国家一级假勤奋大学生整理，希望大家都考高分！

某计算机系统中有5台打印机，有K个进程竞争使用，每个进程最多需要2台打印机。该系统可能会发生死锁的K的最小值是__。

A.2

B.3

C.4

D.5

(发生死锁的条件:进程中所需最大资源数大于资源数和进程数之和。

即2k>=5+k　　k>=5)

系统有某类资源5个，供3个进程共享，为保证系统的安全，应限定每个进程申请的资源数不超过__

A.1

B.2

C.3

D.4

(3K<5+3)

15．某文件共占用 8 个磁盘块 B0~B7，磁盘每道有 8 个扇面，每个扇面可存放一个磁盘块，磁盘旋转一圈的时间是 20ms，程序处理一个磁盘块的时间是 4ms，B0～B7 在一个磁道上优化分布，磁头目前在 B0 起点处。则把 B0～B7 全部读出的时间是（　　D　）

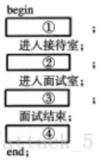
A．62.5ms B．60ms

C．57.5ms D．55ms

53．某单位要对参加招聘的人员进行面试。待聘人员先要从1号门进入接待室。接待室最多可同时容纳20人，当不满20人时，允许在门外的等待者进入。每次允许在接待室等待的一个人通过2号门进入面试室，一个面试者结束面试后再让下一个进入。回答下列问题：

(1)把每个面试者看作一个进程。若用PV操作进行管理，应怎样定义信号量及其初值?

(2)在下面进程的程序中的方框位置填写正确的P、V操作，以保证并发进程的正确执行。



答：(1)S1=20,S2=2

(2) ①____P(S1)_____； ②_____P(S2)_____；

③____V(S2)_____； ④__V(S1)_____。

此部分为考研题

1、 某页式存储管理系统，页内地址为11位， 逻辑地址 总长度为16位，物理地址长度为32位。

1）每页有多少字节?

2）逻辑地址空间有多少页?

3）主存空间又有多少块?

4）假设某作业的大小为5000字节，依次分配的页面为3、7、9。请画出页表，并简单叙述逻辑地址4500的地址转换过程。

4、某银行有人民储蓄业务，由n个柜员负责，有1台取号机。每个顾客进入银行后先取一个号，若有人取号则需等他人取完后才能取，取到号后等待叫号，当一个柜员人员空闲下来，就叫下一个号。试用P，V操作正确编写柜台人员和顾客进程的程序。

5、某系统有同类资源m个，供n个进程使用；如果每个进程对资源的最大需求量为K，问：

（1）为使系统不发生死锁，K的最大值为多少?

（2）按（1）的结果，当n=3,m分别取值2，3，4时，对应的K值是多少，就可以使系统不会发生死锁?

45.（7分）三个进程P1、P2、P3互斥使用一个包含N（N>0）个单元的缓冲区。P1每次用produce（）生成一个正整数并用put（）送入缓冲区某一空单元中；P2每次用getodd（）从该缓冲区中取出一个奇数并用countodd（）统计奇数个数；P3每次用geteven（）从该缓冲区中取出一个偶数并用counteven（）统计偶数个数。请用信号量机制实现这三个进程的同步与互斥活动，并说明所定义的信号量的含义。要求用伪代码描述。

****

1. 若在一基本分页存储管理系统中，某作业的页表如下所示：

页号	块号
0	2
1	3
2	1
3	6

已知页面大小为 1024 字节，试将逻辑地址 1011、2148、4000、5012 转化为相应的物理地址。

解：物理地址由页号 P 和页内地址 W 两部分组成，P 等于逻辑地址除以页面大小的除数，W 等于逻辑地址除以页面大小的余数，物理块号和页面大小相同。

则逻辑地址为 1011 的物理地址算法如下：P=[1011/1024]=0,W=1011,据页表可知页号为 0 的页对应的是物理块号为 2 的块，所以物理地址=2*1024+1011=3059；

同理，逻辑地址为 2148 的物理地址：P=[2148/1024]=2,W=100。页号为 2 对应物理块号 1，物理地址=1*1024+100=1124；

逻辑地址为 4000 的物理地址：P=[4000/1024]=3,W=904。页号为 3 对应物理块号 6，物理地址=6*1024+904=7048；

逻辑地址为 5012 的物理地址：P=[5012/1024]=4。页号为 4 的页面在页表中没有，所以要产生页面中断，请求将外存中的页面调入内存。

22、在银行家算法中，若出现下列资源分配情况：

Process	allocation	need	available
P0	0032	0012	1622
P1	1000	1750	
P2	1354	2356	
P3	0332	0652	
P4	0014	0656	

试问：

1)该状态是否安全？

2）若进程 P2 提出请求 request（1，2，2，2）后，系统是否将资源分配给它？

(1) 该状态是安全的，这时可以找到一个安全序列：P0、P3、P4、P1、P2

设置两个向量①工作向量 work，它表示系统可提供给进程继续运行所需的各类资源数目，在执行算法开始时，work：= Available. ②finish，它表示系统是否有足够的资源分配给进程，使其运行完成。

所以对上述分配资源情况进行分析如下：

Process	Allocation	Need	work	work+Allocation	finish
P0	0032	0012	1622	1654	true
P3	0332	0652	1654	1986	true
P4	0014	0656	1986	199 10	true
P1	1000	1750	199 10	299 10	true
P2	1354	2356	299 10	3 12 14 14	true

(2) 若进程 P2 提出上述请求，系统不能将资源分配给它，因为分配之后系统将进入不安全状态。

P2 请求资源：P2 发出请求向量 Request2（1，2，2，2），系统按银行家算法进行检查：

①Request2（1，2，2，2）≤Need2（2，3，5，6）；

②Request2（1，2，2，2）≤Available（1，6，2，2）；

③系统暂时先假定可为 P2 分配资源，并修改 P2 的有关数据，如下表：

Allocation	Need	Available
2 5 7 6	1 1 3 4	0 4 0 0

可用资源 Available（0，4，0，0）已不能满足任何进程的需要。

设有一个可以装入A、B两种物品的仓库，其容量无限大，但是要求仓库A、B两种物品的数量满足下述不等式：

-m≤A物品数量-B物品数量≤ n。

其中，m和n为正整数，试用信号量和PV操作描述A、B两种物品的入库过程。

虑同步一般在极限状态，极限状态就是没有A或没有B的两种情况。当没有A时，B最多能放M个，否则必须等待；同理没有B时，A最多能放N个，否则必须等待。若既有A也有B时二者只要满足不等式的约束就可以。
所以在放A时必须判断是否超过N，执行P操作；在放B时必须判断是否超过M，执行P操作。

当A为N+1时，A等待，此时由放入第一个B来唤醒；同样当B为M+1时由放入第一个A来唤醒，因此在A和B放入后需要执行一个V操作。

```
Semaphore a=n;//用于判断A是否比B多N个
Semaphore b=m;//用于判断B是否比A多M个
Void main()
{   createprocess(A,…);
    createprocess(B, …)
}
```

```
A物品入库：
Void A(){
    while(1){
        P(a);
        A物品入库；
        V(b);
    }
}
```

```
B物品入库：
Void B(){
    while(1){
        P(b);
        B物品入库；
        V(a);
    }
}
```


3.某图书阅览室有50个座位，进入阅览室的读者需要在登记簿上登记，离开阅览室时注销登记。试用管程实现对阅览室的管理。

```
TYPE reading_room=MONITOR;
VAR seat: integer;
    q: condition;
define enter, leave;
PROCEDURE enter;
BEGIN
   IF seat=0 THEN
      wait(q);
   登记进入时间;
   seat:=seat-1;
END;
```

```
PROCEDURE leave;
BEGIN
   登记离开时间;
   seat:=seat+1;
   signal(q);
END;
BEGIN seat:=50 END;
```

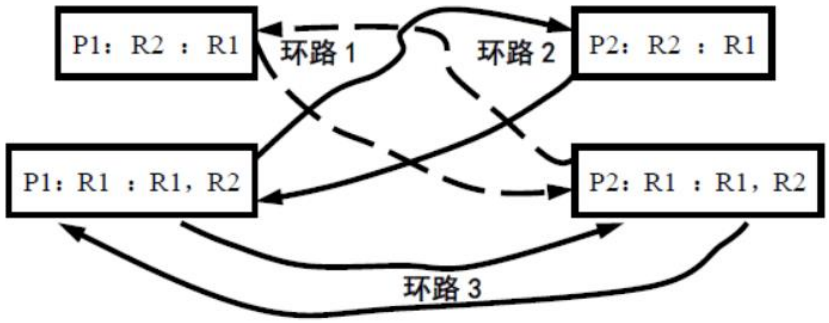
3 设系统有R1和R2两类独占型供进程P1和P2所共享，R1有2台，R2有1台。已知P1和P2均以如下的顺序申请和释放资源：

申请R1；申请R2；申请R1；释放R1；释放R2；释放R1。

问 题： 根据进程P1和P2对资源的请求序列，

画图分析两进程并发运行时可能到达的所有死锁点

解：(1) （4 分）根据 2 个进程有关资源的活动，画出进程资源申请、占用状态图如下：



(2) （6 分）上图中，由于资源 R2 只有 1 个，所以环路 3 两个进程的状态不可同时到达。

图中环路 1 和环路 2 就是 P1 和 P2 并发运行时可能到达的死锁点。

具体各环路表示的两进程可能到达的发生死锁点如下（竖线表示死锁位置）：

环路 1
- P1：申请 R1；| 申请 R2；申请 R1；释放 R1；释放 R2；释放 R1。
- P2：申请 R1；申请 R2；| 申请 R1；释放 R1；释放 R2；释放 R1

环路 2
- P1：申请 R1；申请 R2；| 申请 R1；释放 R1；释放 R2；释放 R1。
- P2：申请 R1；| 申请 R2；申请 R1；释放 R1；释放 R2；释放 R1

1. 某系统T0时刻的资源分配情况如下所示：

1. 某系统 T0 时刻的资源分配情况如下所示：

资源 / 进程	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
A	3	1	1	1	0	0	1	2	0
B	0	0	0	0	1	2			
C	1	1	0	3	0	0			
D	1	0	1	0	1	0			
E	0	0	0	2	1	0			

试问：

（1）该状态是否安全?

存在一个安全序列{A.C,D,E,B}，故系统是安全的。

进程 \ 资源	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
A	1	2	0	1	0	0	3	1	1	4	3	1	T
C	4	3	1	3	0	0	1	1	0	5	4	1	T
D	5	4	1	0	1	0	1	0	1	6	4	2	T
E	6	4	2	2	1	0	0	0	0	6	4	2	T
B	6	4	2	0	1	2	0	0	0	6	4	2	T

(2) 若进程B提出请求RequestB（0，1，0），系统能否将资源分配给它?

RequestB（0,1,0）<= NeedB(0, 1, 2)

RequestB（0,1,0）<= Available(1,2,0)

系统暂时先假定可以为B分配资源，并修改有关数据

进程 \ 资源	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
A	3	1	1	1	0	0	1	1	0
B	0	1	0	0	0	2			
C	1	1	0	3	0	0			
D	1	0	1	0	1	0			
E	0	0	0	2	1	0			

存在一个安全序列{A.C,D,E,B}，故系统是安全的，将B所申请的资源分配给它。

进程 \ 资源	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
A	1	1	0	1	0	0	3	1	1	4	2	1	T
C	4	2	1	3	0	0	1	1	0	5	3	1	T
D	5	3	1	0	1	0	1	0	1	6	3	2	T
E	6	3	2	2	1	0	0	0	0	6	3	2	T
B	6	3	2	0	0	2	0	1	0	6	4	2	T

- 1-8操作系统的资源共享有什么方式？**互斥访问共享;同时访问共享**

1. 用户在程序设计过程中，可通过 **系统调用** 获得操作系统的服务
2. 在设计分时操作系统时,首先要考虑的是 **交互性和响应时间**
3. 在设计批处理操作系统时,首先要考虑的是 **周转时间和系统吞吐量**
4. 在设计实时操作系统时,首先要考虑的是 **实时性和可靠性**

4. 临界区是指进程中访问临界资源的那段代码（√）

12. 信号量的初值不能为负数（√）

13. 线程是CPU调度的基本单位，但不是资源分配的基本单位（√）

14. 在进程对应的代码中使用wait、signal操作后，可以防止系统发生死锁（×）

15. 管程每次只允许一个进程进入（√）

16. wait、signal操作可以解决一切互斥问题（√）

4. 下列算法中，（**FCFS调度算法**)只能采用非抢占调度方式，（**时间片轮转法**)的只能采用抢占调度方式，而其余的 算法既可采用抢占方式，也可采用非抢占方式

5. 我们如果为每一个作业只建立一个进程，则为了照顾短作业用户，应采用(**短作业优先**);为 照顾紧急作业的用户，应采用(**高响应比优先**);为能实现人机交互作用应采用(**时间片轮转法**);为了兼顾短作业和 长时间等待的作业，应采用(**高响应比优先**);为了使短作业、长作业及交互作业用户都比较满意，应采 用(**多级反馈队列调度算法**);为了使作业的平均周转时间最短，应采用(**短作业优先**)算法

1. 在请求调页系统中,地址变换过程可能会因为**逻辑地址越界**、**缺页**和**访问权限错误**等原因而产生中断.

4. 已知某分页系统中，页的大小为1K，则逻辑地址A=2170，则其
   （1）页号是（ 2 ）
   （2）页内偏移量是（ 122 ）
   （3）若查页表得知A对应的物理块号为5，则A的物理地址是（ 5242 ）

解:(1)页号 = 逻辑地址/页的大小 = 2170/1024B = 2

(2)页内偏移量 = 逻辑地址%页的大小 = 2170%1024B = 122

(3)A的物理地址 = 物理块号*页的大小*+*页内偏移量* = 51024+122 = 5242

三. 司机售票员问题

1. 问题描述

公共汽车上。司机和售票员的活动分别如下：
司机的活动：启动车辆，正常行车，到站停车.
售票员的活动： 关车门，售票，开车门。

2. 思路

分析可知。售票员和司机的4个活动存在前后关系。即关车门-->启动汽车。到站停车-->开车门。所以用两个信号量s1,s2分别来控制两组前后关系.

3. 使用记录型信号量解决问题

伪代码

```
semaphore s1 = 1,s2 = 1;
//司机
void driver(){
    signal（s1）
    启动车辆
    正常行驶
    到站停车
    wait(s2)
}

void conductor(){
    关车门
    wait(s1)
    售票
    signal（s2）
    开门
}
```

复制

死锁检查算法

8.某系统采用死锁检测手段发现死锁，设系统中资源类集合为{A，B，C}，资源类A中共有8个实例，资源类B中共有6个实例，资源类C中共有5个实例．又设系统中进程集合为{p1,p2,p3,p4,p5,p6}，某时刻系统状态如下：

	Allocation	Need	Available
	A B C	A B C	A B C
p1:	1 0 0	0 0 0	2 2 1
p2:	3 2 1	0 0 0	
p3:	0 1 2	2 0 2	
p4:	0 0 0	0 0 0	
p5:	2 1 0	0 3 1	
p6:	0 0 1	0 0 0	

1. 在上述状态下系统依次接受如下请求：Request[1]=(1,0,0)；Request[2]=(2,1,0)；Request[4]=(0,0,2)。给出系统状态变化情况，并说明没有死锁。
2. 在由(1)所确定的状态下系统接收如下请求：Request[1]=(0,3,1)，说明此时已发生死锁，并找出参与死锁的进程。

（1）①系统接受请求：Request[1]=(1,0,0)，Request[1]< Available,可以执行分配，那么系统状态变为：

	Allocation	Need	Available
	A B C	A B C	A B C
p1:	2 0 0	0 0 0	1 2 1
p2:	3 2 1	0 0 0	
p3:	0 1 2	2 0 2	
p4:	0 0 0	0 0 0	
p5:	2 1 0	0 3 1	
p6:	0 0 1	0 0 0	

在该状态下运行死锁检测算法，可以找到一个进程序列<*p*4,*p*1,*p*2,*p*3,*p*5,*p*6>，它使Finish[i]=true，对于所有1≤i≤6，因而可以断言系统当前没有进入死锁状态。

②系统接受请求：Request[2]=(2,1,0)，Request[2]>Available,系统只接受请求，无法实现资源分配，那么系统状态变为：

	Allocation	Need	Available
	A B C	A B C	A B C
p1:	2 0 0	0 0 0	1 2 1
p2:	3 2 1	2 1 0	
p3:	0 1 2	2 0 2	
p4:	0 0 0	0 0 0	
p5:	2 1 0	0 3 1	
p6:	0 0 1	0 0 0	

在该状态下运行死锁检测算法，可以找到一个进程序列<*p*4,*p*1,*p*2,*p*3,*p*5,*p*6>，它使Finish[i]=true，对于所有1≤i≤6，因而可以断言系统当前没有进入死锁状态。

③系统接受请求：Request[4]=(0,0,2)，Request[4]>Available,系统只接受请求，无法实现资源分配，那么系统状态变为：

	Allocation	Need	Available
	A B C	A B C	A B C
p1:	2 0 0	0 0 0	1 2 1
p2:	3 2 1	2 1 0	
p3:	0 1 2	2 0 2	
p4:	0 0 0	0 0 2	
p5:	2 1 0	0 3 1	
p6:	0 0 1	0 0 0	

在该状态下运行死锁检测算法，可以找到一个进程序列< *p*1,*p*2,*p*3, *p*4,*p*5,*p*6>，它使Finish[i]=true，对于所有1≤i≤6，因而可以断言系统当前没有进入死锁状态。

(2)在（1）状态下系统接收如下请求：Request[1]=(0,3,1)，Request[1]>Available,系统只接受请求，无法实现资源分配，则系统状态变为：

	Allocation	Need	Available
	A B C	A B C	A B C
p1:	2 0 0	0 3 1	1 2 1
p2:	3 2 1	2 1 0	
p3:	0 1 2	2 0 2	
p4:	0 0 0	0 0 2	
p5:	2 1 0	0 3 1	
p6:	0 0 1	0 0 0	

在该状态下运行死锁检测算法，找不到一个进程序列使Finish[i]=true，对于所有1≤i≤6，因为存在i∈{1，2，3，5}，使Finish[i]=false，因而可以断言系统已经进入死锁状态，进程*p*1,*p*2,*p*3,*p*5卷入死锁.

10.请求分页管理系统中，假设某进程的页表内容如下表所示。

页面大小为4KB，一次内存的访问时间是100ns，一次快表（TLB）的访问时间是10ns，处理一次缺页的平均时间为108ns（已含更新TLB和页表的时间），进程的驻留集大小固定为2，采用最近最少使用置换算法（LRU）和局部淘汰策略。假设：

①TLB初始为空；

②地址转换时先访问TLB，若TLB未命中，再访问页表（忽略访问页表之后的TLB更新时间）；

③有效位为0表示页面不在内存，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行。设有虚地址访问序列137H、565H、15A5H，请问：

页号	页框号	有效位（存在位）
0	101H	1
1	--	0
2	254H	1

（1） 依次访问上述三个虚地址，各需多少时间？给出计算过程。

（2） 基于上述访问序列，虚地址258H的物理地址是多少？请说明理由。

（1）根据页式管理的工作原理，应先考虑页面大小，以便将页号和页内位移分解出来。页面大小为4KB，即212，则得到页内位移占虚地址的低12位，页号占剩余高位。可得三个虚地址的页号P如下（十六进制的一位数字转换成4位二进制，因此，十六进制的低三位正好为页内位移，最高位为页号）：

137H：P=0，访问快表10ns，因初始为空，访问页表100ns得到页框号，合成物理地址后访问主存100ns，共计10ns+100ns+100ns=210ns。

565H：P=0，访问快表，因第一次访问已将该页号放入快表，因此花费10ns便可合成物理地址，访问主存100ns，共计10ns+100ns=110ns。

15A5H：P=1，访问快表10ns，落空，访问页表100ns落空，进行缺页中断处理108ns，合成物理地址后访问主存100ns，共计10ns+100ns+108ns+100ns≈108ns。

（2）访问1565H时,1号页面在内存中,页框号与15A5H相同。前面当访问虚地址15A5H时，访问页表发现1号页不在内存中，产生缺页中断，合法驻留集为2，必须从页表中淘汰一个页面，根据题目的置换算法，最近访问的都是0号页的内容，0号页不应被淘汰。应淘汰2号页面，因此1号页面装入时对应的页框号为254H。由此可得1565H的物理地址为254565H。

页式管理

12.在某页式管理系统，某进程页表如下，已知页面大小为1024B，试将逻辑地址1012、2248、3010、4020、5018转化为相应的物理地址。

页号	页框号
0	5
2	8
2	8
3	1
4	6

解：

逻辑地址	页号	页内地址	页框号	物理地址
1012	0	1012	5	5*1024+1012=6132
2248	2	200	8	8*1024+200=8392
3010	2	962	8	8*1024+962=9154
4020	3	948	1	1*1024+948=1972
5018	4	922	6	6*1024+922=7066

段式存储管理

11.在某个段式存储管理系统中，进程P的段表如下表，求表中各逻辑地址对应的物理地址

段号	段内位移
0	430
1	15
2	500
3	400
4	112

解：

段号	段内位移	物理地址
0	430	680
1	15	2365
2	500	越界
3	400	1750
4	112	越界

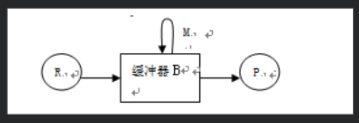
2. 设有3个并发执行的进程：输入进程Pi、计算进程Pc和输出进程Po。其中进程Pi不断地从键盘读入整数，放入缓冲区Buf1，Pc按输入顺序从Buf1中取数据，每次取出2个整数，计算其和，将结果放入缓冲区Buf2。Po负责将Buf2中的数据按顺序输出。设缓冲区Buf1、Buf2可存放的整数个数分别为m、n（m、n>0）。要求利用信号量的P、V操作写出进程Pi、Pc、Po的算法。（10.0分）

设信号量 e1,f1,e2,f2，其初值分别为：e1=m,f1=0,e2=n,f2=0

3 个并发进程分别为：

Cobegin
Pi:
begin
L1:P(e1);
向 Buf1 中输入 1 个数;
V(f1);
Goto L1;
end;
	Pc:
	begin
	L2:P(f1);
	x=从 Buf1 中读 1 个数;
	V(e1);
	P(f1);
	y=从 Buf1 中读 1 个数;
	V(e1);
	z=x+y;
	P(e2);
	将 z 送入 Buf2 中;
	V(f2);
	Goto L2;
	end;
		Po:
		begin
		L3:P(f2);
		W=从 Buf2 中读 1 个数;
		打印 W;
		V(e2);
		Goto L3;
		End;
		Coend;

3. 今有3个并发进程R、M、P，它们共享一个缓冲器B。进程R负责向B中输入数据；进程R每输入一数据，进程M对其进行加工；进程M加工完成后，进程P负责打印输出。缓冲器B中每次只能存放一个数据，数据一旦被打印，进程R又可存放下一个数据，……。它们之间的关系如图所示。用PV操作机制进行控制，写出三个进程的工作流程。



（10.0分）

正确答案：

设信号量 e,f1,f2:semaphore;

初值 e=1;f1=f2=0;

Cobegin
Process R;
Begin
L1:P(e);
生产一数据，放入缓冲区 B;
V(f1);
Goto L1;
End;
	Process M;
	Begin
	L2:P(f1);
	加工缓冲区 B 中数据;
	V(f2);
	Goto L2;
	End;
		Process P;
		Begin
		L3:P(f2);
		打印缓冲区 B 中数据;
		V(e);
		Goto L3;
		End;
		Coend;

4. 系统运行有三个进程：输入进程、计算进程和打印进程，它们协同完成工作。输入进程和计算进程之间共用缓冲区buffer1,计算进程和打印进程之间共用缓冲区buffer2。输入进程接收外部数据放入buffer1中；计算进程从buffer1中取出数据进行计算，然后将结果放入buffer2；打印进程从buffer2取出数据打印输出。

用算法描述这三个进程的工作情况，并用wait和signal原语实现其同步操作。

答题技巧:

①分清楚是同步问题(1个信号量)还是互斥问题(2个信号量)?即确定要定义几个信号量。

②分清楚消费者进程和生产者进程。

③消费者：满锁进，出锁空；生产者：空锁进，出锁满

答案与解析如下：

该问题是同步问题，同步需要两个信号量，互斥问题需要一个信号量。

输入进程—buf1→计算进程→buf2—打印进程

从键盘输入到打印机输出的数据传送过程，可以看作是由键盘输入进程到计算进程,以及由计算进程到打印输出进程这两个数据传送进程所组成。

其中，对键盘输入进程而言，计算进程是消费者进程;而对打印输出进程而言,计算进程又是生产者进程。据此可将它们之间的同步问题描述如下:

```
var:mutex1,mutex2,empty1,empty2,full1,full2:=1,1,1,1,0,0;
```

```
InP:begin
 repeat
  wait(empty1);//生产者测试是否有空闲
  wait(mutex1);//测试是否互斥
     input a data from keyboard;
     Add to buffer1;
  signal(mutex1);//离开并修改互斥标志
  signal(full1);//生产者生产了资源，修改资源个数
 until false
end
```

```
CalP:begin
 repeat
  wait(full1);//消费者测试是否有资源
  wait(mutex1);//测试是否互斥
    Take a data from buffer1;
    Add to ch1;
  signal(mutex1);//离开并修改互斥标志
  signal(empty1);//消费者使用资源，再次有空闲
    calculate ch1;
  wait(empty2);//生产者测试是否有空闲
  wait(mutex2);//测试是否互斥
    Take a data from ch1;
    Add to buffer2;
  signal(mutex2);//离开修改互斥标志
  signal(full2);//生产者生产了资源，修改资源个数
 until false
end
```

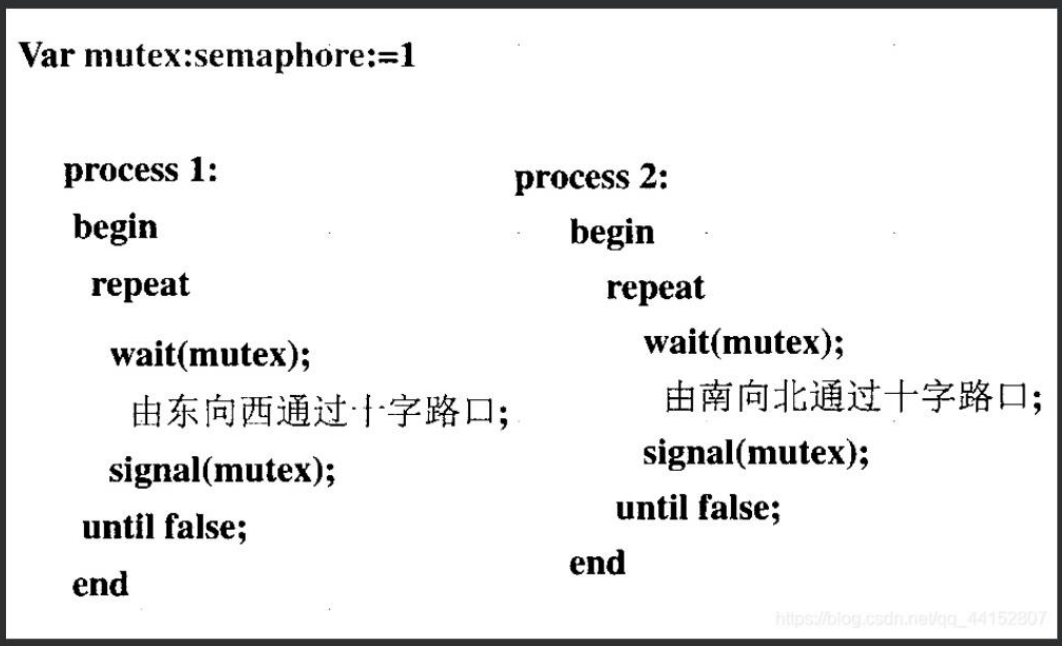
1. 某虚拟存储器的用户编程空间共32(KB)个页面，每页为1KB，内存为16KB。假定时刻t某用户程序的内存页表如下表所示，计算逻辑地址0A5C(H)的物理地址

页号	物理块号
0	1
1	7
2	11
3	8

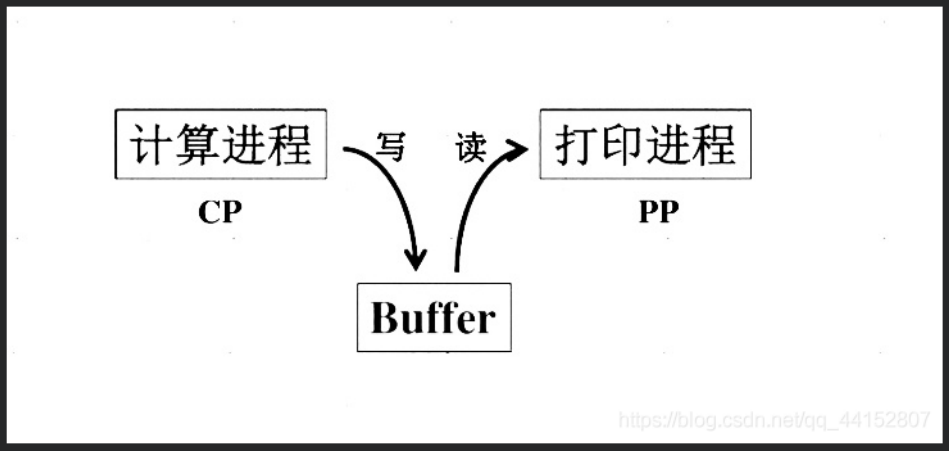
答题技巧:
①据编程空间总页数所占字节算出页号位数
②据每页所占字节数算出页内地址所占位数
③题干的逻辑地址转为二进制，据所占位数找出页内地址及页号部分(先数够页内地址位数，剩下的位数均为页号部分)
④据得到的页号查表得物理块号
⑤将物理块号与页内偏移量拼接
由题可知：
| 页号 | 页内地址 |
2^5=32,则页号占5位
2^{10}=1KB，则页内地址占10位
0A5C(H)的二进制为:0000 10 10 0101 1100 (标黄的为页号，标红的为页内地址)
由上可见页号为0000 10即为2，查表可见页号为2对应的物理块号为11
11的二进制为1011
又因为内存物理块大小=每个页面的大小，故页内地址即就是物理块内的地址(页内地址和物理地址一一对应)
则将物理块号与页内偏移量拼接就是物理地址
拼接：1011 1001 0111 00=2E5C(H)
或：11×1024+604=11868=2E5C(H)（10 0101 1100的十进制=604）

5. 在一个只允许单向行驶的十字路口，分别有若干由东向西，由南向北的车辆在等待通过十字路口。为了安全，每次只允许一辆车通过。当有车辆通过时其它车辆必须等候，当无车辆在路口行驶时则允许辆车通过。

答:



6. 两个进程合作完成数据计算和打印工作，计算进程未计算完就不打印，反之亦然，双方共用一个缓冲区，写出此算法。




```
Var mutex,empty,mutex:semaphore:=1,1,1

process cp:
 begin
  repeat
    wait(empty);
    wait(mutex);
     将计算结果写入buffer;
    signal(mutex);
    signal(full);
 until false;
 end

process 2:
    begin
      repeat
        wait(full);
        wait(mutex);
       将计算结果从buffer取出;
        signal(mutex);
        signal(empty);
     until false;
    end
```

8. 在一辆公共汽车上，司机和售票员各行其职，司机负责开车和到站停车;售票员负责售票和开、关门，当售票员关好车门后，司机才能继续开车行驶。试用P、V操作实现司机与售票员之间的同步。

```
var S1,S2 : semaphore ;
S1=0；S2=0；

Procedure driver          Procedure Conductor
begin                      begin
while TRUE                 while TRUE
begin                      begin
P(S1);                       关车门；
Start;                     v(s1);
Driving;                     售票；
Stop;                      p(s2);
V(S2);                     开车门；
end                        上下乘客；
end                        end
                           end
```

答：解析：在这个问题中没有资源的抢夺，所以无互斥信号量。司机和售票员是同步关系，司机需要接收门是否关好的信号量，而售票员需要接收车是否到站的信号量。
活动顺序：关车门->启动车辆->售票->到站停车->开车门
初始状态为：停车且门未关。
流程：售票员给司机关门的信号，司机收到后开始正常行驶车辆，到站时由司机给售票员停车的信号。
解：设关门信号量为door=0,停车信号量为stop=0。

```
void conductor()
{
    while(true)
    {
        //关门
        signal(door);//售票员给司机关门的信号
        //此阶段为售票时间
        wait(stop);//等待停车信号，一旦停车，则开门
        //开门
    }
}
void driver()
{
    while(true)
    {
        wait(door);//司机等待关门信号，一旦获取信号，则启动车辆
        //此阶段为正常行车时间
        signal(stop);//司机给售票员停车的信号
    }
}
```

例2：若有两个售票员，一人控制一扇门，其余情况不变。用PV模拟售票员和汽车司机的同步行为。

> 设两扇门的信号量为door1,door2;司机与两位售票员都需要通信，告知他们停车的信号，将其设为stop1和stop2。
> 解：设door1=0,door2=0,stop1=0,stop2=0;

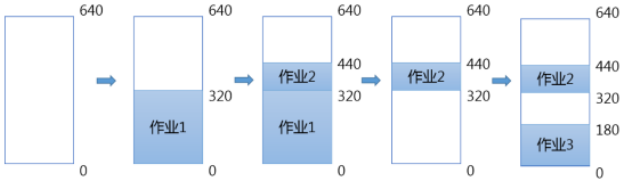
```
void conductor1()
{
    while(true)
    {
        //关门
        signal(door1);//售票员给司机关门的信号
        //此阶段为售票时间
        wait(stop1);//等待停车信号，一旦停车，则开门
        //开门
    }
}
void conductor2()
{
    while(true)
    {
        //关门
        signal(door2);//售票员给司机关门的信号
        //此阶段为售票时间
        wait(stop2);//等待停车信号，一旦停车，则开门
        //开门
    }
}
void driver()
{
    while(true)
    {
        wait(door1);
        wait(door2);//司机等待两扇门的关门信号，一旦获取信号，则启动车辆
        //此阶段为正常行车时间
        signal(stop1);//司机给售票员1停车的信号
        signal(stop2);//司机给售票员2停车的信号
    }
}

```

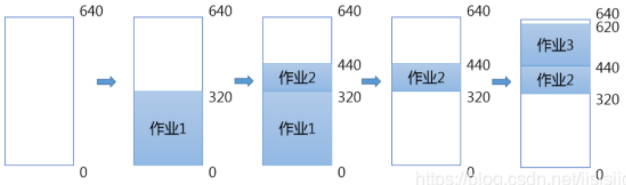
一、动态分区管理

3. 某系统采用动态分区分配方式管理一块内存空间，该内存空间容量为640KB，在内存分配时，系统优先使用空闲区低端的空间。现有如下请求序列：作业1请求320KB，作业2请求120KB，作业1释放320KB，作业3请求180KB，请分别画图表示，使用如下基于顺序搜索的动态分区分配算法进行内存分配和回收后，内存的实际使用情况：（1）首次适应（First Fit，FF）算法；（2）最佳适应（Best Fit，BF）算法。（本题7分）

（1）使用首次适应（FF）算法的内存分配过程如下图所示，最后的内存使用情况示于下图最右侧，两空闲区域的大小分别为 140KB 和 200KB。



（2）使用最佳适应（BF）算法的内存分配过程如下图所示，最后的内存使用情况示于下图最右侧，两空闲区域的大小分别为 320KB 和 20KB。



习题1 假设磁道数为0——199，申请调度的盘块儿分别在98, 183, 37, 122, 14, 124, 65, 67 磁道上。当前硬盘磁头在第53号磁道，向磁道增大方向访问。试采用下列4种磁盘调度算法的平均寻道时间和磁道的访问次序。

（1）先来先服务FCFS

（2）最短寻道时间优先FCFS

（3）电梯调度(扫描算法)SCAN

（4）单向电梯调度(循环扫描算法)C-SCAN

嗜睡的理发师问题

问题描述
一个理发店有N个沙发，1个理发椅；
理发师：

- 持续睡觉，理发，收钱的动作

顾客：

- 若有沙发，进入等待；否则离开。
- 理发椅空，一顾客放弃沙发，去唤醒它理发；
- 理发后付费，付费完毕离开理发椅，离店。

同步关系分析

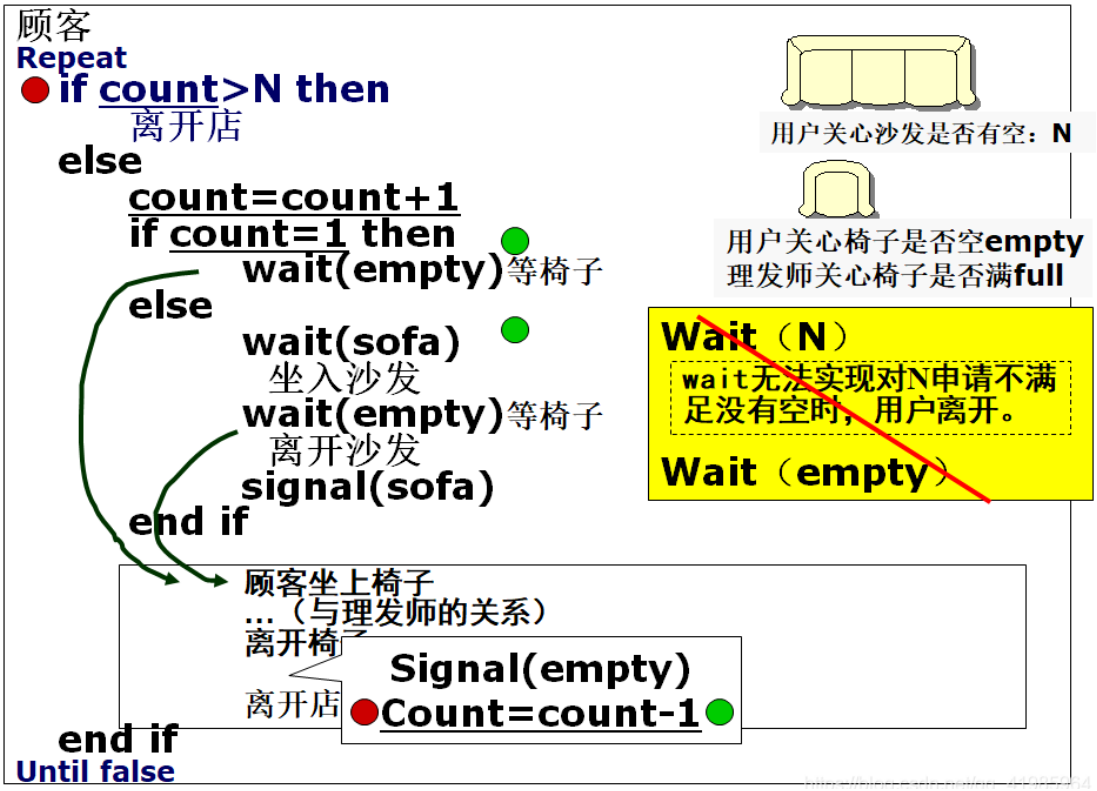
从简单的入手：
理发师与椅子上的顾客间的同步，简单而典型

顾客
Repeat
…
顾客坐上椅子
singal(full)（叫理发师）
wait(cutok)（享受理发中）
付费
signal(payment)
wait(receipt)
离开椅子
离开店

Until false

repeat
wait(full)(阻塞就是在睡觉)
理发
signal(cutok)
wait(payment)
收费找零
Signal(receipt)
Until false

顾客间对椅子、沙发使用导致的相互关系

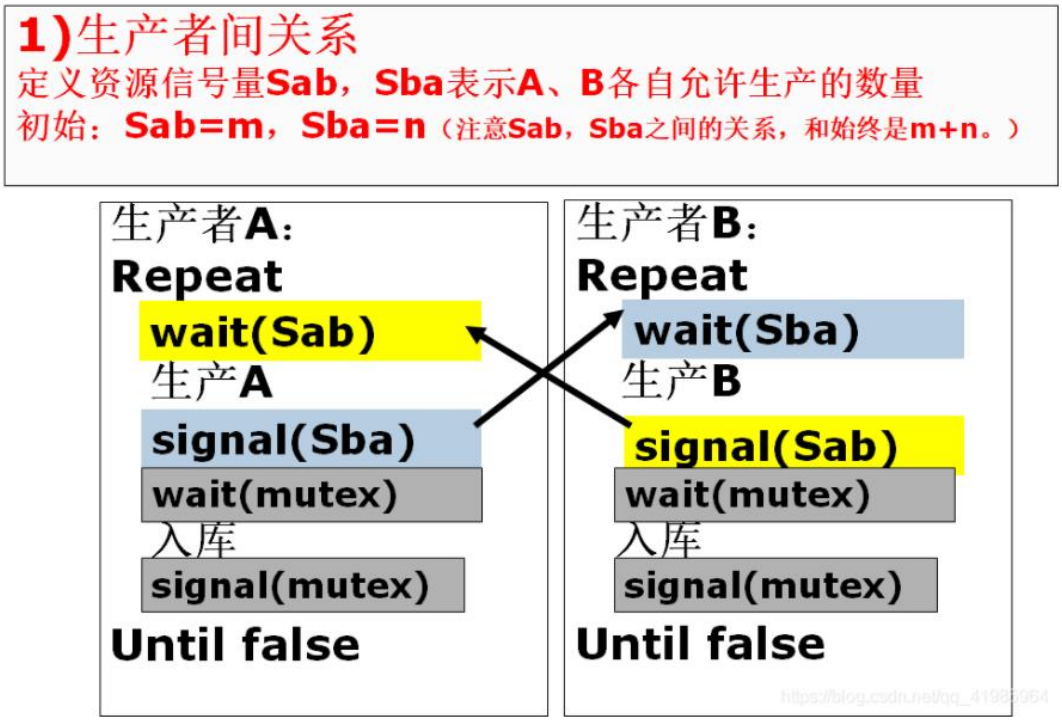


生产与销售问题

问题描述
设一无限大仓库。就1个门，不允许同时入库，也不允许边入库边出库。
1）两个生产者A,B生产各自的产品入库；但要求满足关系Sa-Sb在限定的[-n,m]差值范围内。
2）一个销售者取产品销售，但对两种产品的卖出进度要把握在Ma-Mb也在[-n,m]范围内。

同步关系分析
互斥：三人在对仓库的使用上必须互斥（mutex=1）
顺序关系：

- 1)生产者AB之间：因为假设仓库无限大，不需考虑空的限制。控制其能否生产的关键是再生产一个能否满足Sa-Sb在限定的[-n,m]范围内。
  - A、B关心的都是各自的生产指标数量，即A关心x=A-B这个变量的值，同样B关心y=B-A。
  - 初值：初始若无产品，A最多可生产m个，B最多可生产n个。自己生产一个必然使对方可以多生产一个。
- 2)生产与消费之间
  - 关心卖出的A、B间的差值



若当前状态是仓库中已有3个A、1个B，Sab，Sba值是？A已比B多2 Sab=m-Sba=n+2

2）生产与销售的关系

- 销售要做的：
  - 是否有商品？
    - 设置总商品数S，每次wait（S）
  - 有商品，是A还是B?
    - 设置A,B商品计数的信号量SA,SB。对某种商品执行取就wait（SA）
  - 是否可以取？
    - 取操作要满足销售产品的关系。所以wait（SA/SB）前面需要判断控制---判断差值difference，