

# 习题6

## 1

什么是临界区？对临界区的访问应该遵循什么准则？

临界区是进程中访问临界资源的代码段

准则：

- 空闲让进
- 忙则等待
- 有限等待
- 让权等待

## 2

请谈谈同步信号量的值有什么含义。

信号量其实就是一个变量 (可以是一个整数，也可以是更复杂的记录型变量)，可以用一个信号量 来表示系统中某种资源的数量

## 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。

`mutex` 用来保证互斥访问.保证在一个进程操作BUFF时,另一个进程不会操作BUFF

`msg` 用来显示BUFF中的消息数量

`r1Done`, `r2Done`, `r3Done` 用来表示三个线程是否读完BUFF中的内容

```

1  semaphore mutex = 1;
2  semaphore msg = 0;
3  semaphore r1Done = 0, r2Done = 0, r3Done = 0;
4
5
6  S1() {
7      while (true) {
8          // 生成消息
9          P(mutex);
10         // 将消息放入 BUFF
11         V(mutex);

```

```

12     V(msg);
13     // 等待三个接收者都接收了消息
14     P(r1Done); P(r2Done); P(r3Done);
15     // 清除 BUFF 中的消息
16 }
17 }
18
19
20 R1() {
21     while (true) {
22         P(msg);
23         P(mutex);
24         // 读取 BUFF 中的消息
25         V(mutex);
26         V(r1Done);
27     }
28 }
29
30
31 R2() {
32     while (true) {
33         P(msg);
34         P(mutex);
35         // 读取 BUFF 中的消息
36         V(mutex);
37         V(r2Done);
38     }
39 }
40
41 R3() {
42     while (true) {
43         P(msg);
44         P(mutex);
45         // 读取 BUFF 中的消息
46         V(mutex);
47         V(r3Done);
48     }
49 }
50

```

## 4

桌上有一个空的水果盘，且盘中一次只能放一个水果。爸爸专向盘中放苹果，妈妈专向盘中放桔子，一个儿子专等吃盘子中的桔子，一个女儿专等吃盘子中的苹果。固定每次当盘子空时爸爸或妈妈可向盘中放一个水果，仅当盘中有自己需要的水果时，儿子或女儿可从中取出。请用 PV 操作实现爸爸、妈妈、儿子和女儿四个进程的同步。

```
1 semaphore plate = 1;
2 semaphore apple = 0;
3 semaphore orange = 0;
4
5 dad(){
6     while(true){
7         //准备苹果
8         P(plate);
9         //放入苹果
10        V(apple);
11    }
12 }
13
14 mom(){
15     while(true){
16         //放入橘子
17         P(plate);
18         //放入桔子
19         V(orange);
20     }
21 }
22
23 son(){
24     while(true){
25         P(apple);
26         //取出苹果
27         V(plate);
28         //吃掉苹果
29     }
30 }
31
32 doughter(){
33     while(true){
34         P(orange);
35         //取出橘子
36         V(plate);
37         //吃掉橘子
38     }
39 }
40
41
42
```