

操作系统练习题（二）

四、应用题

1. 答：设有两个信号量 $\text{Sin}=1$ ， $\text{Sout}=0$ ， Sin 表示缓冲区中有无空位置， Sout 表示缓冲区中有无数据。两个进程的同步可以描述如下：

compute:

 得到一个计算结果;

$\text{P}(\text{Sin})$;

 将数放入缓冲区;

$\text{V}(\text{Sout})$;

print:

$\text{P}(\text{Sout})$

 从缓冲区取数;

$\text{V}(\text{Sin})$;

 将数输出到打印机

2. 答：设有四个信号量： $\text{SR}=1$; $\text{SM1}=0$; $\text{SM2}=1$; $\text{SP}=0$

Process read:

 接收的一个记录 X;

$\text{P}(\text{SR})$;

$\text{B1}:=\text{X}$;

$\text{V}(\text{SM1})$;

Process move:

$\text{P}(\text{SM1})$;

$\text{Y}:=\text{B1}$;

$\text{V}(\text{SR})$

 加工 Y

$\text{P}(\text{SM2})$;

$\text{B2}:=\text{Y}$;

$\text{V}(\text{SP})$;

Process print:

$\text{P}(\text{SP})$;

$\text{Z}:=\text{B2}$;

$\text{V}(\text{SM2})$

 打印 Z

3. 答：为了安全起见，显然要求：关车门后才能启动车辆；到站停车后才能开车门。所以司机和售票员在到站、开门、关门、启动车辆这几个活动之间存在着同步关系。用两个信号量 S1 、 S2 分别表示可以开车和可以开门, S1 、 S2 的初值均应为 0。

司机:

$\text{P}(\text{S1})$

启动车辆

售票员:

售票

$\text{P}(\text{S2})$

正常行车
到站停车
V (S2)

开车门
关车门
V (S1)

4. 答:(1)Sr 用于读者计数 rc 的互斥信号量;

(2)if rc=1 then P(S)中的 P(S)用于读写互斥, 写者进程中的 P(S)用于写写互斥, 读写互斥。

(3)程序中增加一个信号量 S5, 初值为 5, P(S5)语句加在读者进程 P(Sr)之前, V(S5)语句加在读者进程第 2 个 V(Sr)之后。

5. 答: (1)定义一信号量 S, 初始值为 20。

意义: $S > 0$ S 的值表示可继续进入售票厅的人数

$S = 0$ 表示售票厅中已有 20 名顾客(购票者)

$S < 0$ |S|的值为等待进入售票厅的人数

(2) PROCESS $P_i(I=1, 2, \dots)$

cobegin

P(S);

进入售票厅;

购票;

退出;

V(S)

coend;

(3)S 的最大值为 20

S 的最小值为 $20 - n$

6. 答: (1) 每个读者都可视为一个进程, 有多少个读者就有多少个进程, 这些进程称为读者进程, 设为 $P_i(I=0,1,2,\dots)$ 。读者进程 P_i 执行的程序包括动作: 登记、阅览、撤消。每个读者的活动都相同, 所以其程序也相同。进程与程序之间的关系是: 各读者进程共享同一个程序。

(2) 在读者进程执行的程序中, 登记、撤消都需要互斥执行, 其信号量 S1 的初值为 1; 而对进入阅览室需判断是否有空座位, 信号量 s2 的初值为 200, 表示空座位数。

读者进程 P_i :

P(S2)

P(S1)

登记

V(S1)

阅览

P(S1)

撤消

V(S1)

V(S2)

7. 答: 设信号量 $S1=1$ (互斥: 每次只能存入一本书), $S2=8$ (还能放 8 本语文书), $S3=7$ (还能放 7 本数学书)。

进程 PA:

进程 PB:

P(S2)

P(S3)

P(S1)	P(S1)
放语文书	放数学书
V(S1)	V(S1)
V(S3)	V(S2)

8. 答:

```

开锁原语 unlock(W)
    W=0;
关锁原语 lock(W)
if(W==1) do no op;
W=1;
利用开关锁原语实现互斥:
int W=0;
cobegin
lock(W);
critical section
unlock(W);
remainder section
coend

```

9. 答: (1) A、B、C 三个进程之间存在互斥的制约关系, 因为打印机属于临界资源, 必须一个进程已用完之后另一个进程才能使用。

(2) mutex: 用于互斥的信号量, 初值为 1。

各进程的代码如下:

进程 A	进程 B	进程 C
P (mutex)	P (mutex)	P (mutex)
使用打印机	使用打印机	使用打印机
V (mutex)	V (mutex)	V (mutex)

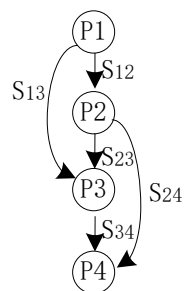
10. 答: (1) A、B 两个进程之间存在互斥的制约关系, 因为打印机属于临界资源, 必须一个进程已用完之后另一个进程才能使用。

(2) mutex: 用于互斥的信号量, 初值为 1。

各进程的代码如下:

进程 A	进程 B
P (mutex)	P (mutex)
使用打印机	使用打印机
V (mutex)	V (mutex)

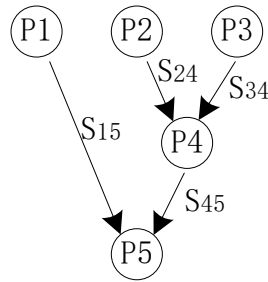
11. 答: 设信号量 $S_{12}=0, S_{13}=0, S_{23}=0, S_{24}=0, S_{34}=0$



进程代码略

12. 答：进程 P1: $A*B$ P2: $C+D$ P3: $E-F$ P4: $(C+D) * (E-F)$

P5: $A*B-(C+D) * (E-F)$



设信号量 $S_{15}=0, S_{24}=0, S_{34}=0, S_{45}=0$

P1:
$A*B;$
$V(S_{15})$

P2:
$C+D;$
$V(S_{24})$

P3:
$E-F;$
$V(S_{34})$

P4:
$P(S_{24})$
$P(S_{34})$
$(C+D) * (E-F)$
$V(S_{45})$

P5:
$P(S_{15})$
$P(S_{45})$
$A*B-(C+D) * (E-F)$

13. 答：

```

#define CHAIRS 6          /*为等候的顾客准备的椅子数*/
semaphore customers=0;
semaphore barbers=0;      /* 用控制理发师与顾客的先后顺序 */
semaphore S=1;           /*用于对 waiting 资源进行互斥*/
int waiting=0;
void barber() {
    while (True) {
        P(customers);
        P(S);
        waiting =waiting -1;
        V(S);
        V(barbers);
        请一个等候的顾客到理发椅上坐，开始理发...
    }
}
void customer()
P(S);
if (waiting<CHAIRS) {
    waiting=waiting+1;
    V(S);
    在等候椅上落座；
    V(customers);
    P(barbers);
    在理发椅子上落座，直到理发完成，最后离开
} else {
    V(S);
}
}
  
```

14. 答: S 的初值: S=3

P(s)

UseFloopy()

V(s)

15. 答: 0 0 1 1

sStorageOver sRecToDisplay sDisplayOver sStorageOver

16. 答: 2 0 0 1

sEmpty sMutex sOrange sApple

17. (1) 发生变迁 2、变迁 3、变迁 4 的原因是什么?

发生变迁 2 的原因是时间片到, 变迁 3 的原因是请求 I/O 服务, 变迁 4 的原因是 I/O 服务完成。

(2)

① 3 —> 1 可能发生, 正在运行的进程由于请求 I/O 服务而阻塞 (变迁 3), 离开 CPU, 调度程序从就绪队列中选择一个进程使用 CPU (变迁 1)。

② 2 —> 1 可能发生, 正在运行的进程由于时间片到而变为就绪 (变迁 2), 调度程序从就绪队列中选择一个进程使用 CPU (变迁 1)。

③ 3 —> 2 不可能发生

④ 4 —> 1 不可能发生