

Gedächtnisprotokoll der Klausur zur Vorlesung Betriebssysteme

vom 30.01.2015

Prof. Claudia Linnhoff-Popien

20 Punkte pro Aufgabe. Bearbeitungszeit 120 Minuten.

Notenschlüssel:

Note	5.0	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0
Ab	0	50	57	64	71	78	85	92	99	106	113

1. Multiple Choice.

<Verschiedene Fragen zu Threads, Speicher, BS allgemein, Deadlocks, ...>

2. Scheduling

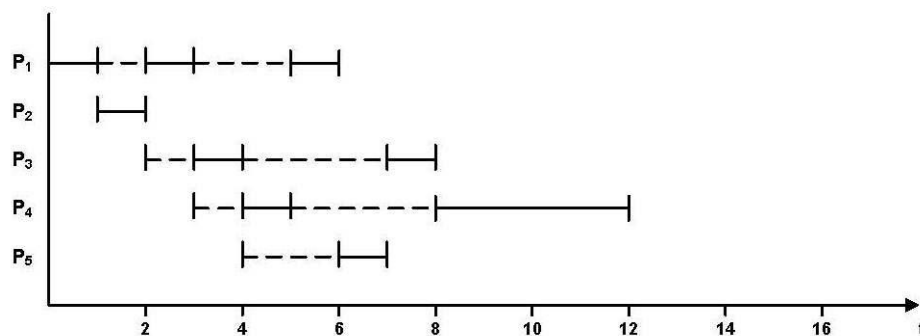
5 Prozesse mit Ankunfts- und Bedienzeit gegeben.

Prozess	Ankunftszeitpunkt	Bedienzeit
P1	0	
P2	0	
P3	2	
P4	4	
P5	5	

Wir gehen davon aus, dass jeder Prozess sein Zeitquantum stets vollständig ausnutzt d.h. kein Prozess gibt den Prozessor freiwillig frei (Ausnahme: bei Prozessende).

Trifft ein Prozess zum Zeitpunkt t ein, so wird er direkt zum Zeitpunkt t berücksichtigt. 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 werden die Prozesse nach aufsteigender Prozess-ID in die Warteschlange eingereiht. Diese Annahme gilt sowohl für neu im System eintreffende Prozesse, als auch für den Prozess, dem der Prozessor u.U. gerade entzogen wird!

Beispiel:

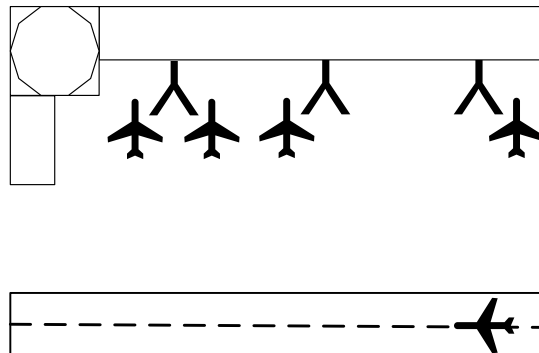


- Tragen Sie die Abarbeitung in das Diagramm nach dem FIFO-Algorithmus ein.
- Tragen Sie die Abarbeitung in das Diagramm nach dem SJF-Algorithmus ein.
- Tragen Sie die Abarbeitung in das Diagramm nach dem RR-Algorithmus mit $Q=2$ ein.
Wird ein Prozess zur gleichen Zeit unterbrochen, wie ein neuer eintrifft, so wird der neu eingetroffene zuerst in die Warteschlange eingefügt.
- Nennen Sie einen Vorteil von FIFO gegenüber SJF!
- Nennen Sie einen Nachteil von FIFO gegenüber SJF!
- Warum ist SJF praktisch nicht realisierbar?

3. Flughafen (Semaphoren / Pseudocode)

An einem Flughafen gibt es sechs Stellplätze für Flugzeuge und genau eine Rollbahn. Die Rollbahn wird gleichzeitig als Start- und Landebahn benutzt. Alle Flugzeuge müssen sie passieren. Um Unfälle zu vermeiden, muss der Betrieb am Flughafen geregelt werden. So darf immer nur genau ein Flugzeug gleichzeitig die Rollbahn in Anspruch nehmen. Ist die Rollbahn belegt, fliegt das Flugzeug eine Warteschleife. Gehen Sie davon aus, dass es hierbei zu keinen Unfällen kommen kann. Stellplätze dürfen natürlich auch nicht von mehreren Flugzeugen zur gleichen Zeit belegt werden. Auch dürfen sich einschließlich der Rollbahn nie mehr als sechs Flugzeuge am Flughafen aufhalten.

Lösen Sie dieses Problem mittels Semaphoren!



Vier Stellplätze und die Rollbahn sind belegt. Nur noch ein Flugzeug hat im Flughafen Platz.

- Welche Semaphoren benötigen Sie, um den Betrieb am Flughafen zu regeln? Geben Sie an wofür diese jeweils gebraucht werden und um welche Art von Semaphore es sich handelt.
- Initialisieren Sie Ihre Semaphore in folgender Notation:
`init(<semaphor>, <value>);`
- Regeln Sie nun den Betrieb am Flughafen, indem Sie unten angegebenen Pseudocoderaahmen entsprechend ergänzen. Verwenden Sie dabei folgende Notation:
`wait(<semaphor>);`
`signal(<semaphor>);`

```
1 flugzeug() {  
2     while(true) {  
3  
4  
5  
6  
7         <landen>;  
8  
9  
10  
11  
12         <stellPlatzBelegen>;  
13  
14  
15  
16  
17         <starten>;  
18  
19  
20  
21     }  
22 }  
23 }
```

- d. Die Gepäckabteilung des Flughafens beschäftigt drei Mitarbeiter. Diese haben verschiedene Ansprüche und Aufgaben. Es soll erreicht werden, dass möglichst viele der Mitarbeiter ihre Aufgaben fertig stellen.

Prozess M1

```

1  m1() {
2      ...
3      wait(a);
4      wait(b);
5      wait(c);
6      <fertig>;
7      signal(b);
8      signal(b);
9      ...
10 }
```

Prozess M2

```

1  m2() {
2      ...
3      wait(c);
4      wait(c);
5      <fertig>;
6      signal(b);
7      ...
8  }

```

Prozess M3

```

1  m3() {
2      ...
3      wait(a);
4      wait(b);
5      wait(b);
6      <fertig>;
7      signal(a);
8      signal(c);
9      ...
10 }
```

Ermitteln Sie für die vier folgenden Initialisierungen der Semaphore a, b und c, in welcher Reihenfolge die Prozesse P1, P2 und P3 geschedult werden müssen, um möglichst viele dieser Prozesse vollständig abzuarbeiten und geben Sie die Bezeichner der vollständig abgearbeiteten Prozesse an. Gehen Sie dabei von einem preemptiven Scheduling Algorithmus aus. Verwenden Sie zur Darstellung der Abläufe, die zu einem Deadlock bzw. zu einem erfolgreichen Ablauf führen, folgende Darstellung und markieren Sie mögliche Deadlocks.

Hinweis: Eine Codezeile soll erst dann als ausgeführt angesehen werden, wenn der entsprechende Funktionsaufruf zurückgekehrt ist.

Initialisierung	i)	ii)
Semaphor a	1	2
Semaphor b	1	0
Semaphor c	1	3

Kommentieren Sie Ihre Lösung ausführlich!

i)	
Prozess	Ausgeführte Codezeile

ii)	
Prozess	Ausgeführte Codezeile

4. Prozessfortschrittsdiagramme

Zwei Prozesse P und Q und sieben gemeinsam genutzte Betriebsmittel gegeben.

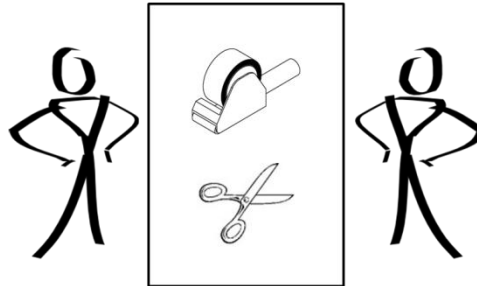
- a. Zeichnen Sie ein Prozessfortschrittsdiagramm. Markieren Sie dabei alle unsicheren und unmöglichen Bereiche.
- b. Zeichnen Sie alle prinzipiell verschiedenen Wege der Abarbeitung von P und Q ein. Nehmen einen preemptiven Scheduling-Algorithmus an und gehen Sie davon aus, dass Dispatching keinerlei Rechenzeit in Anspruch nimmt. Wieviele solcher Wege gibt es?
- c. Wieviele prinzipiell verschiedene Wege führen zu einem Deadlock?
Geben Sie für ein Beispiel an wie lange P und Q dabei aktiv bzw. suspendiert sind.
- d. Gehen Sie davon aus die Prozesse werden auf einem Multiprozessorsystem echt-parallel abgearbeitet. Wieviele Zeiteinheiten werden für die vollständige Abarbeitung beider mindestens benötigt?

5. Lager (Petrinetz)

Zwei Lageristen packen Pakete ein. Um ein Paket einzupacken benötigen Sie Schere und Klebeband. Auf dem Tisch zwischen den beiden liegt genau eine Schere und ein Klebeband. Neben dem Tisch liegt ein unendlich großer Stapel an Paketen.

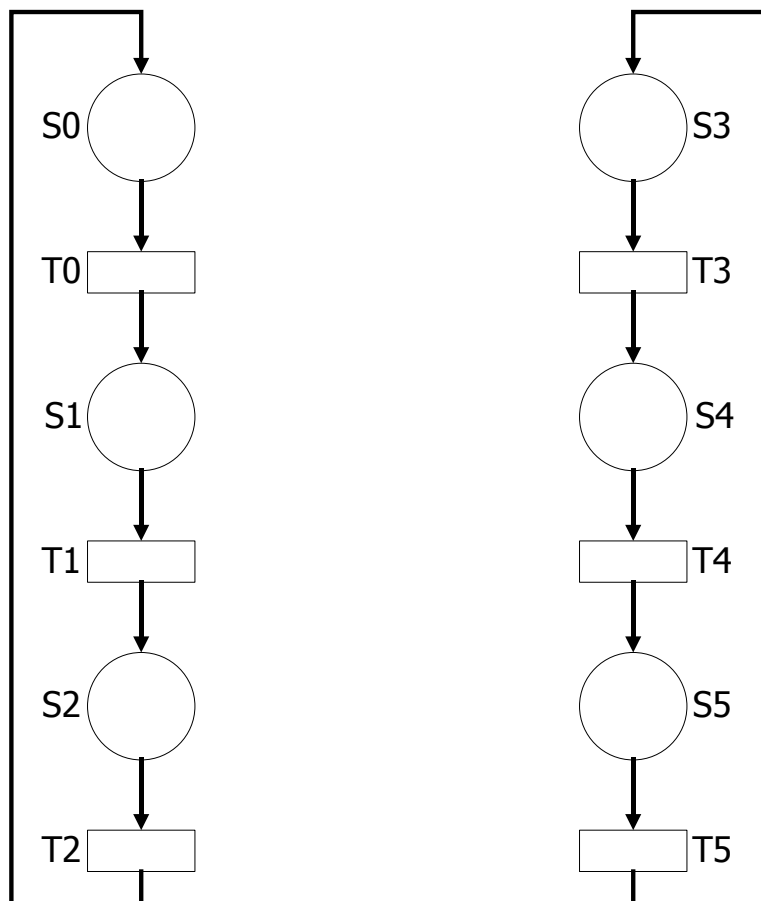
Gehen Sie davon aus, dass beide Lageristen Rechtshänder sind. D.h. Lagerist1 greift zuerst nach der Schere, während Lagerist2 zuerst nach dem Klebeband greift.

Lageristen können drei Zustände haben: "Ruhen", "Hat ein BM" und "Klebt Paket zu".



Unsere Lageristen bei der Arbeit.

- a. Vervollständigen Sie folgendes Petrinetz, sodass es die Situation der Lagerarbeiter vollständig beschreibt. Beschriften Sie jede Stelle und Transition aussagekräftig!



- b. Geben Sie den Erreichbarkeitsgraphen Ihres Petrinetzes an.
- c. Begründen Sie jeweils separat: Kann hier ein Deadlock entstehen? Kann hier ein partieller Deadlock entstehen?

6. Javasynchronisation von Prozessen

In dieser Aufgabe geht es darum einen Kinosaal zu synchronisieren. (Leser-/Schreiber-Problem, leserfreundlich).

In dieser Aufgabe soll ein Kinosaal in einem Kino simuliert werden, in dem aufgrund von Brandschutzvorschriften die Anzahl der aktuellen Besucher zu jedem Zeitpunkt maximal `maxBesucher` betragen darf, einschließlich etwaiger Personen im Drehkreuz. Die einzelnen Besucher (die als Java Threads realisiert werden sollen) besitzen Tickets mit einer eindeutigen Ticketnummer. Sie können den Ausstellungsraum durch ein Drehkreuz am Eingang betreten, indem sie ihr Ticket an einen dort angebrachten Ticketleser halten. Am anderen Ende des Raums können die Besucher den Kinosaal über den Ausgang wieder verlassen, indem Sie ihr Ticket an den Ticketleser des dort installierten Drehkreuzes halten, welches anschließend das Passieren erlaubt. Der Raum ist somit nur in eine Richtung passierbar. Die Drehkreuze sind so konstruiert, dass immer nur genau eine Person passieren kann, nachdem sie ihr Ticket an den jeweiligen Ticketleser gehalten hat. Im Kinosaal dürfen die Besucher für eine beliebige Zeitspanne verweilen - somit können Besucher den Kinosaal in einer anderen Reihenfolge verlassen, als sie ihn betreten haben.

Zusätzlich gibt es eine Reinigungskraft, die den Kinosaal von Zeit zu Zeit reinigt. Sie darf den Saal allerdings nur dann reinigen, wenn sich derzeit kein Besucher mehr in dem Raum aufhält. Während die Reinigungskraft ihrer Arbeit nachgeht, darf der Raum von keinem Besucher betreten werden.

Die Implementierungen der Klassen `Kino` und `Besucher` geben Ihnen einen Überblick wie die Klasse `Kinosaal` verwendet wird.

<Code fehlt, vgl. Code aus der Klausur vom 07.02.2013 (WiSe 2012/13)>

- a. Erklären Sie den Unterschied zwischen Monitoren nach dem Hoare-Konzept und "Java-Monitoren". Gehen Sie dabei auf die Umsetzung der Methoden `notify()` und `wait()` ein.
- b. Was versteht man unter einem kritischen Bereich? Was sind kritische Bereiche in diesem Beispiel?
- c. Implementieren Sie den Konstruktor der Klasse `Kinosaal`. Alle Klassenattribute sind bereits vorgegeben. Initialisieren Sie diese entsprechend und kommentieren Sie Ihren Code ausführlich!
- d. Implementieren Sie die Methode `besucher_betrittSaal(int ticket_nummer)`, welche das Passieren des Drehkreuzes am Eingang des Kinosaals modelliert, sowie die Methode `besucher_verlaesstSaal(int ticket_nummer)`, welche im Gegenzug das Verlassen des Kinosaals über das Drehkreuz am Ausgang modelliert. Beachten Sie bei der Implementierung, dass alle oben genannten Bedingungen eingehalten werden. Schreiben Sie die Methode `besucher_betrittSaal(int ticket_nummer)` so, dass sie die eindeutige Ticketnummer des eintretenden Besuchers sowie die Anzahl der sich gerade im Kinosaal befindlichen Besucher (inklusive des eintretenden Besuchers) auf der Konsole ausgibt. Schreiben sie analog die Methode `besucher_verlaesstSaal(int ticket_nummer)` so, dass sie die Ticketnummer des Besuchers, der gerade das Drehkreuz am Ausgang passiert sowie die Anzahl der sich gerade im Kinosaal befindlichen Besucher (exklusive des Besuchers der den Kinosaal gerade verlässt) auf der Konsole ausgibt. Sie können davon ausgehen, dass die Methoden `besucher_betrittSaal(int ticket_nummer)` bzw. `besucher_verlaesstSaal(int ticket_nummer)` immer in einer sinnvollen Reihenfolge aufgerufen werden (siehe Beispielimplementierung der Klasse `Besucher`). Verwenden Sie dabei den Coderahmen am Ende der Aufgabe und kommentieren Sie Ihre Lösung ausführlich!
- e. Implementieren Sie die Methoden `reinigung_beginnt()` und `reinigung_endet()`. Achten Sie dabei auf oben gegebene Vorschriften und vermeiden Sie Deadlocks! Kommentieren Sie Ihre Lösung ausführlich!
- f. Erläutern Sie wie die kritischen Bereiche in Ihrer Lösung geschützt werden.

<Code fehlt, vgl. Code aus der Klausur vom 07.02.2013 (WiSe 2012/13)>