

Vorlesung Systemnahe Informatik

Sommersemester 2023

Prof. Dr. Peter Martini, Dr. Matthias Frank, Lennart Buhl M.Sc.

6. Übungszettel

Ausgabe: Dienstag, 16. Mai 2023.

Abgabe: Sonntag, 21. Mai 2023

Besprechung: In den Übungen ab Montag, 22. Mai 2023.

Hinweis: Abgabe erfolgt freiwillig per PDF über eCampus, siehe Hinweise in Aufgabe 1 auf dem 1. Übungszettel
Sie können Kontakt zu Ihrer Tutor/in aufnehmen durch E-mail an `cs4+ueb-si-XX@cs.uni-bonn.de` mit `XX` als Gruppennummer.

Besonderheit: Am 24.05.2023 ist Dies Academicus, im Allgemeinen finden keine Vorlesungen und Übungen statt. Teilnehmer/innen unserer Mittwochs-Übungen können in dieser Woche in eine der anderen Präsenz-Übungen gehen, bzw. sprechen das Vorgehen mit der Tutorin ab.

Aufgabe 1: Alpha-Notation, Stacks

Bei dieser Aufgabe stehen Ihnen ausschließlich die folgenden Befehle zur Verfügung ($k \in \mathbb{Z}$, $i, j \in \{n, \bar{x}, h_1, h_2\}$, $u, v \in \{0, 1, 2\}$, $\text{op} \in \{+, -, *, /\}$):

$\alpha_u := \alpha_v$	$\alpha_0 := \alpha_0 \text{ op } \rho(i)$	<code>goto label</code>
$\alpha_u := \rho(i)$	$\alpha_0 := \rho(i) \text{ op } \rho(j)$	<code>call label</code>
$\rho(i) := \alpha_u$	$\rho(i) := \rho(j) \text{ op } \alpha_0$	<code>return</code>
$\alpha_u := k$	$\rho(i) := \rho(j)$	<code>push</code>
	if $\alpha_0 = 0$ then goto label	<code>pop</code>

Des Weiteren steht ein Stack σ zur Verfügung. Auf diesem können folgende Operationen durchgeführt werden, die auf α_0 wirken:

- push
- pop

Arithmetische Operationen (op) auf dem Stack sind nicht erlaubt!

Zusätzlich sind zwei Unterprogramme gegeben:

- NthRT
 - berechnet: $\alpha_0 := \sqrt[n]{\alpha_1}$
 - benutzt zwei zusätzliche Register (α_1 und α_2)

- POT

- berechnet: $\alpha_0 = \alpha_1^{\alpha_2}$
- benutzt zwei zusätzliche Register (α_1 und α_2)

α_1 und α_2 dürfen nur für den Aufruf der Unterprogramme bzw. in diesen genutzt werden. Der Aufrufbefehl für Unterprogramme lautet `call`. Sie dürfen beliebig viele Sprunglabel verwenden. Schreiben Sie ein Unterprogramm „SA“, welches die Standardabweichung (siehe Formel) berechnet:

$$s = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}$$

wobei \bar{x} der arithmetische Mittelwert ist.

Beachten Sie:

- Im Stack σ befinden sich die gegebenen Elemente (mindestens eins). Deren Anzahl ist in Speicherzelle n abgelegt.
- Speicherzelle „ \bar{x} “ enthalte den arithmetischen Mittelwert.
- Sie dürfen zusätzlich die Speicherzellen „ h_1 “ und „ h_2 “ benutzen.
- Das Ergebnis soll bei Rückkehr aus dem Unterprogramm in α_0 stehen.
- Benutzen Sie die Unterprogramme POT und NthRT.
- Nehmen Sie an, dass alle Unterprogramme vom Typ mehrstufig und erneut aufrufbar sind.

Kommentieren Sie Ihr Unterprogramm ausführlich!

Aufgabe 2: Alpha-Notation, Indexierung

Die Fachschaften von Chemie, Mathematik und Informatik haben bei der Orientierungseinheit für Erstsemester das „Trinerdische Turnier“ gespielt. In mehreren Runden mussten immer zwei zufällig ausgewählte Fachschaften gegeneinander antreten. Die Anzahl der Runden sowie ihr Ausgang wurde im Speicher folgendermaßen ab Adresse 100 (bzw. 200, 300) festgehalten und kann mit Speicherzugriff über das Indexregister γ ausgelesen werden.

Speicherstelle	Inhalt
$\rho(\gamma)$	2 // Anzahl der Runden
$\rho(\gamma + 1)$	3 // eigene Punkte Runde 1
$\rho(\gamma + 2)$	1 // Gegnerpunkte Runde 1
$\rho(\gamma + 3)$	0 // eigene Pkt. Runde 2
$\rho(\gamma + 4)$	2 // Gegnerpkt. Runde 2

Tabelle 1: Ergebnisse einer Beispielfachschaft

Bei dieser Aufgabe stehen Ihnen ausschließlich folgende Befehle zur Verfügung ($k \in \mathbb{Z}$, $i, j \in \{0, 1, 2, 3, 4, 5\}$, **op** $\in \{+, -, *, /\}$, **relop** $\in \{=, >, <\}$). Sie dürfen beliebig viele Sprunglabel **label** verwenden. Ihnen stehen die Register $\alpha_0, \dots, \alpha_5$ und das Indexregister γ zur Verfügung, aber keine weiteren Hilfsspeicherzellen.

$\alpha_i := k$	<i>if α_i relop α_j then goto label</i>
$\alpha_i := \alpha_j$	call label
$\alpha_i := \alpha_j$ op k	goto label
$\alpha_i := \rho(\gamma)$	return
$\alpha_i := \gamma$	
$\gamma := \alpha_i$	

- a) Gehen Sie davon aus, dass die Gesamtpunkte wie beim Fußball berechnet werden: der Gewinner einer Runde erhält 3 Punkte, der Verlierer keinen und bei einem Unentschieden erhält jeder einen. Implementieren Sie in Alpha-Notation ein Unterprogramm, das die Gesamtpunkte einer Fachschaft berechnet und in α_4 schreibt. Kommentieren Sie Ihren Code!
- b) Benennen Sie die Art des von Ihnen verwendeten Unterprogramms und erklären Sie, warum Sie sich für diese entschieden haben.

Aufgabe 3: Java Bytecode, Mnemonics

Gegeben sei der folgende Bytecode-Strom.

03 3d 03 3c a7 00 0c 1c 1b 1a 68 60 3d 84 01 01 1b 10 0a a1 ff f4 1c ac

Wandeln Sie den Bytecode-Strom in die mnemonische Darstellung des Bytecodes (z.B. `iconst_0`, `imul`, ...) um. Bei Ihrer Darstellung muss erkenntlich sein, welche Bytes einem Befehl in mnemonischer Darstellung zugeordnet sind.

Aufgabe 4: Java Bytecode, Interpretation

Gegeben sei der Bytecode einer Java Methode `int doIt(int[])` (vgl. Tabelle 2).

Tabelle 2: Java-Bytecode zu Methode `doIt(int[])`.

0: <code>aload_0</code>	15: <code>iload_1</code>
1: <code>iconst_0</code>	16: <code>if_icmple 23</code>
2: <code>iaload</code>	19: <code>aload_0</code>
3: <code>istore_1</code>	20: <code>iload_2</code>
4: <code>iconst_1</code>	21: <code>iaload</code>
5: <code>istore_2</code>	22: <code>istore_1</code>
6: <code>iload_2</code>	23: <code>iinc 2 1</code>
7: <code>bipush 10</code>	26: <code>goto 6</code>
9: <code>if_icmpge 29</code>	29: <code>iload_1</code>
12: <code>aload_0</code>	30: <code>bipush 42</code>
13: <code>iload_2</code>	32: <code>imul</code>
14: <code>iaload</code>	33: <code>ireturn</code>

- a) Was berechnet diese Methode? Formulieren Sie diese Methode in Java und ordnen Sie, sofern dies möglich ist, die Bytecode-Befehle den einzelnen Zeilen ihres Java-Codes zu.

b) Welchen Wert berechnet `doIt()` bei Übergabe eines Integer-Feldes a , das wie folgt initialisiert wird?

i) `int a[] = {0, 1, 1, 2, 3, 5, 8, 13, 21}`

ii) `int a[] = {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}`

iii) `int a[] = {5, 8, 31, 9, 21, 6, 18, 14, 7, 34, 55}`

Aufgabe 5: IT-Sicherheit - Zusatzaufgabe

Seit einigen Jahren gibt es zusätzlich zu den normalen Übungsaufgaben für Systemnahe Informatik einige zusätzliche Aufgaben mit dem Schwerpunkt IT-Sicherheit. Diese Aufgaben gehen über den Inhalt der Vorlesung hinaus und sind somit nicht prüfungsrelevant. Vielmehr möchten wir Anwendungsgebiete vermitteln, wofür Sie die Grundlagen benötigen, die Sie in der Systemnahen Informatik lernen.

Wir versuchen, praxisnahe Aufgaben zu formulieren, an denen Sie tüfteln und herumprobieren können. Manchmal gibt es dafür auch mehrere Lösungen. Wenn Sie bei einer Aufgabe keinen Ansatz finden, fragen Sie uns gern per E-Mail oder sprechen uns persönlich an. Wir freuen uns auf viele Ergebnisse und Ideen.

Nun zur Aufgabe: Finden Sie den Klartext zu folgendem Chifftrat!

Uremyvpura Tyüpxjhafpu,

Fvr unora qvr refgr Grvynhstnor tryöfg. Qnf jne thgr Neorvg! Jve fvaq nhs qre Fhpur anpu thgra xelcgybvtvpura Iresnuera, hz qvr Fvpureurvg hafrere Trfpuäsgfqngra mh trjäuey-rvfgra. Xraara Fvr täatvtr Irefpuyüffryhatfiresnuera haq xöaara haf qvrfr ibefpuynta? Fpuvpxra Fvr Vuer Ibefpuyätr na qra/qvr Ghgbe/va Vuere Hrohatftehccr na pf4+hro-fv-KK@pf.hav-obaa.qr.