

Rechnerarchitektur im Sommersemester 2017

Übungsblatt 7

Abgabetermin: 19.06.2017, 12:00 Uhr

Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 12. – 16. Juni 2017
Besprechung der H-Aufgaben in den Tutorien vom 19. – 23. Juni 2017

Aufgabe 34: (T) Stackzustände

(– Pkt.)

- a. Zeichnen Sie den Stack, wie er nach der Ausführung des folgenden MIPS-Codefragments aussieht. Tragen Sie auch den Stackpointer in Ihre Skizze ein. Welches Problem liegt vor?

```
li    $t0, 1
sw    $t0, ($sp)
li    $t0, 2
sw    $t0, ($sp)
```

- b. Skizzieren Sie alle Inhalte des Stacks nachdem die folgenden Programmzeilen ausgeführt wurden. Kennzeichnen Sie auch die Position des Stackpointers!

```
li    $t0, 1
li    $t1, 2
li    $t2, 3
addi  $sp, $sp, -20
sw    $t0, 20($sp)
sw    $t0, 16($sp)
sw    $t0, 12($sp)
sw    $t1, 8($sp)
sw    $t2, 4($sp)
```

- c. Welche Register sollten *generell* bei einem rekursiven Unterprogrammen mindestens gesichert werden? Wie geht man hierzu vor? (Kurze Antwortsätze!)
- d. Die Fakultätsfunktion $fac(n) = n!$ ist bekanntlich folgendermaßen definiert:

$$fac(n) = \begin{cases} n * fac(n - 1) & : n \geq 1 \\ 1 & : n = 0 \end{cases}$$

Auf der Website zur Vorlesung steht eine unkommentierte SPIM-Implementierung dieser Funktion zum Download bereit.

Bearbeiten Sie anhand dieser Implementierung die folgenden Aufgaben:

- (i) Kommentieren Sie das Programm ausführlich!

- (ii) Skizzieren Sie nun den Inhalt des Stacks für den Fall, dass das obige Programm mit dem Parameter 3 aufgerufen wird. Visualisieren Sie dazu jedes mal den Inhalt des Stacks, unmittelbar nachdem das Programm die Zeile 45, 52 bzw. 62 ausgeführt hat.

Aufgabe 35: (T) Parameterübergabe bei Unterprogrammaufrufen (– Pkt.)

Für die Parameterübergabe bei Prozeduraufrufen existieren verschiedene Möglichkeiten.

- a. Erläutern Sie zunächst die Begriffe *call by value* und *call by reference*. Geben Sie zu beiden Konzepten je ein Beispiel in einer Hochsprache an.
- b. Schreiben Sie nun ein SPIM-Programm, das den Durchschnitt der Werte eines Feldes berechnet. Die Berechnung selbst soll dabei ein Unterprogramm erledigen. Die Übergabe des Feldes soll nach dem Konzept *call by value* erfolgen.

Achtung: Das Hauptprogramm soll dem Unterprogramm **alle** zur Berechnung notwendigen Werte über den Stack zur Verfügung stellen! Sie dürfen bei Ihrer Implementierung davon ausgehen, dass sich das Feld bereits im Speicher befindet.

- c. Schreiben Sie Ihr Programm aus Aufgabe b) so um, dass die Übergabe des Feldes nach dem Konzept *call by reference* funktioniert.

Achtung: Das Hauptprogramm soll dem Unterprogramm **ausschließlich** Speicheradressen zur Berechnung zur Verfügung stellen! Sie dürfen wieder davon ausgehen, dass sich das Feld bereits im Speicher befindet. Sie dürfen zur Übergabe der Adressen an das Unterprogramm die laut Konvention dafür vorgesehenen Register \$a0 - \$a3 verwenden. Das Ergebnis des Unterprogrammaufrufes dürfen Sie dem Hauptprogramm über das Register \$v0 zur Verfügung stellen.

Aufgabe 36: (T) Anwendungen der Digitalisierung

(– Pkt.)

In der Vorlesung „Rechnerarchitektur“ werden Grundlagen der Digitalisierung behandelt. Während diese in Zeiten der Entstehung erster Rechnerarchitekturen zunächst die Hardware betrafen, vollzog sich sukzessive eine Erweiterung hin zur Software und dann zur Digitalisierung von Prozessen und ganzen Branchen. Die Auswirkungen sind heute in unserem Leben und Arbeiten, in den Medien und unserer Gesellschaft sichtbar. Diesen Aspekt wollen wir mit Hilfe des Buchs „Marktplätze im Umbruch“ betrachten.

Beantworten Sie für jedes der folgenden Geleitworte die Frage:

Welche Beobachtungen macht der jeweilige Autor im Zeitalter der Digitalisierung in seinem Arbeitsumfeld?

- Susanne Steidl: Zukunft des Bezahlens – Mobile Technologien im Handel
- Lothar Borrmann: Software-Eco-Systeme

Lesen Sie des Weiteren den folgenden Artikel und beantworten Sie die Frage:

Welche Zukunft sieht der Autor für soziale Medien und welche Auswirkungen ergeben sich für die Marketingstrategie von Unternehmen.

- Patrick Möbert: Chancen und Grenzen des „Social Business“

Hinweis: Als Student können Sie sich ein freies Exemplar des Buches „Marktplätze im Umbruch“ aus dem Internet herunterladen. Dabei haben Sie die folgenden 2 Möglichkeiten:

- a. Rufen Sie aus dem LRZ-Netz den folgenden Link zum Buch auf: <http://link.springer.com/book/10.1007/978-3-662-43782-7>. Verwenden Sie dabei den PAC-Proxy (<https://www.lrz.de/services/netzdienste/proxy/zeitschriftenzugang/>)
- b. Rufen Sie den folgenden Link zum Buch auf: <http://link.springer.com/book/10.1007/978-3-662-43782-7>. Gehen Sie auf „Sign up/ Login“ und dort auf „Log in via Shibboleth or Athens“. Unter „find your institution“ geben Sie „LMU“ ein und klicken Sie auf „Log in via Shibboleth“. Es folgt die Weiterleitung zur LMU und der Login mit Ihrer Campus-Adresse. Nach Bestätigung können Sie auf der Springer-Seite das Buch herunterladen.

Aufgabe 37: (H) Assemblerprogrammierung unter SPIM

(8 Pkt.)

Laden Sie sich das Programm *simple-counter.s* von der Vorlesungs-Webseite herunter und beantworten Sie die folgenden Fragen zum Thema Assemblerprogrammierung des MIPS-Prozessors.

- a. Ordnen Sie dem Programm *simple-counter.s* jeder Zeile, die mit “# Kommentar <nr> Nr. :” versehen ist, den jeweils sinnvollsten der folgenden Kommentare zu. Ein Kommentar kann auch mehrfach benötigt werden. Schreiben Sie dazu eine Auflistung mit der Kommentarnummer und der Zuordnung zu jeweils einem der folgenden möglichen Kommentare (bspw.: Kommentar 13 Nr.: iii):
 - (i) Schaffe Platz auf Stack
 - (ii) while i > 0
 - (iii) Übergebe Argument
 - (iv) Sichere i auf Stack
 - (v) Lade i vom Stack
 - (vi) i := 10
 - (vii) Sichere übergebenes Argument
 - (viii) i := i - 1
 - (ix) Setze Stackgröße zurück
- b. Geben Sie die letzten drei Zeilen an, die das Programm aus Teilaufgabe a) auf der Konsole ausgibt.
- c. Unabhängig von Teilaufgabe a) sei nun die folgende Befehlssequenz gegeben:

```
1      li $t0, 5
2      li $t1, 32
3      li $t2, 7
4      addi $sp, $sp, -16
5      sw $t0, 12($sp)
6      sw $t1, 8($sp)
7      sw $t2, 4($sp)
```

Erläutern Sie, welchen Nachteil diese Befehlssequenz bezüglich des Stacks besitzt und geben Sie die dafür verantwortliche Zeile an. Welche Änderung wäre zur Behebung des Problems nötig?

Aufgabe 38: (H) Einfachauswahlaufgabe: Boolesche Algebra und SPIM

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Sei folgende Wahrheitstafel einer Booleschen Funktion $f : B^2 \rightarrow B$ gegeben. Welcher Ausdruck entspricht der disjunktiven Normalform (DNF)?																							
<table border="1"> <thead> <tr> <th>i</th> <th>x_1</th> <th>x_2</th> <th>$f(x_1, x_2)$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>				i	x_1	x_2	$f(x_1, x_2)$	0	0	0	1	1	0	1	0	2	1	0	0	3	1	1	1
i	x_1	x_2	$f(x_1, x_2)$																				
0	0	0	1																				
1	0	1	0																				
2	1	0	0																				
3	1	1	1																				
(i) $f(x_1, x_2) = \bar{x}_1\bar{x}_2 + x_1x_2$	(ii) $f(x_1, x_2) = (x_1 + \bar{x}_2) \cdot (\bar{x}_1 + x_2)$	(iii) $f(x_1, x_2) = (\bar{x}_1 + x_2) \cdot (x_1 + \bar{x}_2)$	(iv) $f(x_1, x_2) = \bar{x}_1x_2 + x_1\bar{x}_2$																				
b) Sei folgende Wahrheitstafel einer Booleschen Funktion $f : B^2 \rightarrow B$ gegeben. Welcher Ausdruck entspricht der konjunktive Normalform (KNF)?																							
<table border="1"> <thead> <tr> <th>i</th> <th>x_1</th> <th>x_2</th> <th>$f(x_1, x_2)$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>				i	x_1	x_2	$f(x_1, x_2)$	0	0	0	1	1	0	1	0	2	1	0	1	3	1	1	0
i	x_1	x_2	$f(x_1, x_2)$																				
0	0	0	1																				
1	0	1	0																				
2	1	0	1																				
3	1	1	0																				
(i) $f(x_1, x_2) = \bar{x}_1\bar{x}_2 + x_1\bar{x}_2$	(ii) $f(x_1, x_2) = x_1\bar{x}_2 + x_1x_2$	(iii) $f(x_1, x_2) = (x_1 + \bar{x}_2) \cdot (\bar{x}_1 + \bar{x}_2)$	(iv) $f(x_1, x_2) = (\bar{x}_1 + x_2) \cdot (x_1 + x_2)$																				
c) Welche Aussage bei SPIM ist falsch? Der Stack...																							
(i) ...arbeitet nach dem FIFO Prinzip	(ii) ...wächst in Richtung der Speicheradresse 0	(iii) ...hat eine variable Größe	(iv) ...arbeitet nach dem LIFO Prinzip																				
d) Welcher Befehl lädt die Adresse einer Variablen <code>var</code> (zur Realisierung von Call by Reference)?																							
(i) <code>lw \$t0, var</code>	(ii) <code>sw \$t0, var</code>	(iii) <code>beqz \$t0, var</code>	(iv) <code>la \$t0, var</code>																				
e) Welcher Befehl speichert den Inhalt des Registers <code>\$t1</code> unter der Adresse einer Variablen <code>var</code> im Hauptspeicher?																							
(i) <code>lw \$t1, var</code>	(ii) <code>sw \$t1, var</code>	(iii) <code>sw var, \$t1</code>	(iv) <code>la \$t0, var</code>																				