

## Übungen zu „Parallele Programmierung“, SS 2005

Nr. 4, Abgabe der Aufgaben: 12.Mai 2005 vor der Vorlesung

### Hinweise:

Am 11. Mai findet die Vorlesung in HS II statt.

## Aufgaben

### 4.1 Leser-Schreiber-Problem

4 Punkte
----------

Betrachten Sie die folgende Lösung des Leser/Schreiber-Problems.

```

int nr = 0, nw = 0;
sem e = 1;
sem r = 0, w = 0;
int dr = 0, dw = 0;

proc reader(){
while (true) {
P(e);
if (nw == 0)
{ nr += 1; V(r);}
else { dr += 1 }
V(e);
P(r);
# lese Daten
P(e);
nr -= 1;
if (nr == 0 and dw > 0)
{dw -= 1; nw = 1; V(w); }
V(e);
}
}

proc writer(){
while (true) {
P(e);
if (nr == 0 and nw == 0)
{ nw = 1; V(w);}
else { dw += 1 }
V(e);
P(w);
# schreibe Daten
P(e);
nw = 0;
if (dw > 0)
{ dw -= 1; nw = 1; V(w); }
else
{ while (dr > 0) {
dr -= 1; nr += 1; V(r)}
}
V(e);
}
}

```

- Beschreiben Sie die Funktionsweise dieser Synchronisation, insbesondere die Rolle der verwendeten Semaphore.
- Begründen Sie, dass Schreiber exklusiven Zugriff auf die Daten haben. Ist die Synchronisation *fair*, oder werden Schreiber oder Leser bevorzugt?

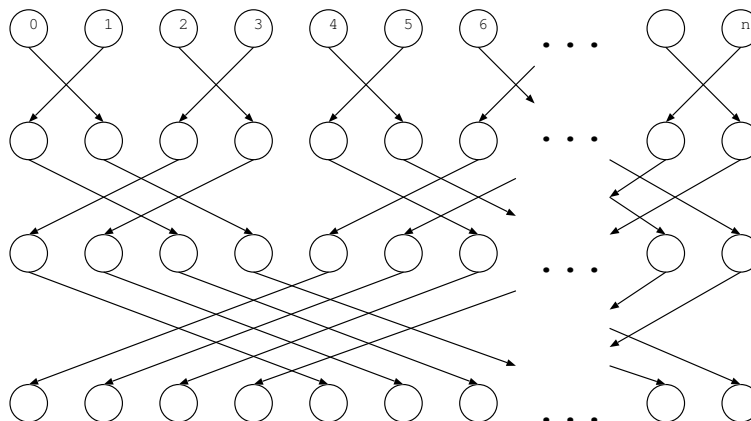
Bitte wenden!

## 4.2 Optimierte Barrieren

4 Punkte

Wie in der Vorlesung angesprochen wurde, ermöglicht die Technik des *rekursiven Doppeln*s effiziente Realisierungen von Barrieren mit Aufwand  $O(\log_2 n)$ .

Programmieren Sie eine Barriere, welche ein sog. *Butterfly-Schema* benutzt:



**Hinweis:** Arbeiten Sie mit einer Folge symmetrischer Barrieren in einem Semaphorenfeld und verwenden Sie bit-Operationen zur Adressierung.

## 4.3 Simulation von Monitoren durch Semaphore

4 Punkte

Erläutern Sie, wie man Monitore durch Semaphore simulieren kann. Dabei können Sie zur Vereinfachung annehmen, daß ein `signal` nur unmittelbar vor dem Verlassen einer Monitorprozedur möglich ist. Das bedeutet, daß die Signalisierungsart 'SX' (signal and exit) benutzt wird.