Übungsblatt I

Brigitte Krenn, ÖFAI
brigitte@ofai.at
`http://www.oefai.at/~brigitte`

1 Einleitung

In den Übungen sollen Textbearbeitungsaufgaben unter Benutzung von einfachen Unix/Linux-Kommandos gelöst werden.

2 Starthilfen

Nachfolgend eine Liste von Befehlen, die wir zum Arbeiten unter Linux/Unix brauchen:

```
ls
cd
mkdir
pwd
less
info
man
apropos
which
rm
vim
```

Die folgenden Befehle brauchen wir vor allem bei großen Files:

```
gzip (bzip2)
zcat (bzcat)
```

Aufgabe: Benutzen Sie den Befehl **info** bzw. **man**, um festzustellen, was die oben gelisteten Befehle tun. Geben Sie für jeden Befehl eine kurze Beschreibung (max. 1 Zeile pro Befehl).

Aufgabe: Was machen

```
ls -la
gzip -d
bunzip2
rm -r
```

Was wir auch noch brauchen:

```
|           pipe
| less      pipe nach less
<           lesen vom input file
>           schreiben in output file
```

3 Zugriffe auf Zeilen

Aufgabe: Wählen Sie die Sätze 1 - 10 aus dem File `~a9700000/vu_lm/Korpora/negra-corpusv2/negra-corpus.sent` aus. Benutzen Sie **tail, sed,** |.

Aufgabe: Wählen Sie die Sätze 20 595 - 20 602 aus dem File `~a9700000/vu_lm/Korpora/negra-corpusv2/negra-corpus.sent` aus. Benutzen Sie **tail**.

4 Zerlegen eines Eingabestrings mittels tr

Aufgabe: Extrahieren Sie Sätze 1 - 10 aus `negra-corpus.sent`, zerlegen Sie die Sätze und geben Sie sie im Format "ein Wort pro Zeile" aus. Satzzeichen sollen wie Wörter behandelt werden. Schreiben Sie das Ergebnis in ein File namens `negra.word`.

5 Sortieren von Wortlisten mit sort, uniq

Aufgabe: Sortieren Sie `negra.word`

- nach lexikalischer Ordnung, d.h.: und > verhelfen > vermischen > viele > zu etc. (verwenden Sie **sort**);
- sodaß jedes Wort nur einmal gelistet wird (verwenden Sie **sort**);
- sodaß für jedes Wort seine Auftretenshäufigkeit angegeben wird (verwenden Sie **sort** und **uniq**); schreiben Sie die derart sortierte Wortliste in ein file namens `negra.word.sort`;
- nach numerischer Ordnung:
  - a) absteigend, d.h.: höherfrequente Wörter kommen vor niederfrequenten;
  - b) aufsteigend, d.h.: niederfrequente Wörter kommen vor höherfrequenten.

6 Zeilenauswahl mittels egrep
Zählen von Zeilen mittels wc

Aufgabe: Extrahieren Sie aus `negra.word.sort` nur die Zeilen, die echte Wörter enthalten, also keine Satzzeichen oder Zahlen.

Aufgabe: Wieviele echte Wörter gibt es im File? Benutzen Sie **wc** zum Zählen.

Aufgabe: Überprüfen Sie, ob das File `negra-corpus.sent` tatsächlich 20 602 Sätze beinhaltet. Beachten Sie: Ein Satz endet mit einem Punkt, Frage oder Ausrufezeichen.

Benutzen Sie **tail, egrep, wc**.

7 Manipulation von Zeilen und Spalten mittels mawk

Awk ist eigentlich eine allgemeine Programmiersprache, die sich aber besonders für das Bearbeiten von Zeilen und Spalteninformation in Files eignet. Mawk ist ein Interpreter für awk.

- Selektion von Spalten pro Zeile im File mittels $ <Spaltennummer>:

$1	1. Spalte
$NF	letzte Spalte
$(NF-1)	vorletzte Spalte

- einige Operatoren

  ```
  >, <, >=, <=, ==, =!, etc.
  ```

  ```
  mawk '$1 > $2 {print $3}' filename
  ```

  z.B. überprüft, ob der Wert (numerisch) in Spalte 1 ($1) größer ist als der in Splate 2 ($2). Wenn ja, wird der Inhalt von Splate 3 ($3) nach STDOUT geschrieben.

  Mehr Information zu mawk gibt es über **info mawk**.

- Die Option **-F** dient zur Spezifizierung von Trennzeichen zwischen den Spalten, z.B. **-F:** definiert **:** als aktuellen Spaltentrenner. Default ist das Leerzeichen.

Aufgabe: Wenden Sie folgende Kommandos

```
mawk -F: '{print $2}'
mawk -F':\t' '{print $2}'
```

auf negra-corpus.sent an. Erklären Sie die Funktionsweise der Kommandos.