B**S**y Praktikum 5 Synchronisationsprobleme

**Kaffee−Automaten**

(FS2020, Gabriele Pace – pacegab1)

# 2.3 Einfache Reihenfolge

## a)

### Schritt 1: Prozesse (Threads) der Problemstellung identifizieren

* Kaffee-Automat
* Kunde A
* Kunde B
* Kunde C
* Kunde D

### **Schritt 2:** Ausführungsschritte der einzelnen Prozesse (Threads) ermitteln

|  |  |
| --- | --- |
| **Kaffe-Maschine** | **Kunde A** |
| Ist bereit | Wartet bis er an der Reihe ist |
| Auf Geld warten | Geld einwerfen |
| Kaffe ausgeben | Auf Kaffe warten |
| Setzt Freigabe | Setzt nächsten Kunden |

### **Schritt 3:** Synchronisationsbedingungen ermitteln



### **Schritt 4:** Benötigte Semaphore definieren

myTurn = sem\_open(MYTURN\_SEMAPHOR, O\_CREAT, 0700, 1);

checkSem(myTurn);

coin = sem\_open(COIN\_SEMAPHOR, O\_CREAT, 0700, 0);

checkSem(coin);

coffee = sem\_open(COFFEE\_SEMAPHOR, O\_CREAT, 0700, 0);

checkSem(coffee);

ready = sem\_open(READY\_SEMAPHOR, O\_CREAT, 0700, 0);

checkSem(ready);

### **Schritt 5:** Prozesse mit Semaphoreoperationen ergänzen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kaffe-Maschine** | **Kunde A** | **Kunde B** | **Kunde C** | **Kunde D** |
| ready(0) | wait(myturn) (wartet bis myturn = 1 und setzt myturn = 0) | wait for myturn | wait for myturn | wait for myturn |
| set ready(1) | wait for ready(1) | wait for myturn | wait for myturn | wait for myturn |
| wait for coin(1) | coin(1) | wait for myturn | wait for myturn | wait for myturn |
| coffee(1) | wait for coffee(1) | wait for myturn | wait for myturn | wait for myturn |
| set ready(0) | post(myturn) (setzt myturn = 1) | wait for myturn | wait for myturn | wait for myturn |

### **Schritt 6:** Implementation

Siehe Code.