

## Betriebssysteme im Wintersemester 2008/2009

### Übungsblatt 14

**Achtung:** Die Klausur findet am **Samstag den 07.02.2009** in den Räumen Audimax und B 101 im Hauptgebäude statt. Um an der Klausur teilnehmen zu können, müssen Sie sich bis **Dienstag den 03.02.2009, 12 Uhr** bei UniWorX anmelden. Anmeldungen nach diesem Termin können nicht mehr berücksichtigt werden, nur angemeldete Studenten sind teilnahmeberechtigt an der Klausur!  
Die Teilnehmerlisten und die Zuteilung der Studenten auf die Hörsäle finden Sie rechtzeitig auf der Webseite. Zur Klausur sind **keinerlei Hilfsmittel** erlaubt. Vergessen Sie nicht, Ihren **Studentenausweis** und einen **amtlichen Lichtbildausweis** mitzubringen. **Beachten Sie bitte auch für kurzfristige Änderungen die Hinweise auf der Webseite.**

#### Aufgabe 55: (K) Zustandsübergänge

(1+1+1 Pkt.)

Geben Sie für jeden der folgenden Zustandsübergänge an, ob der Übergang zulässig ist und auf welche Weise der Übergang stattfindet:

- Ändern des Prozeß-Zustandes von Blocked (Blockiert) zu Running (Aktiv).
- Ändern des Prozeß-Zustandes von Running (Aktiv) zu Blocked (Blockiert).
- Ändern des Prozeß-Zustandes von Ready (Bereit) zu Blocked (Blockiert).

**Aufgabe 56: (K) Multilevel Feedback Queuing (MLFQ)** (12 Pkt.)

Gegeben seien die Prozesse  $P_1, \dots, P_5$  mit den folgenden Ankunfts- und Rechenzeiten:

Prozess	Ankunftszeitpunkt	Rechenzeit
$P_1$	0	3
$P_2$	0	7
$P_3$	3	4
$P_4$	7	5
$P_5$	8	2

Hierbei gilt: Trifft ein Prozess zum Zeitpunkt  $t$  ein, so wird er erst zum Zeitpunkt  $(t + 1)$  berücksichtigt. Damit verbringt jeder Prozess mindestens eine Zeiteinheit im Zustand wartend. Wird ein Prozess zum Zeitpunkt  $t'$  unterbrochen, so reiht er sich auch zum Zeitpunkt  $t'$  wieder in die Warteschlange ein. Sind zwei Prozesse absolut identisch bezüglich ihrer relevanten Werte, so wird nach der Prozess-ID entschieden (eine niedrigere Prozess-ID wird bevorzugt).

- a. Geben Sie für die Strategie **Multilevel Feedback Queuing (MLFQ)** in Form eines Gantt-Charts die Zuteilung der Rechenzeit an die Prozesse an. Gehen Sie von drei Prioritätsklassen mit folgenden Werten aus:

Priorität	Verfahren
0	Round-Robin mit Quantum 2
1	Round-Robin mit Quantum 4
2	First Come First Serve (FCFS)

- b. Geben Sie für die Prozesse  $P_1$  und  $P_4$  jeweils die Verweilzeit und die Wartezeit an.

**Aufgabe 57: (K) Deadlocks: Wiederholung** (7 Pkt.)

Gegeben sei ein Computersystem mit vier relevanten Betriebsmitteln/Ressourcen, die zum betrachteten Zeitpunkt  $t_0$  ungebunden sind. Alle Betriebsmittel sind exklusiv. Drei Prozesse konkurrieren um die Betriebsmittel und tätigen die in der folgenden Tabelle skizzierten Anforderungen. Ein Tabelleneintrag  $A(R_n)$  in der Zeile  $P_i$  und der Spalte  $t$  bedeutet, dass der Prozess  $P_i$  zum Zeitpunkt  $t_0 + t$  auf das Betriebsmittel  $R_n$  zugreifen möchte. Eine Anforderung wird sofort in eine feste Ressourcenbindung umgesetzt, wenn die angeforderte Ressource zum jeweiligen Zeitpunkt frei ist. Andernfalls muss der Prozess so lange warten, bis das Betriebsmittel wieder freigegeben wurde.

	1	2	3	4	5	6	7
$P_1$	$A(R_1)$				$A(R_3)$		
$P_2$			$A(R_4)$	$A(R_3)$			
$P_3$		$A(R_2)$				$A(R_1)$	
$P_4$			$A(R_3)$		$A(R_2)$		

- a. Skizzieren Sie alle Ressourcenbindungen und -anforderungen zum Zeitpunkt  $t_0 + 7$  in einem Resource-Allocation-Graph.
- b. Entscheiden Sie anhand des Graphen, ob sich das System zu diesem Zeitpunkt in einem Deadlock befindet. Begründen Sie Ihre Antwort.

**Aufgabe 58: (K) Multiple Choice: Seitenersetzung** (5 Pkt.)

Die Menge der Seiten sei gegeben durch  $N = \{0, 1, 2, 3, 4\}$  und die Menge der Seitenrahmen, die für die Speicherung der Seiten im Arbeitsspeicher zur Verfügung steht, sei gegeben durch  $\text{Frame}_n = \{f_1, \dots, f_n\}$ . Auf die fünf Seiten der Menge  $N$  werde in folgender Reihenfolge zugegriffen:

$$w = 0 \ 1 \ 2 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4$$

Ein Seitenfehler liegt immer dann vor, wenn sich eine referenzierte Seite nicht im (leeren oder vollen) Arbeitsspeicher befindet. Dieser ist zu Beginn leer. Als Seitenersetzungs-Strategie wurde FIFO gewählt. Welche der folgenden Aussagen sind unter diesen Annahmen korrekt?

- a. Bei einem Speicher mit drei Seitenrahmen, also mit  $\text{Frame}_3 = \{f_1, f_2, f_3\}$  ergeben sich neun Seitenfehler.
- b. Bei einem Speicher mit vier Seitenrahmen treten mehr Seitenfehler als bei einem Speicher mit nur drei Seitenrahmen auf.
- c. Bei einem Speicher mit drei Seitenrahmen muss die Seite 4 genau einmal in den Arbeitsspeicher geladen werden.
- d. Bei einem Speicher mit vier Seitenrahmen muss die Seite 3 genau zwei mal in den Arbeitsspeicher geladen werden.
- e. Bei einem Speicher mit drei Seitenrahmen sind die Seiten 0, 3 und 4 zu keinem Zeitpunkt gleichzeitig im Arbeitsspeicher vorhanden.

**Aufgabe 59: (T) Synchrone & asynchrone Kommunikation** (10 Pkt.)

- a. Zeigen Sie die Richtigkeit der folgenden zwei Behauptungen, und geben Sie jeweils ein Beispiel an.
  - (i) Synchrone, bidirektionale Nachrichten-Kommunikation kann durch asynchrone Kommunikation nachgebildet werden.
  - (ii) Asynchrone Nachrichten-Kommunikation kann durch synchrone Kommunikation nachgebildet werden.
- b. Welches sind die Vor- und Nachteile synchroner und asynchroner Kommunikation?

**Aufgabe 60: (P) Java: Einsatz von Threads**

(12 Pkt.)

Das Java-Programm `Bounce.java` können Sie sich von der Betriebssysteme-Homepage herunterladen. Es lässt einen virtuellen Ball innerhalb einer gegebenen Fläche „tanzen“.

- a. Kompilieren Sie das Programm, und führen Sie es aus. Beschreiben Sie kurz, welche Aufgaben die folgenden drei Methoden der Klasse `Ball` erfüllen:
  - (i) `draw()`
  - (ii) `move()`
  - (iii) `bounce()`
- b. Wie müssten Sie den Rumpf der Methode `bounce()` verändern, sodass sich der Ball langsamer bewegt?
- c. Gibt es noch eine andere Möglichkeit, die Geschwindigkeit des Balls zu beeinflussen?
- d. Wie müssten Sie den Rumpf der Methode `bounce()` verändern, sodass sich der Ball bei gleicher Geschwindigkeit länger im Feld bewegt?
- e. Modifizieren Sie das Programm so, dass mehrere Bälle gleichzeitig tanzen können. Beim Drücken des `Start`-Buttons soll jeweils ein neuer Ball ins Feld geschickt werden.