

## Architektur verteilter Anwendungen – Übung 3

### Aufgabe 1 (Wechselseitiger Ausschluss, Lamport-Algorithmus)

Ziel der Aufgabe ist ein lauffähiges Beispiel zum Lamport-Algorithmus, der den wechselseitigen Ausschluss bzgl. einer Ressource mit Hilfe der Lamport-Zeit und lokalen Request-Queues realisiert ([slides\\_06.pdf](#), 14/15/16)

Die Lösung dieser Aufgabe muss nur auf localhost lauffähig sein, die nur einmal vorhandene Ressource ist eine unter ständiger Modifikation befindliche Datei.

Es sollen sich  $n$  Prozesse den Schreibzugriff auf die Datei  $X$  teilen. Diese enthält zu Beginn den Wert  $000000000$ . Diese 9 Stellen sind notwendig, weil der Wert überschrieben werden soll, ohne den Rest der Datei zu beeinflussen. In C geht das mit

```
fseek(stream, 0L, SEEK_SET) bzw. rewind(stream)
fwrite(...,...,stream)
```

Sehen Sie sich in Ihrer Programmiersprache an, wie man eine Position in einem Filestream setzen kann.

Ein Prozess  $P$  hat folgende Funktionalität:

- er kann von allen anderen das jeweilige Schreibrecht auf die Datei  $X$  anfordern
- nach erhaltenem Schreibrecht modifiziert er den Wert:
  - falls  $P$  eine ungerade Knoten-ID hat, erhöht er den ersten Wert in  $X$  um 1
  - falls  $P$  eine gerade Knoten-ID hat, verringert er den ersten Wert in  $X$  um 1
- hängt seine Prozess-ID an die Datei  $X$  an

Ein Prozess soll (nach dem Schreibvorgang) terminieren, wenn er aus  $X$  das dritte Mal eine 0 gelesen hat.

Am Ende hat  $X$  folgende Form:

```
000000000
1
1
2
1
3
4
2
1
...
```

wobei der Prozess mit ID=1 danach beendet ist (er hat bei allen drei Lesevorgängen eine 0 gesehen).

Der erste Wert wurde von ID=1,1,3,1 erhöht und von 2,4,2 verringert, daher steht an erster Stelle eine 1 und dahinter die beteiligten IDs.