



## Nebenläufiges Programmieren

Sommersemester 2008

Serie 7

21.5. 2008

**Ausgabetermin: 21.5. 2008**

**Abgabe: 30.5. 2008 (11:00)**

**Aufgabe 1 (4 Punkte)** *n-Process Barrier*. Man kann eine wiederbenutzbare Barriere für  $n$  Prozesse unter Benutzung von zwei Semaphoren und einer Zählvariable programmieren. Entwickeln Sie eine solche Lösung.

(Tip: Benutzen Sie die Idee “passing the baton”.)

**Aufgabe 2 (6 Punkte)** Betrachten Sie die folgende Lösung des readers/writers Problems (siehe nächste Seite).

- Erklären Sie sorgfältig wie die Lösung funktioniert und belegen Sie Ihre Argumente durch den Programmcode. Gehen Sie insbesondere darauf ein, wie der exklusive Zugriff einzelner writer sowie zwischen readern und writern gestaltet ist.
- Diskutieren Sie gründlich die folgende Frage: Ist die angegebene Lösung fair oder wird eine Prozessgruppe bevorzugt? Argumentieren Sie auch hier mit dem Programmcode.
- Vergleichen Sie die Lösung mit der in Figure 4.13. Zeigen Sie die Unterschiede auf und vergleichen Sie die Anzahlen der P und V Aufrufe im besten und schlechtesten Fall.

Listing 1: readers/writers

```
1 int nr = 0, nw = 0; # numbers of readers and writers
2 sem e = 1; # mutual exclusion semaphore
3 sem r = 0, w = 0; # delay semaphores
4 int dr = 0, dw = 0; # delay counters
5
6 process Reader[i = 1 to M] {
7     while (true) {
8         P(e);
9         if (nw == 0) { nr = nr+1; V(r); }
10        else dr = dr+1;
11        V(e);
12        P(r); # wait for permission to read
13        read the database;
14        P(e);
15        nr = nr-1;
16        if (nr == 0 and dw > 0)
17            { dw = dw-1; nw = nw+1; V(w); }
18        V(e);
19    }
20 }
21
22 process Writer [j = 1 to N] {
23     while (true) {
24         P(e);
25         if (nr == 0 and nw == 0) { nw = nw+1; V(w); }
26         else dw = dw+1;
27         V(e);
28         P(w);
29         write the database;
30         P(e);
31         nw = nw-1;
32         if (dw > 0) { dw = dw-1; nw = nw+1; V(w); }
33         else
34             while (dr > 0) { dr = dr-1; nr = nr+1; V(r); }
35         V(e);
36     }
37 }
```