Übungsblatt 1 Lösungsbeispiel Fakultät Elektronik und Informatik
Übungen zu Betriebssysteme Prof. Dr. Rainer Werthebach

Aufgabe 1

Ermitteln Sie die aktuelle Konfiguration ihres Logins:

- Bestimmen Sie den Pfadnamen ihres Home-Verzeichnisses.
- Lassen Sie sich den Inhalt ihres Home-Verzeichnis in den verschiedenen Modi kurze/ausführliche Ausgabe bzw. mit und ohne hidden files anzeigen.

```
$ cd
$ pwd
```

change directory (ohne Parameter → Home-Verzeichnis)
print working directory: gibt das aktuelle Verzeichnis aus.

Alternativen:

```
$ echo ~
```

Gibt das Home-Verzeichnis aus, da „~“ in der bash für das Home-Verzeichnis steht.

```
$ echo $HOME
```

Gibt das Home-Verzeichnis aus. Es ist in der Umgebungsvariablen HOME abgespeichert. Mit dem $-Symbol greift man auf den Inhalt einer Variablen zu ($ .ist der Ersetzungsoperator).

```
$ ls
```

list (directory contents)
Gibt die Dateien im aktuellen Verzeichnis (in Kurzform) aus.
Paramter:
-a zeigt auch versteckte Dateien an
Hinweis: Versteckte Dateien erkennt man unter Linux daran, dass ihr Name mit einem „.“ beginnt.
-l zeigt eine lange/ausführliche Ausgabe an

Bsp.: $ ls -al zeigt alle Dateien (also auch versteckte Dateien) in einer ausführlichen Ausgabe an (weitere Optionen, siehe man ls).

Aufgabe 2

Üben von Dateioperationen:

- Erstellen Sie in Ihrem Home-Verzeichnis die Verzeichnisstruktur test/test1 und test/test2.
- Legen Sie im Ordner test/test1 die Dateien 1, 2 und 3 an.
- Kopieren Sie den Inhalt von Ordner test/test1 in den Ordner test/test2.
- Löschen Sie die Testordner.

```
$ mkdir test
$ mkdir test/test1 test/test2
```

mkdir (make directories) erstellt Verzeichnisse.
Hier wird erst das Verzeichnis test im aktuellen Verzeichnis erstellt und darin die Verzeichnisse test1 und test2 innerhalb von test.

Alternative:

nur

```
$ mkdir -p test/test1 test/test2
```

$ cd test/test1

cd (change directory) wechselt in ein Verzeichnis.
Hier wird in das angegebene Verzeichnis test/test1 gewechselt. Wir befinden uns nach Eingabe dieses Befehls darin. (Zur Kontrolle z.B. pwd eingeben.)

`$ touch 1 2 3`

touch erstellt eine leere Datei, wenn der angegebene Dateiname noch nicht existiert.
Hier werden also die Daten mit den Namen „1“, „2“ und „3“ im aktuellen Verzeichnis erstellt.

`$ cd ..`

„..“ gibt das übergeordnete Verzeichnis an. Befinden wir uns in /home/student so, kommen wir mit „cd ..“ in /home. Im Beispiel von „test/test1“ nach „test“.
Wir befinden uns nun in „test“ bzw. in „/home/nutzername/test“.
Hinweis: „.“ ist das aktuelle Verzeichnis.

`$ cp test1/* test2/`

cp (copy) kopiert eine oder mehrere Dateien.
Allgemein: cp Quelle Ziel
Das „*“-Symbol steht in der Bash-Shell für alle Dateien. „test1/*“ steht für alle Dateien, die sich im Verzeichnis test1 befinden. Diese werden in das Verzeichnis „test2“ kopiert.

`$ cd ..`

Siehe oben.
Wir befinden uns nun wieder im Home-Verzeichnis.

`$ rm -r test`

rm (remove) löscht eine oder mehrere Dateien.
Der Paramter „-r“ für „recursiv“ löscht das angegebene Verzeichnis rekursiv. Es werden alle im Verzeichnis enthaltenen Dateien **und** Verzeichnisse gelöscht und am Ende auch das angegebene Verzeichnis selbst.
Alternativ gibt es den Befehl „rmdir“ (remove directory). Dieser kann jedoch nur auf leere Verzeichnisse angewendet werden.

Aufgabe 3

Umgang mit Pack- und Filterprogrammen:

- Entpacken Sie im ersten Schritt die Datei ordner1.tar.gz
  (siehe `gzip` und/oder `tar`). Wieviele Dateien wurden erzeugt und welche Informationen können Sie über diese Dateien herausfinden?
- Versuchen Sie anschließend herauszufinden, was in der Datei 1.txt steht
  (siehe `cat, less, head, tail`).
  Welches Wort steht in der ersten und letzten Zeile?
- Sortieren Sie die Datei aufsteigend in alphabetischer Reihenfolge
  (siehe `sort`).
- Wie könnte die Datei absteigend sortiert werden (siehe `man sort`, welche Option kehrt die Ausgabe um)?
- Finden Sie heraus, wieviele Zeilen, Wörter und Buchstaben die Datei enthält (siehe `wc`).
- Das Kommando wc zeigt automatisch an, wieviele Zeilen, Wörter und Zeichen die Datei enthält. Wie kann man mit wc nur die Anzahl der Wörter einer Datei anzeigen (siehe `man wc`)?

$ tar -xzvf ordner1.tar.gz

tar ist ein Programm zum Erstellen von Dateiarchiven.
Die Paramter:
„-x“ für extract (entpacken).
„-z“ da das Archiv zusätzlich gzip-komprimiert wurde (zu erkennen an „.gz“).
„-v“ für verbose, eine ausführliche Ausgabe von tar.
„-f“ für die Angabe der Datei, die entpackt werden soll (hier „ordner1.tar.gz“).

```
$ ls -hR ordner1
```

ls (list directory contents) listet den Inhalt eines Verzeichnisses auf.
Die Paramter:
„-R“ für rekursives Auflisten (befinden sich Verzeichnisse in „ordner1“ werden rekusiv auch deren Inhalte ausgegeben).
„-h“ für „human readable“. Dateigrößen werden in „menschenlesbaren“ Einheiten wie KB, MB oder GB ausgegeben.

```
$ cat ordner1/ordner2/1.txt
```

cat (concatenate files) gibt die angegebenen Dateien nacheinander auf die Standartausgabe (unser Terminal, unsere Konsole) aus.
Hier wird der Inhalt von der Datei „1.txt“ ausgegeben.
Die Datei enthält den bekannten „Lorem ipsum“-Dummy-Text. In der ersten und letzten Zeile kommt das Wort „consetetur“ vor.
Alternativen:
head gibt den Anfang einer Datei aus.
tail gibt das Ende einer Datei aus.
more gibt eine Datei seitenweise aus (nächste Seite mit <Enter>).
less lässt einen in der Ausgabe navigieren.

```
$ cat ordner1/ordner2/1.txt | sort
```

„|“ ist die sogenannte Pipe. Diese Pipe gibt die Ausgabe von cat an die Eingabe von sort weiter.
Hinweis: Der Paramter „-r“ von sort wird oft gebraucht. Er sortiert „reverse“, d.h. in umgekehrter Reihenfolge.
Alternative: $ sort ordner1/ordner2/1.txt

```
$ cat ordner1/ordner2/1.txt | wc -l
```

wc (print word count) zählt Wörter einer Datei oder des Input über die Standardeingabe.
Wichtige Paramter:
„-l“ für lines (Zeilen).
„-w“ für words (Wörter).
„-c“ für chars (Zeichen).
Alternativ:
$ wc -l ordner1/ordner2/1.txt

Aufgabe 4

Ein-/Ausgabeumlenkung:

- Wie kann das Ergebnis von sort in eine Datei umgelenkt werden (siehe `man bash bzw. siehe Infoblatt`)?
- Wie können in einem Schritt die Wörter der ersten drei Zeilen von 1.txt gezählt werden?

```
$ cat ordner1/ordner2/1.txt | sort > \
    ordner1/ordner2/1_sortiert.txt
```

Mit Hilfe des Ausgabeumleitungsoperators „>“ kann man die Ausgabe eines Befehls in eine Datei umleiten.
„>“ überschreibt dabei die Datei, falls sie schon existiert.
„>>“ hängt die Ausgabe an eine bestehende Datei an.
Hinweis: Der „\“ steht hier nur für den Zeilenumbruch und gehört nicht zum eigentlichen Befehl.

```
$ head -n 3 ordner1/ordner2/1.txt | wc -w
```

head gibt den Anfang einer Datei aus. Mit „-n 3“ gibt man die Anzahl der Zeilen an, hier 3. Die Ausgabe wird über die Pipe an „wc -w“ weitergegeben, welches die Wörter zählt und ausgibt.