

# 2.

## 1-1 答案

```
semaphore mutex=1;  
cobegin  
    process traveler_i() {  
        P(mutex);  
        过桥;  
        V(mutex);  
    }  
coend
```

## 2

## 1-2 答案

```
semaphore mutex= 1; //东西方向互斥使用独木桥
int numE2W=0, numW2E=0; //东向西和西向东的桥上人数
semaphore mutexE2W=1, mutexW2E=1; //互斥改变numE2W和
numW2E
cobegin
```

```
→ process travelerE2W_i() {
    P(mutexE2W);
    numE2W++;
    if (numE2W==1)
        P(mutex);
    V(mutexE2W);
    过桥;
    P(mutexE2W);
    numE2W--;
    if (numE2W==0)
        V(mutex);
    V(mutexE2W);
}
coend
```

```
process travelerW2E_j() {
    P(mutexW2E);
    numW2E++;
    if (numW2E==1)
        P(mutex);
    V(mutexW2E);
    过桥;
    P(mutexW2E);
    numW2E--;
    if (numW2E==0)
        V(mutex);
    V(mutexW2E);
}
```

```
semaphore mutex= 1; //东西方向互斥使用独木桥
int numE2W=0; //东向西的桥上人数
semaphore mutexE2W=1; //互斥改变numE2W
cobegin
```

```
process travelerE2W_i() {
    P(mutexE2W);
    numE2W++;
    if (numE2W==1)
        P(mutex);
    V(mutexE2W);
    过桥;
    P(mutexE2W);
    numE2W--;
    if (numE2W==0)
        V(mutex);
    V(mutexE2W);
}
coend
```

```
process travelerW2E_j() {
    P(mutex);
    过桥;
    V(mutex);
}
```

# 2

## 第二题

```
P1:
while (True){
    x=produce();
    P(empty);
    P(mutex);
    //x=produce();
    put(x);
    V(mutex);
    if x%2==0
        V(even);
    else
        V(odd);
}
```

```
P2:
while (True){
    P(odd);
    P(mutex);
    getodd();
    V(mutex);
    countodd();
    V(empty);
}
```

```
P3:
while (True){
    P(even);
    P(mutex);
    geteven();
    V(mutex);
    counteven();
    V(empty);
}
```



# 2

## 第二题

- ◆ 互斥信号量mutex：实现进程对缓冲区的互斥访问，初值为1
- ◆ 信号量empty：空缓冲区的数量，初值为N
- ◆ 信号量even：数字为偶数的缓冲区数量，初值为0
- ◆ 信号量odd：数字为奇数的缓冲区数量，初值为0

## 2

## 第三题

- ◆ 最短剩余时间优先调度：0时刻P1执行，由于最短剩余时间优先调度是抢占式调度算法，故P1执行到2时P2到达，此时P2剩余时间短于P1，故P2先执行。P2执行期间P3到达，但P3剩余时间多于P2，故P2执行到6结束。此时P4也到达，P1剩余8，P3剩余10，P4剩余5，故先执行P4结束后再执行P1，最后执行P3。因此进程的运行次序为P1, P2, P4, P1, P3。各进程相关时间如下表：

进程	运行时间	到达时间	结束时间	周转时间	带权周转时间
P1	10	0	19	19	1.9
P2	4	2	6	4	1
P3	10	4	29	25	2.5
P4	5	6	11	5	1

- ◆ 平均周转时间为： $(19+4+25+5)/4=13.25$
- ◆ 带权平均周转时间为： $(1.9+1+2.5+1)/4=1.6$

## 2

## 第四题

- ◆ 每个作业运行将经过两个阶段：作业调度（SJF算法）和进程调度（优先数抢占式）。此外，批处理最多容纳2道作业，其余作业都在后备队列等待。
- ◆ A作业10:00到达，此时只有1个作业直接进入内存执行，到10:20时，B作业到达，因为B的优先数高于A，因此A进入就绪队列，B执行至10:50结束。
- ◆ 10:50时，作业D到达。此时B执行结束，根据SJF算法，调度D进入内存，然后再根据优先数调度算法，调度A执行，A执行至11:10结束。
- ◆ 此时调度C进入内存，根据优先数调度算法，先调度执行C，C执行至12:00结束，最后再执行D，D执行至12:20结束。



# 2

## 第四题

- ◆ 综上所述，所有作业的到达时间、进入内存时间、运行结束时间以及周转时间如下表所示：

作业名	到达时间	进入内存时间	结束时间	周转时间
A	10:00	10:00	11:10	70
B	10:20	10:20	10:50	30
C	10:30	11:10	12:00	90
D	10:50	10:50	12:20	90

- ◆ 平均周转时间为  $(70+30+90+90)/4=70$  分钟



## 2

## 第五题

- ◆ (1) T0时刻的五个进程资源缺口情况及已分配资源情况如下表:

进程	已分配资源量			资源缺口			系统剩余资源数量		
	A	B	C	A	B	C	A	B	C
P1	2	1	2	3	4	7	2	3	3
P2	4	0	2	1	3	4			
P3	4	0	5	0	0	6			
P4	2	0	4	2	2	1			
P5	3	1	4	1	1	0			

- ◆ 剩余资源满足P4和P5，先将P4的已分配资源释放，得到剩余资源量为(4, 3, 7)，然后再将P5的已分配资源释放，得到剩余资源量为(7, 4, 11)，满足剩余三个进程P1、P2和P3的资源缺口。
- ◆ 故T0时刻是安全状态，其中一个安全序列为P4, P5, P1, P2, P3 (**安全序列不唯一**)

# 2

## 第五题

- ◆ (2) 进程P4申请资源 (2, 0, 1)，判断申请有效性（分别和系统剩余资源数量、P4资源缺口进行比较）后进行尝试分配，分配后各进程的已分配资源及缺口情况见下表：

进程	已分配资源量			资源缺口			系统剩余资源数量		
	A	B	C	A	B	C	A	B	C
P1	2	1	2	3	4	7	0	3	2
P2	4	0	2	1	3	4			
P3	4	0	5	0	0	6			
P4	4	0	5	0	2	0			
P5	3	1	4	1	1	0			

- ◆ 剩余资源满足P4，先将P4的已分配资源释放，得到剩余资源量为 (4, 3, 7)，剩余资源满足P2、P3、P5的资源缺口，依次释放后满足P1的资源缺口。
- ◆ 故P4的资源申请是安全的，尝试分配后存在安全序列P4, P2, P3, P5, P1 (安全序列不唯一)



◇ (3) 进程P1申请资源 (0, 2, 0)，判断申请有效性（分别和系统剩余资源数量、P1资源缺口进行比较）后进行尝试分配，分配后各进程的已分配资源及缺口情况见下表：

进程	已分配资源量			资源缺口			系统剩余资源数量		
	A	B	C	A	B	C	A	B	C
P1	2	3	2	3	2	7	0	1	2
P2	4	0	2	1	3	4			
P3	4	0	5	0	0	6			
P4	4	0	5	0	2	0			
P5	3	1	4	1	1	0			

◇ 发现剩余资源数量不能满足任何一个进程的需求缺口，无法找到一个安全序列，将导致系统进入不安全状态，因此不能按照进程P1的申请进行资源分配。

# 2

## 第1题

- ◆ 一个 32 位地址的计算机系统使用二级页表，虚拟地址为：顶级页表占9位，二级页表占11位。请问：页面长度为多少？虚拟地址空间有多少个页面？
- ◆ 虚拟地址共32位，顶级页表占9位，二级页表占11位，那么页内偏移占 $32-9-11=12$ 位，因此页面长度是 $2^{12}\text{B}$ ，即4KB。虚拟地址空间的页面数量为 $2^{20}=1\text{M}$ 个。



# 2

## 第2题

- ◆ 用户编程空间共32个页面，则在虚拟地址中页号占5位。每页为1KB，则在虚拟地址中10位。
- ◆ 逻辑地址0A5C (H) 转化为二进制是0000 1010 0101 1100，其中页内偏移量是10 0101 1100，转化为十进制是604，页号是10，转化为十进制是2，对应物理块号是4。
- ◆ 最后的物理地址为 $4 \times 1024 + 604 = 4700$ （或者是125CH）

# 2

## 第3题

◆ FIFO:

页面	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	4			4	4	3	3	3	6	6	6	2	2	2	3	3
		2	2	2			5	5	5	1	1	1	7	7	7	4	4	4	2
			3	3			3	6	6	6	4	4	4	5	5	5	1	1	1
缺页				是			是	是	是	是	是	是	是	是	是	是	是	是	是

◆ 缺页中断次数为14次，缺页率为14/19

## 2

## 第3题

◇ OPT:

页面	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	4			4	4		4			4	4	4			3	
		2	2	2			5	6		6			7	5	2			2	
			3	3			3	3		1			1	1	1			1	
缺页				是			是	是		是			是	是	是			是	

◇ 缺页中断次数为8次，缺页率为8/19

## 2

## 第3题

◆ LRU:

页面	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	4			5	5		1	1	1	7	7	7	4	4	4	2
		2	2	2			2	6		6	4	4	4	5	5	5	1	1	1
			3	3			3	3		3	3	6	6	6	2	2	2	3	3
缺页				是			是	是		是	是	是	是	是	是	是	是	是	是

◆ 缺页中断次数为13次，缺页率为13/19



## 2

## 第4题

(1) 页面大小为4KB，则在虚拟地址中页内偏移量占低12位

逻辑地址	页号	页内偏移量
2362H	2	362H
1565H	1	565H
25A5H	2	5A5H

- ◆ 访问2362H: 10(访问块表) + 100(访问页表) + 100(访问物理地址) = 210ns
- ◆ 访问1565H: 10(访问块表) + 100(访问页表) + 108(处理缺页) + 100(访问物理地址) = 318ns
- ◆ 访问25A5H: 10(访问块表) + 100(访问物理地址) = 110ns
- ◆ (2) 访问虚拟地址1565H，产生缺页中断，驻留集为2，必须淘汰1个页面。根据局部淘汰策略和LRU算法，0号页被淘汰，1号页被置换进入内存，则1号页对应的页框号为101H。故逻辑地址1565H的物理地址是101565H

## 2

## 第5题

- ◆ 分配给该作业的主存共300字，页的大小为100字，因此分配给该作业3个物理块。
- ◆ 用户作业访问的字地址序列是：115, 228, 120, 88, 446, 102, 321, 432, 260, 167
- ◆ 对应的页号是：1, 2, 1, 0, 4, 1, 3, 4, 2, 1

## 2

## 第5题

◆ FIFO:

页面	1	2	1	0	4	1	3	4	2	1
	0	0			4		4			4
	1	1			1		3			3
		2			2		2			2
淘汰					0		1			2
缺页					是		是			是

◆ 缺页中断次数为3次，缺页中断率为3/10

## 2

## 第5题

◆ LRU:

页面	1	2	1	0	4	1	3	4	2	1
	0	0			0		3		3	1
	1	1			1		1		2	2
		2			4		4		4	4
淘汰					2		0		1	3
缺页					是		是		是	是

◆ 缺页中断次数为4次，缺页中断率为4/10



- ◆ (1) 页的大小为1KB，页内偏移量占低10位。逻辑地址为17CAH，转化为二进制位0001 0111 1100 1010，故页号为5（0101）。
- ◆ (2) 操作系统使用固定分配局部置换策略，页面5不在内存，故需要进行页面置换。采用FIFO置换算法，故将页面0淘汰，将其页框7分配给页面5。因此其对应的物理地址为0001 1111 1100 1010，转化为16进制为1FCAH。
- ◆ (3) 采用Clock置换算法，且当前指针指向2号页框，从2号页框开始，依次检测页框的访问位，第一次循环过后，发现所有页框访问位都为1，故将所有页框的访问位都置0。在第二次循环时，将2号页框对应的页置换，因此对应的物理地址为0000 1011 1100 1010，转化为16进制为0BCAH。