Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Claudia Linnhoff-Popien

mobile and distributed systems group

Betriebssysteme im Wintersemester 2017/2018
Übungsblatt 2

Abgabetermin: 06.11.2017, 18:00 Uhr

Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 30. Oktober – 03. November 2017
Besprechung der H-Aufgaben in den Tutorien vom 06. – 10. November 2017

Aufgabe 6: (T) Betriebssystem-Schichten

(– Pkt.)

In dieser Aufgabe sollen Sie sich klar machen, wie das Betriebssystem mit den restlichen Komponenten eines Rechners verbunden ist. Finden Sie für die folgenden Begriffe eine sinnvolle Kategorisierung, und stellen Sie die sich ergebenen Rechner-Schichten grafisch dar:

Web-Browser, CPU, Dateisystem, Office-Programme, Ein- und Ausgaberoutinen, Festplatte, Hauptspeicher, Gerätemanagement, Benutzer, Scheduler, Shell, Speicherverwaltung, Unix-Compiler, Drucker, Windows-Systemsteuerung.

Aufgabe 7: (T) Grundlagen von Threads

(– Pkt.)

a. Nennen Sie zwei Gründe, warum es nicht sinnvoll ist, zu viele Threads zu verwenden.

b. Nennen Sie zwei Gründe, warum Threads sinnvoll/wichtig sind.

Aufgabe 8: (T) Threads in Java

(– Pkt.)

a. Betrachten Sie das folgende Java-Programm

```
1  public class SimpleThread extends Thread {
2
3          String msg;
4          int cycles;
5
6          SimpleThread(String m, int c) {
7                  msg = m;
8                  cycles = c;
9          }
10
11         // Overrides run() in Thread class to define object's
12         // behavior.
13         public void run() {
```

```
14                  for (int i = 0; i < cycles; i++) {
15                          System.out.println(msg + " cycle " + i);
16                  }
17          }
18
19          // Command-line argument is the number of cycles c
20          // which must be converted from String to int.
21          // Builds and starts two threads of type SimpleThread.
22          // Continues for c cycles.
23
24          public static void main(String[] args) {
25
26                  if (args.length < 1) {
27                          System.out.println("Arguments are:");
28                          System.out.println("  cycles");
29                          System.exit(-1);
30                  }
31
32                  int c = Integer.parseInt(args[0]);
33
34                  SimpleThread t1 = new SimpleThread("Thread 1", c);
35                  SimpleThread t2 = new SimpleThread("Thread 2", c);
36
37                  t1.start();
38                  t2.start();
39          }
40  }
```

Als Eingabeparameter verlangt es eine Integer-Zahl. Wie hängt diese Zahl mit der Ausgabe zusammen? Welche Art der Ausgabe erwarten Sie für verschiedene Integer-Eingaben (zum Beispiel 1, 2, 100 oder 10000)?

b. Basierend auf dem Java-Programm aus Teilaufgabe a:
Geben Sie in Abhängigkeit von `c` eine allgemeine Formel für die Anzahl der möglichen Konsolenausgaben an.

Aufgabe 9: (H) E/A-Operationen mit Hilfe von Interrupts

(6 Pkt.)

In dieser Aufgabe sollen Sie verschiedene Arten der Steuerung von E/A-Operationen gegenüberstellen.

a. Was sind Interrupts und wie wirken sie sich auf den Befehlszyklus im Prozessor aus?

b. Was versteht man unter programmiertem Warten bzw. Busy Waiting? Erläutern Sie diesen Begriff!

c. Nennen Sie je einen Vor- und einen Nachteil von programmiertem Warten im Vergleich zur Steuerung mittels Interrupts.

Aufgabe 10: (H) Einfachauswahlaufgabe: Einführung in Betriebssysteme

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe explizit 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.

<table>
<tr><td colspan="4">a) Bei welcher Belegung (x_1, x_2, x_3, x_4) ergibt die Boolesche Funktion
$f(x_1, x_2, x_3, x_4) = (x_1 \cdot x_2 \cdot \overline{x_3}) + (x_3 \cdot x_4) + \overline{x_2}$ den Wert 1?</td></tr>
<tr><td>(i) $(1, 1, 1, 0)$</td><td>(ii) $(0, 1, 1, 0)$</td><td>(iii) $(0, 0, 0, 0)$</td><td>(iv) $(0, 1, 0, 1)$</td></tr>
<tr><td colspan="4">b) Was ist im Allgemeinen kein Systembus über den die CPU mit anderen Hardwarekomponenten kommuniziert?</td></tr>
<tr><td>(i) Statusbus</td><td>(ii) Steuerbus</td><td>(iii) Datenbus</td><td>(iv) Adressbus</td></tr>
<tr><td colspan="4">c) Welche Speichereinheit beschreibt die Speicheradresse des nächsten auszuführenden Befehls?</td></tr>
<tr><td>(i) AR (address register)</td><td>(ii) BR (buffer register)</td><td>(iii) Hauptspeicher</td><td>(iv) PC (program counter)</td></tr>
<tr><td colspan="4">d) Was ist die korrekte Aussage bezüglich kurzer E/A-Operationen mittels Interrupts (eine E/A-Operation ist abgeschlossen, bevor die nächste auftritt)?</td></tr>
<tr><td>(i) Es besteht kein Unterschied zum Ablauf ohne die Verwendung von Interrupts.</td><td>(ii) Das Nutzerprogramm kann echt parallel zur E/A-Operation arbeiten.</td><td>(iii) Das Nutzerprogramm bleibt blockiert, bis die E/A-Operation abgeschlossen ist.</td><td>(iv) Das Nutzerprogramm muss trotz Interrupts warten, bis die vorhergehende E/A-Operation beendet ist.</td></tr>
<tr><td colspan="4">e) Was ist keine Java Direktive, um das Betreten bzw. Verlassen von <code>synchronized</code> Methoden/Blöcken zu organisieren?</td></tr>
<tr><td>(i) wait()</td><td>(ii) notify()</td><td>(iii) notifyAll()</td><td>(iv) run()</td></tr>
</table>