1. 有一个二维数组：Var A:ARRAY[1...100,1..100] OF integer;按先行后列的次序存储。对一采用 LRU 置换算法的页式虚拟存储器系统，假设每页可存放 200 个整数。若分配一个进程的内存块数为 3，其中一块用来装入程序和变量 i、j，另外两块专门用来存放数组（不作它用），且程序段已在内存，但数据页尚未装入内存。请分别就下列程序计算执行过程中的缺页次数。

   程序 1：

   ```
   FOR i:=1 TO 100 DO
   FOR j:=1 TO 100 DO
   A[i,j]:=0
   ```

   程序 2：

   ```
   FOR j:=1 TO 100 DO
   FOR i:=1 TO 100 DO
   A[i,j]:=0
   ```

   解：

   对程序 1，首次缺页中断（访问 A[0，0]时产生）将装入数据的第 1、2 行共 200 个整数，由于程序是按行对数组进行访问的，只有在处理完 200 个整数后才会再次产生缺页中断；以后每调入一页，也能处理 200 个整数，因此，处理 100*100 个整数共将发生 100*100/200=50 次缺页。

   对程序 2，首次缺页中断同样将装入数组的第 1、2 行共 200 个整数，但由于程序是按列对数组进行访问的，因此在处理完 2 个整数后又会再次产生缺页中断；以后每调入一页，也只能处理 2 个整数，因此，处理 100*100 个整数共将发生 100*100/2=5000 次缺页。

2. 某虚拟存储器的用户空间共有 32 个页面，每页 1K，主存 16K。假定某时刻系统为用户的第 0、1、2、3 页分配的物理块号为 5、10、4、7，而该用户作业的长度为 6 页，试将十六进制的虚拟地址 0A5C、103C、1A5C 转换成物理地址。

   解：

   根据题意，虚拟存储器的用户空间共有 32 个页面，每页 1K，则页号占 $\log_2 32=5$ 位，页面大小 $\log_2 1024=10$ 位，即该系统的逻辑地址有 15 位，其中高 5 位为页号，低 10 位为页内地址；物理地址有 14 位，其中高 4 位为块号，低 10 位为块内地址。

   又：

   用户作业的长度为 6 页，页号=[虚拟地址/页面大小] （[]表示向下取整）

   页号与物理块对应表

页号	物理块号
0	5
1	10
2	4
3	7

   所以：

   (0A5C)H 的页号为 2，页号合法，页表对应物理块号为 4，则用物理块号替换页号为：(01001001011100)B=(125C)H；

   (103C)H 的页号为 4，页号合法，但该页未装入内存，故产生缺页中断；

   (1A5C)H 的页号为 6，为非法页号，故产生越界中断。

3. 考虑一个请求调页系统，它采用全局置换策略和平均分配内存块的算法（即若有 m 个内

存块和 n 个进程，则每个进程分得 m/n 个内存块）。如果在该系统中测得如下的 CPU 和对换盘的利用率，请问能否用增加多道程序的度数来增加 CPU 的利用率，为什么？

1) Cpu 利用率 13%，盘利用率 97%；
2) Cpu 利用率 87%，盘利用率 3%；
3) Cpu 利用率 13%，盘利用率 3%；

解：

1).这种情况下表示系统在进行频繁地置换，以致大部分时间被花在页面置换上，此时增加多道程序的度数会进一步增加缺页率，使系统进一步恶化，所以不能增加多道程序的度数来增加 CPU 的利用率，反而应减少内存中的作业道数。

2).在这种情况下，CPU 的利用率已相当高，但对换盘的利用率却相当低，这表示运行进程的缺页率很低，可以适当增加多道程序的度数来增加 CPU 的利用率。

3).在这种情况下，CPU 的利用率相当低，而且对于换盘的利用率也非常低，表示内存中可运行的程序数不足，此时应该增加多道程序的度数来增加 CPU 的利用率。

4. 现有一请求调页系统，页表保存在寄存器中。若一个被替换的页未被修改过，则处理一个缺页中断需要 8ms；若被替换的页已被修改过，则处理一个缺页中断需要 20ms。内存存取时间为 1us，访问页表的时间可忽略不计。假定 70%被替换的页被修改过，为保证有效存取时间不超过 2us，可接受的最大缺页率是什么？

解：

在缺页中断处理完成，调入请求页面后，还需要 1us 的存取访问，即当：

1).当未缺页时，直接访问内存，用时 1us；
2).当缺页时，如果未修改，则用时 8ms+1us;
3).当缺页时，而且修改了，则用时 20ms+1us;

因此，设最大缺页中断率为 p,则有：

(1-p)*1us+(1-70%)*p*(1us+8ms)+70%*p*(1us+20ms)<=2us
即 1us+(1-70%)*p*8ms+70%*p*20ms<=2us
解得 p<=0.00006

5. 假如一个程序的段表如下所示，其中存在位为 1 表示段在内存，存取控制字段中 W 表示可写，R 表示可读，E 表示可执行。对下面的指令，在执行时会产生什么样的结果？

(1)STORE R1,[0,70]
(2)STORE R1,[1,20]
(3)LOAD R1.[3,20]
(4)LOAD R1.[3,100]
(5)JMP [2,100]

段表

段号	存在位	内存地址	段长	存取控制	其他信息
0	0	500	100	W	
1	1	1000	30	R	
2	1	3000	200	E	
3	1	8000	80	R	
4	0	5000	40	R	

解：

(1).从段表中可读出第 0 段的存在位为 0，表示此段未装入内存，因此产生缺页中断；
(2).从段表中可读出第 1 段在内存中，但存取控制为 R，即只读，而此指令要求写，故

访问权限不合法，产生保护性中断；

(3).从段表中可读出第 3 段在内存中，此指令中段内位移小于段长，存取控制合法，求出其内存地址为 8000+20=8020，指令将该单元的内容读到寄存器 R1 中；

(4).从段表中可读出第 3 段在内存中，但此指令中段内位移大于段长，产生越界中断；

(5).从段表中可读出第 2 段在内存中，此指令中段内位移小于段长，存取控制合法，求出其内存地址为 3000+100=3100，指令执行后，跳转到内存单元 3100 处继续执行。

6. 某软盘有 40 个磁道，磁头从一个磁道移至另一个磁道需要 6ms，文件在磁盘上非连续存放，逻辑上相邻数据块的平均距离为 13 个磁道，每块的旋转延迟时间及传输时间分别为 100ms 和 25ms，则读取一个 100 块的文件需要多少时间？如果系统对磁盘进行了整理，让同一个磁盘块尽可能靠拢，从而使逻辑上相邻的数据块的平均距离降为 2 磁道，这时读取一个 100 块的文件需要多少时间？

解：

磁盘访问时间 t_a=寻道时间 t_s+旋转延迟时间 t_r+传输时间 t_t。

根据题意，t_r=100ms，t_t=25ms。

(1).文件在磁盘上非连续存放，寻道时间 t_s=13*6=78ms，

磁盘访问时间 t_a=78+100+25=203ms，

读取 100 块的时间=100*203=20300ms；

(2).整理文件后，文件在磁盘上仍非连续存放，寻道时间 t_s=2*6=12ms，

磁盘访问时间 t_a=128+100+25=137ms，

读取 100 块的时间=100*137=13700ms。

7. 已知某分页系统，主存容量为 64K，页面大小为 1K，对一个 4 页大的作业，其 0，1，2，3 页分别被装入到主存的 2，4，6，7 块中。

1）十进制的逻辑地址 1023、2500、3500、4500 转换成物理地址。

2）以十进制的逻辑地址 1023 为例，画出地址变换过程图。

解：

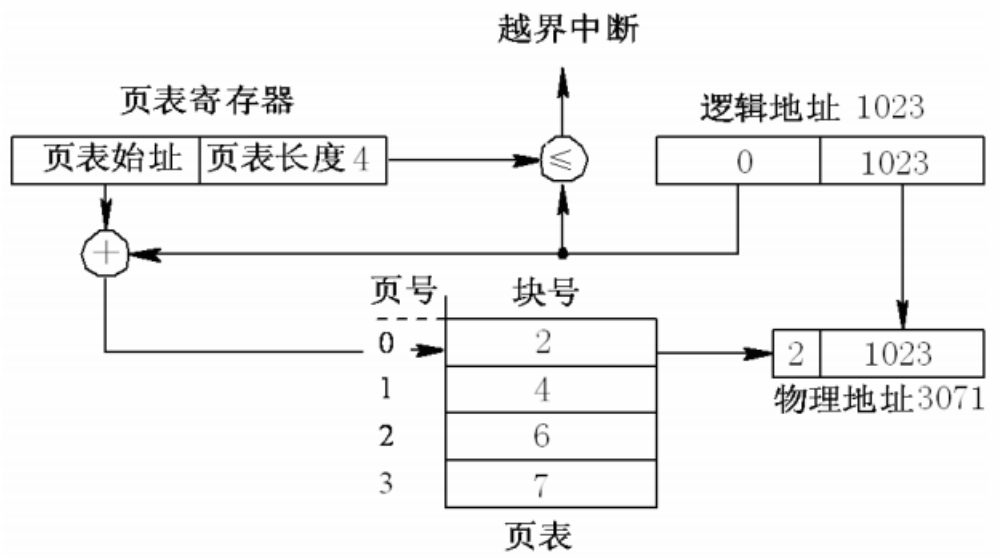
(1).逻辑地址 1023：1023/1K 得到页号为 0，页内地址为 1023，查页表找到对应的物理块号为 2，故物理地址为 2*1K+1023=3071；

逻辑地址 2500：2500/1K 得到页号为 2，页内地址为 452，查页表找到对应的物理块号为 6，故物理地址为 6*1K+452=6596；

逻辑地址 3500：3500/1K 得到页号为 3，页内地址为 428，查页表找到对应的物理块号为 7，故物理地址为 7*1K+428=7596；

逻辑地址 4500：4500/1K 得到页号为 4，页内地址为 404，因页号不小于页表长度，故产生越界中断。

(2). 逻辑地址 1023 地址变换过程如图所示：



8. 对一个将页表放在内存中的分页系统：
(1) 如果访问内存需要 0.2μs，有效访问时间为多少?
(2) 如果增加一个快表，且假定在快表中找到页表项的概率高达 90%，则有效访问时间又是多少（假定查找快表需花的时间为 0）？
解：
（1）有效访问时间为：2 * 0.2=0.4μs
（2）有效访问时间为：0.9 *0.2+（1-0.9）*2*0.2=0.22μs

9. 桌上有个能剩得下五个水果的空盘子。爸爸不停地向盘中放苹果和桔子，儿子不停地从盘中取出桔子享用，女儿不停地从盘中取出苹果享用。规定三人不能同时从盘子中取放水果。试用信号量实现爸爸、儿子和女儿这三个循环进程之间的同步。

设置信号量 semaphore empty = 5, orange = 0, apple = 0, mutex = 1;（初始 5 个盘子都为空，儿子和女儿都没拿到水果，三人不能同时操作，意味着只有一个临界资源。）

```
Dad() {
  while (1) {
    wait(empty);
    wait(mutex);
    将水果放入盘中;
    signal(mutex);
    if (放入的是桔子)
    {
      signal(orange);
    }
    else signal(apple);
  }
}

Son() {
  while (1) {
```

```
        wait(orange);
        wait(mutex);
        从盘中取一个桔子;
        signal(mutex);
        signal(empty);
        享用桔子;
      }
    }

    Daughter() {
      while (1) {
        wait(apple);
        wait(mutex);
        从盘中取一个苹果;
        signal(mutex);
        signal(empty);
        享用苹果;
      }
    }
```

10. 某杂技团进行走钢丝表演。在钢丝的 A、B 两端各有 n 名演员（n>1）在等待表演。 只要钢丝上无人时便允许一名演员从钢丝的一端走到另一端。现要求两端的演员交替地走钢丝，且从 A 端的一名演员先开始。请问，把一名演员看作一个进程时，怎样用 PV 操作来进行控制？请写出能进行正确管理的程序。

```
sa=1,sb=0
cobegin
    process    A
    begin
    P(sa)
    走钢丝
    V(sb)
    end
    process    B
    begin
    P(sb)
    走钢丝
    V(sa)
end  coend
```

11. LRU 算法思想。
12. 银行家算法。

    在银行家算法中，若出现下述资源分配情况，试问：

Process	Allocation	Need	Available
P0	0032	0012	1622

P1	1000	1750	
P2	1354	2356	
P3	0332	0652	
P4	0014	0656	

(1)该状态是否安全？

解：该状态是安全的，因为存在一个安全序列<P0,P3,P4,P1,P2>。

下表为该时刻的安全序列：

进程	Work	Need	Allocation	Work+Allocation	Finish
P0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P3	1 6 5 4	0 6 5 2	0 3 3 2	1 9 8 6	true
P4	1 9 8 6	0 6 5 6	0 0 1 4	1 9 9 10	true
P1	1 9 9 10	1 7 5 0	1 0 0 0	2 9 9 10	true
P2	2 9 9 10	2 3 5 6	1 3 5 4	3 12 14 14	true

(2).若进程 P2 提出请求 Request(1,2,2,2)后，系统能否将资源分配给它？

解：若进程 P2 提出请求 Request(1，2，2，2)后，系统不能将资源分配给它，若分配给进程 P2，系统还剩的资源情况为（0，4，0，0），此时系统中的资源将无法满足任何一个进程的资源请求，从而导致系统进入不安全状态，容易引起死锁的发生。

13. 假设某系统中有 3 种资源(R1,R2,R3)，在某时刻系统中共有 4 个进程，进程(P1,P2,P3,P4)的最大资源需求数向量和此时已分配的资源数向量分别为：

进程	最大资源需求	当前已分配到资源
P1	3 2 2	1 0 0
P2	6 1 3	5 1 1
P3	3 1 4	2 1 1
P4	4 2 2	0 0 2

系统中当前可用资源向量为(1,1,2)，问：

(1) 计算还需要资源数组；

	MAX	Allocation	Need	Available
P1	3 2 2	1 0 0	2 2 2	1 1 2
P2	6 1 3	5 1 1	1 0 2	
P3	3 1 4	2 1 1	1 0 3	
P4	4 2 2	0 0 2	4 2 0	

(2) 系统此时是否安全？

	Work	Need	Allocation	Work+Allocation	Finish
P2	1 1 2	1 0 2	5 1 1	6 2 3	True
P3	6 2 3	1 0 3	2 1 1	8 3 4	True
P4	8 3 4	4 2 0	0 0 2	8 3 6	True
P1	8 3 6	2 2 2	1 0 0	9 3 6	True

存在安全序列 P2、P3、P4、P1，即系统此时安全

(3)如果进程 P2 发出资源请求向量(1,0,1)，系统能否将资源分配给它？

① Request2(1, 0, 1)≤Need2(1, 0, 2)

② Request2(1, 0, 1)≤Available2(1, 1, 2)

	MAX	Allocation	Need	Available
P1	3 2 2	1 0 0	2 2 2	0 1 1

P2	6 1 3	6 1 2	0 0 1	
P3	3 1 4	2 1 1	1 0 3	
P4	4 2 2	0 0 2	4 2 0	

利用安全性算法检测是否安全：

	Work	Need	Allocation	Work+Allocation	Finish
P2	0 1 1	0 0 1	6 1 2	6 2 3	True
P3	6 2 3	1 0 3	2 1 1	8 3 4	True
P4	8 3 4	4 2 0	0 0 2	8 3 6	True
P1	8 3 6	2 2 2	1 0 0	9 3 6	True

存在安全序列 P2、P3、P4、P1，即系统此时安全

(4)如果进程 P1 发出资源请求向量(1,0,1)，系统能否将资源分配给它？

① Request1(1, 0, 1)≤Need1(2, 2, 2)

② Request1(1, 0, 1)≤Available1(1, 1, 2)

	MAX	Allocation	Need	Available
P1	3 2 2	2 0 1	1 2 1	0 1 1
P2	6 1 3	5 1 1	1 0 2	
P3	3 1 4	2 1 1	1 0 3	
P4	4 2 2	0 0 2	4 2 0	

进行安全性检查，可用资源 Available(0 1 1)已不能满足任何进程需要，故系统进入不安全状态。

14. 假设某系统中有 4 种资源，在某时刻系统中共有 5 个进程，进程(P0,P1,P2,P3,P4)的最大资源需求数向量和此时已分配的资源数向量分别为：

进程	最大资源需求	当前已分配到资源
P0	0 0 1 2	0 0 1 2
P1	2 7 5 0	2 0 0 0
P2	6 6 5 6	0 0 3 4
P3	4 3 5 6	2 3 5 4
P4	0 6 5 2	0 3 3 2

系统中当前可用资源向量为(2,1,0,0)，问：

(1) 计算进程还需要请求的资源向量；

	MAX	Allocation	Need	Available
P0	0 0 1 2	0 0 1 2	0 0 0 0	2 1 0 0
P1	2 7 5 0	2 0 0 0	0 7 5 0	
P2	6 6 5 6	0 0 3 4	6 6 2 2	
P3	4 3 5 6	2 3 5 4	2 0 0 2	
P4	0 6 5 2	0 3 3 2	0 3 2 0	

(2) 系统当前是处于安全状态么？

	Work	Need	Allocation	Work+Allocation	Finish
P0	2 1 0 0	0 0 0 0	0 0 1 2	2 1 1 2	True
P3	2 1 1 2	2 0 0 2	2 3 5 4	4 4 6 6	True
P4	4 4 6 6	0 3 2 0	0 3 3 2	4 7 9 8	True
P1	4 7 9 8	0 7 5 0	2 0 0 0	6 7 9 8	True
P2	6 7 10 8	6 6 2 2	0 0 3 4	6 7 12 12	True

存在安全序列 P0、P3、P4、P1、P2，即系统此时安全

(2)当进程 P2 申请(0,1,0,0)时，系统能立即满足么？

① Request2(0，1，0，0)≤Need2(6，6，2，2)

② Request2(0，1，0，0)≤Available1(2，1，0，0)

	MAX	Allocation	Need	Available
P0	0 0 1 2	0 0 1 2	0 0 0 0	2 0 0 0
P1	2 7 5 0	2 0 0 0	0 7 5 0	
P2	6 6 5 6	0 1 3 4	6 5 2 2	
P3	4 3 5 6	2 3 5 4	2 0 0 2	
P4	0 6 5 2	0 3 3 2	0 3 2 0	

利用安全性算法检测是否安全：

	Work	Need	Allocation	Work+Allocation	Finish
P0	2 0 0 0	0 0 0 0	0 0 1 2	2 0 1 2	True
P3	2 0 1 2	2 0 0 2	2 3 5 4	4 3 6 6	True
P4	4 3 6 6	0 3 2 0	0 3 3 2	4 6 9 8	True
P1	4 6 9 8	0 7 5 0	2 0 0 0		False
P2		6 5 2 2	0 1 3 4		

不存在安全序列

填空题：

1. 产生死锁的原因是：竞争临界资源和进程推进不当。
2. 采用多道程序设计能充分发挥 CPU 与外围设备并行工作的能力。
3. 用信号量 S 实现对系统中 4 台打印机的互斥使用，S.value 的初值应设置为 4，若 S.value 的当前值为-1，则表示 S.L 队列中有 1 个等待进程。
4. 在生产者，消费者问题中，应设置互斥信号量 mutex、资源信号量 full 和 empty，它们的初值分别应为 1，0，n。
5. 对于请求式页式管理系统中，提取页面的策略有：预调页策略和请求调页策略。
6. 在引入线程的操作系统中，把线程作为分配和调度的基本单位，把进程作为资源拥有的基本单位。
7. 在首次适应算法中，空闲区以地址递增的次序拉链；在最佳适应算法中，空闲区以空闲区大小递增的次序拉链。

简答题

1. 进程的三个状态即转换原因。

   就绪状态：进程已分配到除 CPU 以外的所有必要资源后，只要再获得 CPU，便可立即执行。

   执行状态：进程已获得 CPU，其程序正在执行的状态。

   阻塞状态：正在执行的程序由于发生某事件暂时无法继续执行时的状态。

   就绪—>执行：处于就绪状态的程序，在调度程序为其分配处理机后，进程便由就绪状态转换为执行状态

   执行—>就绪：正在执行的进程，因分配给它的时间片已用完而被剥夺处理机暂停执行时，进程从执行状态转换为就绪状态

   执行—>阻塞：正在执行的程序由于发生某事件暂时无法继续执行时，进程从执行状态

转换为阻塞状态

阻塞—>就绪：处于阻塞状态的进程，由于等待事件已发生，进程从阻塞状态转换为就绪状态。

2. 为什么要引入线程？好处？

由于进程是资源的拥有者，所以在创建、撤销、切换操作中需要较大的时空开销，限制了并发程度的进一步提高。为减少进程切换的开销，把进程作为资源分配单位和调度单位这两个属性分开处理，即进程还是作为资源分配的基本单位，但是不作为调度的基本单位，把调度执行与切换的责任交给“线程”。

这样做的好处不但可以提高系统的并发度，还能适应新的对称多处理机（SMP）环境的运行，充分发挥其性能。

3. 什么是死锁？产生原因？

如果一组进程中的每一个进程都在等待仅由该组进程中的其它进程才能引发的事件，那么该组进程是死锁的。

原因：

（1） 因为系统资源不足。

（2） 进程运行推进的顺序不合适。

（3） 资源分配不当等。

4. 什么叫抖动？产生原因？

抖动(Thrashing)就是指当内存中已无空闲空间而又发生缺页中断时，需要从内存中调出一页程序或数据送磁盘的对换区中，如果算法不适当，刚被换出的页很快被访问，需重新调入，因此需再选一页调出，而此时被换出的页很快又要被访问，因而又需将它调入，如此频繁更换页面，以致花费大量的时间，我们称这种现象为"抖动"；

产生抖动的原因是由于 CPU 的利用率和多道程序度的对立统一矛盾关系引起的，为了提高 CPU 利用率，可提高多道程序度，但单纯提高多道程序度又会造成缺页率的急剧上升，导致 CPU 的利用率下降，而系统的调度程序又会为了提高 CPU 利用率而继续提高多道程序度，形成恶性循环，我们称这时的进程是处于"抖动"状态。

5. 什么是缓冲区管理？原因？

缓冲区的管理主要功能是组织好这些缓冲区，并提供获得和释放缓冲区的手段。

原因：

(1).缓和 CPU 与 I/O 设备间速度不匹配的矛盾；

(2).减少对 CPU 的中断频率，放宽对 CPU 中断响应时间的限制；

(3).解决数据粒度不匹配的问题；

(4).提高 CPU 和 I/O 设备之间的并行性。

6. 什么是虚拟存储器？特点？

虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器体系。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。

特点：

1.多次性：一个作业中的程序和数据无需在作业运行时一次性地全部装入内存，而是允许被多次调入内存运行，即只需将当前要运行的那部分程序和数据装入内存即可运行。

2.对换性：一个作业中的程序和数据，无须在作业运行时一直常驻内存，而是允许在作业的运行过程中进行换进、换出，即在进程运行期间，允许将那些暂不使用的代码和数据从内存调至外存的交换区，待以后需要时再将它们从外存调至内存。

3.虚拟性：能够从逻辑上扩充容量，使用户看到的内存容量远大于实际内存容量。

7. 假脱机技术是什么？

通过共享设备来模拟独享设备所采用的操作是假脱机操作，即在联机情况下外部设备设备同时操作。所使用的假脱机技术称之为假脱机技术。