

AI-B.41: Verteilte Systeme

Übungsaufgaben [Ü9]

Viel Erfolg!

C. Schmidt | SG AI | FB 4 | HTW Berlin

Stand: WiSe 18/19

Urheberin: Prof. Dr. Christin Schmidt

Verwertungsrechte: keine außerhalb des Moduls

Ü1: Bewertung: Prozesse vs. Threads

Wenn Domänentransitionen auch beim Threading „teuer“ sind, warum nutzt man Threads überhaupt (einfache Prozesse könnten ja ebenso genutzt werden)?

Ü2: Evaluation von Prozesszuständen (single-core Prozessor)

Gegeben sei ein single-core Prozessor mit drei aktiven Prozessen {A, B, C}. Bewerten Sie die nachfolgenden Kombinationen der Prozesszustände dahingehend, ob diese auftreten können oder nicht. Begründen Sie ihre Antwort!

Prozess	Kombination Prozesszustände					
	a	b	c	d	e	f
A	Running	Ready	Running	Blocked	Ready	Blocked
B	Ready	Blocked	Running	Ready	Ready	Blocked
C	Ready	Ready	Ready	Running	Ready	Blocked

Ü3: Zustandssequenzen

Gegeben sei ein Prozess mit den folgenden Zustandssequenzen:

- a. Ready -> Running -> Blocked -> Ready -> Running -> Blocked -> Running
- b. Ready -> Running -> Ready -> Running -> Blocked -> Ready -> Running
- c. Ready -> Running -> Blocked -> Ready -> Blocked -> Ready -> Running
- d. Ready -> Blocked -> Ready -> Running -> Blocked -> Running -> Blocked
- e. Ready -> Running -> Ready -> Running -> Blocked -> Running -> Ready

Welche der Sequenzen sind realistisch / möglich? Begründen Sie ihre Antwort!

Ü4 [Prog]: Threads & Monitore am Beispiel eines Parkhauses in Java

Programmieren Sie mittels JAVA ein Parkhaus.

- Das Parkhaus soll so abstrahiert werden, dass sein Zustand mittels der Anzahl der noch freien Plätze festgehalten wird
- Mehrere Autos sollen einfahren und ausfahren können
 - » Beim Einfahren wird die Zahl der freien Plätze um eins vermindert
 - » Beim Ausfahren wird die Zahl der freien Plätze um eins erhöht
- Die Anzahl der freien Plätze kann nicht um eins vermindert werden, wenn die freien Plätze gleich null sind
- Bedingungen:
- Nutzen Sie Threads und Synchronisierung mit den Methoden wait() und notifyAll() ohne Semaphoren
- Das Parkhaus soll fair sein, d.h. es soll sichergestellt sein, dass Autos in der Reihenfolge, in der Sie beim Parkhaus ankommen, einfahren dürfen

```
Auto 26: eingefahren
Auto 7: eingefahren
Auto 39: eingefahren
Auto 10: eingefahren
Auto 36: eingefahren
Auto 5: eingefahren
Auto 19: eingefahren
Auto 22: eingefahren
Auto 36: ausgefahren
Auto 24: eingefahren
Auto 39: ausgefahren
Auto 9: eingefahren
Auto 7: ausgefahren
Auto 18: eingefahren
Auto 12: eingefahren
Auto 16: eingefahren
Auto 16: ausgefahren
Auto 13: eingefahren
Auto 18: ausgefahren
Auto 27: eingefahren
Auto 5: ausgefahren
Auto 33: eingefahren
```

Beispiel-Ausgabe auf der Kommandozeile (Ihre Autos müssen nicht in dieser Reihenfolge einfahren!):