```
1 void reader(int i)
2 {
3         message rmsg;
4         while(true)
5         {
6                 rmsg.id = i;
7                 send(readrequest, rmsg);
8                 receive(mbox[i], rmsg);
9                 READUNIT();
10                rmsg.id = i;
11                send(finished, rmsg);
12        }
13 }
14
15 void writer(int j)
16 {
17        message wmsg;
18        while(true)
19        {
20                wmsg.id = j;
21                send(writerequest, wmsg);
22                receive(mbox[j], wmsg);
23                WRITEUNIT();
24                wmsg.id = j;
25                send(finished, wmsg);
26        }
27 }
28
29
```

```
30 //Count wird initalisiert mit einem Wert, der größer ist als
31 //die max. Anzahl an Leser (hier 100)
32 void controller()
33 {
34       while(true)
35       {
36       //keine Schreiber warten, evtl. Leser aktiv
37                if(count>0)
38                {
39                         //alle FINISH-Nachrichten abarbeiten
40                         if (!empty(finished))
41                         {
42                                  receive(finished, msg);
43                                  count++;
44                         }
45                         //Schreibanfragen bearbeiten
46                         else if(!empty(writerequest))
47                         {
48                                  receive(writerequest, msg);
49                                  writer_id = msg.id;
50                                  count = count - 100;
51                         }
52                         //Leseanfragen bearbeiten
53                         else if(!empty(readrequest))
54                         {
55                                  receive(readrequest, msg);
56                                  count--;
57                                  send(msg.id, "OK")
58                         }
59                }
60
61                //die einzige offenstehende Anfrage ist eine
62                //Schreibanfrage
63                if(count == 0)
64                {
65                         send(writer_id, "OK");
66                         receive(finished, msg);
67                         count = 100;
68                }
69
70                // eine Schreibanfrage ist eingetroffen,
71                // der Schreiber wartet auf die Beendigung aller
72                // Leseprozesse
73                while(count < 0)
74                {
75                         receive(finished, msg);
76                         count++;
77                }
78       }
79 }
```