InfM-VIS: Verteilte Systeme

WiSe 2018/19 · Übungsblatt 3 (VSYS 2) · Abgabe: Do. 22.11.2018 (bis 12:00 Uhr)

Bitte beachten Sie, dass *Präsenzaufgaben* zwar nicht schriftlich bearbeitet werden müssen und nicht bewertet werden, jedoch als Vorbereitung für den Übungstermin und für die Klausur notwendig sind.

Die Abgabe der *Pflichtaufgaben* erfolgt in Teams von 3–5 Studierenden online als exakt eine PDF-Datei spätestens bis zum oben genannten Termin. Eine spätere Abgabe ist nicht möglich. Bitte verwenden Sie dabei den Zugriffscode einer vorherigen Abgabe, wenn die Zusammensetzung Ihrer Kleingruppe unverändert geblieben ist. Achten Sie in jedem Fall auf die gleiche Schreibweise der Namen!
`https://svs.informatik.uni-hamburg.de/submission/for/vis18-3`

Die Übungen zu diesem Blatt finden vom 26.11.2018 bis 27.11.2018 statt.

Aufgabe 1: UDP und TCP

(Präsenzaufgabe) Im Internet werden auf der Transportschicht zwei Protokolle eingesetzt: Das *User Datagram Protocol* (UDP) und das *Transmission Control Protocol* (TCP). Beschreiben Sie kurz die Funktionsweise beider Protokolle. Geben Sie außerdem Beispiele nicht-trivialer Anwendungen für jedes einzelne Protokoll und erläutern Sie die jeweiligen Vorteile gegenüber dem anderen Protokoll.

Aufgabe 2: Deadlocks (3 Punkte)

a) (Präsenzaufgabe) Erklären Sie den Begriff *Deadlock* (*Verklemmung*) und nennen Sie dabei die vier notwendigen Bedingungen, unter denen Deadlocks auftreten. Grenzen Sie den Begriff außerdem zum Begriff *Livelock* ab.

b) (Präsenzaufgabe) Erläutern Sie die verschiedenen Strategien zum Umgang mit Deadlocks.

c) (Pflichtaufgabe) Erläutern Sie eine Strategie zum Erkennen von verteilten Deadlocks. (3 Punkte)

Aufgabe 3: Threads (6 Punkte)

a) (Pflichtaufgabe) Beschreiben Sie kurz und präzise, was ein *Thread* ist. Erklären Sie dabei die Unterschiede bzw. den Zusammenhang zum *Prozess*-Begriff. (2 Punkte)

b) (Präsenzaufgabe) Beschreiben Sie ein konkretes Anwendungsszenario, in welchem die Nutzung von Threads sinnvoll ist.

c) (Pflichtaufgabe) Was ist das Problem bei der Verwendung der `sleep()`-Methode eines Java-Threads zur Lösung von Nebenläufigkeitsproblemen? (2 Punkte)

d) (Pflichtaufgabe) Warum ist Thread.sleep() nicht für das Timing in Programmen geeignet, z.B. zur Planung einer Aufgabe, die alle 5ms wiederholt werden soll? (2 Punkte)

InfM-VIS: Verteilte Systeme

WiSe 2018/19 · Übungsblatt 3 (VSYS 2) · Abgabe: Do. 22.11.2018 (bis 12:00 Uhr)

Aufgabe 4: Nebenläufigkeit (11 Punkte)

Der folgende Java-Quellcode stellt die Klasse `Downloader` dar, die das Herunterladen einer Datei durch die Methode `startDownload(String url)` anbietet. Das Herunterladen selbst wird dabei in einem neuen Thread mit Namen *Downloadthread* ausgeführt, um den Programmablauf des Aufrufers nicht zu verzögern.

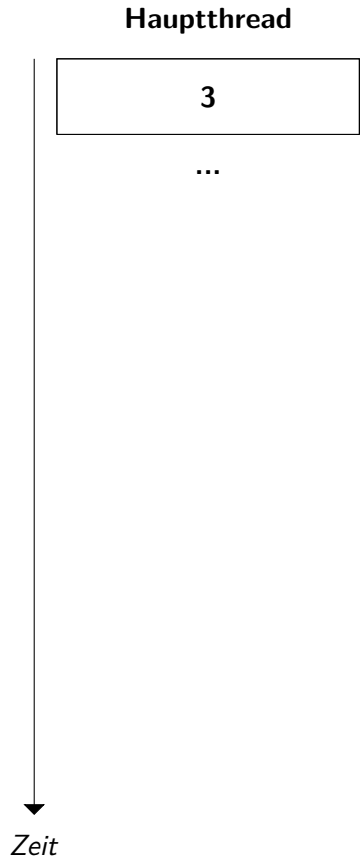
```
public static class Downloader
{
  protected byte[] result;

  public void startDownload(String url)
  {
    new Thread(new Runnable()
    {
      public void run()
      {
1       result = download(url);
      }
2   }, "Downloadthread" ).start();
  }

  public byte[] getResult()
  {
    return result;
  }

  protected byte[] download(String url)
  {
    // Hier geschieht der eigentliche Download
  }
}
```

Im Folgenden wird der Downloader aus einem anderen Thread (*Hauptthread*) genutzt, um ein Bild herunterzuladen:

```
  Downloader downloader = new Downloader();
3 downloader.startDownload("https://goo.gl/QbDGQC");
4 byte[] result = downloader.getResult();
  System.out.println("Download finished. Downloaded " +
     result.length + " bytes.");
5 // Bild anzeigen
```

a) (Pflichtaufgabe) Vervollständigen Sie die folgende Skizze eines möglichen zeitlichen Ablaufs von Haupt- und Download-Thread. Ordnen Sie dazu die mit den roten Ziffern 1-5 markierten Ausführungsschritte einem der beiden Threads zu. Die dargestellte Zeitskala ist global, d.h. Einträge auf gleicher Höhe laufen auch gleichzeitig ab. Achten Sie daher auf korrekte Darstellung der Start- und Endzeitpunkte der eingetragenen Schritte!

(5 Punkte)



b) (Pflichtaufgabe) Der Ausführungsschritt 5 soll das heruntergeladene Bild anzeigen. Erklären Sie, wieso dies mit dem gezeigten Quellcode nicht immer gelingen wird. (4 Punkte)

c) (Pflichtaufgabe) Eine Lösung für das Nebenläufigkeitsproblem könnte die Nutzung der Methode `join()` der Thread-Klasse sein. Was bewirkt der Aufruf dieser Methode? Erklären Sie kurz und präzise! (2 Punkte)

d) (Präsenzaufgabe) Nennen und Beschreiben Sie zwei weitere aus der Vorlesung bekannte Synchronisationsmöglichkeiten für (Java-)Threads.