ulm university universität
uulm

Institut für Angewandte Informationsverarbeitung
Dr. Andreas F. Borchert
Stefan Lenz

31. Januar 2013
Blatt 12

Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 7. Februar 2013, 16:00 Uhr

Lernziele:

- Sichere Strings mit der Stralloc-Bibliothek
- Hashtabellen

Aufgabe 17: Komponistenkette (unbewertet)

Auf der Homepage findet Ihr die Datei `composers.txt`, in der sich die Namen sowie Geburts- und Todesjahr einiger Komponisten befinden. (Quelle: engl. Wikipedia)

Ziel der Aufgabe ist es, ein Spiel zu programmieren, in dem man nacheinander verschiedene Komponisten nennen soll, wobei die Lebensspannen zweier hintereinander Genannter sich jeweils überschneiden müssen. Zu Beginn kann man mit einem beliebigen Komponisten starten. Falls mehrere Komponisten mit dem gleichen Nachnamen in der betreffenden Zeitspanne gelebt haben, soll dies nicht akzeptiert werden und eine entsprechende Meldung ausgegeben werden. Zusätzlich zum Nachnamen soll in der Kollisionsliste auch nach dem Beginn des Vornamens gesucht werden können.

Um eine möglichst schnelle Suche zu gewährleisten, sollen die Daten aus der Datei beim Start des Programms in eine Hashtabelle eingelesen werden, wobei als Schlüssel der Nachname verwendet werden soll. (Eine kurze Wiederholung von Hashtabellen findet Ihr unten auf dem Blatt.) Das Programm soll bis zum Eingabeende von der Standardeingabe lesen und danach die erreichte Punktzahl, d. i. die Anzahl der erfolgreich eingegebenen Komponisten, ausgeben.

Für das Arbeiten mit Strings soll die Stralloc-Bibliothek verwendet werden. Dazu sind zum Compilieren auf Thales die Option `-I/usr/local/diet/include` und zum Linken die Optionen `-lowfat` und `-L/usr/local/diet/lib` notwendig.

Beispiel:

```
thales$ composers
6997 dead composers loaded which lived in the time from 880 to 2013.
*** Composer Chain Game ***
```

```
Start by naming a composer and then name another one
whose live span intersects that of the former composer.
You get a point for each extension of the chain. No composer
may be named twice.
Name: Bach
Bach is ambiguous.
Name: Johann Sebastian Bach
Johann Sebastian Bach 1685-1750
Name: Vivaldi
Antonio Vivaldi 1678-1741
Name: Mahler
Nobody found with that name living in the time from 1678 to 1741.
Name: Zorro
Nobody found with that name.
Name: You got 1 points.
thales$
```

Hashtabellen

Das Prinzip von Hashtabellen ist, dass Daten so in einem Array abgelegt werden, dass ihr Index durch das gesuchte Datum mittels einer Hashfunktion bestimmt werden kann. Dadurch kann das Entfernen und Ablegen von Daten - bei geeignet großem Array und geeigneter Hashfunktion - mit konstantem Aufwand erledigt werden. Da der Fall auftreten kann, dass zwei Daten auf der gleichen Stelle abgelegt werden, darf das Array die Daten nicht direkt beherbergen, sondern muss Listen beinhalten, deren Elemente dann jeweils den gleichen Hashwert haben.

Hier ein Beispiel für eine Hashfunktion, die Ihr verwenden könnt:

```
#define HASHSTART 5381

static unsigned int hashadd(int hashval, unsigned char ch) {
   hashval += hashval << 5;
   return hashval ^ ch;
}

static unsigned int compute_hash(const char* buf) {
   int hashval = HASHSTART;
   while (*buf) {
      hashval = hashadd(hashval, *buf++);
   }
   return hashval;
}
```

Der Schlüssel, aus dem der Hashwert berechnet wird, ist hier also einfach ein C-String. Der so errechnete Wert kann (modulo Größe des Arrays) als Index verwendet werden.

Viel Erfolg!