Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Linnhoff-Popien

mobile and distributed systems group

Übungsblatt 9
Betriebssysteme im WiSe 21/22

Zum Modul J

Abgabetermin: am 19.12.2021 bis 17:59 auf Uni2Work

Besprechung: vom 20. – 23. Dezember 2021 in den Übungsgruppen

Aufgabe Ü23: Algorithmus von Peterson

(14 Pkt.)

Druckaufträge werden vom Betriebssystem in einer (FIFO-)Warteschlange `W` verwaltet. Die Warteschlange verwaltet selbst lediglich eine Liste von Zeigern, die auf den Speicherbereich verweisen, an dem die zu druckenden Daten liegen. Die Variable `next` enthält den Index der nächsten freien Position in der Warteschlange.
Gegeben seien zwei Prozesse P_1 und P_2, die jeweils eine Datei drucken möchten. Die folgende Tabelle illustriert die Programmausschnitte, die die Prozesse P_1 bzw. P_2 dazu jeweils ausführen.

Prozess $\mathbf{P_1}$

```
1  ...
2  W[next] = pointer_file1;
3  next = next + 1;
4  ...
```

Prozess $\mathbf{P_2}$

```
1  ...
2  W[next] = pointer_file2;
3  next = next + 1;
4  ...
```

a. Welches Problem kann auftreten, wenn P_1 und P_2 im Mehrprogrammbetrieb parallel ausgeführt werden? Modellieren Sie einen Ablauf, der dieses Problem illustriert. Verwenden Sie dazu folgende Darstellung:

aktiver Prozess	ausgeführte Codezeile	Inhalt von `W`	Wert von `next`	Kommentar
...	...	...	...	...

Stellen Sie den Inhalt von `W` als Liste der Form `[ptr_1, ptr_2, ...]` dar.

b. Synchronisieren Sie die Prozesse mit dem Algorithmus von Peterson. Ergänzen Sie dazu (in Analogie zu den Code-Ausschnitten der Angabe) den fehlenden Code der Prozesse P_1 und P_2.

c. Welchen erheblichen Nachteil hat der Peterson-Ansatz?

Aufgabe Ü24: Einfachauswahlaufgabe: Prozess-Koordination

(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 dieser Schritte ist **kein** Teil des Peterson Algorithmus?			
(i) Ich zeige an, dass ich will.	(ii) Falls du willst und du darfst, so lasse ich dir den Vortritt.	(iii) Ich nehme mir den Vortritt, d.h. `ich darf`.	(iv) Ich räume dem Anderen Vortritt ein, d.h. `du darfst`.
b) Welche dieser Aussagen über den Decker Algorithmus (Erster Ansatz) ist korrekt?			
(i) Die Prozesse können den kritischen Bereich nur abwechselnd betreten.	(ii) Sobald einer der Prozesse terminiert, läuft der Algorithmus einwandfrei.	(iii) Die Bedingung Mutual Exclusion wird nicht erfüllt.	(iv) Die Bedingung Progress wird erfüllt.
c) Welche dieser Aussagen über den Decker Algorithmus (Zweiter Ansatz) ist korrekt?			
(i) Der Algorithmus kann problemlos ohne Abänderungen für mehr als zwei Prozesse verwendet werden.	(ii) Unter Verwendung dieses Algorithmus wird immer eine Deadlocksituation entstehen.	(iii) Jeder Prozess erhält im Vergleich zum ersten Ansatz nun zusätzlich eine Variable `flag`.	(iv) Der Algorithmus erfüllt alle drei Bedingungen MutEx, Progress, BoundedWaiting.
d) Wie lautet die korrekte Funktion des `test-and-set`-Befehls auf einer Integer-Variable `i`?			
(i) `i=0 bewirkt i:=0 return false` und `i=1 bewirkt i:=0 return true`	(ii) `i=0 bewirkt i:=1 return false` und `i=1 bewirkt i:=1 return true`	(iii) `i=0 bewirkt i:=0 return true` und `i=1 bewirkt i:=0 return false`	(iv) `i=0 bewirkt i:=1 return true` und `i=1 bewirkt i:=1 return false`
e) Wie nennt man es, wenn ein Prozess immer wieder überprüfen muss, ob eine bestimmte Bedingung erfüllt ist?			
(i) Scheduling	(ii) Busy Waiting	(iii) Check and Load	(iv) Deadlock