

Aufgabe 07

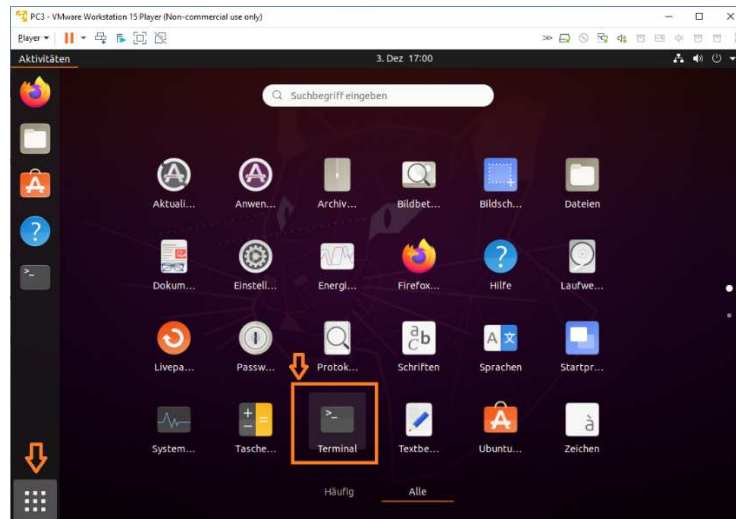
Benutzerverwaltung und Dateisystemrechte im Filesystem

Aufgaben

a) Lesen Sie sich auf der netacad.com – Seite im Kurs “NDG Linux Unhatched” die Kapitel 6-10 durch.

b) Übungen in der Ubuntu-VM

Starten Sie ihre Ubuntu-VM und öffnen Sie ein „Terminal“.



Führen Sie in diesem Terminal folgende Übungen durch bzw. Befehle aus:

Als welcher Benutzer sind sie gerade eingeloggt?

`whoami`

Wechseln Sie zum Benutzer root (Das eigentliche Kommando wäre `su -`, da dann aber das Kennwort für diesen Benutzer root einzutippen wäre und wir für root bei der Installation keines gesetzt haben, müssen wir folgendes Kommando verwenden.

`sudo su -`

Mit sudo vor einem Befehl, wird der nachfolgende Befehl als root ausgeführt. Damit dies funktioniert, muss der aktuelle Benutzer in der Linux-Gruppe „sudo“ sein (was unser bei der Installation angelegte Benutzer ist). Zur Sicherheit wird auch hier ein Kennwort abgefragt – allerdings das des aktuellen sudo-Benutzers.

`whoami`

Legen Sie nun als root zwei neue Benutzer an: schueler,lehrer

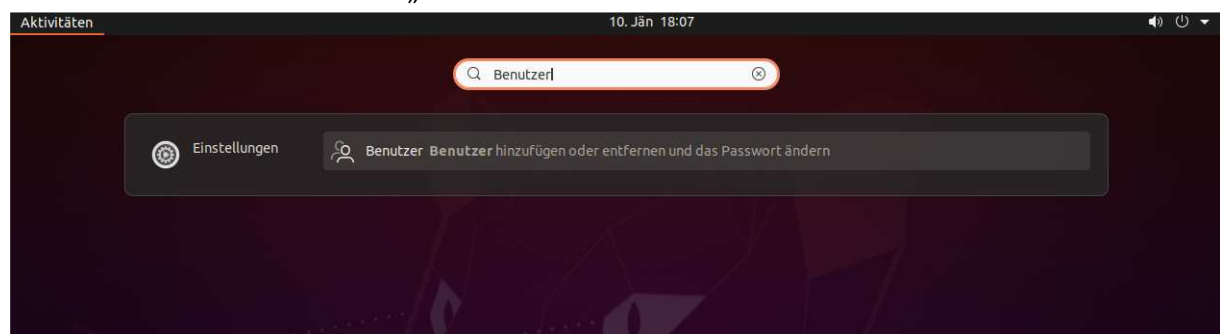
2 Möglichkeiten gäbe es (useradd bzw. adduser). Ich empfehle adduser, da dabei auch gleich das Kennwort gesetzt wird. (Man könnte aber mit „passwd <benutzer>“ auch später ein neues Kennwort setzen)

`adduser schueler`

`adduser lehrer`

Interaktiv wird 2x nach dem Kennwort für den neuen Benutzer gefragt, der Rest kann einfach nur mit <Enter> bestätigt werden.

Alternativ kann beim grafischen Ubuntu auch mittels GUI ein Benutzer angelegt werden. Links unten im Menü klicken und nach „Benutzer“ suchen.

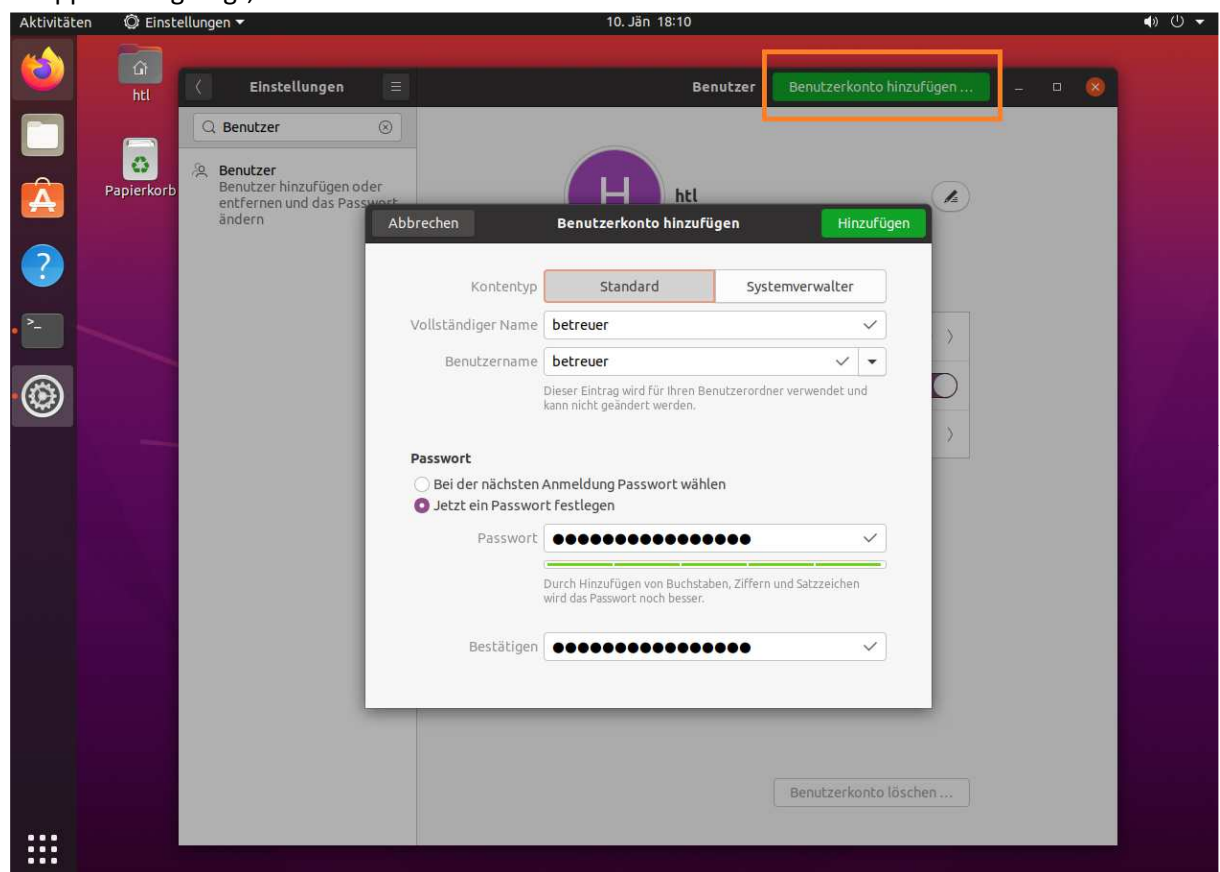


Dann sollte der entsprechende Einstellungen-Dialog kommen.

Danach rechts oben auf „Entsperren“ klicken – und Kennwort für den aktuellen (sudo-) Benutzer eintippen.

Nun kann mittels „Benutzerkonto hinzufügen“ ein neuer Benutzer angelegt werden.

Info: Wird bei Kontentyp „Systemverwalter“ gewählt, wird der Benutzer automatisch zur sudo-Gruppe hinzugefügt, d.h. er kann als root Befehle ausführen.



Wechseln Sie nun in das Verzeichnis /home und zeigen Sie den Inhalt des Verzeichnisses an.

```
drwxr-xr-x  2 betreuer betreuer 4096 Jän 10 18:14 betreuer
drwxr-xr-x 35 htl      htl      4096 Jän 10 17:29 htl
drwxr-xr-x  2 lehrer  lehrer   4096 Jän 10 18:14 lehrer
drwxr-xr-x  2 schueler schueler 4096 Jän 10 18:14 schueler
```

Man sieht, dass Ubuntu für alle Benutzer ein Homeverzeichnis angelegt hat. Man sieht, dass Ubuntu nicht nur den Besitzer des Verzeichnisses gesetzt hat, sondern daneben auch eine gleichnamige Gruppe angelegt worden ist, in der auch nur der angelegte Benutzer eingetragen ist. Dies ist nicht bei jeder Linux-Distribution so, aber eben bei Ubuntu.

Mit dem Befehl `id` kann man die Gruppen, in denen ein Benutzer ist kontrollieren:

```
id schueler
```

```
id lehrer
```

```
id betreuer
```

Man kann aber jederzeit neue Gruppen anlegen, und dann die Benutzer in diese Gruppe eintragen:

```
groupadd schule
```

```
usermod -G schule schueler (Großes -G, bei kleinem -g würde die primäre Gruppe geändert werden)
```

```
usermod -G schule betreuer
```

```
usermod -G schule lehrer
```

Vergleichen Sie die Ausgaben mit vorhin:

```
id schueler
```

```
id lehrer
```

```
id betreuer
```

Im nächsten Teil sollen die Dateirechte getestet werden:

Wechseln Sie als root ins Verzeichnis `/var` und erstellen Sie einen Ordner `demo`.

```
cd /var
```

```
mkdir demo
```

```
cd demo
```

Legen Sie 4 Dateien an und ein weiteres Verzeichnis an

```
echo eins >datei1.txt
```

```
echo zwei >datei2.txt
```

```
echo ls -l >skript.sh
```

```
mkdir verz
```

```
echo drei >verz/datei3.txt
```

```
ls -l
```

```
-rw-r--r-- 1 root root  5 Jän 10 19:03 datei1.txt
-rw-r--r-- 1 root root  5 Jän 10 19:03 datei2.txt
-rw-r--r-- 1 root root  6 Jän 10 19:03 skript.sh
drwxr-xr-x 2 root root 4096 Jän 10 19:03 verz
```

Der Besitzer von `datei1.txt` soll nun „schueler“ werden:

```
chown schueler datei1.txt
```

Die Gruppe von `datei2.txt` soll nun „schule“ werden:

```
chgrp schule datei2.txt
```

Sollen Besitzer und Gruppe gleichzeitig geändert werden, wäre auch folgende Kombination möglich:

```
chown lehrer:schule skript.sh (statt : kann auch . verwendet werden)
```

```
chown lehrer.schueler verz
```

`ls -l`

```
-rw-r--r-- 1 schueler root      5 Jän 10 19:03 datei1.txt
-rw-r--r-- 1 root    schule     5 Jän 10 19:03 datei2.txt
-rw-r--r-- 1 lehrer  schule     6 Jän 10 19:03 skript.sh
drwxr-xr-x 2 lehrer  schueler 4096 Jän 10 19:03 verz
```

Setzen der Dateirechte entweder numerisch

`chmod 600 datei1.txt` (1.Ziffer Besitzer, 2.Ziffer Gruppe, 3. Ziffer – Others)

Bsp: Ziffer 6 (binär 110 gedacht für rwx) ergibt rw-

oder mit mit Symbolen

`chmod u=rw,g=rw,o= datei2.txt` (u user, g group, o others)

`chmod g-x verz` (x-Recht für group entfernt)

`chmod u+x skript.sh` (x-Recht für user hinzugefügt)

`ls -l`

```
-rw----- 1 schueler root      5 Jän 10 19:03 datei1.txt
-rw-rw---- 1 root    schule     5 Jän 10 19:03 datei2.txt
-rwxr--r-- 1 lehrer  schule     6 Jän 10 19:03 skript.sh
drwxr--r-x 2 lehrer  schueler 4096 Jän 10 19:16 verz
```

Beenden Sie die „Arbeit“ als root.

`exit`

und wechseln Sie zu Benutzer lehrer:

`su - lehrer`

`cd /var/demo`

`ls -l`

Testen Sie nun als lehrer den Zugriff auf die Dateien. (Der Befehl cat gibt den Inhalt von Dateien aus.)

`cat datei1.txt` (Sollte nicht erlaubt sein)

`cat datei2.txt` (Sollte funktionieren, da lehrer Mitglied der Gruppe schule ist, und für die Gruppe das r-Recht gesetzt ist)

`./skript.sh` (Sollte funktionieren (da das execute-Recht gesetzt ist) und das Shell-Skript ausführen – im File steht ls, daher wird dieses Kommando ausgeführt)

`cd verz` (Sollte erlaubt sein, da das x-Recht bei einem Verzeichnis bedeutet, dass man hineinwechseln darf)

`cd ..`

Darf lehrer den Besitzer einer Datei ändern?

`chown schueler verz` (Obwohl lehrer Besitzer ist, darf er nicht den Besitzer wechseln – nur root darf das)

Darf lehrer die Dateirechte ändern?

`chmod o= verz` (Dateirechte darf der Besitzer ändern)

Darf lehrer die Datei im Unterordner verz löschen (Befehl rm (remove) löscht die Datei)?

`ls -l verz`

```
-rw-r--r-- 1 root root 5 Jän 10 19:16 datei3.txt
```

`rm verz/datei3.txt`

`ls -l verz`

Das Verzeichnis sollte leer sein. Warum war das Löschen erlaubt? Weil das Löschen einer Datei nicht mit den Dateirechten zusammenhängt, sondern mit dem w-Recht des Verzeichnisses! (w(rite) – der Benutzer darf etwas (um)schreiben in dem Verzeichnis, d.h. Dateien anlegen, aber auch löschen)

exit

Wechseln Sie zu Benutzer schueler:

su - schueler

cd /var/demo

ls -l

Testen Sie nun als schueler den Zugriff auf die Dateien.

cat datei1.txt (Sollte diesmal erlaubt sein)

cat datei2.txt (Sollte funktionieren, da schueler ebenfalls Mitglied der Gruppe schule ist)

./skript.sh (Sollte nicht erlaubt sein, da schueler kein x-Recht hat)

cd verz (Sollte nicht erlaubt sein, da das x-Recht beim Verzeichnis fehlt)

exit

c) Fragen für die Vorbereitung auf die nächste Stundenwiederholung

- Mit welchem Befehl kann ein neuer Benutzer angelegt werden?
- Mit welchem Befehl kann eine neue Gruppe angelegt werden?
- Wofür stehen die einzelnen Zeichen bei der Ausgabe von "ls -l" ?

```
-rw----- 1 schueler root      5 Jän 10 19:03 datei1.txt
-rw-rw---- 1 root      schule    5 Jän 10 19:03 datei2.txt
-rwxr--r-- 1 lehrer    schule    6 Jän 10 19:03 skript.sh
drwxr--r-x 2 lehrer    schueler 4096 Jän 10 19:16 verz
```

- Welcher Befehl öffnet eine neue Shell, in der man als root-Benutzer eingeloggt ist?
- Mit welchem Befehl kann der Besitzer (owner) einer Datei geändert werden?
- Wie können die Datei-/Verzeichnisrechte gesetzt werden?
- Braucht man root-Rechte, um den Besitzer einer Datei zu ändern?