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

mobile and distributed systems group

Übungsblatt 8
Betriebssysteme im WiSe 21/22

Zum Modul I

Abgabetermin: am 12.12.2021 bis 17:59 auf Uni2Work

Besprechung: vom 13. – 17. Dezember 2021 in den Übungsgruppen

Aufgabe Ü21: Wechselseitiger Ausschluss

(14 Pkt.)

Von einer **Race Condition** spricht man, wenn das Ergebnis nebenläufiger Prozesse von der Reihenfolge der Prozessaktivierung abhängt.

Im Folgenden seien die beiden Prozesse *Ehemann* und *Ehefrau* gegeben. Die beiden Eheleute besitzen ein gemeinsames Konto. Die Ehefrau will an einem Geldautomaten 400 Euro auf das gemeinsame Konto einzahlen. Zum gleichen Zeitpunkt will der Ehemann 100 Euro an einem anderen Geldautomaten abheben. Der Kontostand `k` sei eine globale Variable, die von beiden Prozessen verändert werden kann. Zu Beginn sei der Kontostand `k` bei 200 Euro. Die Variable `b` stellt eine lokale Variable dar, von der jeder Prozess seine eigene Instanz besitzt.

Prozess **Ehemann**

```
1 ...
2 b = k;
3 b = b - 100;
4 k = b;
5 ...
```

Prozess **Ehefrau**

```
1 ...
2 b = k;
3 b = b + 400;
4 k = b;
5 ...
```

a. Geben Sie eine zeitliche Abfolge der Prozesse Ehefrau und Ehemann an, so dass eine Race Condition eintritt und der Kontostand `k` am Ende beider Transaktionen inkonsistent wird. Verwenden Sie dazu folgende Darstellung:

aktiver Prozess	ausgeführte Codezeile	Inhalt von `k`	Inhalt von `b` (Ehemann)	Inhalt von `b` (Ehefrau)	Kommentar
...	...	...	...	...	...

b. Nennen Sie die drei Bedingungen, die erfüllt sein müssen, um Race Conditions zu verhindern.

c. Betrachten Sie nun die folgenden Vorschläge, das Problem zu lösen:

(i) Der Prozess Ehemann setzt die globale boolesche Variable `x` auf `true`, bevor er den kritischen Bereich betritt. Vor dem Setzen von `x` wird geprüft, ob die Variable nicht bereits vom Prozess Ehefrau auf `true` gesetzt wurde. Ist dies der Fall, wird kein Eintritt in den kritischen Bereich gewährt. Der Prozess Ehefrau verwendet die globale boolesche Variable `y` und geht analog vor.

Prozess **Ehemann**

```
1 ...
2 x = false;
3 while(y) { /* do nothing */ }
4 x = true;
5 b = k;
6 b = b - 100;
7 k = b;
8 x = false;
9 ...
```

Prozess **Ehefrau**

```
1 ...
2 y = false;
3 while(x) { /* do nothing */ }
4 y = true;
5 b = k;
6 b = b + 400;
7 k = b;
8 y = false;
9 ...
```

Zeigen Sie an einem Beispielablauf, dass dieses Vorgehen das Auftreten einer Race Condition nicht verhindern kann. Warum funktioniert dieser Ansatz nicht? Welche der drei Bedingungen wird verletzt?

(ii) Um das im ersten Ansatz auftretende Problem zu beheben, verwenden wir nun zusätzlich eine Integervariable:

Prozess **Ehemann**

```
1 ...
2 x = true;
3 z = 0;
4 while(z == 1 || y) {
5     /* do nothing */
6 }
7 b = k;
8 b = b - 100;
9 k = b;
10 x = false;
11 ...
```

Prozess **Ehefrau**

```
1 ...
2 y = true;
3 z = 1;
4 while(z == 0 || x) {
5     /* do nothing */
6 }
7 b = k;
8 b = b + 400;
9 k = b;
10 y = false;
11 ...
```

Zeigen Sie wieder an einem Beispielablauf, dass auch dieses Vorgehen das Problem nicht behebt. Welche der drei Bedingungen wird hier verletzt?

Aufgabe Ü22: Einfachauswahlaufgabe: Prozesskoordination

(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) Wie bezeichnet man einen Speicher, der in Form eines eindimensionalen Arrays unter Verwendung der Modulo-Funktion realisiert wird?			
(i) Ringpuffer	(ii) Linearpuffer	(iii) Wechselpuffer	(iv) Sparpuffer
b) Wie bezeichnet man die Eigenschaft einer korrekten Lösung des wechselseitigen Ausschlusses, die besagt, dass sich zu jedem Zeitpunkt nur ein Prozess im kritischen Bereich befinden darf?			
(i) Bounded Waiting	(ii) Mutual Exclusion	(iii) Progress	(iv) Circular Wait
c) Für eine korrekte Lösung des wechselseitigen Ausschlusses müssen die drei Bedingungen Mutual Exclusion, Progress, und Bounded Waiting erfüllt sein. Welche Bedingung(en) erfüllt der Algorithmus von Decker (erster Ansatz) nicht?			
(i) Progress	(ii) Mutual Exclusion	(iii) Bounded Waiting	(iv) alle drei
d) Für eine korrekte Lösung des wechselseitigen Ausschlusses müssen die drei Bedingungen Mutual Exclusion, Progress, und Bounded Waiting erfüllt sein. Was trifft auf den Algorithmus von Peterson zu?			
(i) Er erfüllt keine der Bedingungen.	(ii) Er erfüllt alle Bedingungen.	(iii) Er erfüllt nur die Mutual Exclusion Bedingung.	(iv) Er erfüllt nur die Progress Bedingung.
e) Welche Art von Prozessen macht die Synchronisation der Prozesse untereinander erforderlich?			
(i) unabhängige Prozesse	(ii) sequenzielle Prozesse	(iii) nebenläufige Prozesse	(iv) parallele aber unabhängige Prozesse