

Übungen zur VL Betriebssysteme

Blatt 5

Aufgabe 1:

Versuchen Sie zur Wiederholung folgende Fragen aus dem Kopf, d. h. ohne nochmaliges Blättern und Lesen zu beantworten.

- a) Was bezeichnet man in der Prozessverwaltung als Blockieren, Verklemmen und Verhungern?
- b) Bei Betriebssystemen, bei systemnahen Programmen, aber auch bei Anwendungssoftware muss man sich als Entwickler von nebenläufig auszuführenden Aktionen mit kritischen Abschnitten befassen. Was versteht man unter einem kritischen Abschnitt?
- c) Was sind Semaphore? Warum müssen die Semaphor-Operationen $P()$ und $V()$ selbst wieder ununterbrechbar sein?
- d) Welche Auswirkung hat die folgende, falsche Nutzung eines binären Semaphors?
`V(); ... kritischer Abschnitt ...; P();`
- e) Welche vier Kriterien sind nach Dijkstra für die Behandlung kritischer Abschnitte zu beachten?
- f) Was ist ein Mutex im Sinne der Synchronisation konkurrierender Prozesse?
- g) Nennen Sie die vier notwendigen und hinreichenden Bedingungen, die zu einem Deadlock führen!
- h) Welche Maßnahmen sind zu treffen, dass es beim Durchlaufen eines kritischen Abschnitts nicht zu Inkonsistenzen kommt? Gehen Sie dabei auf den Begriff des gegenseitigen Ausschlusses (mutual exclusion) ein!

Aufgabe 2:

Implementieren Sie ein Thread-Beispiel und erzeugen Sie in `main(...)` 100 separate Threads. Die `run()`-Methode jedes Threads soll dabei die Thread-ID in einer Endlosschleife mittels

```
System.out.println(Thread.currentThread().getId());
```

anzeigen. Sie erhalten dann auf dem Bildschirm Meldungen der folgenden Art

```
1
2
1
2
3
2
```

- a) Wie viele Threads lassen sich erzeugen, bis die Leistungsfähigkeit Ihres Systems merklich beeinträchtigt wird?
- b) Beobachten Sie den Speicherverbrauch mit z.B. dem Windows Task-Manager. Lässt sich abschätzen, was ein Thread "kostet"?

Tipp: Die Kapitel 14.2.1 und 14.2.2 im Buch „Java ist auch eine Insel“ (http://openbook.rheinwerk-verlag.de/javainsel9/javainsel_14_002.htm) könnten Ihnen hier gut weiterhelfen.

Aufgabe 3:

Übernehmen Sie das folgende Programm.

a) Welche Funktion hat dieses Programm?

b) Welches Verhalten können Sie bei der Ausführung beobachten? Begründen Sie!

```
public class Adder implements Runnable {
    public int a = 0;

    public void run() {
        for(int i = 0; i < 10000000; i++) {
            a = a + 1;
        }
    }

    public static void main(String[] args) throws Exception {
        Adder value = new Adder();
        Thread t1 = new Thread(value);
        Thread t2 = new Thread(value);

        t1.start();
        t2.start();
        t1.join();
        t2.join();

        System.out.println("Expected a = 20000000, " +
            " but a = " + value.a);
    }
}
```

Aufgabe 4:

Schreiben Sie eine Methode, welche vier ganzzahlige Parameter für die Werte Jahr, Monat, Tag und Differenz übergeben bekommt und das durch die Einzelwerte festgelegte Datum anhand der Differenz verschiebt.

Eine Bildschirmausgabe des Ergebnisses ist hinreichend.

Überprüfen Sie des Weiteren, ob die Parameterwerte ein gültiges Datum beschreiben. Der Betrag der Differenz darf maximal 1000 (Tage) betragen.

Beispiele:

```
VerschiebeDatum(2011,5,1, 2) hat das Ergebnis 03.05.2011  
VerschiebeDatum(2011,4,10, -12) hat das Ergebnis 29.03.2011  
VerschiebeDatum(2010,12,24, 20) hat das Ergebnis 13.01.2011  
VerschiebeDatum(2012,2,20, 10) hat das Ergebnis 01.03.2012
```

Die entsprechenden Funktionen der Klassenbibliothek von Java (z.B. „`plusDays(1)`“) dürfen nur zur eigenen Kontrolle verwendet werden.