

Programmieren in C / C++

Wintersemester 2007/08

Hubert Feyrer, Florian Lutz

Übung 7: Zahlensysteme, Binärarithmetik, Module

Übung 7.1: Zahlensysteme

- Schreiben Sie zwei C-Funktionen, die sich mit der Umwandlung von int-Werten in Zeichenketten, die die Zahl als Binärzahl darstellt, beschäftigen:

```
int bintoint(char buffer[]);
```

wandelt die angegebene Zeichenkette in einen gültigen int-Wert, z.B. "10110" wird in die int-Zahl 22 konvertiert

```
int inttobin(char buffer[], int inttoconvert);
```

wandelt `inttoconvert` in eine Zeichenkette um, und speichert diese in das übergebene Array `buffer`, z.B. 127 wird in "1111111" gewandelt. Rückgabewert ist die Länge der Zeichenkette.

- Schreiben Sie äquivalente C-Funktionen `hextoint()` und `inttohex()` die analog von/nach hexadezimal konvertieren.
- Schreiben Sie ein Hauptprogramm, das die Zahlen von 0 bis $2^{32}-1$ in dezimal, hexadezimal und binär ausgibt, indem die oben geschriebenen Routinen verwendet werden.

Übung 7.2: Binärarithmetik

- Schreiben Sie eine Funktion `void kompl1(char in[], char out[])`; die als Eingabe `in` eine Zeichenkette von Binärziffern erwartet, und in `out` das Einerkomplement dieser Ziffern schreibt.
- Schreiben Sie eine Funktion `int binadd(char a[], char b[], char erg[])`; die zwei Strings, die Binärziffern enthalten, addiert. Es soll z.B. nach `binadd('1000', '0011', c)` in `c="1011"` stehen. Rückgabewert der Funktion soll ein eventueller Übertrag sein.
- Schreiben Sie mit Hilfe der Funktionen `kompl1()` und `binadd()` eine Funktion `int binneg(char num[], char neg[])`; die in `num` eine Folge von Binärziffern erwartet, und eine Zeichenkette in `neg` ablegt, die dem Zweierkomplement (1er-Komplement + 1) von `num` entspricht.

- Schreiben Sie eine Funktion `int binsub(char a[], char b[], char erg[]);`, die die in den Strings `a` und `b` enthaltenen Zeichenketten als Binärstrings interpretiert, und in `erg` das Ergebnis von `a-b` ablegt. Benutzen Sie zur Implementierung die Funktionen `binadd()` und `neg()`!
- Schreiben Sie ein Hauptprogramm, das zwei Binärzahlen in Form von Zeichenketten von der Tastatur einliest und sowohl Summe als auch Differenz dieser Zahlen bildet. Benutzen Sie dazu die oben erstellten Funktionen!
- Skizzieren Sie für drei der o.g. Funktionen den Stackaufbau inkl. Variablennamen und korrektem Platzbedarf (in Bytes)!

Übung 7.3: Modularisierung

Modularisieren Sie ihr oben geschriebenes Programm zur Addition und Subtraktion von Binärzahlen-Strings:

- Schreiben Sie ein Modul `binfunc.c`, das die Funktionen `binadd()` und `binsub()` zur Verfügung stellt
- Erstellen Sie ein Header-File `binfunc.h`, das die Prototypen der Funktionen `binadd()` und `binsub()` wie oben angegeben zur Verfügung stellt.
- Die o.g. Hilfsfunktionen für `binadd()` und `binsub()` sollen im Modul `binfunc.c` enthalten sein, jedoch nicht von ausserhalb des Moduls sichtbar sein! (\rightarrow `static`)
- Das Hauptprogramm `bincalc.c` soll die oben genannte Addition und Subtraktion ausführen!
- Erstellen Sie ein Makefile, das die Module übersetzt und linkt!