

## Betriebssysteme im Wintersemester 2019/2020

### Übungsblatt 4

**Abgabetermin:** 18.11.2019, 18:00 Uhr

**Besprechung:** Besprechung der T-Aufgaben in den Tutorien vom 11. – 15. November 2019  
Besprechung der H-Aufgaben in den Tutorien vom 18. – 22. November 2019

#### Aufgabe 18: (T) Praxis-Scheduling

(– Pkt.)

Sie werden beauftragt, für eine Zahnarztpraxis mit einem Arzt und  $n$  Behandlungszimmern einen Scheduler zu programmieren: Dieser soll dem Arzt sagen, in welchem Zimmer er welchen Patienten behandeln soll.

- Betrachten Sie die Patienten als Prozesse und bilden Sie das Zahnarztpraxisbeispiel auf das 7-Zustandsprozessmodell für Prozesse ab. Was geschieht mit den Patienten in den einzelnen Zuständen und wo befinden sie sich jeweils (Behandlungszimmer, Empfangsschalter, Wartezimmer)? Wie sehen konkret die Zustandsübergänge aus?
- Diskutieren Sie, ob preemptive Scheduling-Algorithmen überhaupt angewandt werden können. Geben Sie an, ob hierzu Modifikationen gegenüber dem Prozess-Scheduling eines Betriebssystems vorgenommen werden müssen. Erläutern Sie gegebenenfalls diese Modifikation.
- Erläutern Sie, wie sich die Anwendung der Algorithmen FCFS, SJF, RR, PS und MLFQ auf die Wartezeiten der Patienten mit unterschiedlich aufwendigen Behandlungen auswirken.
- Es wird vorgeschlagen, die Wahl des nächsten Patienten vom Zufall abhängen zu lassen. Ist dies hier fair? Ändert sich die Fairness, wenn man eine solche Wahl bei preemptiven Prozess-Scheduling verwendet?

#### Aufgabe 19: (T) Preemptives Scheduling

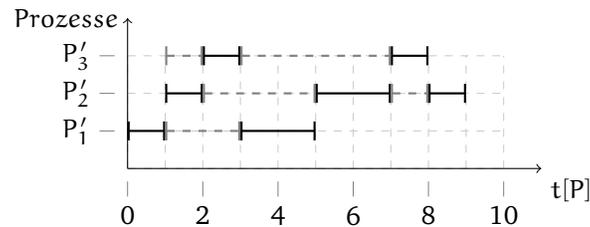
(– Pkt.)

In dieser Aufgabe sollen zwei Scheduling-Strategien untersucht werden: die preemptive Strategie **SRPT (Shortest Remaining Processing Time)** und die preemptive Strategie **RR (Round Robin)**. Dazu seien die u.g. Prozesse (nach dem Beispiel) mit ihren Ankunftszeitpunkten und Bedienzeiten (in beliebigen Zeiteinheiten) gegeben.

**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  $P'_3$ :



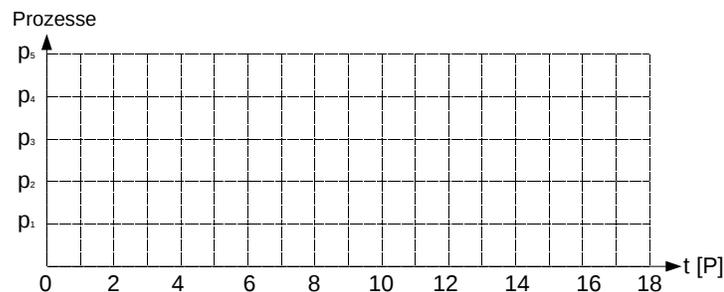
- Trifft ein Prozess zum Zeitpunkt  $t$  ein, so wird er direkt zum Zeitpunkt  $t$  berücksichtigt.
- Wird ein Prozess zum Zeitpunkt  $t'$  unterbrochen, so reiht er sich auch zum Zeitpunkt  $t'$  wieder in die Warteschlange ein.
- Sind zwei Prozesse absolut identisch bezüglich ihrer relevanten Werte, 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).

Geben sei nun folgende Prozesstabelle:

Prozess	Ankunftszeitpunkt	Bedienzeit
$P_1$	0	10
$P_2$	2	1
$P_3$	1	2
$P_4$	3	1
$P_5$	5	2

Bearbeiten Sie unter den gegebenen Voraussetzungen nun die folgenden Aufgaben:

- a. Verwenden Sie nun die **preemptive Strategie SRPT** und erstellen Sie entsprechend dem vorherigen Beispiel ein Diagramm, das für die Prozesse  $P_1$ – $P_5$  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. Geben Sie für die **preemptive Strategie RR** an, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren, indem Sie Ihre Lösung wie in der vorherigen Teilaufgabe a) darstellen. Die Dauer einer **Zeitscheibe betrage 2 Zeiteinheiten**. Gehen Sie davon aus, dass jeder Prozess die Dauer seiner Zeitscheibe stets vollständig ausnutzt, sofern er nicht terminiert. Terminiert ein Prozess vor Ablauf seiner Zeitscheibe, gibt er den Prozessor zum Zeitpunkt der Terminierung sofort frei. Trifft genau nach Ende einer Zeitscheibe ein neuer Prozess ein, so wird der neue Prozess **vor** dem gerade unterbrochenen Prozess in die Warteschlange eingereiht.

- c. Berechnen Sie als Dezimalzahl mit zwei Nachkommastellen die mittlere Verweil- und Wartezeit für die zwei Verfahren SRPT und RR.

## Aufgabe 20: (H) Preemptives Scheduling

(13 Pkt.)

In dieser Aufgabe sollen zwei Scheduling-Strategien untersucht werden: die preemptive Strategie **SRPT (Shortest Remaining Processing Time)** und die preemptive Strategie **RR (Round Robin)**. Dazu seien die folgenden Prozesse mit ihren Ankunftszeitpunkten und Rechenzeiten (in beliebigen Zeiteinheiten) gegeben.

Prozess	Ankunftszeitpunkt	Rechenzeit
P <sub>1</sub>	0	6
P <sub>2</sub>	1	2
P <sub>3</sub>	1	3
P <sub>4</sub>	4	4
P <sub>5</sub>	2	1

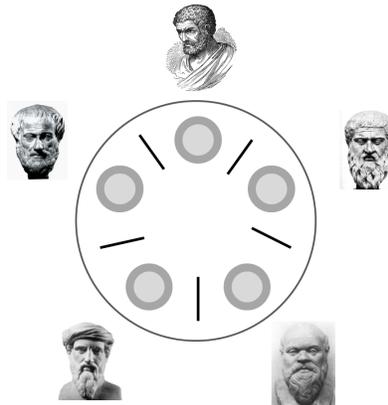
Gehen Sie von den gleichen Voraussetzungen und der gleichen Darstellungsform für das Scheduling wie in Aufgabe 19) aus. Beachten Sie insbesondere die zusätzliche Bedingung, die bei der **Round Robin** Strategie hinzukommt.

- a. Erstellen Sie entsprechend des Beispiels in Aufgabe 19) ein Diagramm für die **preemptive Strategie SRPT**, das für die Prozesse P<sub>1</sub>–P<sub>5</sub> angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit.
- b. Erstellen Sie entsprechend des Beispiels in Aufgabe 19) ein Diagramm für die **preemptive Strategie RR**, das für die Prozesse P<sub>1</sub>–P<sub>5</sub> angibt, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren. Kennzeichnen Sie zudem für jeden Prozess seine Ankunftszeit. Die Dauer einer **Zeitscheibe betrage 2 Zeiteinheiten**. Gehen Sie davon aus, dass jeder Prozess die Dauer seiner Zeitscheibe stets vollständig ausnutzt, sofern er nicht terminiert. Terminiert ein Prozess aber vor Ablauf seiner Zeitscheibe, gibt er den Prozessor zum Zeitpunkt der Terminierung sofort frei. Trifft genau nach Ende einer Zeitscheibe ein neuer Prozess ein, so wird der neue Prozess **vor** dem gerade unterbrochenen Prozess in die Warteschlange eingereiht.
- c. Berechnen Sie als Dezimalzahl mit zwei Nachkommastellen die mittlere Verweil- und Wartezeit für die zwei Verfahren SRPT und RR.
- d. Welcher Nachteil entsteht, wenn man für Round Robin eine zu lange Zeitscheibe wählt?

## Aufgabe 21: (H) Deadlocks

(5 Pkt.)

Beim Philosophenproblem sitzen fünf Philosophen an einem runden Tisch beim Essen (siehe Abbildung). Vor jedem Philosophen befindet sich ein Teller voller Reis und zwischen je zwei Tellern befindet sich ein Stäbchen. Ein Philosoph, der hungrig ist, benötigt sowohl das linke als auch das rechte Stäbchen, um essen zu können. Hat ein Philosoph zwei Stäbchen, so isst dieser, bis er satt ist, erst dann legt er die Stäbchen an die ursprünglichen Plätze zurück, so dass seine Nachbarn davon Gebrauch machen können.



Wenn man beim Philosophenproblem keine Einschränkung beim Zugriff auf die Stäbchen formuliert, kann es zu einem Deadlock kommen.

- Beschreiben Sie informell einen Ablauf für das Philosophenproblem, der zu einem Deadlock führt.
- Beschreiben Sie *kurz* zwei Möglichkeiten, wie sich die Deadlocksituation im speziellen Fall des Fünf-Philosophenproblems vermeiden lässt.

## Aufgabe 22: (H) Einfachauswahlaufgabe: Prozesse

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Was ist kein erlaubter direkter Zustandsübergang im 5-Zustands-Prozessmodell?			
(i) running → ready	(ii) ready → running	(iii) new → exit	(iv) blocked → ready
b) Wie bezeichnet man einen vollständigen Prozesswechsel, bei welchem ein laufender Prozess unterbrochen und das Betriebssystem einem anderen Prozess den Zustand „Running“ zuordnet?			
(i) System-Switch	(ii) Kontext-Switch	(iii) Input-Switch	(iv) User-Switch
c) Wie wird das Auslagern von Prozessen aus dem Hauptspeicher in den Hintergrundspeicher (z.B. Festplatte) bezeichnet?			
(i) Scheduling	(ii) Dispatching	(iii) Swapping	(iv) Blocking
d) Wie bezeichnet man die grundlegende Scheduling-Variante, bei der Prozesse zu jedem Zeitpunkt unterbrochen (suspendiert) werden können, so dass ein anderer Prozess zur Ausführung kommen kann?			
(i) Preemptives Scheduling	(ii) Endloses Scheduling	(iii) Running Scheduling	(iv) Dispatching Scheduling
e) Ein Computer habe vier Bandlaufwerke und n Prozesse, von denen jeder zwei Bandlaufwerke gleichzeitig für seine Ausführung benötigt. Bei einer Anfrage bekommt ein Prozess ein beliebiges freies Bandlaufwerk zugewiesen. Für einen Prozess ist es irrelevant welche Bandlaufwerke er verwendet, solange es zwei sind. Nachdem er die zwei Bandlaufwerke erhalten hat, terminiert er nach endlicher Zeit. Für welchen Wert von n besteht die Möglichkeit eines Deadlocks?			
(i) 4	(ii) 3	(iii) 2	(iv) 1