

3. Aufgabenblatt

Abgabe 13.11.15

Allgemeine Hinweise:

- Bitte geben Sie zu jeder Aufgabe entweder die Beantwortung oder Testläufe auf Papier ab.
- Quellcode geben Sie bitte unkomprimiert und kommentiert im KVV ab.
- Beantworten Sie alle Aufgaben mit Ihren eigenen Worten.

Problem 1: Zahlensysteme

1. Konvertieren Sie die Zahlen 1234, 23.77, 256 vom Dezimal- ins Hexadezimalsystem.
2. Konvertieren Sie die Zahlen 0x77, 0x80, 0x66.01 ins Dezimalsystem.

Problem 2: Assembler-Pseudo-Codezeilen

Geben Sie nasm Assembler-Befehle zur Simulation folgender Pseudo-Codezeilen an:

1. `while X > Y do S;`
2. `if A = B then begin X:=X+1; Y:=Z end else A:=B;`
3. `for J:= LAST downto FIRST do S;`
4. `function f(A, B) return 7 + A * 2 + B * 30;`

Problem 3: Fibonacci-Zahlen

Die Fibonacci-Zahlen seien wie folgt definiert:

$\text{fib}(0) = 0$, $\text{fib}(1) = 1$, $\text{fib}(n > 1) = \text{fib}(n-2) + \text{fib}(n-1)$.

Um die n -te Fibonacci-Zahl iterativ zu berechnen ist, können drei Laufvariablen x_1 , x_2 und k benutzt werden, die zu Beginn mit 0, 1 und 0 initialisiert sind. Am Ende jedes Schleifendurchlaufes soll gelten,

- dass der neue Wert von x_1 gleich dem alten Wert von x_2 ist,
- der neue Wert von x_2 gleich dem alten Wert von k ist,
- und $k = x_1 + x_2$ ist.

Die Schleife soll n mal wiederholt werden (für $n = 0$ also keinmal, für $n = 1$ einmal, usw.).

1. Überlegen Sie sich zunächst eine Implementierung in Pseudo-Code.
2. Übersetzen Sie Ihre Lösung nach x86-Assembler.
3. *Freiwillige Zusatzaufgabe:* Implementieren Sie Ihre Lösung in C. Vergleichen Sie mit Hilfe des Benchmarks Ihre Lösung in C und Ihre handgeschriebene Assembler-Lösung!