“!

4. Posix Rechte und Ausgabe von „`ls -l`“

4.1. Ausgabe von „`ls -l`“

Wenn du in deinem Homeverzeichnis den Befehl „`ls -l`“ ausführst, sollte folgende Zeile darin vorkommen:

```

1 2      3      4      5      6      7      8      9      10
- - - - - - - - - -
-rw-r--r-- 1 root root 21 Mar 29 19:33 neueDatei

```

- 1 : Hier wird angegeben, ob es sich bei dem Eintrag in dieser Zeile um ein Verzeichnis oder eine „normale“ Datei handelt. In unserem Fall ist hier einfach eine Textdatei aufgeführt, deswegen steht hier nur ein „-“. Würde es sich um ein Verzeichnis handeln, würde hier ein „d“ stehen.
- 2 : Hier sind die Rechte des Besitzers der Datei aufgeführt.
- 3 : Hier sind die Rechte der Gruppe der Datei aufgeführt.
- 4 : Hier sind die Rechte von allen anderen Benutzern des Systems aufgeführt.
- 5 : Das ist die Anzahl der Links im Dateisystem auf diese Datei.
- 6 : Das ist der Besitzer der Datei.
- 7 : Das ist die Gruppe der Datei.
- 8 : Das ist die Dateigröße, in diesem Fall also 21 Bytes.
- 9 : Zu diesem Zeitpunkt wurde die Datei das letzte Mal geschrieben wurde.
- 10: Das ist der Dateiname.

4.2. Besitzer und Gruppe ändern

In GNU/Linux hat jede Datei, wie du oben schon gesehen hast, einen Besitzer und eine Gruppe. Heutiger Standard ist, dass es für jeden Benutzer auch eine eigene Gruppe gibt, die gleich heißt wie der Benutzer. Jede neue Datei, die ein Benutzer anlegt, hat standardmäßig ihn selbst als Besitzer und die gleichnamige Gruppe als Gruppe.

Um den Besitzer und die Gruppe einer Datei zu ändern gibt es den Befehl „`chown`“ [*change owner*]. Der Aufruf funktioniert so:

```
„chown neuerBesitzer:neueGruppe dateiname“.
```



Mit welchem Befehl änderst du den Besitzer der Datei „neueDatei“ auf „nobody“ und die Gruppe auf „sudo“?

```
root@debian-client-jakob:/home# chown nobody:sudo neueDatei
```



Führe nochmals den Befehl „ls -l“ aus! Du solltest jetzt sehen, dass die Datei „neueDatei“ einen neuen Besitzer und eine neue Gruppe hat!

4.3. Posix Rechte

Es gibt drei Posix Rechte:

- „r“ [read]: Wer dieses Recht besitzt, darf die Datei lesen.
- „w“ [write]: Wer dieses Recht besitzt, darf die Datei schreiben.
- „x“ [execute]: Wer dieses Recht besitzt, darf die Datei ausführen. Das setzt natürlich voraus, dass es sich um eine ausführbare Datei handelt.

Welche der drei Rechte jemand besitzt, lässt sich in einer einzigen Zahl ausdrücken. Hierfür zählt das Recht „r“ 4 Punkte, das Recht „w“ 2 Punkte und das Recht „x“ 1 Punkt. Diese Punkte werden dann einfach addiert. Die möglichen Kombinationen sind also (In Klammer immer die dazugehörige Ausgabe von „ls -l“):

- 0: Gar keine Rechte („---“)
- 1: Nur ausführen. („--x“)
- 2: Nur schreiben. („-w-“)
- 3: Schreiben und ausführen. („-wx“)
- 4: Nur lesen. („r--“)
- 5: Lesen und ausführen. („r-x“)
- 6: Lesen und schreiben. („rw-“)
- 7: Alle Rechte, also lesen, schreiben und ausführen. („rwx“)

Diese Rechte gibt es für jede Datei dreimal. Einmal für den Besitzer, einmal für die Gruppe und einmal für alle anderen Benutzer des Systems.

Im obigen Beispiel waren die Rechte mit „rw-r--r--“ angegeben. Der Besitzer darf die Datei also

lesen und schreiben, die Gruppe die Datei nur lesen und alle anderen auch nur lesen. Ausführen darf die Datei niemand. Die Zahlenschreibweise für die Rechte auf diese Datei wäre also „644“.



Wie lautet die Zahlenschreibweise für die Rechte „**rwxr-xr-x**“?



Für welche Rechte steht die Zahlenschreibweise „511“?

5. Händisch einen User anlegen

5.1. Die Datei „/etc/passwd“

In der Datei „/etc/passwd“ steht, welche Benutzer es auf unseren GNU/Linux System gibt. Für jeden Benutzer gibt es in dieser Datei eine Zeile. Eine Zeile könnte zum Beispiel lauten:

```

1      2      3      4      5      6      7
user1234:x:1001:1001:Vol1ler Name 12,,,:/home/user1234:/bin/bash
```

Wie du siehst, sind die einzelnen Argumente in der Zeile durch Doppelpunkte voneinander getrennt. Die Argumente bedeuten folgendes:

- 1: Das ist der Benutzername.
- 2: Dieses Argument gibt es nur aus historischen Gründen. Hier steht immer ein „x“.
- 3: Jeder Benutzer hat eine im System eindeutige Nummer.
- 4: Das ist die primäre Gruppe des Benutzers. Auch sie hat eine eindeutige Nummer.
- 5: Hier handelt es sich um ein Kommentarfeld. Eingebürgert hat es sich, den vollen Namen des Benutzers hier anzugeben.
- 6: Hier steht, was das Homeverzeichnis des Benutzers ist.
- 7: Dieses Programm wird ausgeführt, wenn sich der Benutzer auf der Kommandozeile einloggt.

Unser neuer Benutzer soll folgende Daten haben:

- Benutzername: „**user1**“
- Eindeutige Nummer: „2000“
- Eindeutige Nummer der Gruppe: „2000“

- Voller Name: „**Mein neuer Benutzer**“
- Homeverzeichnis: „**/home/user1**“
- Loginprogramm: „**/bin/bash**“



Öffne die Datei „**/etc/passwd**“ mit dem Editor „**nano**“! Füge am Ende der Datei eine neue Zeile mit den Daten unseres neuen Benutzers ein! Wie lautet die Zeile?

```
user1:x:2000:2000:Mein neuer Benutzer:/home/user1:/bin/bash
```

5.2. Die Datei „**/etc/group**“

In der Datei „**/etc/group**“ ist gespeichert, welche Gruppen es in unserem GNU/Linux System gibt. Für jede Gruppe gibt es eine eigene Zeile. Eine Zeile könnte zum Beispiel lauten:

```
gruppe1:x:1001:user1234,user1235
```

Wie in der Datei „**/etc/passwd**“ sind die Argumente in einer Zeile durch Doppelpunkte getrennt. Am Anfang steht der Name der Gruppe. Das zweite Feld gibt es wie schon oben nur aus historischen Gründen. Hier steht immer ein „**x**“. Als drittes Argument kommt die im System eindeutige Nummer der Gruppe. Am Schluss steht eine durch Kommas getrennte Liste aller Mitglieder der Gruppe.



Füge am Ende der Datei „**/etc/group**“ eine neue Zeile mit den Daten für die Gruppe unseres neuen Benutzers ein! Wie lautet diese Zeile?

```
2000:x:2000:user1
```

5.3. Ein Passwort setzen

Mit dem Befehl „**passwd BENUTZERNAME**“ kann man einem Benutzer ein neues Passwort setzen. Man wird zweimal aufgefordert, das neue Passwort einzugeben. Achtung! Dabei bleibt die Anzeige am Bildschirm leer, es werden als auch keine „*****“ oder sonstiges angezeigt!



Setze unserem neuen Benutzer das Passwort „**str3ngGeheim!**“!

5.4. Erstellen des Homeverzeichnisses



Wechsle in das Verzeichnis „/home“!

Mit dem Befehl „**mkdir** **VERZEICHNISNAME**“ lässt sich ein Verzeichnis anlegen.



Lege für unseren neuen Benutzer das Homeverzeichnis „**user1**“ an!



Führe den Befehl „**ls -l**“ aus!

```
root@debian-client-jakob:/home# ls -l

total 8

-rw-r--r-- 1 nobody sudo      7 Oct  3 09:06 neueDatei
drwxr-xr-x 2 root    root 4096 Oct  3 09:31 user1
```

Du siehst also, dass der Besitzer den neuen Verzeichnisses „**root**“ und die Gruppe auch „**root**“ ist.



Mit welchem Befehl änderst du den Besitzer des Verzeichnisses „**user1**“ auf „**user1**“ und die Gruppe auch auf „**user1**“?

```
root@debian-client-jakob:/home# chown user1:2000 user1
```

5.5. Den neuen Benutzer verwenden

Du kannst dich jetzt ausloggen und mit dem Benutzernamen und Passwort des neuen Benutzers einloggen. Gratulation, du kennst dich jetzt mit GNU/Linux schon gut aus!