## Übungen zu Computergrundlagen WS 2013/2014

# Übungsblatt 13: Programmieren in C

23.1.2014

### Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
  - **Dienstag, 28.1.2014, 10:00** für die Übungsgruppen am Donnerstag und Freitag
  - **Mittwoch, 29.1.2014, 10:00** für die Übungsgruppen am Montag und Dienstag
- Schickt die Lösungen bitte per Email an Euren Tutor.

### Aufgabe 13.1: Wörter zählen in C (10 Punkte)

Ziel dieses Übungsblattes und Teil des nächsten Übungsblattes ist es, das Pythonskript `/group/cgl/2013/13/occurrence.py` in C zu implementieren. Das Pythonskript zählt die Häufigkeit der verschiedenen Wörter in einem Text und gibt diese aus.

In der letzten Version wird das C-Programm deutlich länger sein, als das Pythonskript und es wird trotzdem nicht ganz dessen Fähigkeiten haben. Beispielsweise wird es seine Ergebnisse nicht nach Häufigkeit sortiert ausgeben.

**Hinweise:**

- Zum Testen des Programmes könnt Ihr die Dateien `/group/cgl/2013/13/gpl-3.0.txt` (General Public License) und `/group/cgl/2013/13/mobydick.txt` (Moby Dick) verwenden.
- Dokumentation der meisten C-Bibliotheksfunktionen bekommt man über den `man`-Befehl. So zeigt der Befehl `man getline` die Hilfe für die C-Funktion `getline` an.
- Zum Kompilieren Eures C-Programms verwendet am besten den Befehl
  `gcc -std=c99 -O3 -o occurrence -lm occurrence.c`

### Aufgaben

- **13.1.1** Schreibt ein C-Programm, das eine Datei zeilenweise einliest und den Inhalt auf den Bildschirm ausgibt. Der Dateiname soll auf der Kommandozeile übergeben werden können. (2 Punkte)

  **Hinweise:**

  - Verwendet den Befehl `getline`.
  - Lest den Abschnitt "Example" in der Manpage von `getline`. Er zeigt ein Beispielprogramm.
  - Vergesst nicht die Zeile `#define _GNU_SOURCE` am Anfang des Programmes!
- **13.1.2** Erweitert das Programm aus der vorigen Aufgabe so, dass es die einzelnen Zeilen mit Hilfe der Funktion `strtok` in einzelne Wörter zerlegt. Verwendet als *delim* dafür die Zeichenkette `delimiters` aus dem Pythonskript. (1 Punkt)

- **13.1.3** Erweitert das Programm aus der vorigen Aufgabe so, dass es die einzelnen Wörter mit Hilfe der Funktion `tolower` vollständig in Kleinbuchstaben umwandelt. (2 Punkte)
- **13.1.4** Erweitert das Programm aus der vorigen Aufgabe so, dass es die verschiedenen Wörter zählt und das Ergebnis ausgibt. (4 Punkte)

  **Hinweise:**

  - Definiert ein **`struct`**, das jeweils ein Wort und einen Zähler für seine Häufigkeit enthält.
  - Erzeugt einen dynamischen Array aus diesen **`struct`**s mit den verschiedenen Zählern.
  - Wenn ein Wort im Text gefunden wird, dann muss das Programm zunächst schauen, ob das Wort bereits in der Liste ist. Wenn das der Fall ist, muss der entsprechende Zähler erhöht werden.
  - Wenn das Wort nicht existiert, muss der Array erweitert werden und ein neues **`struct`** an die Liste angehängt werden.
  - Caveat Emptor! Wenn das Wort in das neue struct eingetragen wird, muss die Funktion `strdup` verwendet werden, sonst sind die Ergebnisse vermutlich etwas seltsam.
- **13.1.5** Messt die Laufzeit des Pythonskripts und des C-Programmes. Welches davon ist schneller? Warum? (1 Punkt)