

Entwurf und Implementierung paralleler Programme

Aufgabe 1

Gegeben sei das Leser- und Schreibersystem von Übungsblatt 10. (Das FSP-Modell dazu ist auf der Rückseite angegeben).

- (a) Welche Fortschrittsseigenschaften soll das System erfüllen?
- (b) In welcher Situation kann es bei der Ausführung des Systems zu unfairen Abläufen kommen? Modellieren Sie diese Situation durch Einführung von Prioritäten!
- (c) Welche Fortschrittseigenschaften sind nun verletzt? Beweisen Sie Ihre Aussage für den Fall $N_{read} = 2$. (Zur Vereinfachung können bei den Prozessen *READER* und *WRITER* die Aktionen *examine* und *modify* weggelassen werden.)
- (d) Revidieren Sie das Modell des Leser-Schreiber Systems so, dass alle Fortschrittseigenschaften gelten (und das System weiterhin deadlockfrei ist und die Sicherheitseigenschaften von Übungsblatt 8 erfüllt).

Aufgabe 2

Implementieren Sie das revidierte Leser-Schreiber Modell in Java.

```
const Nread = 3
const Nwrite = 2
range Tread = 1..Nread
range Twrite = 1..Nwrite
range BOOL = False..True
```

```
READER = (acquire → examine → release → READER).
WRITER = (acquire → modify → release → WRITER).
|| RW = (reader[Tread]:READER || writer[Twrite]:WRITER).
```

```
RW_LOCK = RWL[0][False]
RWL[readers:0..Nread][writing:Bool] =
  (when (!writing) reader[Tread].acquire → RWL[readers+1][writing]
   | reader[Tread].release → RWL[readers-1][writing]
   | when(readers==0) && !writing) writer[Twrite].acquire → RWL[readers][True]
   | writer[Twrite].release → RWL[readers][False]).
```

```
|| RW_SYS = (RW || RW_LOCK).
```