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 2015/2016
Übungsblatt 13

Abgabetermin: Freiwillige Abgabe von Fragen zum Stoff bis 28.01.2016 16.00 Uhr

Besprechung: Besprechung der Aufgaben in den Tutorien vom 25. – 29. Januar 2016

Ankündigungen: Am Montag, den **1. Februar 2016 findet von 16.00 – 18.00 Uhr im Hörsaal E 004 im Hauptgebäude am Geschwister-Scholl-Platz 1 ein Sondertutorium** für alle Studenten statt, an dem gezielt nochmals Fragen zum Stoff gestellt werden können!
In der Woche vom 1. – 5. Februar 2016 finden keine Übungen und auch keine Vorlesung statt!
Die **Klausur** findet am **5. Februar 2016 von 18.30 - 20.30 Uhr** statt. Bitte melden Sie sich **bis spätestens 2. Februar 2015** zur Klausur über Uniworx an.

Aufgabe 55: (T) SJF versus SRPT

(– Pkt.)

In dieser Aufgabe sollen zwei Scheduling-Strategien untersucht werden: die nicht-preemptive Strategie SJF (Shortest Job First) und die preemptive Strategie SRPT (Shortest Remaining Processing Time). Dazu seien die folgenden Prozesse mit ihren Ankunftszeitpunkten und Bedienzeiten (in beliebigen Zeiteinheiten) gegeben.

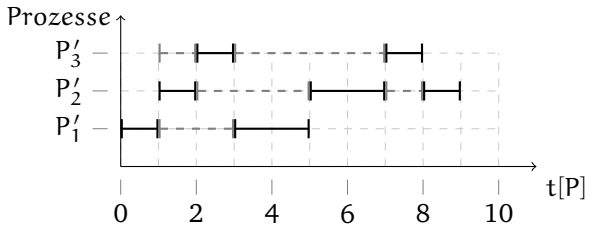
Prozess	Ankunftszeitpunkt	Bedienzeit
P_1	0	6
P_2	2	4
P_3	2	2
P_4	4	1
P_5	8	7
P_6	9	3

- 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 der Warteschlange eingereiht (Prozess P_i vor Prozess P_{i+1}, usw.). Diese Annahme gilt sowohl für neu im System eintreffende Prozesse, als auch für den Prozess, dem der Prozessor u.U. gerade entzogen wird!
- Jeder Prozess nutzt sein Zeitquantum stets vollständig aus d.h. kein Prozess gibt den Prozessor freiwillig frei (Ausnahme: bei Prozessende).

Beispiel: Es seien folgende Ankunfts– und Bedienzeiten für die drei Beispielprozesse P'_1, P'_2 und P'_3 gegeben:

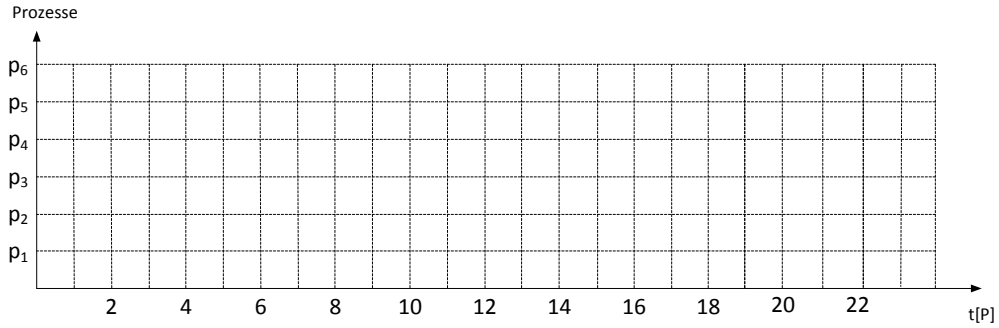
Prozess	Ankunftszeitpunkt	Bedienzeit
P'_1	0	3
P'_2	1	4
P'_3	1	2

Das folgende Diagramm veranschaulicht ein beliebiges Scheduling der drei Prozesse P'_1, P'_2 und P'_3:



Bearbeiten Sie unter den gegebenen Voraussetzungen nun die folgenden Aufgaben:

a. Verwenden Sie nun die **nicht-präemptive Strategie SJF** und erstellen Sie entsprechend dem vorherigen Beispiel ein Diagramm, das für die Prozesse P_1–P_6 angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit. Erstellen Sie Ihre Lösung basierend auf folgender Vorlage:



b. Verwenden Sie nun die **preemptive Strategie SRPT** und stellen Sie Ihre Lösung, wie in der vorherigen Teilaufgabe a), dar.

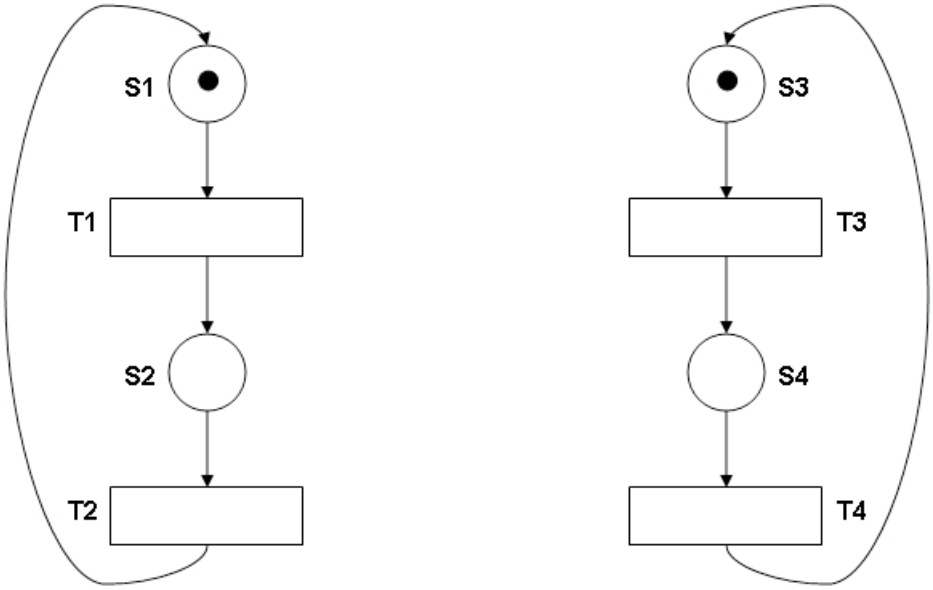
c. Berechnen Sie als Dezimalzahl mit einer Nachkommastelle die mittlere Verweil- und Wartezeit für die zwei Verfahren SJF und SRPT.

d. Wodurch lässt sich der Unterschied zwischen SJF und SRPT in Bezug auf die mittlere Wartezeit begründen?

Aufgabe 56: (T) Deadlockerkennung

(– Pkt.)

Wenn Prozesse nebenläufig ausgeführt werden, macht das eine geeignete Synchronisation dieser Prozesse erforderlich.
Beispielhaft für **zwei** Prozesse P_1 und P_2 soll dieses Synchronisationsproblem durch ein Petrinetz modelliert werden. Die Stellen können als Zustände interpretiert werden. Dabei entspricht S_1 (S_3) dem Zustand "P_1 (P_2) rechnet im unkritischen Bereich", und S_2 (S_4) dem Zustand "P_1 (P_2) rechnet im kritischen Bereich".



a. Ergänzen Sie dieses Petrinetz um weitere Stellen, Transitionen und Kanten (sofern jeweils erforderlich), sodass zu jedem Zeitpunkt höchstens ein Prozess in seinem kritischen Bereich rechnen kann. Zeichnen Sie direkt in das obige Petrinetz.

b. Geben Sie den Erreichbarkeitsgraphen für Ihr Petrinetz an, und beschriften Sie darin alle Übergänge mit den Namen der jeweiligen Transitionen. Begründen Sie, ob das Petrinetz verklemmt ist oder nicht.

Aufgabe 57: (T) Wechselseitiger Ausschluss

(– Pkt.)

Das in der vorangegangenen Aufgabe dargestellte Problem der **zwei** Prozesse P_1 und P_2, die auf einen kritischen Abschnitt zugreifen, soll nun zu einem Erzeuger/Verbraucher-Szenario erweitert werden. Um den wechselseitigen Ausschluss der Zugriffe auf kritische Bereiche zu realisieren, können Binärsemaphore wie folgt eingesetzt werden:

```
BinarySemaphore mutex;
init(mutex, 1);

P:
REPEAT
  <rechne im unkritischen Bereich>
  wait(mutex);
  <rechne im kritischen Bereich>
  signal(mutex);
  <rechne im unkritischen Bereich>
UNTIL FALSE;
```

Der Prozess P_1 erzeugt Daten und schreibt diese in einen gemeinsamen Speicher mit 5 Plätzen. Der Prozess P_2 liest diese Daten. Dazu werden die beiden Zählsemaphoren `platz` und `bestand` eingeführt und wie folgt initialisiert:

```
Semaphore platz, bestand;
init(platz, 5);
init(bestand, 0);
```

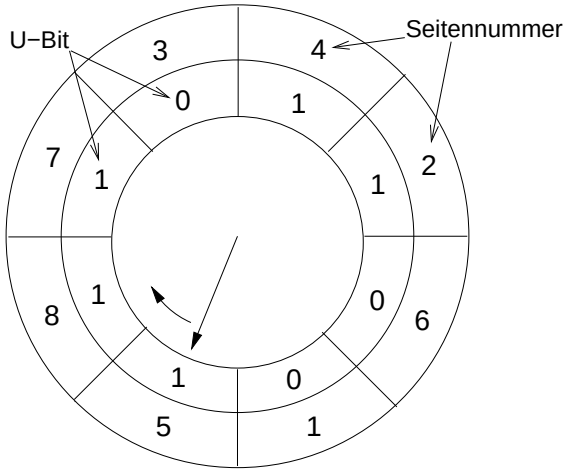
Ergänzen Sie die folgenden Prozessdefinitionen unter Verwendung dieser Semaphoren so, dass P_1 nicht in den vollen Speicher schreiben und P_2 nicht aus einem leeren Speicher lesen kann.

```
P1:                                  P2:
<rechne im unkritischen Bereich>     <rechne im unkritischen Bereich>

wait(mutex);                         wait(mutex);

<erzeuge Element>                    <lies Element>

signal(mutex);                       signal(mutex);

<rechne im unkritischen Bereich>     <rechne im unkritischen Bereich>
```

Aufgabe 58: (T) Seitenersetzung: Second Chance

(– Pkt.)

Der Second-Chance-Algorithmus (eine Variante des Clock-Algorithmus) verwendet für die Auswahl der zu verdrängenden Seiten eine zyklische Datenstruktur wie die hier skizzierte:



Der einzige Unterschied zum Clock-Algorithmus besteht darin, dass der Zeiger immer auf die **zuletzt eingelagerte** Seite verweist. Bei einem Zugriff auf eine Seite wird das dazugehörige U-Bit (Use-Bit) von der Hardware auf 1 gesetzt.

a. Überlegen Sie sich, wie eine sinnvolle, die Seitenfehlerzahl unter Ausnutzung des Lokalitätsprinzips minimierende Paging-Strategie unter den gegebenen Rahmenbedingungen (zyklische Datenstruktur, genau ein Zeiger, U-Bits für jede Seite) aussehen müsste. Geben Sie also in natürlicher Sprache die Arbeitsweise des Second-Chance-Algorithmus an.

b. Eine Seite mit der Nummer 10 soll in den Hauptspeicher geladen werden. Welche Seite wird dafür aus dem Hauptspeicher verdrängt?

c. Skizzieren Sie die obige Datenstruktur nach dem Einlagern der neuen Seite.

d. Was passiert, wenn die U-Bits aller Seiten auf 1 gesetzt sind und ein Zugriff auf eine nicht im Hauptspeicher befindliche Seite erfolgt?

e. Wie könnte der Second-Chance-Algorithmus verbessert werden, sodass er Least Recently Used (LRU) besser approximiert?

Aufgabe 59: (T) Einfachauswahlaufgabe: Speicher

(– Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Was ist keine Aufgabe der Speicherverwaltung?			
(i) Relocation	(ii) Destruction	(iii) Sharing	(iv) Protection
b) Wie bezeichnet man das Problem, wenn z.B. der Hauptspeicher in feste Partitionen gleicher Größe aufgeteilt ist aber der von einem Programm benötigte Speicher geringer als solch eine Partition ist?			
(i) interne Fragmentierung	(ii) externe Fragmentierung	(iii) integrierte Fragmentierung	(iv) dedizierte Fragmentierung
c) Was ist keine Strategie, um bei dynamischer Partitionierung des Speichers einen passenden freien Speicherbereich für eine Speicheranforderung bereitzustellen?			
(i) Best-Fit	(ii) First-Fit	(iii) No-Fit	(iv) Next-Fit
d) Angenommen ein System verfügt über einen Adressbus mit der Breite von 32 Bit und ein Seitenrahmen hat die Größe von 4 KByte. Wie viele solcher Seitenrahmen können adressiert werden?			
(i) 2^{20}	(ii) 2^{15}	(iii) 2^{10}	(iv) 2^{5}
e) Was ist keine Paging-Strategie?			
(i) Demand Paging	(ii) Demand Prepaging	(iii) Look-Ahead-Paging	(iv) Backward Paging

Aufgabe 60: (T) Systeme in der Praxis

(– Pkt.)

Im Rahmen der Vorlesung „Betriebssysteme“ hatte Dr. Harald Rölle einen Vortrag zum Thema „Operating systems in the wild“ gehalten. Eine ähnliche Thematik wird in dem Artikel „Software-Eco-Systeme“ von Dr. Lothar Borrmann, Siemens betrachtet. Lesen Sie diesen Artikel und überlegen Sie, welche Kompetenzen zur Erstellung eines Software-Eco-Systems erforderlich sind.

Hinweis: Als Student können Sie sich ein freies Exemplar des Buches „Marktplätze im Umbruch“ aus dem Internet herunterladen. Dabei haben Sie die folgenden 2 Möglichkeiten:

a. Rufen Sie aus dem LRZ-Netz den folgenden Link zum Buch auf: `http://link.springer.com/book/10.1007/978-3-662-43782-7`. Verwenden Sie dabei den PAC-Proxy (`https://www.lrz.de/services/netzdienste/proxy/zeitschriftenzugang/`)

b. Rufen Sie den folgenden Link zum Buch auf: `http://link.springer.com/book/10.1007/978-3-662-43782-7`. Gehen Sie auf „Sign up/ Login“ und dort auf „Log in via Shibboleth or Athens“. Unter „find your institution“ geben Sie „LMU“ ein und klicken Sie auf „Log in via Shibboleth“. Es folgt die Weiterleitung zur LMU und der Login mit Ihrer Campus-Adresse. Nach Bestätigung können Sie auf der Springer-Seite das Buch herunterladen.

Aufgabe 61: (H) Nachgefragt

(– Pkt.)

Diese Aufgabe dient dazu, sich nochmals gezielt Fragen über den Stoff zu überlegen!
Bitte formulieren Sie **auf freiwilliger Basis** Fragen, die Ihnen beim Durcharbeiten Ihrer Vorlesungsmitschriften (bzw. des Skripts) oder bei der Bearbeitung der Übungsblätter bisher unbeantwortet geblieben sind.
Laden Sie Ihre Fragen bitte bis **spätestens 28.01.2016 16.00 Uhr** als Lösung zu diesem Blatt bei Uniworx hoch. Strukturieren Sie Ihre Fragen bitte folgendermaßen:

<frage>*Text der ersten Frage...* </frage>
<frage>*Text der zweiten Frage...* </frage>
...
Ihre eingereichten Fragen werden dann in einem zusätzlich Sondertutorium beantwortet. Dieses findet satt am: **1. Februar 2016 von 16.00 – 18.00 Uhr im Hörsaal E 004 im Hauptgebäude am Geschwister-Scholl-Platz 1 statt.**.
Wir wünschen Ihnen viel Erfolg bei den Vorbereitungen auf die Klausur!