Universität Trier
Fachbereich IV – Informatik
Dr. Michael Ley

Trier, den 5.5.2017

# C-Übungsblatt zum kleinen Studienprojekt (Praktikum)
## Sommersemester 2017

**Abgabe bis 26.6.2017 14 Uhr an `ley@uni-trier.de` Betreff: `Praktikum`**

Geben Sie Ihren Namen an!!! Bei allen Aufgaben müssen lauffähige Programme abgegeben werden. Die Funktionalität sollte jeweils durch einige Testfälle demonstriert werden. Ergänzendes Material zu dieser Übung finden Sie unter `http://dblps.uni-trier.de/~ley/kp17/` und auf dem Informatik-CIP-Pool unter `~ley/Cexamples`. Verwenden Sie als Mail-Anhänge nur `.txt`, `.c`, `.h` oder (später) `.java` Dateien, jedoch KEINE `.tar`, `.zip`, `.jar`, `.rar`, usw. Dateien.

### Ü 1: trim — 20 Punkte

Implementieren Sie zwei Funktionen `char *trim...(char *s)`, die aus dem String `s` eventuell am Anfang oder Ende stehende Leerzeichen, Tabs etc. entfernt. Schauen Sie sich als Vorbild die Java-Methode `String.trim` an. Geben Sie zwei Versionen vom `trim...()` an: (1) `trimInPlace()` überschreibt den ursprünglichen String mit dem neuen String, d.h. der ursprüngliche String geht verloren, es wird jedoch kein neuer Speicherplatz gebraucht. (2) `trimCopy()` erzeugt einen neuen String, der ursprüngliche String bleibt erhalten.

### Ü 2: utf8 String Länge — 10 Punkte

Unicode ist ein 21-Bit Code, der die Zeichen aller bekannten Schriftsysteme vereinigen soll. UTF-8 definiert die Speicherung oder Übertragung von Sequenzen von Unicode-Zeichen in Byte-Sequenzen. Im "Request for Comments" (RFC) 3629 finden Sie die genaue Spezifikation des UTF-8 Formats.

Implementieren Sie ein Funktion `int utf8strlen(char *s)`, die die logische Länge eines utf8-kodierten Strings, d.h. die Anzahl der Unicode-Codepunkte, berechnet.

Beim String "`hallo`" liefern `strlen()` und `utf8strlen()` denselben Wert, bei den Strings "`Längentest`", "`daß`", "`strange: \xf0\x9f\x9a\xbe\xd0\xbb\xf0\x9f\x8d\x9f`", ... sollten sich die Werte unterscheiden.

### Ü 3: Word Clouds für Informatik-Zeitschriften — 60 Punkte

`dblp.xml` ist eine große ($> 1.8 GBytes$) xml/Text-Datei, die alle bibliographischen Sätze (Records) des dblp-Literaturservers enthält.

Für Ihre Aufgabe sind nur Records relevant, die Zeitschriftenveröffentlichungen beschreiben. Solche `article`-Records beginnen stehts mit einer Zeile der Form <`article` *`Attribute`*>, also z.B. <`article mdate="2015-04-14" key="journals/cacm/Vardi15"`>. Der Wert des immer vorhandenen `key`-Attributs wird benötigt, andere Attribute sind für die Aufgabe unwichtig. Jeder `article`-Record endet mit einer Zeile der Form <`/article`>.

Der Aufbau der Schlüssel (keys) der Records entspricht der Pfad-Syntax von Unix-Dateinamen. Beachten Sie nur `article`-Records, deren Schlüssel mit `journals` beginnt und aus drei Teilen besteht. Der zweite Teil, also in unserem Beispiel `cacm`, bezeichnet eine Zeitschrift. Alle

Records mit Schlüsseln `journals/cacm/*` sind also in der Zeitschrift `cacm` erschienen. Die zu einer Zeitschrift behörenden Records stehen in der dblp-Datei direkt hintereinander.

In jedem (`article`-)Record stehen ein variable Anzahl von Feldern, wie `author`, `title`, `year`, `pages` usw. Uns interessieren hier jedoch nur `title`-Felder. Sie haben den Aufbau <`title`>*Titel*</`title`>. Im *Titel* kann es vereinzelt XML-Tags geben, die Sie jedoch ignorieren können. In einem Record gibt es jeweils nur ein `title`-Feld.

Ihr Programm soll die Erzeugung von „Word Clouds“ für Informatik-Zeitschriften vorbereiten. Für die Erstellung einer „Word Cloud“ gibt es fertige Software und Internetdienste. Die „Word Cloud“-Generatoren benötigen als Eingabe jeweils ein Liste der wichtigsten Worte mit ihren Häufigkeiten, z.B.

```
30 database
20 SQL
10 Oracle
...
```

Ihr Programm soll die komplette `dblp.xml` lesen und für jede „große“ Zeitschrift (z.B. $> 500$ Aufsätze) eine zur Erzeugung einer World Cloud geeignete Statistik ausgeben.

Hinweise: Sie müssen die Titel in Worte zerlegen und pro Zeitschrift eine Wort-Statistik erstellen. Sie sollten bei der Ausgabe die Länge der Listen beschränken, also nur die häufigsten $x$ Worte berücksichtigen. Sie können einige Word Clouds erzeugen, dies ist jedoch nicht Teil der Aufgabe.