



Technische Universität  
München

Fakultät für Informatik  
Forschungs- und Lehrereinheit Informatik VI

# Übung zur Vorlesung Echtzeitsysteme

## Aufgabe 9 – Message Queues und Timer

Simon Barner  
barner@in.tum.de

Irina Gaponova      Stephan Sommer  
gaponova@in.tum.de      sommerst@in.tum.de

Wintersemester 2009/10

## Aufgabe 9: Message Queues und Timer

### Allgemeines

In der aktuellen Übung soll die Aufgabe (Philosophen) vom letzten Übungsblatt als Basis dienen und an geeigneten Stellen erweitert werden.

Folgendes könnte für das Lösen der Aufgaben nützlich sein:

- Message Queue: `mqd_t`, `struct mq_attr`, `mq_open`, `mq_send`, `mq_receive`, `mq_getattr`
- Timer: `timer_t`, `struct sigevent`, `struct itimerspec`, `sigset_t`, `sigemptyset`, `sigaddset`, `timer_create`, `timer_settime`, `sigwait`

### Message Queue

Wie aus der letzten Aufgabe bekannt, sitzen die Philosophen an einem Tisch. Jeder der Philosophen hat Initial einen leeren Teller und es werden die Zustände *denken*, *hungrig* und *essen* unterschieden. Immer wenn ein Philosoph hungrig ist, bestellt er bei einem Kellner etwas zu Essen.

- a) Erweitern Sie Ihr Programm um eine Message Queue (`waiter_queue`). Immer wenn ein Philosoph hungrig ist und bestellt, soll eine Nachricht in die Message Queue gestellt werden mit der ID des Philosophen. Verwenden Sie folgende Einstellungen für die MessageQueue:

- `mq_flags = 0;`
- `mq_maxmsg = 10;`
- `mq_msgsize = 4;`

- b) Fügen Sie zu Ihrem Programm einen eigenen Thread für den Kellner hinzu. Dieser soll ein Element aus der Message Queue entnehmen und den Text *"Philosoph ID ist hungrig und hat etwas bestellt"* auf der Konsole ausgeben.

### Timer

Nachdem bereits die Bestellungen über die Konsole ausgegeben werden, sollen diese jetzt vom Koch abgearbeitet werden.

Erweitern Sie dazu Ihr Programm ...

- a) um eine zweite Message Queue (*cook\_queue*) für die Kommunikation zwischen Kellner und Koch. Wenn der Kellner eine Nachricht aus der (*waiter\_queue*) entnommen und ausgegeben hat, soll er 5 Sekunden warten und anschließend die Nachricht an die *cook\_queue* weiterleiten.
- b) um Semaphoren für jeden Teller am Tisch.
- c) so, dass die Philosophen, nach dem Bestellen warten, bis ihr Teller gefüllt ist.
- d) um eine Methode (den Koch), die:
  - Eine Nachricht aus der *cook\_queue* nimmt. Falls die Message Queue leer ist, soll hier nicht blockierend gewartet werden. Prüfen Sie dazu vor dem Auslesen einer Nachricht, wie viele Elemente in der MQ sind.
  - Dem Philosophen ein Essen serviert und anschließend 5 Sekunden wartet.
- e) **Zusatz:** Realisieren Sie die Koch-Methode mittels eines Timers.