读者/写者问题

1. 读者与读者进程（或线程）之间可以共享数据。
2. 读者与写者进程（或线程）之间互斥使用数据。
3. 写者与写者进程（或线程）之间互斥使用数据。
4. 当数据文件被读时，其为共享文件；被写时其为互斥文件

**简单调度**

**S=1：数据是否空闲**

**读者进程**

P(s)或down(s)

读操作

V(s) 或up(s)

**写者进程**

P(s)或down(s)

写操作

V(s) 或up(s)

**r1 r2 w1 r3 w2 w3 w4 r4 w5 r5**

**效率调度（读者优先）**

**sr=1：用于控制xr相关的临界区**

**Int xr=0;正在读操作的进程数量**

**读者进程**

P(sr) //获得xr相关临界区权限

If(xr==o) P(s);//当获得s控制的缓冲区权限后，所有读进程都无需再提出申请直接访问缓冲区即可，但写进程会被阻塞

Xr=xr+1;

V(sr)

读操作

P(sr)

xr=xr-1;

If(xr==0) V(s);//当xr为0证明该进程为最后一个读进程，则释放s所控制的缓冲区权限

V(sr) //释放sr控制的xr权限

**写者进程**

P(s)或down(s)

写操作

V(s) 或up(s)

算法缺陷：公平性较差，只要有读者访问则一直占用缓冲区权限，实际上读者优先

修改方案：设sw作为写进程变量，xw为sw信号量控制xw临界区（当所有读进程和其它写进程提出申请前都应访问xw所以xw临界区十分危险）但当读进程意识到xw=1后其应该在何处sleep并被何者唤醒时间复杂的问题