



— Grundlagen der Betriebssysteme WiSe 2017/18 —

## 6. Übungsaufgabe

Ausgabe: 10.01.2018

Abgabe: 16.01.2018

### 1 Parallele Programmierung: Semaphore

**Punkte: 8**

Gegeben seien drei Prozesse  $P_a$ ,  $P_b$  und  $P_c$ , in denen jeweils ein gleichnamiger Anweisungsblock in einer Endlosschleife ausgeführt wird:

$$\begin{aligned} P_a &:: \text{while ( TRUE ) } \{ A; \} \\ P_b &:: \text{while ( TRUE ) } \{ B; \} \\ P_c &:: \text{while ( TRUE ) } \{ C; \} \end{aligned}$$

Erweitern Sie die drei Schleifenrumpfe um Synchronisierungsmechanismen, so dass die Anweisungsblöcke A, B und C in der Reihenfolge ausgeführt werden, die durch die folgenden regulären Ausdrücke angegeben werden. Zur Verfügung stehen Ihnen Semaphoren und Integer-Variablen.

a) (3P)

$$( \text{ACCCB} )^+$$

b) (5P)

$$( ( C^+ \mid A ) B )^+$$

### 2 Parallele Programmierung: Barriere-Problem

**Punkte: 4**

In der Vorlesung GdB wurde eine Lösung des Barriere-Problems vorgestellt, die einen Prozess Q vorsieht, der n Threads synchronisiert. Das Schema ist unten zu sehen:

```
1 while(TRUE) do{
2   P(t);
3   cnt++;
4   if (cnt==N)
5     for(i=N-1, i>0, i--) {cnt--; V(s)};
6 }
```

Diese Lösung ist jedoch nicht vollständig, da sie erlaubt, dass Prozesse schon in die nächste Parallelanweisung eintreten können, bevor eine exklusive Operation S vollendet ist.

Konstruieren Sie eine Barriere, die die oben beschriebene Problematik verhindert und mit möglichst wenigen Semaphoren auskommt. Skizzieren Sie auch die notwendigen Anweisungen (sprich: die notwendigen Semaphoreoperationen), die die n Threads ausführen müssen, um das vorangehende PAREND zu überwinden, und erst dann das nächste PARBEGIN überwinden (zwei Phasen), nachdem Operation S ausgeführt ist.