

# Nichtsequentielle und verteilte Programmierung

## Übungsblatt 2

Prof. Dr. Claudia Müller-Birn, Barry Linnert  
Tutorium 01 (Alexander Rudolph)

---

### Aufgabe 1

```
Brücke: besetzt; Auto Nr: 99998; Kollision als Thread: 2; Crash Nummer: 57587  
Counter: 99998  
Brücke: frei; Auto Nr: 99999: Erfolgreich überquert als Thread 1  
Counter: 99999  
Brücke: besetzt; Auto Nr: 100000; Kollision als Thread: 0; Crash Nummer: 57588  
Counter: 100000  
Brücke: besetzt; Auto Nr: 100001; Kollision als Thread: 2; Crash Nummer: 57589  
Counter: 100001  
Gesamtzahl Crashes: 57590,  
Gesamtzahl Crashes: 57590,  
Brücke: frei; Auto Nr: 100002: Erfolgreich überquert als Thread 3  
Counter: 100002  
Gesamtzahl Crashes: 57590,  
Brücke: frei; Auto Nr: 100003: Erfolgreich überquert als Thread 0  
Counter: 100003  
Gesamtzahl Crashes: 57590,
```

### Aufgabe 2

Mit Peterson Locking

```
Brücke: frei; Auto Nr: 99984: Erfolgreich überquert als Thread 1  
Counter: 99984  
Brücke: frei; Auto Nr: 99985: Erfolgreich überquert als Thread 0  
Counter: 99985  
Brücke: frei; Auto Nr: 99986: Erfolgreich überquert als Thread 1  
Counter: 99986  
Brücke: frei; Auto Nr: 99987: Erfolgreich überquert als Thread 0  
Counter: 99987  
Brücke: frei; Auto Nr: 99988: Erfolgreich überquert als Thread 1  
Counter: 99988  
Brücke: frei; Auto Nr: 99989: Erfolgreich überquert als Thread 0  
Counter: 99989  
Brücke: frei; Auto Nr: 99990: Erfolgreich überquert als Thread 1  
Counter: 99990  
Brücke: frei; Auto Nr: 99991: Erfolgreich überquert als Thread 0  
Counter: 99991  
Brücke: frei; Auto Nr: 99992: Erfolgreich überquert als Thread 1  
Counter: 99992  
Brücke: frei; Auto Nr: 99993: Erfolgreich überquert als Thread 0  
Counter: 99993  
Brücke: frei; Auto Nr: 99994: Erfolgreich überquert als Thread 1  
Counter: 99994  
Brücke: frei; Auto Nr: 99995: Erfolgreich überquert als Thread 0  
Counter: 99995  
Brücke: frei; Auto Nr: 99996: Erfolgreich überquert als Thread 1  
Counter: 99996  
Brücke: frei; Auto Nr: 99997: Erfolgreich überquert als Thread 0  
Counter: 99997  
Brücke: frei; Auto Nr: 99998: Erfolgreich überquert als Thread 1  
Counter: 99998  
Brücke: frei; Auto Nr: 99999: Erfolgreich überquert als Thread 0  
Counter: 99999  
Brücke: frei; Auto Nr: 100000: Erfolgreich überquert als Thread 1  
Counter: 100000  
Brücke: frei; Auto Nr: 100001: Erfolgreich überquert als Thread 0  
Counter: 100001  
Gesamtzahl Crashes: 0,
```

### Aufgabe 3

Mit Peterson Locking und 3 Threads (beliebige aber feste Anzahl, funktioniert ebenso mit z.B. 4 oder mehr Threads)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Brücke: frei; Auto Nr: 99979: Erfolgreich überquert als Thread 3
Counter: 99979
Brücke: frei; Auto Nr: 99980: Erfolgreich überquert als Thread 1
Counter: 99980
Brücke: frei; Auto Nr: 99981: Erfolgreich überquert als Thread 0
Counter: 99981
Brücke: frei; Auto Nr: 99982: Erfolgreich überquert als Thread 2
Counter: 99982
Brücke: frei; Auto Nr: 99983: Erfolgreich überquert als Thread 3
Counter: 99983
Brücke: frei; Auto Nr: 99984: Erfolgreich überquert als Thread 1
Counter: 99984
Brücke: frei; Auto Nr: 99985: Erfolgreich überquert als Thread 0
Counter: 99985
Brücke: frei; Auto Nr: 99986: Erfolgreich überquert als Thread 2
Counter: 99986
Brücke: frei; Auto Nr: 99987: Erfolgreich überquert als Thread 3
Counter: 99987
Brücke: frei; Auto Nr: 99988: Erfolgreich überquert als Thread 1
Counter: 99988
Brücke: frei; Auto Nr: 99989: Erfolgreich überquert als Thread 0
Counter: 99989
Brücke: frei; Auto Nr: 99990: Erfolgreich überquert als Thread 2
Counter: 99990
Brücke: frei; Auto Nr: 99991: Erfolgreich überquert als Thread 3
Counter: 99991
Brücke: frei; Auto Nr: 99992: Erfolgreich überquert als Thread 1
Counter: 99992
Brücke: frei; Auto Nr: 99993: Erfolgreich überquert als Thread 0
Counter: 99993
Brücke: frei; Auto Nr: 99994: Erfolgreich überquert als Thread 2
Counter: 99994
Brücke: frei; Auto Nr: 99995: Erfolgreich überquert als Thread 3
Counter: 99995
Brücke: frei; Auto Nr: 99996: Erfolgreich überquert als Thread 1
Counter: 99996
Brücke: frei; Auto Nr: 99997: Erfolgreich überquert als Thread 0
Counter: 99997
Brücke: frei; Auto Nr: 99998: Erfolgreich überquert als Thread 2
Counter: 99998
Brücke: frei; Auto Nr: 99999: Erfolgreich überquert als Thread 3
Counter: 99999
Brücke: frei; Auto Nr: 100000: Erfolgreich überquert als Thread 1
Counter: 100000
Brücke: frei; Auto Nr: 100001: Erfolgreich überquert als Thread 0
Counter: 100001
Gesamtzahl Crashes: 0,
```

### Aufgabe 4

#### Auftretende Effekte

Bei der Programmierung ist es vor allem immer wieder zu Live-Locks bzw. Deadlocks gekommen, dies begründet sich darin, dass wir nicht auf dem selben Maschinenmodell arbeiten wie es zunächst in der Vorlesung gezeigt wurde (1 cpu core). Unsere Maschinen besitzen bis zu 4 und 10 Kerne, weshalb simple locking Methoden immer wieder zu Lock Situationen führen, da die Threads schneller und vor allem „gleichzeitig“ versuchen auf die Variable im kritischen Abschnitt zuzugreifen. In unserem Fall hat es gereicht, den kritischen Abschnitt mithilfe eines Peterson Locks zu sichern. Diese in Aufgabe 2 implementierte Lösung erfüllt bereits die Kriterien der Aufgabe 3, da wir nur die Anzahl der Threads erhöhen müssen, um die beliebige aber feste Anzahl Autos zu simulieren. Ebenso ist das Logging des Outputs schwierig zu handhaben, da die Threads terminieren (und somit auch das Programm) bevor

der Output komplett in das Logfile geschrieben wurde. In unserem Fall wurden von 100 Durchläufen nur die ersten 63 in das Log geschrieben bzw. bei einer Anzahl von Durchläufen  $\leq 50$  wurde gar kein Output generiert. Dies könnte man umgehen indem wir z.b. ein Buffer Array erstellen in dem die Ausgaben zunächst gebuffert werden bevor sie nach Ende der Berechnung in das File geschrieben werden. Wir haben uns dagegen entschieden und lassen uns die Ergebnisse ausprinten.