

Rechnersysteme und -netze: Quiz 3

Christian Borgelt, Christoph Doell,
Heiko Dreyer, Alice Hildebrand, Josua Sattler, Simon Suckut

Winter 2017/2018
02.02.2018

Name		Übung	Immatrikulationsnummer
Aufgabe 1	Aufgabe 2	Bonusaufgabe	Summe
/ 26	/ 24	/ 6	/ 50(+6)

Aufgabe 1 Hack-Assemblersprache (4 + 4 + 8 + 6 + 4 Punkte)

- @a
M=1
@b
M=0
@R0
D=M
@i
M=D
(LOOP)
@i
MD=M-1
@STORE
D; JLT
@a
D=M
@b
MD=D+M
@a
M=D-M
@LOOP
O; JMP
(STORE)
@b
D=M
@R1
M=D
(END)
@END
O; JMP
- Gegeben sei das links gezeigte Programm in Hack-Assemblersprache.
- Welchen Inhalt hat die Symboltabelle des Hack-Assemblers am Ende der Übersetzung dieses Programms? (Geben Sie von den vordefinierten Symbolen nur die an, die auch im links gezeigten Programm auftreten!)
 - Übersetzen Sie die beiden Hack-Assemblerbefehle „MD=M-1“ und „D; JLT“, die im links gezeigten Programm auftreten, in Hack-Maschinensprache!
 - Das links gezeigte Programm werde ausgeführt, wobei zu Beginn im virtuellen Register R0 der Wert 6 stehen möge. Füllen Sie die folgende Iterationstabelle aus (es werden ggf. nicht alle Zeilen benötigt):

Zeitpunkt/Variablen	i	a	b
Initialisierung			
1. Durchlauf			
2. Durchlauf			
3. Durchlauf			
4. Durchlauf			
5. Durchlauf			
6. Durchlauf			
7. Durchlauf			
8. Durchlauf			

Die Zeitpunkte beziehen sich auf das erste bzw. jedes weitere Erreichen der Markierung (LOOP) (bzw. des durch sie markierten Befehls).
Welcher Wert steht am Ende der Ausführung im virtuellen Register R1?
 - Erläutern Sie die Berechnung des Programms, indem Sie eine Funktion `int f (int i)` in Pseudocode (oder Jack oder Java oder ...) angeben, die die Berechnung auf äquivalente Weise durchführt!

Welche Funktion berechnet das Programm allgemein?
(Beschreibung in einem Satz!)
 - Nennen Sie zwei Vor- und zwei Nachteile von Assemblersprache!

Aufgabe 2 Virtuelle Maschine (6 + 8 + 6 + 4 Punkte)

```
function Main.f 1
label LOOP
    push argument 0
    push constant 0
    lt
    if-goto STORE
    push local 0
    push constant 1
    add
    pop local 0
    push argument 0
    push argument 0
    add
    push constant 1
    add
    pop argument 0
    goto LOOP
label STORE
    push local 0
    return
```

Gegeben sei die links gezeigte Funktion in der Sprache der virtuellen Maschine des Hack-Systems.

a) Übersetzen Sie die Anweisung „`pop argument 0`“, die in der links gezeigten Funktion auftritt, in Hack-Assemblersprache!

b) Die links gezeigte Funktion werde durch die Anweisungen

```
push constant 2048
call Main.f 1
```

aufgerufen. Füllen Sie die folgende Iterationstabelle aus (es werden ggf. nicht alle Zeilen benötigt):

Zeitpunkt/Variablen	argument 0	local 0
Initialisierung		
1. Durchlauf		
2. Durchlauf		
3. Durchlauf		
4. Durchlauf		
5. Durchlauf		
6. Durchlauf		

Die Zeitpunkte beziehen sich auf das erste bzw. jedes weitere Erreichen der Markierung `LOOP`. Welchen Wert gibt die Funktion `Main.f` bei diesem Aufruf zurück?

(Hinweis: Es ist u.U. vorteilhaft, mit einer Binär- oder Hexadezimaldarstellung der Werte zu arbeiten. Dies erleichtert auch das Lösen der folgenden Teilaufgabe.)

c) Erläutern Sie die Berechnung der Funktion, indem Sie eine Funktion `int f (int i)` in Pseudocode (oder Jack oder Java oder ...) angeben, die die Berechnung auf äquivalente Weise durchführt!

Was berechnet die Funktion `Main.f` allgemein?

(Beschreibung in einem Satz!)

d) Nennen Sie zwei Vor- und zwei Nachteile von virtuellen Maschinen!

Bonusaufgabe (2 + 2 + 2 Punkte)

a) Wie lauten Titel und Untertitel des Buches, dem die Vorlesung „Rechnersysteme und -netze“ folgt?

b) Wofür stehen in der Sprache der virtuellen Maschine das n und das k in den Anweisungen

```
function f $n$ name  $n$     und    call f $n$ name  $k$     ?
```

(Antworten Sie mit „Das n steht für ...“ und „Das k steht für ...“!)

c) Zeichnen Sie einen Parse-Baum für den folgenden Jack-Ausdruck:

```
2+min(x+3*y,0)
```