

Übungen PdP-Praktikum — Blatt 2

Aufgabe 1:

- a) Die Nutzerin `juser` habe die folgende Datei in ihrem Homeverzeichnis:

```
-rw-rw---- 1 juser stud 6 4. Nov 11:43 dokument.txt
```

Wie kann sie die Zugriffsberechtigungen ändern, damit sie wie folgt aussehen?

```
-rw-r--r-- 1 juser stud 6 4. Nov 11:43 dokument.txt
```

- b) `juser` habe ein Verzeichnis `ablage`:

```
dr-xr-xr-x 2 juser stud 4096 4. Nov 11:57 ablage
```

Im Verzeichnis `ablage` befindet sich eine Datei `test.txt` mit den folgenden Berechtigungen:

```
-rw-r--r-- 1 juser stud 12 4. Nov 11:58 test.txt
```

- Kann sie den Inhalt der Datei `test.txt` mit einem Editor ändern?
- Kann sie im Verzeichnis `ablage` eine neue Datei anlegen?

- c) `juser` sei Mitglied der Gruppe `projekt`.

Wie muss sie den Besitzer und die Zugriffsrechte des obigen Verzeichnisses `ablage` mindestens verändern, damit alle Mitglieder der Gruppe `projekt` darin Dateien anlegen können?

Wie kann sie zusätzlich sicherstellen, dass alle neuen Dateien automatisch der Gruppe `projekt` gehören?

Aufgabe 2: Unter der folgenden URL können Sie das Dateiarchiv `dateien02.tar.gz` herunterladen:

<https://www.math.uni-bielefeld.de/~frettlloe/teach/unix/dateien02.tar.gz>

Das Archiv können Sie anschließend mit dem Befehl `tar xzf dateien02.tar.gz` in das aktuelle Verzeichnis auspacken. Es entsteht ein Unterverzeichnis `uebungen02/`. Die folgenden Aufgaben beziehen sich alle auf dieses Unterverzeichnis.

- a) Verwenden Sie den Befehl `ls` zusammen mit einer Wildcard, um alle Dateien anzuzeigen, die mit dem Wort `tag` beginnen.
- b) Was muß man tun, um das Ergebnis des Befehle `ls -la` in eine Datei `listing.txt` umzuleiten?
- c) Wie kann man mit Hilfe des Befehls `grep` alle Zeilen in der Datei `listing.txt` ausgeben, die das Wort `tag` enthalten.
- d) Wie kann man die Aufgabenteile b) und c) mit Hilfe einer Pipe gemeinsam lösen, also ohne die Datei `listing.txt` zu erzeugen? (Weiter nächste Seite)

Aufgabe 3: Die (fiktive) Firma *Geowatch* nimmt jeden Tag Messungen an der Erdkruste in Städten Deutschlands vor. Diese Daten werden nachts in zufälliger Reihenfolge an die Zentrale übermittelt und bekommen die Namen `tag20121020.txt`, `tag20121021.txt` usw. Diese Dateien sind in dem in Aufgabe 2 genannten tar-Archiv enthalten.

- a) Wie muß man den Befehl `grep` benutzen, um aus den beiden Dateien alle Einträge für Bielefeld in der folgenden Form zu extrahieren?

```
tag20121020.txt:Bielefeld 1188.25626
tag20121021.txt:Bielefeld 1187.9329
```

- b) Wie kann man den Befehl `sort` verwenden, um sich die Werte der Datei `tag20121020.txt` alphabetisch nach Städtenamen sortiert anzusehen?
- c) Wie kann man aus den beiden Dateien `tag20121020.txt` und `tag20121021.txt` die nach Städtenamen alphabetisch sortierten Dateien `sort20121020.txt` und `sort20121021.txt` erzeugen?

Aufgabe 4: Schreiben Sie ein Shellskript "`worte.sh`", das die ihm übergebenen Worte sowie deren Anzahl ausgibt.

Beispielaufruf:

```
$ ./worte.sh Heute scheint in Bielefeld die Sonne
Heute scheint in Bielefeld die Sonne
Das sind 6 Worte.
```

Lösungshinweis: Die Aufgabe lässt sich alleine durch Auswertung der Shell-Parameter (siehe aktuelle Vorlesung 4) und durch Benutzung des `echo`-Befehls lösen. Das Programm `wc` wird zur Lösung *nicht* benötigt.

Aufgabe 5: Schreiben Sie ein Shell-Skript `ls-sort` das die Dateien im aktuellen Verzeichnis aufsteigend nach Größe sortiert. Verzeichnisse sollen nicht in der sortierten Ausgabe vorkommen. Das Shell-Skript soll sich wie folgt verhalten:

```
$ ./ls-sort.sh
-rwxr--r-- 1 juser stud 105 11. Nov 11:04 brian.sh
-rwxr--r-- 1 juser stud 214 11. Nov 13:44 ls-count.sh
-rwxr--r-- 1 juser stud 264 11. Nov 12:09 ls-sort.sh
```

Aufgabe 6:

Die Datei `verbrauch.txt` (aus dem Archiv von Aufgabe 2) enthält Verbrauchswerte für die Städte Bielefeld, Gütersloh und Paderborn.

- a) Geben Sie eine Pipe von `grep`-Befehlen an, um aus der Datei die Werte für Bielefeld vom Februar zu extrahieren.
- b) Geben Sie einen Aufruf von `head` an, um aus der Datei die erste Zeile mit der Überschrift zu extrahieren.
- c) Verbinden Sie die Befehle aus den Aufgabenteilen a) und b) zu einer Pipe, um eine Tabelle für die Stadt Bielefeld für den Monat Februar zu erstellen. Die Tabelle soll auf die Kommandozeile ausgegeben werden und zu Beginn wie folgt aussehen:

Stadt	Tief	Hoch	Datum
Bielefeld	24631	55637	1. Februar 2020
Bielefeld	14912	19484	2. Februar 2020
Bielefeld	22040	29291	3. Februar 2020
u.s.w			

Aufgabe 6:

Die Datei `bielefeld.txt` aus dem Archiv von Aufgabe 2 enthält die Aufteilung der Stadt Bielefeld nach Flächenarten (Quelle: Wikipedia). Geben Sie eine Pipe-Folge von Befehlen an, um die Tabelle wie nachfolgend gezeigt nach der zweiten Spalte (Fläche in km^2) zu sortieren:

Fläche nach Nutzungsart	: Fläche in Km2	: Anteil an Gesamtfläche
sonstige Nutzung	: 1,19	: 0,46 %
Wasserfläche	: 1,86	: 0,72 %
Sport- und Grünfläche	: 11,79	: 4,57 %
Siedlungs- und Verkehrsfläche	: 24,89	: 9,65 %
Waldfläche	: 52,22	: 20,25 %
Gebäude-, Frei-, Betriebsfläche	: 70,20	: 27,22 %
Landwirtschaftsfläche	: 95,75	: 37,13 %

Wie müssen Sie den Befehl ändern, damit die Tabelle in absteigender Reihenfolge der Flächen sortiert wird?

Lösungshinweis: Lesen Sie (z.B. in der Manualpage zu `sort`) nach, wie absteigend sortiert wird, und wozu die Option `-t` dient. Gibt es ein entsprechendes Merkmal bzw. Zeichen in der Tabelle, das Sie entsprechend als Spaltentrenner verwenden können?

Zusatzaufgabe:

In dem Ordner aus Aufgabe 2 finden Sie eine kleine Sammlung von Bildern im `pbm`-Format. Öffnen Sie eines in einem Texteditor und finden heraus, wie dieses Dateiformat funktioniert (oder wikipedieren sie es oder...)

Finden Sie dann eine Befehlsfolge, die ein solches Bild invertiert (also schwarz wird zu weiß und umgekehrt) und das Ergebnis in eine Datei `invers.pbm` schreibt.

Schreiben Sie dann ein Shellskript, das das tut. Gestalten Sie das Shellskript so, dass es zwei Argumente nutzt: `invert datei.pbm name` soll die existierende `pbm`-Datei `datei.pbm` invertieren und das invertierte Bild in die Datei `name.pbm` schreiben.

Hinweis: Sie können das Programm `gimp` benutzen, um Bilder als `pbm` zu speichern (“Exportieren”) oder um die `pbm`-Bilder zu öffnen und anzusehen (und zu bearbeiten, aber das erfordert etwas Einarbeitung).