

第六章主观题

1. 什么是临界区？对临界区的访问应该遵循什么准则？

临界区:访问临界资源的那段代码。

准则:

- ✧ 互斥，如果有进程在某个临界区执行，其他进程将被排斥在临界区外，有相同临界资源的临界区都需要互斥，无相同临界资源的临界区不需要互斥。
- ✧ 有空让进，临界区内无进程执行，不能无限期地延长下一个要进入临界区的进程的等待时间。
- ✧ 有限等待，每个进程进入临界区前的等待时间必须有限，不能无限等待。

2. 请谈谈同步信号量的值有什么含义。

- ✧ 大于 0 时，表示这一资源没有被使用的数量，或者释放的数量，可分配请求使用这个资源的进程。
- ✧ 等于 0 时，表示这一资源分配完毕，或者没有被释放，如果现在有进程申请这个资源，必须等待。
- ✧ 小于 0 时，表示请求这一资源而没有得到正在等待这一资源的数目。

3. 有四个进程 S1、R1、R2 和 R3，其中 S1 向缓冲区 BUFF 发送消息，R1、R2 和 R3 从缓冲区中接收消息。发送和接收的规则如下：

- 1) 缓冲区 BUFF 任何时候只能存放 1 个消息；
- 2) R1、R2 和 R3 每次可取 S1 存放在缓冲区中的消息；
- 3) 每个存放在缓冲区中的消息必须被 R1、R2 和 R3 均接收后才能清除。

请用信号量机制来实现这 4 个进程间的同步。

| | | | |
|-------------------------------------|---------------|---------------|---------------|
| 三、信号量初值S1=1,S2=0,S3=0,S4=0,MUTEX=1; | | | |
| int Count=0; | | | |
| S1 | R1 | R2 | R3 |
| repeat | repeat | repeat | repeat |
| P(S1) | P(S2) | P(S3) | P(S4) |
| Send message | get message | get message | get message |
| P(MUTEX) | P(MUTEX) | P(MUTEX) | P(MUTEX) |
| Count=0 | Count=Count+1 | Count=Count+1 | Count=Count+1 |
| V(S2) | if(Count=3) | if(Count=3) | if(Count=3) |
| V(S3) | V(S1) | V(S1) | V(S1) |
| V(S4) | | | |
| V (MUTEX) | V(MUTEX) | V(MUTEX) | V(MUTEX) |
| until false | until false | until false | until false |

4、下面是用 wait/signal 机制解决写者优先的 Readers-Writers Problem 的伪代码。
在下面空格处填空，并回答该题目最后面的问题。
所谓写者优先的 Readers-Writers Problem 定义如下：

```
int rcount, wcount; (initial value = 0)  
semaphore m1, m2, m3, w, r; (initial value = 1)
```

READER:

```
wait(m1); //所有的 readers 竞争抢 w 信号量的权利  
wait(r); //所有抢 readers 竞争 w 信号量的过程是互斥的  
wait(m2);  
rcount := rcount + 1;  
if rcount = 1 then wait(w); //只有第一个 reader 能够有资格抢 w 信号量  
signal(m2);  
signal(r);  
signal(m1);  
reading  
wait(m2);  
rcount := rcount - 1;  
if rcount = 0 then signal(w); //还有 readers 在等，就不放 w 信号量  
signal(m2);
```

WRITER:

```
wait(m3);  
wcount := wcount + 1;  
if wcount = 1 then wait(r);  
signal(m3);  
  
wait(w);  
writing  
signal(w);  
  
wait(m3);  
wcount := wcount - 1;  
if wcount = 0 then signal(r);  
signal(m3);
```

在上面这个解决方案中，有饥饿现象发生，例如可以发生在这样的情况下：

如果一直有写者进入临界区，读者始终无法进入临界区，会产生饥饿现象