

操作系统 2022 课后应用题作业 2

姓名：刘晓旭 学号：201250123

提醒：直接在本文档填写解题答案，

提交作业的文件名命名规范为【学号_姓名_作业 2.doc】

题序	1 满分 8 分	2 满分 6 分	3 满分 8 分	4 满分 8 分	5 满分 6 分	6 满分 8 分	7 满分 16 分	8 满分 16	9 满分 16	10 满分 8	总分
分值											

1、假定磁盘有 200 个柱面，编号 0~199，当前移动臂位于 143 号柱面上，并刚刚完成 125 号柱面的服务请求。如果请求队列的先后顺序是：86，147，91，177，94，150，102，175，130；试问：为了完成上述请求，下列算法移动臂所移动的总量分别是多少？并给出移动臂移动的顺序。①先来先服务算法；②最短查找时间优先算法；③双向扫描算法；④电梯调度算法。

答：（12 分，每小题 3 分）

① 先来先服务算法：

$$143 \xrightarrow{57} 86 \xrightarrow{61} 147 \xrightarrow{56} 91 \xrightarrow{86} 177 \xrightarrow{83} 94 \xrightarrow{56} 150 \xrightarrow{48} 102 \xrightarrow{73} 175 \xrightarrow{45} 130$$
$$57 + 61 + 56 + 86 + 83 + 56 + 48 + 73 + 45 = 565$$

② 最短查找时间优先算法：

$$143 \xrightarrow{4} 147 \xrightarrow{3} 150 \xrightarrow{20} 130 \xrightarrow{28} 102 \xrightarrow{8} 94 \xrightarrow{3} 91 \xrightarrow{5} 86 \xrightarrow{89} 175 \xrightarrow{2} 177$$
$$4 + 3 + 20 + 28 + 8 + 3 + 5 + 89 + 2 = 162$$

③ 双向扫描算法：

$$143 - 147 - 150 - 175 - 177 - 199 - 130 - 102 - 94 - 91 - 86$$
$$(199 - 143) + (199 - 86) = 169$$

④ 电梯调度算法：

$$143 - 147 - 150 - 175 - 177 - 130 - 102 - 94 - 91 - 86$$
$$169 - 22 \times 2 = 125$$

2、有一个磁盘组共有 10 个盘面，每个盘面有 100 个磁道，每个磁道有 16 个扇区。若以扇区为分配单位，现问：(1)用位示图管理磁盘空间，则位示图占用多少空间？(2)若空白文件目录的每个目录项占 5 个字节，则什么时候空白文件目录大于位示图？

答：（6 分，每小题 3 分）

(1) 一个盘块 \leftrightarrow 一个扇区.

$$10 \times 100 \times 16 = 16000 \text{ bits} = 2000 \text{ Bytes}$$

$$(2) 2000/5 = 400$$

当空白文件目录项 > 400 , 空白文件目录大于位示图。

3、假设在 Unix 文件系统中, inode 节点中分别含有 10 个直接地址的索引和一、二、三级间接索引。若设每个盘块有 512B 大小, 每个盘块中可存放 128 个盘块地址, 则(1)一个 1MB 的文件占用多少间接盘块? (2) 一个 25MB 的文件占用多少间接盘块? (8 分, 每小题 4 分)

答:

$$(1) 10 \text{ 个直接地址索引: } 10 \times 512 \text{ B} / 1024 = 5 \text{ KB}$$

$$\text{一级间接索引: } 128 \times 512 \text{ B} / 1024 = 64 \text{ KB}$$

$$\text{二级间接索引: } 128^2 \times 512 \text{ B} / 1024 = 8192 \text{ KB}$$

$$\text{三级间接索引: } 128^3 \times 512 \text{ B} / 1024 = 1048576 \text{ KB}$$

$$1 \text{ MB} = 1024 \text{ KB} \quad 1024 \text{ KB} - 5 \text{ KB} - 64 \text{ KB} = 955 \text{ KB} \quad 955 \text{ KB} / 512 \text{ B} = 1910$$

\therefore 占用 128 一次间接盘块和 1910 个二次间接盘块。

$$(2) 25 \text{ MB} = 25600 \text{ KB} \quad 25600 \text{ KB} - 5 \text{ KB} - 64 \text{ KB} - 8192 \text{ KB} = 17339 \text{ KB} \quad \frac{17339 \text{ KB}}{512 \text{ B}} = 34678$$

\therefore 占用 128 一次间接盘块, 16384 二次间接盘块, 34678 三次间接盘块。

4、【基本概念】(8 分, 每小题 4 分)

设有 n 个进程共享一个互斥段, 如果: ①每次只允许一个进程进入互斥段; ②每次最多允许 m 个进程 ($m \leq n$) 同时进入互斥段。

试问: 以上两种情况下所采用的信号量初值是否相同? 试给出信号量值的变化范围。

答:

初值不相同。

①: 初值为 1, 每有一个进程进入/等待进入互斥段, 信号值-1, 所以信号量的变化范围是 $[1-n, 1]$ 。

②: 初值为 m , 每有一个进程进入/等待进入互斥段, 信号值-1, 所以信号量的变化范围是 $[m-n, m]$ 。

5、【基本概念】(6 分)

有两个优先级相同的进程 P1 和 P2, 其各自程序如下, 信号量 S1 和 S2 的初值均 0。试问 P1、P2 并发执行后, x 、 y 、 z 的值各为多少?

P1() { $y=1$; $y=y+3$; V(S1);	P2() { $x=1$; $x=x+5$; P(S1);
--	--

$z=y+1;$ $P(S2);$ $y=z+y;$ $\}$	$x=x+y;$ $V(S2);$ $z=z+x;$ $\}$
--	--

答：定义左边的代码为 1-1, 1-2, ……，1-6。右边的代码为 2-1, 2-2, ……，2-6。

几种可能的顺序：

P1 先执行：

1-1>1-2>1-3>1-4>1-5>2-1>2-2>2-3>2-4>2-5>1-6>2-6; (x10,y9,z15)

1-1>1-2>1-3>1-4>1-5>2-1>2-2>2-3>2-4>2-5>2-6>1-6。 (x10,y19,z15)

P2 先执行：

2-1>2-2>2-3>1-1>1-2>1-3>2-4>2-5>2-6>1-4>1-5>1-6; (错误)

2-1>2-2>2-3>1-1>1-2>1-3>1-4>1-5>2-4>2-5>1-6>2-6; (x10,y9,z15)

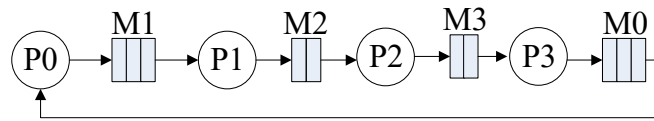
2-1>2-2>2-3>1-1>1-2>1-3>1-4>1-5>2-4>2-5>2-6>1-6。 (x10,y19,z15)

所以 xyz 的值有两种：

X=10, Y=9, Z=15;

X=10, Y=19, Z=15。

6、【PV】四个进程 P_i ($i=0\dots3$) 和四个信箱 M_j ($j=0\dots3$)，进程间借助相邻信箱传递消息，即 P_i 每次从 M_i 中取一条消息，经加工后送入 $M_{(i+1)\bmod 4}$ ，其中 M_0 、 M_1 、 M_2 、 M_3 分别可存放 3、3、2、2 个消息。初始状态下， M_0 装了三条消息，其余为空。试以 P、V 操作为工具，写出 P_i ($i=0\dots3$) 的同步工作算法。



答：(8 分)

首先定义进程对邮箱的占用信号量：

Semaphore $m0o, m1o, m2o, m3o = 1$; // 信号量为 1，最多允许 1 个占用

然后定义邮箱的剩余空间信号量：

Semaphore $empty0 = 0, empty1 = 3, empty2 = empty3 = 2$; // 信号量为 0 进入队列

然后定义邮箱的已装载邮件信号量：

Semaphore $full0 = 3, full1 = full2 = full3 = 0$;

然后定义放入取出邮件的位置：

Int $in0, out0, in1, out1, in2, out2, in3, out3 = 0$;

思路：检查邮箱直至有消息并减小邮件占用数——检查邮箱占用状态直至能占用——取消息并更改下一次取消息位置——释放邮箱占用——增加邮箱的空闲数——加工消息——投递邮件（相当于加工消息前的反向过程）

cobegin

```

process P0() {
while(true) {
P(full0);
P(m0o);
//取消息代码
out0=(out0+1) % 3;
V(m0o);
V(empty0);
//加工消息代码
P(empty1);
P(m1o);
//存消息代码
in1=(in1+1) % 3;
V(m1o);
V(full1);
}
}
process P1() {
while(true) {
P(full1);
P(m1o);
//取消息代码
out1=(out1+1) % 3;
V(m1o);
V(empty1);
//加工消息代码
P(empty2);
P(m2o);

```

```

//存消息代码
in2=(in2+1) % 2;
V(m2o);
V(full2);
}
}
process P2( ) {
while(true) {
P(full2);
P(m2o);
//取消息代码
out2=(out2+1) % 2;
V(m2o);
V(empty2);
//加工消息代码
P(empty3);
P(m3o);
//存消息代码
in3=(in3+1) % 2;
V(m3o);
V(full3);
}
}
process P3( ) {
while(true) {
P(full3);
P(m3o);
//取消息代码
out3=(out3+1) % 2;
V(m3o);
V(empty3);
//加工消息代码
P(empty0);
P(m0o);
//存消息代码
in0=(in0+1) % 3;
V(m0o);
V(full0);
}
}
coend

```

7、【PV、管程】有一个阅览室，读者进入时必须先在一张登记表上登记，此表为每个座位列出一个表目，包括座位号、姓名，读者离开时要注销登记信息；假如阅览室共有 100 个座位。试用：①信号量和 PV 操作；②管程，实现用户进程的同步算法。

（满分 16 分，每小题 8 分，即 PV 题 8 分，管程 8 分）

答：

PV:

```
struct {
    char name[10];int number;
}seats[100]; for(int i=0;i<100;i++){seats[i].number=i;seats[i].name=null;}
semaphore mutex=1,seatcount=100;
cobegin
process p0(char name[]){//思路，有座位才能填表，表格同时只能由一人填写。
    P(seatcount);
    P(mutex);
    For(int i=0;i<100;i++){
        If (seats[i].name==null)seats[i].name=name;break;
    }
    V(mutex);
    P(mutex);
    A[i].name=null;
    V(mutex);
    V(seatcount);
}
coend
```

管程:

```
type readbook=MONITOR{
semaphore R;Int R_count,seat_count;seat_count=0;
char name[100];
InterfaceModule IM;
DEFINE readbook(),readerleave();
USE enter(),leave(),wait(),signal();
void readercome(char readername[]){
    enter(IM);
    if(seat_count>=100)wait(R,R_count,IM);
    seat_count=seat_count+1;
    for(int i=0;i<100;i++){if(name[i]==null) name[i]=readername;break;}
    leave(IM);}
void readerleave(char readername[]){
    enter(IM);
    seatcount--;
    for(int i=0;i<100;i++){if(name[i]==readername)name[i]=null;break;}
    signal(R,R_count_IM);}
cobegin
process reader i(){
    readbook.readercome(readername);
    readbook.readerleave(readername);}
coend
```

8、【PV、管程】在一个盒子里，混装了数量相等的黑白围棋子。现在用自动分拣系统把黑子、白子分开，设分拣系统有二个进程 P1 和 P2，其中 P1 拣白子；P2 拣黑子。规定每个进程每次拣一子；当一个进程在拣时，不允许另一个进程去拣；当一个进程拣了一子时，必须让另一个进程去拣。试分别使用 PV 操作和管程方法写出两进程 P1 和 P2 能并发正确执行的程序。（**满分 16 分，每小题 8 分**）

答：

PV:

```
semaphore S1,S2;S1=0;S2=0;
cobegin//先从拣其中一个子开始。
process P1(){
    while(1){P(S1);白子;V(S2);}
}
process P2(){
    while(1){P(S2);黑子;V(S1);}
}
```

管程:

```
type pickup=MONITOR{
boolean flag=true;
semaphore black,white;
int black_count;white_count;
InterfaceModule IM;
DEFINE pickb,pickw;
USE enter,leave,wait,signal;

void pickb(){//flag 决定了谁正在使用
    enter(IM);
    if(flag){wait(black,black_count,IM);}
    flag=true;
    黑子;
    signal(white,white_count,IM);
    leave(IM);
}
void pickw(){//flag 决定了谁正在使用
    enter(IM);
    if(!flag){wait(white,white_count,IM);}
    flag=false;
    白子;
    signal(black,black_count,IM);
    leave(IM);
}}

cobegin
pickup.pickb();pickup.pickw();
coend
```

9、【PV、管程】一组生产者进程和一组消费者进程共享 9 个缓冲区，每个缓冲区可以存放一个整数。生产者进程每次一次性地向 3 个缓冲区中写入整数，消费者进程每次从缓冲区取出一个整数。请用：①信号量和 PV 操作；②管程，写出能够正确执行的程序。（满分 16 分，每小题 8 分）

答：

PV:

```
int buffer[9];for(item in buffer){item=0;}
int putpos,fetchpos=0;
semaphore Sproducer,Sconsumer=0;
int producetimes,consumetimes=0;
cobegin
process producer(){
    while(true){
        int num1,num2,num3;
        P(Sproducer);
        if(producetimes*3-consumetimes<6){
            if(buffer[putpos]!=nullptr)
                buffer[putpos]=num1;buffer[(putpos+1)%9]=num2;buffer[(putpos+2)%9]=num3;
            putpos=(putpos+3)%9;
        }
        V(Sproducer);
    }
}
process consumer(){
    while(true){
        P(Sconsumer);
        if(consumetimes-producetimes*3<0){
            if(buffer[fetchpos]!=nullptr)y=buf[fetchptr];buf[fetchptr]=nullptr;fetchptr=(fetchptr+1)%9;
        }
        V(Sconsumer);
    }
}
coend
```

管程:

```
type get_put=MONITOR{
    int buf[9];int count,getptr,putptr=0;
    semaphore SP,SG;int SP_count,SG_count;
    InterfaceModule IM;
    DEFINE put,get;
    USE wait,signal,enter,leave;
procedure put(int a1,int a2,int a3){
    enter(IM);
    if(count>6)wait(SP,SP_count,IM);
    count=count+3;
    buf[putptr]=a1;buf[(putptr+1)%9]=a2;buf[(putptr+2)%9]=a3;putptr=(putptr+3)%9;
    signal(SG,SG_count,IM); signal(SG,SG_count,IM); signal(SG,SG_count,IM);
    leave(IM);
}
procedure get(int* b){
```



```

    enter(IM);
    if(count==0)wait(SG,SG_count,IM);
    b=buf[getptr];getptr=(getptr+1);count--;
    if(count<=6)signal(SP,SP_count,IM);
    else if(count>0)signal(SG,SG_count,IM);
    leave(IM);
}
}
process producer(){
    while(true){int a1,int a2,int a3;}get_put.put(a1,a2,a3);
}
process consumer(int* a){
    while(true){get_put.get(a);}
}
cobegin
producer();
consumer(int* a);
coend

```

10、【银行家算法】系统有 A、B、C、D 共 4 种资源，在某时刻进程 P₀、P₁、P₂、P₃ 和 P₄ 对资源的占有和需求情况如表，试解答下列问题：（满分 8 分，4+4）

Process	Allocation				Claim				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P ₀	0	0	3	2	0	0	4	4	1	6	2	2
P ₁	1	0	0	0	2	7	5	0				
P ₂	1	3	5	4	3	6	10	10				
P ₃	0	3	3	2	0	9	8	4				
P ₄	0	0	1	4	0	6	6	10				

(1)系统此时处于安全状态吗？试给出一个可能的安全序列。（4 分）

(2)若此时进程 P₂ 发出 request₁(1, 2, 2, 2)，系统能分配资源给它吗？为什么？（4 分）

答：

(1)系统处在安全状态，存在安全序列有：

P₀ P₃ P₄ P₁ P₂

P₀ P₃ P₁ P₄ P₂

P₀ P₃ P₁ P₂ P₄

(2)不能，这样做会使系统处于不安全状态。