

Betriebssysteme(BESYST)

05 Übungsblatt

Prof. Dr. Patrick Draheim

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Stellen wir uns für diese Situation den Alltag einer Familie mit drei Kindern vor, die im Winter draußen spielen möchten. Damit sie sich nicht erkälten, benötigt jedes Kind eine Mütze, ein Paar Handschuhe und einen Schal, um im Freien spielen zu können. Bedauerlicherweise hat ihre Oma bisher noch nicht genug gestrickt, sodass jedes Kind momentan nur eines dieser drei Kleidungsstücke besitzt, und zwar jeweils ein unterschiedliches. Zusätzlich haben die Kinder im Kleiderschrank eine alte Mütze, ein altes Paar Handschuhe und einen großen Schal entdeckt. Wenn nun eines der Kinder nach draußen möchte, benötigt es noch zwei der Gegenstände aus dem Schrank.

- Das jeweilige Kind benötigt entweder Mütze und Schal ODER
- Das jeweilige Kind benötigt Handschuhe ^{und} oder Mütze ODER, ~~X~~
- Das jeweilige Kind benötigt Schal oder Handschuhe

2M, 2H, 2S

M H SKind A |1

Aufgabenstellung: Fragestellungen zur Vorlesung

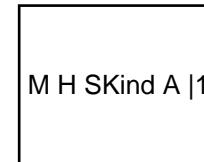
- Beantworten Sie folgende Fragen:
- Konzipieren Sie eine Lösung für diese Aufgabenstellung basierend auf mehreren Semaphoren. Stellen Sie sicher, dass sie mehrere Threads konzipieren, wobei jeder Thread einem der drei Kinder entspricht. Geben Sie die Lösung im Pseudocode an.
 - Überlegen Sie, ob sich das von Ihnen angegebene Threadsystem verklemmen kann.

Muetzen = 1

Handschuhe = 1

Schaale = 1

P(H); P(M); P(M);
P(S); P(S); P(H);
//spielen //spielen //spielen
P(S); P(S); P(H);
P(H); P(M); P(M);



Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Betrachten Sie die Lösung aus der Vorlesung:

```
Semaphore mutex = 1;  
Semaphore full = 0;  
Semaphore empty = N;
```

Erzeuger:

```
while (true)  
{  
    item = Produce();  
    P(empty);  
    P(mutex);  
    insert_item(item);  
    V(mutex);  
    V(full);  
}
```

Verbraucher:

```
while (true)  
{  
    P(full);  
    P(mutex);  
    item = remove_item();  
    V(mutex);  
    V(empty);  
    Consume(item);  
}
```

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:
- d. Ist die Reihenfolge der Semaphor-Operationen in der gezeigten Umsetzung wichtig? Zeigen Sie am Gegenbeispiel, warum ja/nein.

Wechseln wir P(full) mit P(mutex),
so gerät man in einen Deadlock

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Betrachten Sie das folgende Threadsystem:

// Semaphore und globale Variable

```
Semaphore S = 2;
```

```
Semaphore T = 0;
```

// Thread A

```
while (true)
```

```
{
```

```
    P(S);
```

```
    print("A");
```

```
    V(T);
```

```
}
```

// Thread B

```
while (true)
```

```
{
```

```
    P(S);
```

```
    print("B");
```

```
    V(T);
```

```
}
```

// Thread C

```
while (true)
```

```
{
```

```
    P(T);
```

```
    print("C");
```

```
    V(S);
```

```
}
```

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Welche Aussagen zu dem Threadsystem sind richtig?

1. Alle drei Threads können gleichzeitig die Prozedur print() ausführen. Nein
2. Es wird immer zuerst ein A ausgegeben. Nein
3. Es kann nie zuerst ein C ausgegeben werden. Ja
4. Die Ausgabe kann mit ABA ... beginnen. Nein
5. Die Ausgabe kann mit ABCC ... beginnen. Ja
6. Die Ausgabe kann die Folge ... CCCCC ... enthalten. Nein
7. In dieser Umsetzung sind Verklemmungen möglich. Nein

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Ein Warenautomat enthält zu Beginn 200 Chips-Tüten (er ist damit vollständig gefüllt) und führt eine Endlosschleife aus, deren Rumpf die folgenden Schritte enthält:

- Warten, wenn der Automat leer ist,
- Warten, bis ein Kunde 5 Euro eingeworfen hat,
- Erniedrigen eines Zählers für die Anzahl der Chips-Tüten und Chips-Tüte ausgeben.

Es gibt mehrere Kunden, die jeweils folgende Schritte ausführen:

- Warten, solange ein anderer Kunde den Automaten benutzt,
- den Betrag in fünf einzelnen Eurostücken einwerfen,
- Warten, bis die Ware kommt (das kann durchaus länger dauern, da erst der Lieferant kommen muss, wenn der Automat leer ist),
- den Automaten wieder freigeben.

P(besetzt)	Lieferant:	Automat:	chips = 200
5x V(bezahlt)	//2 Tage sind vergangen	while(true){	bezahlt = 0
P(ware)	while(chips < 200){	5x P(bezahlt)	besetzt = 1
V(besetzt)	V(chips)	counter += 5	counter = 0
	}	P(chips)	ware = 0
		V(gibChips)	
		}	

Aufgabenstellung: Fragestellungen zur Vorlesung

- Beantworten Sie folgende Fragen:

Zudem gibt es einen Lieferanten, der alle zwei Tage kommt und den Automaten wieder vollständig (mit maximal 200 Chips-Tüten) nachfüllt.

- Geben Sie eine Lösung in einer Pseudo-Programmiersprache für die drei Prozesse (Automat, Kunde und Lieferant) an.
- Identifizieren Sie alle Synchronisationsbedingungen zwischen den Prozessen. Benutzen Sie selbst definierte Semaphore, um eine korrekte Synchronisation zu implementieren.