



**Technische Universität
München**

**Fakultät für Informatik
Forschungs- und Lehrereinheit Informatik VI**

Übung zur Vorlesung Echtzeitsysteme

Aufgabe 5 – Semaphoren

Philipp Heise Steffen Wittmeier
heise@in.tum.de steffen.wittmeier@in.tum.de

Christoph Staub Michael Jäntsch
staub@in.tum.de michael.jaentsch@in.tum.de

Wintersemester 2012/13

Aufgabe 5: Semaphoren

In der Vorlesung wurde die Verwendung von Semaphoren besprochen. Ziel des aktuellen Aufgabenblattes ist es, das *Dining Philosophers Problem* umzusetzen.

Philosophen

Die Philosophen sitzen am Tisch (vgl. Abb. 1), denken über philosophische Probleme nach und nehmen ab und an einen Happen zu sich. Sie speisen jedoch nur mit Gabeln, Messer gibt es nicht am Tisch. Wenn einer hungrig wird, greift er zuerst die Gabel links von seinem Teller, dann die auf der rechten Seite und beginnt zu essen. Wenn er satt ist, legt er die Gabeln wieder zurück und beginnt wieder zu denken. Sollte eine Gabel nicht an ihrem Platz liegen, wenn der Philosoph sie aufnehmen möchte, so wartet er, bis die Gabel wieder verfügbar ist.



Abbildung 1: Speisende Philosophen

- a) Implementieren Sie die Funktion *eating()* fertig, die von der Funktion *philosopher()* aufgerufen wird und folgenden Ablauf ausführen soll

```
eating(int i){ //i: Nummer des Philosophen
    // Linke Gabel nehmen
    // Rechte Gabel nehmen
    // Der Philosoph ißt ... zufällige Zeit
    // Linke Gabel zurücklegen
    // Rechte Gabel zurücklegen
}
```

Zur Auswahl der linken/rechten Gabel verwenden Sie die bereitgestellten Funktionen *left(int)* und *right(int)*.

Verwenden Sie zum Schutz der kritischen Bereiche (=Gabeln) Semaphoren. Definieren Sie hierzu ein globales Array für die Semaphoren, dessen Größe der Anzahl der Philosophen (und dementsprechend auch der Gabeln) entspricht. Folgende Funktionen können Ihnen dabei hilfreich sein:

- *sem_init()*: Funktion zum Initialisieren eines Semaphors
- *sem_wait()*: Funktion zum Locken eines Semaphors
- *sem_post()*: Funktion zum Freigeben eines Semaphors
- *sem_close()*: Funktion zum Beenden eines Semaphors

Solange nur einzelne Philosophen hungrig sind, funktioniert dieses Verfahren wunderbar. Es kann aber passieren, dass sich alle fünf Philosophen gleichzeitig entschließen zu essen. Sie ergreifen also alle gleichzeitig ihre linke Gabel und nehmen damit dem jeweils links von ihnen sitzenden Kollegen seine rechte Gabel weg. Nun warten alle fünf darauf, dass die rechte Gabel wieder auftaucht. Dies passiert aber nicht, da keiner der fünf seine linke Gabel zurücklegt. Die Philosophen verhungern (*Deadlock*). Deadlock oder Verklemmung bezeichnet in der Informatik einen Zustand, bei dem ein oder mehrere Prozesse auf Betriebsmittel warten, die dem Prozess selbst oder einem anderen beteiligten Prozess zugeteilt sind.

- b)** Da es in der Funktion `eating()` zu einem Deadlock kommen kann, müssen Sie die Funktion erweitern. Überlegen Sie sich, wie man diesen vermeiden kann und implementieren Sie Ihre Lösung.