ulm university universität
uulm

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

8. November 2012
Blatt 3

Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 15. November. 2012, 16:00 Uhr

Lernziele:

- Präprozessor: `#include`, `#define`, `#undef`, `#ifdef`
- Zahlendarstellung und Bitoperationen

Aufgabe 4: Zur Funktionsweise des C-Präprozessors (6 Punkte)

Um die Funktionsweise und Möglichkeiten des C-Präprozessors genauer kennen zu lernen, wollen wir in dieser Aufgabe den Text Liedes „Let it be“ von den Beatles betrachten. Diesen wollen wir in mehrere Dateien aufspalten und komprimieren, indem wir für mehrfach vorkommende Textpassagen *Makros* definieren. Anschließend soll der C-Präprozessor wieder den Liedtext zusammensetzen. Weil der Text kein Programmcode ist und also nicht übersetzt werden soll, muss der `gcc` mit der Option `-E` aufgerufen werden.

- Sucht drei Textstücke, die mindestens einmal wiederholt werden, und ersetzt diese durch Makros. Zudem sollen Autor und Titel des Lieds durch Makros ersetzt werden. Makros können mit der Präprozessordirektive `#define` definiert werden, z. B.

  ```
  #define LIB let it be
  ```

- Die Makrodefinitionen sollen in eine eigene Datei `makros.h` verlegt werden. Der Refrain soll in der Datei `refrain.h` stehen. Verwendet aber auch dort die vorhin definierten Makros.

- Da die Strophen einen ähnlichen Aufbau haben, kann man diese ebenfalls vereinfachen. Verwendet dafür die Datei `strophe.h` von der Vorlesungshomepage, indem Ihr erst die passenden Makros definiert und danach die Datei einbindet. Die Datei `strophe.h` soll nicht verändert werden und es dürfen auch keine Warnings generiert werden. Hinweise: An die Stelle, wo `#include "datei.h"` steht, schreibt der Präprozessor den Inhalt der Datei `datei.h`. Eine Makrodefinition kann mit `#undef` entfernt werden.

- Als Parodie von „Let it be“ gibt es einen alternativen Liedtext namens „Write in C“. Hier wollen wir lediglich das Satzstück „let it be“ durch „write in C“ersetzen - allerdings nur, wenn das Makro `WRITEINC` definiert ist! Makros können nicht nur in Dateien, sondern auch auf der Kommandozeile via `gcc -DWRITEINC=...` definiert werden. (Falls der Wert des Makros weggelassen wird, ist er standardmäßig als 1 gesetzt.) Hinweis: bedingte Übersetzung mit `#ifdef`. Beispielausgabe für diese Aufgabe:

```
$ gcc -E letitbe.c
[...] let it be [...]

$ gcc -DWRITEINC -E letitbe.c
[...] write in C [...]
```

```
thales$ submit ss1 4 letitbe.c makros.h refrain.h team
```

Aufgabe 5: Umrechnung in verschiedene Zahlensysteme (2 Punkte)

Gesucht ist ein Programm, das zwei Ganzzahlen a und b (im Dezimalsystem) einliest und die Zahl a im Zahlensystem zur Basis b ausgibt. (Dabei soll $a \geq 0$ und $36 \geq b \geq 2$ gelten.) Die Buchstaben des Alphabets sollen zur Darstellung von zusätzlichen Ziffern benutzt werden, falls $b > 10$ ist. Hinweis: Verwende für die einzelnen Ziffern des Ergebnisses ein Array. Wie groß muss es sein?

```
thales$ submit ss1 5 conversion.c team
```

Aufgabe 6: Binärzahl ausgeben (2 Punkte)

Löse Aufgabe 5 für den Fall $b = 2$, diesmal aber ohne Arrays zu verwenden! Hinweis: Zuerst die Bitreihenfolge von a umdrehen.

```
thales$ submit ss1 6 digits.c team
```

Viel Erfolg!