

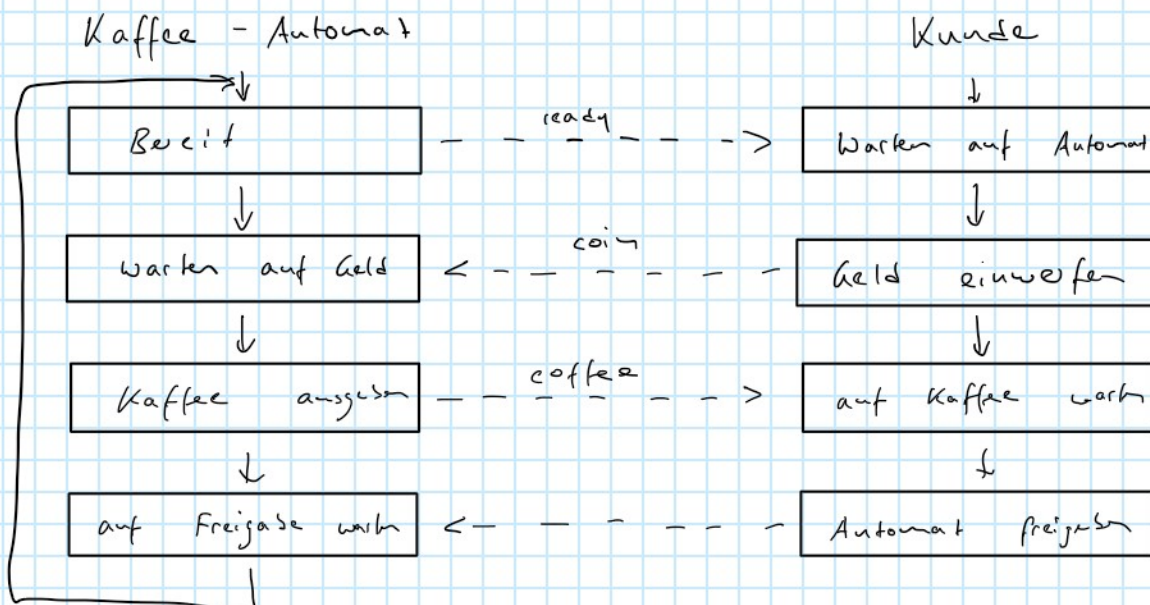
Praktikum 08

Dienstag, 26. April 2022 11:10

4.1

3. Im Thread des Kaffee-Automaten wird an verschiedenen Orten mehrmals auf die gemeinsamen Daten in *cD* zugegriffen. Wenn sie die gemeinsamen Daten in lokale Variablen kopieren und dann nur noch auf diese lokalen Variablen zugreifen würden, könn-ten sie dann auf die Synchronisation mit dem Mutex verzichten?
 4. Wie oft kann ein einzelner Kunde einen Kaffee beziehen, bis der nächste Kunde an die Reihe kommt? Hier reicht eine qualitative Aussage.
-
3. Nein: Die Threads teilen sich den Speicherbereich. Somit sind die lokalen Variablen ebenfalls identisch. Die Zugriffe würden dementsprechend überschreibend sein.
 4. Grundsätzlich 1 Mal. Es kommt auf das Scheduling an. Falls ein Kunde direkt wieder Prozessor-Zeit zugewiesen bekommt ist auch mehrmals nacheinander denkbar.

4.2



5.2

```
vagrant@snp:~/snp/P10_Sync/Sync/banking/a1$ ./startApp

Running 1 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 1.04s

Running 2 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1007418325 ... not correct
Run time 0.64s

Running 4 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1007337282 ... not correct
Run time 0.61s
```

Der Zugriff auf das Bankkonto erfolgt nicht synchronisiert.
 -> Lost Updates, Race Conditions, etc. sind die Folge

```
vagrant@snp:~/snp/P10_Sync/Sync/banking/a1$ ./startApp

Running 1 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 1.54s

Running 2 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 0.93s

Running 4 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 0.94s
```

Wir locken den Account, damit wir nicht eine ganze Filiale blockieren bei einer Transaktion

3.3

```
vagrant@snp:~/snp/P10_Sync/Sync/banking/a2$ ./startApp

Running 1 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 3.33s

Running 2 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 2.12s

Running 4 threads
Balance of accounts is: 1006995400
Balance of accounts is: 1006995400 ... correct
Run time 2.08s
```


Das ist nicht effizient.

Die vielen Locks sind unübersichtlich, beanspruchen viel Zeit und es gibt keinen Nutzen!

Der Mitarbeiter ist nicht so schlau. Die Locks müssen immer von detailliert zu weniger detailliert gelockt werden. Die Locks können sonst nicht mehr freigegeben werden!

```
vagrant@snp:~/snp/P10_Sync/Sync/banking/a2$ ./startApp  
  
Running 1 threads  
Balance of accounts is: 1006995400  
Balance of accounts is: 1006995400 ... correct  
Run time 2.57s  
  
Running 2 threads  
Balance of accounts is: 1006995400  
^C  
vagrant@snp:~/snp/P10_Sync/Sync/banking/a2$ ^C
```

5.4

Die Anforderung ist nicht erfüllt, weil die Informationen nicht konsistent überwiesen werden.

Zuerst wird abgehoben und dann gutgeschrieben. Bei einem Programmabsturz könnte so Geld verloren gehen.

Die Synchronisation mit Transfer wurde bereits in A 5.2 umgesetzt! Die Aufgabe gilt daher als erledigt.