

Aufgabe 1a)

Erstellen Sie in Ihrem HOME-Verzeichnis ein Unterverzeichnis

Uebungen/Uebung2/Programme. Setzen Sie die Zugriffsrechte auf dieses Verzeichnis so, dass nur Sie als Besitzer dort Lese-, Schreib- und Ausführungsrechte haben (siehe mkdir, chmod).

```
$ mkdir -p Uebungen/Uebung2/Programme
```

„mkdir -p“, alternativ „--parents“, erstellt übergeordnete Verzeichnisse (Elternverzeichnisse) je nach Bedarf. Hier werden also die Verzeichnisse Uebung und Uebung2 mit erstellt, wenn sie zuvor nicht extra angelegt wurden.

```
$ chmod 700 Uebungen/Uebung2/Programme
```

chmod (change file mode bits) wird unter anderem dazu genutzt die Rechte einer Datei bzw. eines Verzeichnisses zu setzen bzw. zu ändern.

Die „700“ bedeutet: Volle Rechte für den User/Besitzer der Datei (7), keine Rechte für die Gruppe (0) und keine Rechte für alle anderen Benutzer (0).

Hinweis: chmod kennt auch alternative Schreibweisen für die Rechte. So kann man z.B. der Gruppe mit „chmod g+r Datei/Verzeichnis“ Leserechte (r=read) geben.

```
$ ls -l Uebungen/Uebung2/
```

Zeigt einem die ausführliche Ausgabe von ls und somit auch die Zugriffsrechte. (Dient nur der Kontrolle, eigentlich nicht Teil der Aufgabe an sich.)

Alternativ: ls -ld Uebung/Uebung2/Programme

Aufgabe 1b)

Erzeugen Sie in Ihrem Home-Verzeichnis einen Verweis (symbolischen Link) mit dem Namen bin, der auf das Systemverzeichnis /usr/bin, verweist (siehe ln).

```
$ ln -s /usr/bin bin
```

ln erstellt sogenannte Links auf/zwischen Dateien oder Verzeichnissen.

Der Option „-s“ steht hier für „symbolic“, womit angegeben wird, dass ein symbolischer Link (Symmlink) erstellt werden soll.

Ein Link ist ein Verweis. Der Symmlink „bin“, der im aktuellen Verzeichnis erstellt wird, verweist auf das Verzeichnis „/usr/bin“.

```
$ ls -l
```

(Dient nur zur Kontrolle, eigentlich nicht Teil der Aufgabe an sich.)

Aufgabe 1c)

Kopieren Sie aus dem Systemverzeichnis `/usr/bin` alle Programme, die mit dem Buchstaben `l` oder `z` beginnen, in das neu angelegte Verzeichnis hinein. Ermitteln Sie nun die Anzahl der Dateien im Verzeichnis `Programme`. Wie sieht der Befehl aus, wenn nur die ausführbaren Programme kopiert werden sollen? (man `find`).

```
$ cp /usr/bin/[lz]* Uebungen/Uebung2/Programme
```

„`[lz]*`“ entspricht allen Dateien/Verzeichnissen, deren Name mit „`l`“ oder „`z`“ beginnt (also einem der Buchstaben in „`[]`“) und denen eine beliebige Zeichenkette (`*`) folgt.

```
$ ls Uebungen/Uebung2/Programme | wc -w
```

Hier werden einfach alle Dateien, die im „`Programme`“-Verzeichnis liegen von „`ls`“ über eine Pipe an „`wc`“ weitergereicht, welches die Wörter („`-w`“) zählt. (Ein Wort = eine Datei.)

Hinweis: Der `wc`-Paramter „`-l`“ für „`lines`“ wäre hier falsch, da `ls` ohne Paramter auch mehrere Dateien in einer Zeile aufzählt.

```
$ cp $(find /usr/bin/ -name [lz]* -perm -111) \
Uebungen/Uebung2/Programme/
```

„`$(...)`“ wird durch die Ausgabe des darin befindlichen Befehls (hier „`find`“) ersetzt. Somit werden die Dateien, die „`find`“ findet, als Paramter an „`cp`“ übergeben.

„`find`“ bekommt als ersten Paramter den Ort, an dem es suchen soll. („`/usr/bin`“)

„`-name`“ gibt an wie der Name der gesuchten Datei(en) lauten soll. (Hier also alle Dateien deren Name mit „`l`“ oder „`z`“ beginnt und mit einer beliebigen Zeichenkette fortgesetzt wird.)

„`-perm -111`“ gibt an, dass die Dateien mindestens („`-`“) die Rechte „`111`“ haben sollen, also mindestens alle Benutzer Rechte für das Ausführen haben sollen. (Alternativ z.B. „`100`“ wenn mindestens der Besitzer der Dateien Rechte für das Ausführen haben soll.)

Hinweis: „`\`“ steht hier nur für den Zeilenumbruch.

Alternativ:

```
$ find /usr/bin/ -name [lz]* -perm -111 -exec cp {} \
Uebungen/Uebung2/Programme/ \;
```

Aufgabe 1d)

Generieren Sie durch eine geeignete E/A -Umleitung in Ihrem `Uebung2` Verzeichnis eine Datei namens `index`, welche das Inhaltsverzeichnis des Verzeichnisses `Programme` enthält (`ls`). Erzeugen Sie analog eine weitere Datei `index2`, welche das komplette Inhaltsverzeichnis von `/usr/bin` enthält.

Die Inhaltsverzeichnisse `index` und `index2` sollen jeweils nur einen Programmnamen pro Zeile enthalten (man `ls`).

```
$ ls -1 Uebungen/Uebung2/Programme > Uebungen/Uebung2/index
```

```
$ ls -1 /usr/bin > Uebungen/Uebungen2/index2
```

Mit der Ausgabeumleitung „`>`“ wird die Ausgabe von „`ls`“ in die Datei `index` umgeleitet. Mit „`-1`“ listet „`ls`“ nur einen Programmnamen pro Zeile auf.

Hinweis: „`>`“ überschreibt die Datei, falls sie schon existiert. „`>>`“ würde die Ausgabe an das Ende der Datei anhängen, falls sie schon existiert.

Aufgabe 1e)

Hängen Sie den Inhalt der Datei `index2` an die Datei `index` an. Sortieren Sie die Datei `index` nach Programmnamen, wobei Sie eine neue Datei `index.sortiert` erzeugen, die keine Duplikate mehr enthält. (siehe `sort`, `uniq`)

```
$ cat index2 >> index
```

Die Datei „index2“ wird mit `cat` ausgegeben und mit der Ausgabeumleitung „>>“ an die Datei „index“ angehängt.

```
$ cat index | sort | uniq > index.sortiert
```

„sort“ sortiert die Zeilen, die es über die Pipe von der Standardeingabe liest, alphabetisch und schreibt sie auf die Standardausgabe.

„uniq“ ersetzt mehrfach vorkommende Zeilen durch ein einmaliges Vorkommen.

Wichtig: Es ist wichtig erst „sort“ und anschließend „uniq“ zu verwenden, da „uniq“ mehrfach vorkommende Zeilen nur erkennt, wenn diese direkt nacheinander vorkommen.

Alternativ: `sort -u index > index.sortiert`

„-u“ für „unique“ hat dieselbe Funktionalität wie das Programm „uniq“.

Aufgabe 1f)

Suchen Sie in der Datei `index2` nach allen Zeilen, die den Teilstring „ch“ enthalten (siehe `grep`).

```
$ grep ch index2
```

„grep“ sucht in der Datei „index2“ nach dem Muster „ch“ und gibt die gefundenen Zeilen auf die Standardausgabe, also das Terminal, aus.

Aufgabe 2a)

Sorge dafür, dass für alle neuangelegten Dateien die Berechtigungen `rw-rw-rw-` und für alle Verzeichnisse die Rechte `rxwxrwx` gelten (siehe `umask`).

Hinweis: aufgrund der Konfiguration der Poolrechner lassen sich im Home-Verzeichnis nur die Rechte `rw-rw---` bzw. `rxwxrwx---` erreichen; in `/tmp` funktioniert es jedoch wie erwartet.

```
$ umask 000
```

Von den Standardberechtigungen wird 000 abgezogen. Daraus ergibt sich für Dateien (666) und für Verzeichnisse (777) als Berechtigung beim Anlegen.

Aufgabe 2b)

Legen Sie das Verzeichnis `abc` und darin die Datei `x` an (siehe `mkdir`, `touch`). Legen Sie einen symlink „sym“ und einen hardlink „hard“ auf `x` an (siehe `ln`).

```
$ mkdir abc
```

```
$ touch x
```

```
$ ln -s x sym
```

```
$ ln x hard
```

Aufgabe 2c)

Versehen Sie `abc` mit dem sticky bit (siehe `chmod`).

```
$ chmod +t abc
```

Aufgabe 2d)

Ändern Sie die Berechtigungen für hard so, dass außer dem Eigentümer niemand mehr Zugriff darauf hat und lassen Sie den Inhalt des Verzeichnisses ausführlich anzeigen. Was fällt dabei auf?

```
$ chmod go-rwx hard
$ ls -l
```

x hat dieselben Berechtigungen wie hard (hard und x verweisen auf denselben inode).

Aufgabe 2e)

Schreiben Sie einen beliebigen kurzen Text in die Datei x. Löschen Sie x und geben Sie den Inhalt von hard und sym aus.

```
$ echo "irgendein Text" > x
$ echo "noch ein Text" >> x
$ rm x
$ cat sym
$ cat hard
```

„cat sym“ erzeugt eine Fehlermeldung, da der symbolische Link ins Leere zeigt.
„cat hard“ gibt den geschriebenen Text aus, da beim Löschen von x nur der Verzeichniseintrag von x gelöscht wurde. Die durch die inode referenzierte Datei existiert nach wie vor.

Aufgabe 3a)

Geben Sie aus der /etc/passwd nur den jeweils zweiten und dritten Buchstaben der Home-Verzeichnisse aus, sortieren Sie sie alphabetisch und speichern Sie die ersten 5 Zeilen des Ergebnisses in die Datei top5 (siehe head, cut, sort).

```
$ cut -d: -f6 -s /etc/passwd|cut -b2-3|sort|head -n5>/tmp/top5
```

Aufgabe 3b)

Ersetzen Sie alle Zeilenumbrüche in dieser Datei durch Leerzeichen und speichere das Ergebnis in einer neuen Datei top5space (siehe tr).

```
$ tr "\n" " " <top5 >top5space
```

Aufgabe 3c)

Öffnen Sie die Datei top5space im vim, ersetzen Sie alle Leerzeichen durch „#“-Zeichen und speichern Sie sie unter top5hash.

```
:%s/ /#/g
:w top5hash
:q!
```

Aufgabe 3d)

Lassen Sie sich die Unterschiede zwischen den beiden Dateien top5space und top5hash anzeigen (siehe diff -u).

```
$ diff -u top5space top5hash
```