

Systemprogrammierung

WS 2004/2005

Übungsblatt 5

Abgabe der Lösungen: ab 22.11.2004 in den Übungen

Aufgabe 1 (2+3+3=8 Punkte): Semaphore

Italienische Banken verfügen über Sicherheitsschleusen bestehend aus zwei Türen, die potentielle Bankräuber an der Flucht hindern sollen. In den Raum zwischen den Türen paßt jeweils nur eine Person. Es läßt sich nur jeweils eine der beiden Türen öffnen. Zum Betreten der Schleuse drückt der Kunde auf den Türgriff. Dadurch wird die Tür entriegelt und die gegenüberliegende Tür verriegelt. Die Verriegelung wird erst aufgehoben, wenn die erste Tür wieder ins Schloß gefallen ist. Der Kunde kann nun den zweiten Türgriff betätigen und die Schläuse verlassen.

Bei einem Überfall können Mitarbeiter den Bank einen Schalter betätigen, der das Öffnen der äußeren Türen verhindert.

- (a) Welche Sensoren bzw. Relais (also Ein- und Ausgaben) werden von einem Steuerprogramm für diese Schleuse benötigt?
- (b) Schreiben Sie einen Steuerprogramm für diese Türanlage. Benutzen Sie Semaphore.
Hinweis: Benutzen Sie einen C-Pseudocode. Funktionen, welche die Ein- und Ausgabe betreffen, brauchen nicht ausformuliert werden, sondern können abstrakt verwendet werden.
- (c) Um das Bankgeheimnis zu wahren, dürfen sich maximal n Personen in der Bank aufhalten. Zu diesem Zweck ist die Schleuse mit einem Sensor ausgestattet, der darüber Auskunft gibt, ob sich eine Person in ihr aufhält. Modifizieren Sie ihr Programm so, daß dieses berücksichtigt wird.

Aufgabe 2 (7 Punkte): Semaphore

Schreiben Sie ein Steuerprogramm für einen Fahrkartenautomaten (es gibt nur einen Typ Fahrkarten). Ein Prozeß überwacht die Menge des eingeworfenen Geldes und gibt evtl. Wechselgeld. Ein zweiter Prozeß ist für das Ausdrücken der Fahrkarte zuständig. Benutzen Sie eine Semaphore zur Überwachung des Geldeinwurfs und des korrekten Programmablaufs. Stellen Sie außerdem sicher, daß jeweils nur eine Fahrkarte ausgedruckt wird.

Aufgabe 3 (3,5+1,5=5 Punkte): Deadlocks und Semaphore

Ein Anthropologie-Student mit Nebenfach Informatik hat nach dem Abschluß seiner Forschungsarbeit über afrikanische Paviane ein neues Projekt gestartet. In diesem soll das Verhalten von südostasiatischen Gibbons zur Vermeidung von Verklemmungen untersucht werden. Dazu wurde über eine Schlucht ein Seil gespannt, so daß sich die Gibbons darüberhangeln können. Mehrere Gibbons können gleichzeitig hinüberhangeln, sofern sie sich alle in derselben Richtung bewegen. Befindet sich jedoch bereits ein ihnen entgegenkommender Artgenosse auf dem Seil, so müssen sie warten. Da Gibbons ein ausgeprägtes Sozialverhalten besitzen, betritt kein neu eintreffendes Tier das Seil, wenn auf der anderen Seite bereits ein Gibbon wartet. Wenn an beiden Seiten Gibbons warten, dann überquert abwechselnd jeweils ein Tier von jeder Seite die Schlucht.

- (a) Schreiben Sie eine Funktion `ape_ow()`, die einen von Osten nach Westen ziehenden Gibbon realisiert und Verklemmungen vermeidet. Die Existenz einer analog aufgebauten Funktion `ape_wo()` wird vorausgesetzt. Folgende Vereinbarungen, Variablen und Funktionen stehen Ihnen dabei zur Verfügung:

```
1      const EMPTY=0;
2      const WO=1;
3      const OW=2;
4
5      int dir;           // was ist auf dem Seil los
6      int climb_count;   // Anzahl der Affen auf dem Seil
7      int wo_count;      // Anzahl im Westen wartender Affen
8      int ow_count;      // Anzahl im Osten wartender Affen
9
10     semaphore mutex, s_wo, s_ow;
11
12     void wait(semaphore s);
13     void signal(semaphore s);
14
15     void climb(void);    // Hinüberklettern eines einzelnen Affen
```

- (b) Wie müssen die Semaphor-Variablen `mutex`, `s_wo` und `s_ow` initialisiert werden?