

```
1  int readers;                                // Negative value => active writer
1.5 int readWaiters = 0;
2  pthread_mutex_t lock;
3  pthread_cond_t busy;                        // Use one condition variable to
4                                           // indicate whether data is busy
5  AcquireExclusive()
6  {
7      pthread_mutex_lock(&lock);              // This code suffers from spurious
8      while(readers!=0)                       // wake-ups!!!
9      {
10         pthread_cond_wait(&busy, &lock);
11     }
12     readers=-1;
13     pthread_mutex_unlock(&lock);
14 }
15
16 AcquireShared()
17 {
18     pthread_mutex_lock(&lock);
19     while(readers<0)
20     {
20.5         readWaiters++;
21         pthread_cond_wait(&busy, &lock);
21.5         readWaiters--;
22     }
23     readers++;
24     pthread_mutex_unlock(&lock);
25 }
26
27 ReleaseExclusive()
28 {
29     pthread_mutex_lock(&lock);
30     readers=0;
30.5     if (readWaiters==0)                    // If there are no waiting readers
30.6         pthread_cond_signal(&wBusy);      // Wake up a writer
30.7     else
31         pthread_cond_broadcast(&rBusy);    // Wake up all readers
32     pthread_mutex_unlock(&lock);
33 }
34
35 ReleaseShared()
36 {
37     pthread_mutex_lock(&lock);
38     readers--;
39     if (readers==0)
40     {
41         pthread_cond_signal(&busy);
42     }
43     pthread_mutex_unlock(&lock);
44 }
```