

Betriebssysteme im Wintersemester 2018/2019

Übungsblatt 5

- Abgabetermin:** 26.11.2018, 18:00 Uhr
- Besprechung:** Besprechung der T-Aufgaben in den Tutorien vom 19. – 23. November 2018
Besprechung der H-Aufgaben in den Tutorien vom 26. – 30. November 2018
- Ankündigungen:** Die Vorlesung am Mittwoch, den 21. November 2018 behandelt Nebenläufigkeit mit Java. Dabei werden über das Skript hinausgehende Folien zum Einsatz kommen. Diese werden am Tag vor der Vorlesung auf der Webseite veröffentlicht.

Aufgabe 22: (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 23: (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 folgenden Prozesse mit ihren Ankunftszeitpunkten und Bedienzeiten (in beliebigen Zeiteinheiten) gegeben.

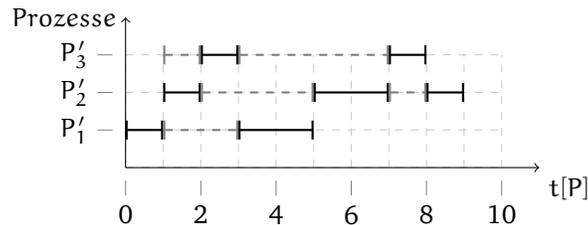
Prozess	Ankunftszeitpunkt	Bedienzeit
P ₁	0	10
P ₂	2	1
P ₃	1	2
P ₄	3	1
P ₅	5	2

- 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:

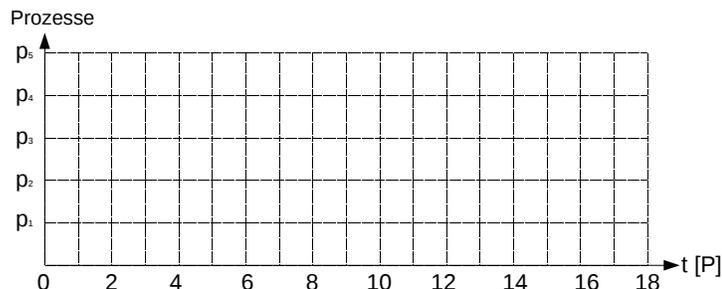
Prozess	Ankunftszeitpunkt	Bedienzeit
P' ₁	0	3
P' ₂	1	4
P' ₃	1	2

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:

- 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 einer Nachkommastelle die mittlere Verweil- und Wartezeit für die zwei Verfahren SRPT und RR.

Aufgabe 24: (H) Preemptives Scheduling

(15 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 ₁	0	6
P ₂	1	2
P ₃	1	3
P ₄	4	4
P ₅	2	1

Gehen Sie von den gleichen Voraussetzungen und der gleichen Darstellungsform für das Scheduling wie in Aufgabe 23) aus. Beachten Sie insbesondere die zusätzliche Bedingung, die bei der **Round Robin** Strategie hinzukommt.

- a. Erstellen Sie entsprechend des Beispiels in Aufgabe 23) ein Diagramm für die **preemptive Strategie SRPT**, das für die Prozesse P₁–P₅ 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 23) ein Diagramm für die **preemptive Strategie RR**, das für die Prozesse P₁–P₅ 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 einer Nachkommastelle 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 25: (H) Einfachauswahlaufgabe: Prozesse, Threads und Scheduling

(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) Welcher Zustand wird im 9-Zustands-Modell im Vergleich zum 7-Zustands-Modell aufgespalten, um zu unterscheiden, ob sich der Prozess im Kernel oder User Mode befindet?			
(i) new	(ii) running	(iii) blocked	(iv) ready
b) Welche Aussage bezüglich des Thread-Konzepts ist falsch?			
(i) Durch den getrennten Adressraum der Threads eines Prozesses sind die Daten einzelner Threads bezüglich anderer Threads sicher.			
(ii) Kontextwechsel unter Threads innerhalb eines Prozesses erfordern weniger Zeit als Kontextwechsel von Prozessen.			
(iii) Zur Generierung eines neuen Threads in einem existierenden Prozess ist wesentlich weniger Zeit notwendig, als zur Generierung eines neuen Prozesses.			
(iv) Terminiert der übergeordnete Prozess, so terminieren mit ihm alle Threads.			
c) Welcher der folgenden Zustände wird nicht explizit für die Zustandsmodellierung von Threads verwendet?			
(i) ready	(ii) running	(iii) blocked	(iv) suspended
d) Was ist keine Funktion, für die eine Thread-Bibliothek für User-Level-Threads Code bereitstellen muss?			
(i) Generierung und Terminierung von Threads	(ii) Zuordnung mehrerer Threads eines Prozesses auf verschiedene Prozessoren	(iii) Nachrichten- und Datenübermittlung zwischen Threads	(iv) Sichern und Löschen von Threadkontexten
e) 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) Running Scheduling	(ii) Endloses Scheduling	(iii) Preemptives Scheduling	(iv) Dispatching Scheduling