

## – Praktikumsaufgabe 2 –

### Thema: *Textarbeit unter Unix, einfache Skripte*

**Zielstellung:** Kennenlernen wichtiger Kommandos zur Arbeit mit ASCII-Texten, Umleitung von Ein- und Ausgabe, Verkettung von Kommandos mittels Pipe-Operator, Schreiben eines ersten Shellskriptes

1. Weitere wichtige Kommandos für die Arbeit mit Dateien sind `rm`, `cp`, `mv`, `rmdir` und `cat`. Machen Sie sich mit diesen vertraut, indem Sie ein paar Verzeichnisse anlegen, Dateien hineinkopieren und wieder löschen. Vergessen Sie nicht, auch einmal in die zugehörigen man-Pages zu schauen.
2. Lassen Sie sich das Konzept von Standardeingabe, Standardausgabe, Ausgabeumleitung und Verkettung mittels Pipe-Operator (`|`) erläutern.
3. Erarbeiten Sie selbständig, wozu die Kommandos `find`, `grep` und `sort` und `wc` benutzt werden.
4. Nutzen Sie `find`, um herauszufinden, wieviele verschiedene Dateien namens `test` auf Ihrem Rechner existieren.
5. Laden Sie (z. B. mittels `wget`) unter

<http://www.informatik.htw-dresden.de/~robge/bs1/prak/bibel.txt>

den Text der Einheitsübersetzung der Bibel<sup>1</sup> herunter.

- a) Wie ist die Datei strukturiert?
  - b) Aus wieviel Zeilen und wieviel Worten besteht der Text?
  - c) Ermitteln Sie, wievielmals „Schlange“, „Maus“ bzw. „Löwe“ im Text vorkommen.
  - d) Nutzen Sie das Kommando `cut`, um jeweils nur das erste Wort jeder Zeile (also z. B. „Gen“ in Zeile 1) anzuzeigen. Hinweis: Sie müssen mit der Option `-d' '` das Leerzeichen als Trenner explizit angeben.
  - e) Sortieren Sie alle Sätze alphabetisch. Hinweis: Sie müssen die Kapitelangabe jeweils abschneiden (`cut` haben wir gerade gelernt) und den verbleibenden Rest sortieren.
6. Schreiben Sie ein einfaches Shellskript, das das Zählen der Tiernamen „Schlange“, „Maus“ und „Löwe“ in der Bibel automatisiert.

---

<sup>1</sup>Weder will ich Sie zum Glauben hin- noch von diesem wegführen. Der Text ist aber ideal für unser Praktikum: wohlstrukturiert, ziemlich groß, und keiner weiß *genau*, was wo steht.

7. Da es witzlos ist, immer das gleiche zu zählen, erweitern Sie Ihr Skript um eine Parameterübergabe:

- a) Das zu zählende Tier soll dem Kommando übergeben werden.
- b) Zusätzlich soll der Name der zu analysierenden Datei übergeben werden.

**Hinweis:** Um innerhalb des Shellskriptes auf übergebene Kommandozeilenparameter zuzugreifen, gibt es die vordefinierten Variablen \$1, \$2, ... \$9, die bei Abarbeitung durch die entsprechenden konkreten übergebenen Werte, die jeweils durch Leerzeichen voneinander getrennt sind, substituiert werden. Rufen wir also z. B. das Shellskript `foo` so auf:

```
robge@ipo51:~$ ./foo Das ist ein Satz.
```

, so sind *innerhalb* des Skriptes folgende Variablen mit folgenden Werten definiert:

<i>Variable</i>	<i>Wert</i>
\$1	Das
\$2	ist
\$3	ein
\$4	Satz.

- 8.\* Sie benötigen für Testzwecke eine Datei, die genau 1 MByte zufällige Daten enthält. Wie können Sie diese anlegen? (Hinweis: Nutzen Sie `dd` und das Pseudogerät `/dev/urandom`<sup>2</sup>) Wie gehen Sie vor, wenn stattdessen, die Datei nur Nullbytes enthalten soll?
9. Lösen Sie Aufgabe 7. aus dem 1. Praktikum, jedoch als Shell-Skript. (Achtung! Kleiner Scherz!)
- 10.\* Erweitern Sie die Aufgabe 7. aus dem 1. Praktikum, indem Sie beliebige Unterverzeichnisse einbeziehen.

---

<sup>2</sup>Achtung: es gibt auch noch das Gerät `/dev/random`, was einen kryptografisch sicheren Zufallszahlengenerator implementiert, jedoch viel (für unsere Zwecke zu langsam) langsamer arbeitet.