

Lösung Übungsblatt 3

Christoph van Heteren-Frese (Matr.-Nr.: 4465677),

Sven Wildermann (Matr.-Nr.: 4567553)

Tutor: Alexander Steen, eingereicht am 2. Mai 2013

Aufgabe 1

Eine binäre Semaphore kann wie ein Schloss eingesetzt werden. Die Spezifikation einer binärer Semaphore ist in [1, S. 54] und die Implementierung von `Lock` und `Unlock` in [1, S. 55] zu finden. Initialer Wert für `s`: `true` (bzw. 1).

Aufruf	Auswirkung
P1: s.P	<code>s = false</code> ; P1 betritt kritischen Bereich
P2: s.P	<code>s = false</code> ; P2 muss warten: <i>aktiv</i> \rightarrow <i>blockiert</i>
P3: s.P	<code>s = false</code> ; P3 muss warten: <i>aktiv</i> \rightarrow <i>blockiert</i>
P1: s.V	P1 verlässt kritischen Bereich; <code>s = true</code> ; P3: <i>blockiert</i> \rightarrow <i>bereit</i> ; P3 betritt kritischen Bereich; <code>s = false</code>
P4: s.P	<code>s = false</code> ; P4 muss warten: <i>aktiv</i> \rightarrow <i>blockiert</i>
P3: s.V	P3 verlässt kritischen Bereich; <code>s = true</code> ; P2: <i>blockiert</i> \rightarrow <i>bereit</i> ; P2 betritt kritischen Bereich; <code>s = false</code>
P5: s.P	<code>s = false</code> ; P5: <i>aktiv</i> \rightarrow <i>blockiert</i>

Tabelle 1: Protokoll der Aufrufe

Erläuterung: Die Tatsache, dass P3 in diesem Beispiel als zweiter Prozess den kritischen Bereich betritt, ergibt sich durch die vorgegebene Reihenfolge der Aufrufe, da P3 der nächste Prozess ist, der eine V-Operation ausführt und „[...] Prozesse P- und V-Operationen stets paarig – in dieser Reihenfolge – ausführen [müssen][vgl. 1, S. 55].“

Grundsätzlich gilt: „[...] ein Aufruf von `s.V()` realisiert den Übergang von *blockiert* \rightarrow *bereit* für einen der auf `s` blockierten Prozesse [...] [vgl. 1, S. 55].“ Welcher Prozess das ist, hängt von der Prozessverwaltung des Betriebssystems ab.

Aufgabe 2

a)

b)

c)

Aufgabe 3

Aufgabe 4

Literatur

- [1] Christian Maurer. *Nichtsequentielle Programmierung mit Go 1 Kompakt*. Springer Vieweg, 2012. ISBN 978-3642299681.