

Aufgabe 1 – *ssh und scp*

Oft genug kommt man in die Situation, dass man sich auf einem anderen Rechner einloggen muss, um dort zu arbeiten, zum Beispiel weil auf diesem Dateien oder Programme liegen, auf die man sonst keinen Zugriff hat. Das Kommando, mit dem man sich auf einem anderen Rechner anmelden kann, kennst du bereits aus der Vorlesung.

- Melde dich mit Hilfe von **ssh** auf dem Rechner **cipterm0** an und schaue, was auf diesem Rechner in **/var/tmp/linuxkurs/** liegt.
Hinweis: Du kommst wieder auf den lokalen Computer zurück, indem du **exit** eintippst.
- Lege auf **deinem** Rechner im Verzeichnis **/tmp/** ein neues Verzeichnis mit deinem Loginnamen an und kopiere (mittels **scp**) das, was du auf der **cipterm0** gefunden hast, in dieses Verzeichnis. (*Tipps: Die Rechnernamen stehen auf den Monitoren oder Rechnern. Wenn du auf den lokalen Rechner kopierst, darfst du beim Ziel keinen Rechnernamen angeben, sondern nur den Pfad wo die Datei hinkopiert werden soll.*)
- Entpacke das Archiv, das du dir gerade kopiert hast, mit dem Befehl **tar xf** und schau dir die entpackten Dateien an.

Einige der entpackten Dateien kannst du nicht betrachten, da du dazu nicht berechtigt bist. Lesbar machen kannst du sie mit **chmod**.

- Um Dateien für dich lesbar zu machen, musst du für dich (**u**, für user) das read-Flag (**r**) setzen: **chmod u+r <Dateiname>**
- Manchmal ist es praktisch, alle Dateien in einem Unterordner lesbar zu machen. Wie du dies bewerkstelligst, kannst du in der manpage nachlesen (**man chmod**, Stichwort: *recursive*).
- Lösche die Datei **Linux-Kurs/Musik/DELETEME**; dazu musst du die Rechte des Verzeichnisses **Musik** anpassen (Tipp: Löschen braucht Schreibrechte).

Hinweis: Wenn **chmod** über fehlende Rechte klagt, schau dir das **x**-Flag auf den Verzeichnissen an.

Aufgabe 2 – *chmod und Mensaskript*

Nun wollen wir **chmod** praktisch anwenden, um ein Skript ausführbar zu machen. Dazu nehmen wir uns das Mensaskript aus der vorherigen Übungsstunde vor.

- Bis jetzt kann man das Skript mit **perl mensa** ausführen, aber dazu muss man natürlich vorher wissen, in welcher Skriptsprache es geschrieben ist. Besser ist es die Datei ausführbar zu machen. Dazu setzt du für dich das *executable*-Flag (**x**): **chmod u+x mensa**
- Das Ergebnis deiner Berechtigungsänderung kannst du mit **ls -l mensa** betrachten.
- Nun kannst du das Skript direkt mit **./mensa** aufrufen.

Aufgabe 3 – *Failed Backup*

Im Verzeichnis `/proj/ciptmp/fsi/linuxkurs/uebung/aufgabe_failedbackup/archive/` findest du ein Archiv mit den Namen `failedbackup.tar.bz2`. In dieses Archiv wurden „versehentlich“ falsche Dateien eingepackt.

Kopiere das Archiv `failedbackup.tar.bz2` in ein temporäres Verzeichnis (z. B. `/tmp/<username>`) und entpacke es **dort**. Wechsle nun in das entstandene Verzeichnis `failedbackup`.

- Wie viele Dateien sind im Verzeichnis `dir2`¹? (Bitte nicht per Hand zählen²!)

Nun wollen wir das Backup bereinigen und ein paar Dateien löschen:

- Lösche alle Dateien, die mit `old_` beginnen, aus dem Verzeichnis `dir2`.
- Lösche alle Dateien, die mit `arts` beginnen, aus dem Verzeichnis `dir1`³.
- Lösche alle Dateien, die auf `.bak` enden, aus den *beiden* Verzeichnissen `dir1` und `dir2`.
- Wie viele Dateien sind nun noch in den beiden Verzeichnissen⁴?

Um herauszufinden, ob du die richtigen Dateien entfernt hast, kannst du dir eine Liste mit den Dateinamen erzeugen und eine Prüfsumme darüber berechnen lassen.

- Wechsle ins Verzeichnis `failedbackup`.
- Verwende `find`, um alle Dateien anzeigen zu lassen. Sortiere die Ausgabe mit `sort` und leite die Ausgabe an das Programm `md5sum`⁵ weiter.

Die Aufgabe ist korrekt erledigt, wenn die Prüfsumme `274455e39230db7fb0d2514ea2302485` ist. Solltest du eine andere Prüfsumme haben, dann prüfe, ob noch zu löschende Dateien existieren. Oder hast du vielleicht versehentlich eine neue Datei angelegt?

Erstelle nun aus dem Verzeichnis ein neues Archiv mit dem Namen `fixedbackup_ok.tar.bz2`.

Aufgabe 4 – *Die Kontrolle behalten*

Starte in einer Konsole die Vorlesungsfolien mit `evince`. Wechsle zurück zur Konsole und stoppe das Programm mit `Ctrl-Z`. Schicke `evince` nun in den Hintergrund (`bg`). Nun kannst du deine Konsole für ein anderes Programm benutzen. Starte zum Beispiel `Thunderbird`. Schicke dieses nun auch in den Hintergrund. Mit `jobs` kannst du jederzeit sehen, welche Jobs du gerade laufen hast.

¹Ich biete 2298.

²Tipp: `ls` und `wc` geschickt „verbinden“

³Das klappt nicht? Da fehlen wohl irgendwelche Rechte. Nur welche?

⁴Ich biete 1529 und 766.

⁵`md5sum` liest von `stdin` Daten ein und bildet eine Prüfsumme über die Daten.

Aufgabe 5 – *Kill me!*

Führe zuerst das Programm `/proj/ciptmp/fsi/linuxkurs/killme` aus. Öffne danach eine neue Shell und versuche das Programm zu beenden. *Tipp: Lies dir die Ausgabe des Programms durch!* Falls du dich schon in dem Verzeichnis `/proj/ciptmp/fsi/linuxkurs` befindest, kannst du das Programm mit `./killme` starten.

Als nächstes führst du das Programm `/proj/ciptmp/fsi/linuxkurs/tetris` aus. Lässt sich das Programm wie das `killme`-Programm beenden? Wenn nein, dann erinner dich an die Vorlesungsfolien⁶.

Und die Moral von der Geschicht': Nicht einfach Programme von anderen Leuten ausführen, wenn man nicht weiß, was sie tun. Manchmal trügt der Schein. ☺⁷

Aufgabe 6 – *Quota*

In deinem *Home*-Verzeichnis hast du anfangs 1 GB (als Informatik-Student oder informatiknaher Student 2 GB) Speicherplatz zur Verfügung – und der ist überraschend schnell voll. Vor allem wenn du das *tetris*-Skript aus Aufgabe 5 ausgeführt hast. Falls du die vorherige Aufgabe übersprungen hast, so bearbeite nun wenigstens deren zweiten Teil (*tetris*).

Um zu überprüfen, wie viel Platz noch frei ist, kannst du (natürlich in einer Shell) den `quota`-Befehl verwenden:

- `blocks`: der belegte Speicher (in KiB)
- `quota`: der zur Verfügung stehende Speicher (in KiB)

Schau in der Manpage von `quota` nach, wie man die Ausgabe auf menschenlesbar („human-readable“) stellen kann! Alternativ kannst du auch das CIP-spezifische Programm `cip-quota` verwenden, was eine schönere Ausgabe hat.

Falls du herausgefunden hast, dass du zu viel Speicher belegst ☹, kannst du mit dem Programm `ncdu` nachschauen (mit `q` beendet man es wieder), das dir eine sortierte Anzeige der Speicherplatzfresser im momentanen Verzeichnis anzeigt und es dir auch erlaubt, durch Unterverzeichnisse zu navigieren.

Nun möchtest du die Dateien, die das amoklaufende *tetris*-Programm erstellt hat, wieder löschen. Die Dateien sind einigermaßen erkennbar benannt, also sollte es für dich nicht all zu schwer sein, sie zu identifizieren.

Hinweis für das Arbeiten in der Uni:

Wenn man sein Quota überschritten hat, ist eine graphische Anmeldung am PC oft nicht mehr möglich. In diesem Fall kann man mit `Strg+Alt+F1` auf ein anderes *virtuelles Terminal* wechseln und sich dort anmelden und ein paar Dateien löschen. Ist genug Platz, **melde dich wieder ab** (`exit`) und wechsele mit `Strg+Alt+F7` wieder zurück zur grafischen Oberfläche.

⁶Tipp: -9

⁷Falls es dich interessiert, beide Programme sind Skripte, die du mit `cat` betrachten kannst.

Zusatzaufgaben

Noch mehr Aufgaben

Du hast immer noch Interesse an weiteren Übungsaufgaben? Dann guck doch mal unter folgender Adresse vorbei:

<https://fsi.cs.fau.de/informationen/linuxkurs/aufgabensammlung/>