

3. Aufgabenblatt

Abgabe 16.05.14

Hinweis zu Programmieraufgaben: Bitte machen Sie zu jeder Programmieraufgabe Testläufe, die die Funktionalität Ihrer Programme ausreichend dokumentieren. Die Testläufe und den Quellcode drucken Sie bitte aus, und geben beides mit der jeweiligen Übung auf Papier ab. Der Quellcode muss zusätzlich in Form einer assemblierbaren Datei in das Repository <https://svn.imp.fu-berlin.de/cst-teaching-schmitt/SS14/TI-2/Tut-<NUM>/grp<NUM>> eingereicht werden. Ihr jeweiliger Tutor wird Ihnen Näheres mitteilen.

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. Implementieren Sie Ihre Lösung in C.
3. Übersetzen Sie Ihre Lösung in x86-Assembler.
4. *Freiwillige Zusatzaufgabe:* Vergleichen Sie mit Hilfe des Benchmarks Ihre Lösung in C und Ihre handgeschriebene Assembler-Lösung!