


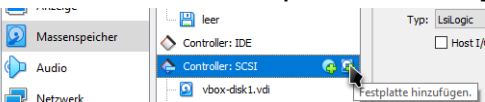
Übung 5 – Partitionen, Dateisysteme, Rechte

Aufgabe 0: Einbau einer neuen virtuellen Festplatte

Dieses Aufgabenblatt geht davon aus, dass Sie die virtuelle Maschine (VMware) verwenden, die vom Lehrenden bereitgestellt wurde.

Einrichtung einer weiteren virtuellen Festplatte in **VirtualBox**:

[Alternativ kann auch die Datei ue05.vdi aus Stud.IP in den Ordner der virtuellen Maschine geschoben werden, also dort, wo auch die .vbox-Datei liegt.]

1. Der virtuelle Rechner muss beendet und das virtuelle Betriebssystem heruntergefahren sein, also z. B. mit `sudo poweroff` beendet worden sein.
2. Wähle die VM in der Liste aus und klicke auf „Ändern“ 
3. Navigiere in der Liste zum Punkt „Massenspeicher“ und wähle in der Liste „Controller: SCSI“ aus, klicke „Festplatte hinzufügen“ und im nächsten Fenster auf „Erzeugen“:

4. Wähle „VDI“, bei „Art der Speicherung“ nichts und dann den Dateinamen und eine Größe für die Festplatte aus (z. B. 128 MB), klicke anschließend auf „Erzeugen“
5. Markiere die erstellte Festplatte in der Liste und klicke auf „Auswählen“, unter „Controller: SCSI“ sollte nun die neu erstellte Festplatte zu sehen sein.

Einrichtung einer weiteren virtuellen Festplatte im **VMware Player**:

1. Der virtuelle Rechner muss beendet und das virtuelle Betriebssystem heruntergefahren sein, also z. B. mit `sudo poweroff` beendet worden sein.
2. Lade die virtuelle Festplatte desk512.vmdk aus Stud.IP herunter und lege diese im Stammordner Ihrer virtuellen Maschine ab [also dort, wo sich z. B. ubuntu-server-9.04-i386.vmx befindet].
3. Öffne die (in diesem Beispiel) Datei ubuntu-server-9.04-i386.vmx in einem Text-Editor und füge Sie folgende Zeilen hinzu:

```
scsi0:1.present = "TRUE"  
scsi0:1.fileName = "disk512.vmdk"  
scsi0:1.writeThrough = "TRUE"
```

Starte den virtuellen Rechner. Die Festplatte ist jetzt unter `/dev/sdb` verfügbar.
[Prüfen mit `ls /dev/sdb`]

Aufgabe 1: Partitionieren und Dateisystem anlegen

- a) Zeigen Sie alle aktiven Festplatten und Partitionen an:

```
sudo fdisk -l
```

1. Was macht der Befehl `sudo`?
2. Wozu dient das Verzeichnis `/dev`?
3. Was ist der Unterschied zwischen `/dev/sda`, `/dev/sda1-2`, `/dev/sdb` usw.?

- b) Legen Sie eine primäre Partition vom Typ „Linux“ [83] auf der neuen Festplatte an, verwenden Sie dazu den Befehl `cfdisk`:

```
sudo cfdisk /dev/sdb
```

Sie müssen selbstständig durch das Menü navigieren. Dies kann durchaus etwas Zeit in Anspruch nehmen. Also bitte nicht aufgeben, wenn es nicht sofort funktioniert!

1. Was ist Partitionieren?
2. Was ist der Unterschied zwischen primären und logischen Partitionen?

- c) Dateisystem [ext3] einrichten:

```
man mkfs  
man mkfs.ext3  
sudo mkfs.ext3 /dev/sdb1
```

1. Wofür steht `mkfs`?
2. Was ist ein Dateisystem?
3. Was bedeutet die Ausgabe von `mkfs`?
4. Welche bekannten Dateisysteme gibt es noch?

Aufgabe 2: mount und unmount

a) „Mounten“ Sie das Dateisystem:

```
sudo mkdir /mnt/myfs
sudo mount /dev/sdb1 /mnt/myfs
mount
```

1. Was bedeutet „mounten“?
2. Was gibt der Befehl mount im letzten Schritt aus?

```
ls -la /mnt/myfs
```

Was sehen Sie? Woher kommt das Verzeichnis lost+found?

b) „Unmounten“ Sie das Dateisystem:

```
sudo umount /mnt/myfs
ls -la /mnt/myfs
```

(Hinweis: umount nicht unmount)

c) Überdeckung existierender Dateien oder Ordner:

```
sudo touch /mnt/myfs/hallo.txt
ls -lai /mnt/myfs
sudo mount /dev/sdb1 /mnt/myfs
ls -lai /mnt/myfs
```

1. Was wird bei den beiden ls-Befehlen ausgegeben? Warum?
2. Was zeigt hier die Option -i für ls und was sind die Unterschiede?

d) Unmounten ist teilweise nicht möglich:

```
cd /mnt/myfs
sudo umount /mnt/myfs
```

Was passiert? Erklären Sie den Effekt, denn Sie beobachten. Nutzen Sie dazu den folgenden Befehl als Hilfe und erklären Sie die Ausgabe:

```
fuser -u /mnt/myfs
```

Aufgabe 3: Besitzer und Rechte

Tipp: In dieser Aufgabe wird viel mit absoluten Pfaden gearbeitet. Wenn Sie sich Tipparbeit sparen und mehr Übung im Umgang mit relativen Pfaden haben möchten, wird empfohlen, direkt vom Verzeichnis /mnt/myfs aus zu arbeiten, damit Sie diesen Pfad nicht immer eintippen müssen.

a) Kopieren als **root**:

```
sudo cp -r /etc /mnt/myfs/etc-root
```

1. Welchem Benutzer gehören die Dateien in „/mnt/myfs/etc-root“? Warum?
2. Wem gehörten die Dateien ursprünglich in /etc?

b) Kopieren als **notroot**:

```
sudo chmod a+rwX /mnt/myfs  
cp -r /etc /mnt/myfs/etc-notroot
```

1. Was bewirkt der Befehl `chmod a+rwX /mnt/myfs`?
2. Welche Ausgabe sehen Sie? Warum?
3. Welchem Benutzer gehören die Dateien in „/mnt/myfs/etc-notroot“? Warum?

c) Löschen als **notroot**:

```
ls -l /mnt/myfs/etc-root/hosts  
rm -f /mnt/myfs/etc-root/hosts
```

1. Wozu dient die Option `-f` bei `rm`?
2. Wird die Datei `hosts` gelöscht? Warum oder warum nicht?
3. Würde sich etwas ändern, wenn die Rechte von `hosts` durch folgenden Befehl verändert werden?

```
sudo chmod o+wx /mnt/myfs/etc-root/hosts
```

Kann das komplette Verzeichnis `etc-root` gelöscht werden?

```
ls -la /mnt/myfs  
rm -fr /mnt/myfs/etc-root  
ls -la /mnt/myfs/etc-root
```

Warum oder warum nicht? Was wird ausgegeben?

Experimentieren sie mit einer Kopie von `/etc/shadow` in `/mnt/myfs` :

```
sudo cp /etc/shadow /mnt/myfs/shadow  
ls -l /mnt/myfs/shadow  
cat /mnt/myfs/shadow  
echo test > /mnt/myfs/shadow
```

Warum kann die Datei `shadow`, die `root` gehört, weder gelesen noch überschrieben werden? Kann die Datei gelöscht werden?

```
rm -f /mnt/myfs/shadow
```

Warum oder warum nicht?

Aufgabe 4: Besitzer- und Gruppenvergabe

chown	Eigentümer setzen
chgrp	Gruppe setzen

- a) Lesen Sie die man pages nach eigenem Ermessen: (Alternativ: Über Internet-„Recherche“ herausfinden, wozu die beiden Programme genutzt werden.)

```
man chown
man chgrp
```

- b) Besitzer ändern:

```
ls -l /mnt/myfs
chown notroot /mnt/myfs/etc-root
sudo chown notroot /mnt/myfs/etc-root
ls -l /mnt/myfs
```

Warum muss sudo benutzt werden bzw. können Sie ein Szenario beschreiben, in der der Befehl chown missbraucht werden könnte, wenn es nicht mit sudo ausgeführt werden muss? Welches „böartige“ Szenario kann man dadurch verhindern, dass nur mit sudo der Befehl ausgeführt werden kann?

In der vorherigen Aufgabe wurde versucht, das Verzeichnis etc-root zu löschen, was ändert sich dieses Mal, wenn folgender Befehl ausgeführt wird?

```
rm -rf /mnt/myfs/etc-root
```

Tipp: Sehen Sie sich auch die Dateien in etc-root an!

- c) Gruppe ändern:

```
chgrp src /mnt/myfs/etc-notroot
chgrp adm /mnt/myfs/etc-notroot
```

Funktionieren beide Befehle?
Erklären Sie Ihre Beobachtung!

Tipp: Benutzen Sie den Befehl „id -Gn“! Was gibt dieser Befehl aus?