

Reader/Writer Problem

A data object is to be shared among several concurrent processes. Some of these may only want to read the data structure (the readers), while others want to write/modify it (the writers). The constraints are as follows:

- Many readers can be reading simultaneously.
- Only one writer can be writing at a time.
- No readers can be reading while a writer is writing.

Sounds simple, doesn't it?

Try it now.

Here's the solution:

```
Semaphore w = new Semaphore(1);  
Semaphore r = new Semaphore(1);  
int readcount = 0;
```

Writer:

```
Forever do {  
    ...//whatever  
    w.acquire();  
    ...//do writing  
    w.release();  
    ...//whatever  
}
```

Reader:

```
Forever do {  
    ...//whatever  
    r.acquire();  
    readcount++;  
    if (readcount==1) w.acquire();  
    r.release();  
    ...//do reading  
    r.acquire();  
    readcount--;  
    if (readcount==0) w.release();  
    r.release()  
}
```

Notice any problems? Readers can starve out writers.

Another solution, not given here, gives preference to a writer who wants in. That solution can starve readers. A starvation-free solution is much more complicated.