操作系统答案

目录

习题一

一．思考题

3. 什么是操作系统？计算机系统中配置操作系统的主要目标是什么？

答(p1)：操作系统是管理系统资源、控制程序执行、改善人机界面、提供各种服务，合理组织计算机工作流程和为用户有效使用计算机提供良好运行环境的一种系统软件。
配置操作系统的主要目标可归结为：
(1) 方便用户使用(2) 扩大机器功能(3) 管理系统资源(4) 提高系统效率(5) 构筑开放环境。

5. 操作系统要为用户提供哪些基本和共性的服务？

答(p25)：操作系统提供给程序和用户的共性服务大致有：
(1) 创建程序(2) 执行程序(3) 数据 I/O(4) 信息存取(5) 通信服务(6) 错误检测和处理

9．试叙述系统调用的实现原理。

答（p28）系统调用的实现有以下几点：⑴编写系统调用处理程序 ⑵设计一张系统调用入口地址表，每个入口地址都指向一个系统调用的处理程序，有的系统还包含系统调用自带参数的个数 ⑶陷入处理机制，需开辟现场保护，以保存发生系统调用时的处理器现场。

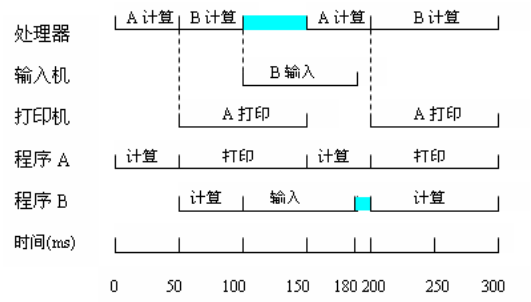
10.试叙述系统调用与过程调用的主要区别。

答（p29）⑴调用形式不同 ⑵被调用代码的位置不同 ⑶提供方式不同 ⑷调用的实现不同

二．应用题

2. 一个计算机系统，有一台输入机和一台打印机，现有两道程序投入运行，且程序 A 先开始做，程序 B 后开始运行。程序 A 的运行轨迹为：计算 50ms、打印 100ms、再计算 50ms、打印 100ms，结束。程序 B 的运行轨迹为：计算 50ms、输入 80ms、再计算 100ms，结束。试说明(1)两道程序运行时，CPU 有无空闲等待？若有，在哪段时间内等待？为什么会等待？(2)程序 A、B 有无等待 CPU 的情况？若有，指出发生等待的时刻。

答：画出两道程序并发执行图如下：



(1) 两道程序运行期间，CPU 存在空闲等待，时间为 100 至 150ms 之间(见图中有色部分)。

(2) 程序 A 无等待现象，但程序 B 有等待。程序 B 有等待时间段为 180ms 至 200ms 间(见图中有色部分)。 学生补充：程序 B 在 0~50ms 时也存在等待。

习题二

一．思考题

26.什么是进程？计算机操作系统中为什么引入进程？（教材 113~教材 114 页）

进程的定义：进程是一个可并发执行的具有独立功能的程序关于某个数据集合的一次执行过程，也是操作系统进行资源分配和保护的基本单位。
引入进程的原因：一是刻画系统的动态性，发挥系统的并发性，提高资源利用率。
二是解决共享性，正确描述程序的执行状态。

28.进程最基本的状态有哪些？哪些事件可能引起不同状态之间的转换？

进程最基本的状态有三种：
运行态：进程占有处理器正在运行。
就绪态：进程具备运行条件，等待系统分配处理器以便运行。

等待态：又称为阻塞（zǔsè）态或睡眠态，指进程不具备运行条件，正在等待某个事件的完成。

进程状态转换的具体原因：
运行态→等待态 等待使用资源或某事件发生，如等待外设传输、等待人工干预。
等待态→就绪态 资源得到满足或某事件已经发生，如外设传输结束；人工干预完成。
运行态→就绪态 运行时间片到，或出现有更高优先权进程。
就绪态→运行态 CPU 空闲时被调度选中一个就绪进程执行。

34. 叙述组成进程的基本要素，并说明它的作用。（教材 120 页）

每个进程有 4 个要素组成：控制块、程序块、数据块和堆栈。
（1）进程控制块 每一个进程都将捆绑一个进程控制块，用来存储进程的标志信息、现场信息和控制信息。进程创建时建立进程控制块，进程撤销时回收进程控制块，它与进程一一对应。
（2）进程程序块 即被执行的程序，规定了进程一次运行应完成的功能。通常它是纯代码，作为一种系统资源可被多个进程共享。
（3）进程数据块 即程序运行时加工处理对象，包括全局变量、局部变量和常量等的存放区以及开辟的工作区，常常为一个进程专用。
（4）系统/用户堆栈 每一个进程都将捆绑一个系统/用户堆栈，用来存储进程的标志信息、现场信息和控制信息。进程创建时建立进程控制块，进程撤销时回收进程控制块，它与进程一一对应。

38. 什么是进程的上下文？简述其主要内容。（教材 120 页）

操作系统中把进程物理实体和支持进程运行的环境合称为进程上下文（process context）。
它包括三个组成部分：
（1）用户级上下文（user-level context）。 由用户进程的程序块、用户数据块（含共享数据块）和用户堆栈组成的进程地址空间。

（2）系统级上下文（system-level context）。包括进程控制块、内存管理信息、进程环境块，以及系统堆栈等组成的进程地址空间。
（3）寄存器上下文（register context）。由程序状态字寄存器、各类控制寄存器、地址寄存器、通用寄存器、用户栈指针等组成。

52. 试从调度、并发性、拥有资源和系统开销四个方面对传统进程和线程进行比较。

	线程	进程
调度	是操作系统中的基本调度和分派单位，具有唯一的标识符和线程控制块。	进程具有独立的虚地址空间，以进程为单位进行任务调度，系统必须交换地址空间，切换时间长。
并发性	同一进程的多个线程可在一个/多个处理器上并发或并行地执行	许多多任务操作系统限制用户能拥有的最大进程数目，这对许多并发应用来说是不够的。
拥有资源	同一进程的所有线程共享但不拥有进程的状态和资源，且驻留在进程的同一个主存地址空间中，可以访问相同的数据，通信和同步的实现十分方便。	是系统中资源分配和保护的基本单位，也是系统调度的独立单位。每个进程都可以各自独立的速度在 CPU 上推进。
系统开销	作为系统调度和分派的基本单位，会被频繁地调度和切换。同一进程中的多线程共享同一地址空间，能使线程快速切换。	对多个进程的管理（创建、调度、终止等）系统开销大，如响应客户请求建立一个新的服务进程的服务器应用中，创建的开销比较显著。

58. 什么是内核级线程、用户级线程和混合式线程？对它们进行比较。

内核级线程 线程管理的所有工作由操作系统内核来做。
优点：（1）在多处理器上，内核能够同时调度同一进程中多个线程并行执行；（2）若其中的一个线程被阻塞了，内核能调度同一进程的其它线程占有处理器并运行，也可以运行其它进程中的线程。（3）由于内核线程仅有很小的数据结构和堆栈，KLT 的切换比较快，内核自身也可以用多线程技术实现，从而，能提高系统的执行速度和效率。
缺点：应用程序线程在用户态运行，而线程调度和管理在内核实现，在同一进程中，控制权从一个线程传送到另一个线程时需要用户态——内核态——用户态的模式切换，系统开销较大。
用户级线程 线程管理的全部工作都由应用程序来做，在用户空间内实现，内核是不知道线程的存在的。
优点：（1）线程切换不需要内核特权方式。（2）按应用特定需要允许进程选择调度算法。
缺点：（1）在传统的基于进程操作系统中，大多数系统调用将阻塞进程。
（2）在纯 ULT 中，多线程应用不能利用多重处理的优点。

二．应用题

11.有 5 个批处理作业 A 到 E 均已到达计算中心，其运行时间分别 10、6、2、4 和 8 分钟；各自的优先级分别被规定为 3、5、2、1 和 4，这里 5 为最高级。若不考虑系统切换开销，计算出平均作业周转时间。（1）FCFS(按 A、B、C、D、E)；(2)优先级调度算法，(3)时间片轮转法(每个作业获得相同的 2 分钟长的时间片)。

(1)FCFS 调度算法（调用次序：A、B、C、D、E）

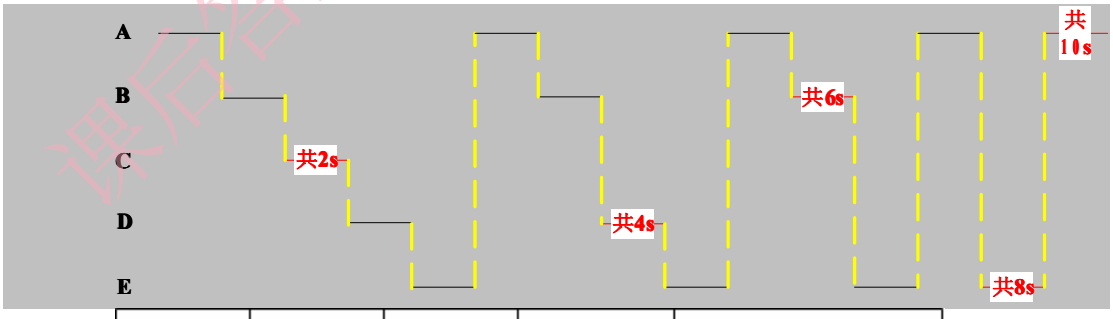
执行次序	执行时间	等待时间	周转时间	带权周转时间
A	10	0	10	1
B	6	10	16	2.66
C	2	16	18	9
D	4	18	22	5.5
E	8	22	30	3.75
作业平均周转时间 作业平均带权周转时间			T=(10+16+18+22+30)/5=19.2 W=(1+2.66+9+5.5+3.75)/5=4.38	

(2)优先级调度算法（调用次序：B（5）、E（4）、A（3）、C（2）、D（1））

执行次序	执行时间	等待时间	周转时间	带权周转时间
B	6	0	6	1
E	8	6	14	1.75
A	10	14	24	2.4
C	2	24	26	13
D	4	26	30	7.5
作业平均周转时间 作业平均带权周转时间			T=(6+14+24+26+30)/5=20 W=(1+1.75+2.4+13+7.5)/5=5.13	

(3)时间片轮转法（调用次序：按照 2s 的时间间隔循环）

按次序 A B C D E A B D E A B E A E A 轮转执行。



作业	执行时间	等待时间	周转时间	带权周转时间
A	10	20	30	3
B	6	16	22	3.66
C	2	4	6	3
D	4	12	16	4
E	8	20	28	3.5
作业平均周转时间 作业平均带权周转时间			T=(30+22+6+16+28)/5=20.4 W=(3+3.66+3+4+3.5)/5=3.43	

14.单道批处理系统中，下列三个作业采用先来先服务调度算法和最高响应比优先算法进行调度，哪一种算法性能较好？请完成下表：

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10：00	2：00				
2	10：10	1：00				
3	10：25	0：25				
平均作业周转时间＝ 平均作业带权周转时间 W=						

FIFO

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10：00	2：00	10:00	12:00	2	120/120=1
2	10：10	1：00	12:00	13:00	2:50=**2.83**	~~145~~(170)/60=2.83
3	10：25	0：25	13:00	13:25	3	180/25=7.2
平均作业周转时间＝2.61＝（2＋2.8333＋3）/ 3＝2.61（小时）＝156.6（分钟） 平均作业带权周转时间 W=3.54＝（1＋2.83＋7.2）/3＝3.678（小时）						

HRRF

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10：00	2：00	10:00	12:00	2	120/120=1
2	10：10	1：00	12:25	13:25	3:15(195)=**3.25**	195/60=3.25
3	10：25	0：25	12:00	12:25	2	120/25=4.8
平均作业周转时间＝2.41＝（2＋3:15＋2）/3＝2.416667=2.42（小时）=145（分） 平均作业带权周转时间 W=3.02＝(1＋3.25＋4.8)/3＝3.016667＝3.02(小时)=181(分钟)						

开始时只有作业 1，作业 1 被选中，执行时间 2 小时。
作业 1 执行完毕后，作业 2 的时间响应比为 1+（2-1:50）/1 = 1.1667
作业 3 的时间响应比为 1+（2-1:35）/0:25 = 1+0.416667 故选 3
然后选择作业 2。
可见 HRRF 比 FIFO 要好。

本次的作业出现如下四个问题：
1)结果未能全部化为小数　　2)单位要统一
3)小数点四舍五入　　4)大多数同学没有能够画图说明

习题三

一．思考题

8.解释进程的竞争关系和协作关系。（教材 213 页）

竞争关系：系统中的多个进程之间彼此无关，它们并不知道其它进程的存在，并且也不接受其它进程执行的影响。
协作关系：某些进程为完成同一任务需要分工协作，由于合作的每一个进程都是独立地以不可预知的速度推进，这就需要相互协作的进程在某些协调点上协调各自的工作。当合作进程中的一个到达协调点后，在尚未得到其伙伴进程发来的消息或信号之前应阻塞自己，知道其它合作进程发来协调信号或消息后方被唤醒并继续执行。这种协作进程之间相互等待对方消息或信号的协调关系称为进程同步。

9．试说明进程的互斥和同步两个概念之间的异同。

进程的互斥是解决进程间竞争关系（间接制约关系）的手段。进程互斥是指若干个进程要使用同一资源时，任何时刻最多允许一个进程去使用，其它要使用该资源的进程必须等待，直到占有资源的进程释放该资源。
进程的同步是解决进程间协作关系（直接制约关系）的手段。进程同步指两个以上进程基于某个条件来协调它们的活动。一个进程的执行依赖于另一个协作进程的消息或信号，当一个进程没有得到来自于另一个进程的消息或信号时则需等待，直到消息或信号到达才被唤醒。

10．什么是临界区和临界资源？对临界区管理的基本原则是什么？

临界区——并发进程中与共享变量有关的程序段。
临界资源——共享变量代表的资源。
临界区管理的基本原则：
Ø （1）一次至多一个进程能够在它的临界区内；
Ø （2）不能让一个进程无限地留在它的临界区内；
Ø （3）不能强迫一个进程无限地等待进入它的临界区。特别，进入临界区的任一进程不能妨碍正等待进入的其它进程的进展；

19．试比较管程与进程的不同点（教材 237 页）。

19. （1）管程是由局部于自己的若干公共变量及其说明和所有访问这些公共变量的过程所组成的软件模块；进程是一个可并发执行的具有独立功能的程序关于某个数据集合的一次执行过程，也是操作系统进行资源分配和保护的基本单位。
20. （2）管程可以作为程序设计语言的一个成分，采用管程作为同步机制便于用高级语言来书写程序，也便于程序正确性验证。有相对固定的代码编写模式；进程则可以采用中级语言（C 语言）也可以采用高级语言实现，其代码编写模式相对自由。
21. （3）管程提供了一种互斥机制，进程可以互斥地调用这些过程；进程可以并发地执行，进程的并发性能改进资源利用率提高系统效率。
22. （4）管程把分散在各个进程中互斥地访问公共变量的那些临界区集中了起来，提

供对他们的保护；进程既是系统中资源分配和保护的基本单位，也是系统调度的独立单位。

23. （5）管程是一段管理临界区资源的代码，而进程则有生命周期。

24. 什么是管道？如何通过管道机制实现进程间通信？（教材 254 页）

管道（pipeline）是连接读写进程的一个特殊文件，允许进程按先进先出传送数据，也能使进程同步执行操作。

发送进程视管道文件为输出文件，以字符流形式把大量数据送入管道；接收进程将管道文件视为输入文件，从管道中接收数据，所以，也叫管道通信。

管道中的消息是无界的，它存于外存。

此外，还需要一定的机制协调读写进程。

25. 什么是消息队列机制，叙述其工作原理。

消息队列本身是操作系统核心为通信双方进程建立的数据结构，两个用户进程间通过发送和接收系统调用来借助消息队列传递和交换消息，这样通信进程间不再需要共享变量。

如图 3-11（教材 263 页）所示，进程间的通信通过消息队列进行。消息队列可以是单消息队列，也可以是多消息队列（按消息类型）；既可以单向，也可以双向通信；既可以仅和两个进程有关，也可以被多个进程使用。

28. 什么是死锁？什么是饥饿？试举日常生活中的例子说明之。

如果在一个进程集合中的每个进程都在等待只能由该集合中的其它一个进程才能引发的事件，则称一组进程或系统此时发生了死锁（教材 268 页）。

饥饿是指一个进程由于其它进程总是优先于它而被无限期拖延（教材 214 页）。

29. 叙述产生死锁的必要条件。

（1）互斥条件（mutual exclusion）：进程应互斥使用资源，任一时刻一个资源仅为一个进程独占，若另一个进程请求一个已被占用的资源时，它被置成等待状态，直到占用者释放资源。

（2）占有和等待条件（mutual exclusion）：进程应互斥使用资源，任一时刻一个资源仅为一个进程独占，若另一个进程请求一个已经被占用的资源时，它被置成等待状态，直到占用者释放资源。

（3）不剥夺条件（no preemption）：任一进程不能从另一进程那里抢夺资源，即已被占用的资源，只能由占用进程自己来释放。

（4）循环等待条件（circular wait）：存在一个循环等待链，其中，每一个进程分别等待它前一个进程所持有的资源，造成永远等待。

二．应用题

2.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≤n）同时进入互斥段。试问：所采用的信号量初值是否相同？信号量值的变化范围如何？

所采用的互斥信号量初值不同。
1）互斥信号量初值为 1，变化范围为 [-n+1 ， 1]。
当没有进程进入互斥段时，信号量值为 1；当有 1 个进程进入互斥段但没有进程等待进入互斥段时，信号量值为 0；当有 1 个进程进入互斥段且有一个进程等待进入互斥段时，信号量值为-1；最多可能有 n-1 个进程等待进入互斥段，故此时信号量的值应为-(n-1)也就是-n+1。
2）互斥信号量初值为 m，变化范围为 [-n+m ， m]。
当没有进程进入互斥段时，信号量值为 m；当有 1 个进程进入互斥段但没有进程等待进入互斥段时，信号量值为 m-1；当有 m 个进程进入互斥段且没有一个进程等待进入互斥段时，信号量值为 0；当有 m 个进程进入互斥段且有一个进程等待进入互斥段时，信号量值为-1；最多可能有 n-m 个进程等待进入互斥段，故此时信号量的值应为-(n-m)也就是-n+m。

4.有一阅览室，读者进入时必须先在一张登记表上登记，该表为每一座位列出一个表目，包括座号、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 P、V 操作；2）管程，来实现用户进程的同步算法。

1) 使用信号量和 P、V 操作：

```
var    A: array[1..100] of Rec;
 Rec=record
     number:integer;
     name:string;
 end;
i:integer;
for i:=1 to 100 do {A[i].number:=i; A[i].name:=null;}
mutex,seatcount:semaphore;  //semaphore 中文含义：信号量
mutex:=1;seatcount:=100;
cobegin
 process    readeri(var readername:string)(i=1,2,…){
     P(seatcount);
     P(mutex);
     for i:=1 to 100 do {
       i++;
       if A[i].name=null then A[i].name:=readername;/*读者登记*/
     }
     //必须采用这种方式,因为该空位是随机产生的。我们无法知道哪个
     //读者何时离开。
     V(mutex)
     进入阅览室，座位号 i，坐下读书;
```

```
    P(mutex); //读书完毕，需要退场
    A[i] name:=null;
    V(mutex);
    V(seatcount);
    离开阅览室;
 }
coend.
```

2) 使用管程操作：

```
TYPE   readbook=monitor
VAR   R:condition;
i,seatcount:integer;
name:array[1..100] of string;
DEFINE readercome,readerleave;
USE check,wait,signal,release;

procedure readercome(readername)
begin
    check(IM);
    if seatcount≥100   wait(R,IM)
    seatcount:=seatcount+1;
    for i=1 to 100 do i++
       if name[i]==null then name[i]:=readername;
    get the seat number=i;
    release(IM);
end

procedure readerleave(readername)
begin
   check(IM);
   seatcount--;
   for i=1 to 100 do i++
   if name[i]==readername then name[i]:=null;
   release(IM);
end

begin
   seatcount:=100;name:=null;
end

cobegin
   process readeri(i=1,2.…)
      begin
        readercome(readername);
```

read the book;
readerleave(readername);
leave the readroom;
end
coend.
如何做管程题目？参考书上的例题程序一步步做下来即可。

21．系统有同类资源 m 个，被 n 个进程共享，问：当 m＞n 和 m≤n 时，每个进程最多可以请求多少个这类资源时，使系统一定不会发生死锁？

当 m≤n 时，每个进程最多请求 1 个这类资源时，系统一定不会发生死锁。当 m>n 时，如果 m/n 不整除，每个进程最多可以请求”商+1”个这类资源，否则为”商”个资源，使系统一定不会发生死锁。

28．把死锁检测算法用于下面的数据，并请问：

$$Available=(1,0,2,0)$$

$$Need=\begin{bmatrix}1&1&0&0\\0&1&1&2\\3&1&0&0\\0&0&1&0\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\1&1&0&1\\0&0&0&0\end{bmatrix}$$

（1）此时系统此时处于安全状态吗？
（2）若第二个进程提出资源请求 request2(0，0，1，0)，系统能分配资源给它吗？
（3）若第五个进程提出资源请求 request5(0，0，1，0)，系统能分配资源给它吗？
答：
(1)此时可以找出进程安全序列：P4，P1，P5，P2，P3。故系统处于安全状态。

$$Available=(1,0,2,0)$$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}1&1&0&0\\0&1&1&2\\3&1&0&0\\0&0&1&0\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\1&1&0&1\\0&0&0&0\end{bmatrix}$$

$Available=(1,0,2,0)+(1,1,0,1)=(2,1,2,1)$执行P4后

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}1&1&0&0\\0&1&1&2\\3&1&0&0\\*&*&*&*\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\*&*&*&*\\0&0&0&0\end{bmatrix}$$

$Available=(2,1,2,1)+(3,0,1,1)=(5,1,3,2)$执行P1后

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\0&1&1&2\\3&1&0&0\\*&*&*&*\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\0&1&0&0\\1&1&1&0\\*&*&*&*\\0&0&0&0\end{bmatrix}$$

$Available=(5,1,3,2)+(0,0,0,0)=(5,1,3,2)$执行P5后

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\0&1&1&2\\3&1&0&0\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\0&1&0&0\\1&1&1&0\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

$Available=(5,1,3,2)+(0,1,0,0)=(5,2,3,2)$执行P2后

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\*&*&*&*\\3&1&0&0\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\*&*&*&*\\1&1&1&0\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

$Available=(5,2,3,2)+(1,1,1,0)=(6,3,4,2)$执行P3后

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

(2)可以分配，存在安全序列：P4，P1，P5，P2，P3。

$Available = (1,0,2,0)$

$$Need = \begin{matrix} \text{P1} \\ \text{P2} \\ \text{P3} \\ \text{P4} \\ \text{P5} \end{matrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad Allocation = \begin{bmatrix} 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

假设执行request2(0,0,1,0)则相当于 $Available = (1,0,1,0)$

$$Need = \begin{matrix} \text{P1} \\ \text{P2} \\ \text{P3} \\ \text{P4} \\ \text{P5} \end{matrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 2 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad Allocation = \begin{bmatrix} 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

P4可以执行。$Available = (1,0,1,0) + (1,1,0,1) = (2,1,1,1)$

$$Need = \begin{matrix} \text{P1} \\ \text{P2} \\ \text{P3} \\ \text{P4} \\ \text{P5} \end{matrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 2 \\ 3 & 1 & 0 & 0 \\ * & * & * & * \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad Allocation = \begin{bmatrix} 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ * & * & * & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

P1可以执行。$Available=(2,1,1,1)+(3,0,1,1)=(5,1,2,2)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\0&1&2&2\\3&1&0&0\\*&*&*&*\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\0&1&0&0\\1&1&1&0\\*&*&*&*\\0&0&0&0\end{bmatrix}$$

P5可以执行。$Available=(5,1,2,2)+(0,0,0,0)=(5,1,2,2)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\0&1&2&2\\3&1&0&0\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\0&1&0&0\\1&1&1&0\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

P2可以执行。$Available=(5,1,2,2)+(0,1,0,0)=(5,2,2,2)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\*&*&*&*\\3&1&0&0\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\*&*&*&*\\1&1&1&0\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

P3可以执行。$Available=(5,2,2,2)+(1,1,1,0)=(6,3,3,2)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}*&*&*&*\\*&*&*&*\\3&1&0&0\\*&*&*&*\\*&*&*&*\end{bmatrix}\quad Allocation=\begin{bmatrix}*&*&*&*\\*&*&*&*\\1&1&1&0\\*&*&*&*\\*&*&*&*\end{bmatrix}$$

(3)不可分配，系统进入不安全状态。[1]

[1] 本题应当理解为在执行了 request2()之后再次提出资源请求 request5()的条件判断。如果将第三问和第二问单独分析，则存在以 P4 为首的序列满足第三问的需求。

$Available=(1,0,2,0)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}1&1&0&0\\0&1&1&2\\3&1&0&0\\0&0&1&0\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\1&1&0&1\\0&0&0&0\end{bmatrix}$$

假设执行request2(0,0,1,0)则相当于$Available=(1,0,1,0)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}1&1&0&0\\0&1&2&2\\3&1&0&0\\0&0&1&0\\2&1&1&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\1&1&0&1\\0&0&0&0\end{bmatrix}$$

再次提出request5(0,0,1,0)则相当于$Available=(1,0,0,0)$

$$Need=\begin{matrix}P1\\P2\\P3\\P4\\P5\end{matrix}\begin{bmatrix}1&1&0&0\\0&1&2&2\\3&1&0&0\\0&0&1&0\\2&1&2&0\end{bmatrix}\quad Allocation=\begin{bmatrix}3&0&1&1\\0&1&0&0\\1&1&1&0\\1&1&0&1\\0&0&0&0\end{bmatrix}$$

此时*Available*不能满足P1~P5任一进程的执行条件，出现了不安全状态。

习题四

一．思考题

3．什么是逻辑地址（空间）和物理地址（空间）（教材 306~307 页）?

用户目标程序使用的地址单元称为逻辑地址（相对地址），一个用户作业的目标程序的逻辑地址稽核称为该作业的逻辑地址空间。
主存中的实际存储单元称为物理地址（绝对地址），物理地址的总体相应构成了用户程序实际运行的物理地址空间。

4．何谓地址转换（重定位）?有哪些方法可以实现地址转换（教材 307 页）?

为了保证程序的正确运行，必须把程序和数据的逻辑地址转换为物理地址，这一工作称为地址转换或重定位。
地址转换有两种方式，一种方式是在作业装入时由作业装入程序（装配程序）实现地址转换，称为静态重定位；这种方式要求目标程序使用相对地址，地址变换在作业执行前一次完成；
另一种方式是在程序执行过程中，CPU 访问程序和数据之前实现地址转换，称为动态重定位。

5．分区存储管理中常用哪些分配策略？比较它们的优缺点。

常用的分配策略有两种：固定分区存储管理及可变分区存储管理。
固定分区存储管理
优点：
①预先将主存分割成若干个连续区域，分割时各区在主存分配表中可按地址顺序排列。其主存分配算法十分简单。
②解决单道程序运行在并发环境下不能与 CPU 速度很好匹配的问题。
③解决了单道程序运行主存空间利用率低的问题。
缺点：
①预先规定了分区大小，使得大程序无法装入，用户不得不采用覆盖等技术补救，不但加重用户负担，而且极不方便。
②主存空间的利用率不高，往往一个作业不可能恰好填满分区。
③因为分区的数目是在系统初启时确定的，限制了多道运行的程序数。
可变分区存储管理
优点：
①克服固定分区方式中的主存空间的浪费，进一步提高了主存资源利用率。
②有利于多道程序设计。
③实现了多个作业对主存的共享。
缺点：
①回收算法复杂。
②各种分配算法都有一定的缺陷，难以避免内存碎片的产生。
③采用动态重定位装入作业，作业程序和数据的地址转换需要专门硬件寄存器的支持。

13．试比较分页式存储管理和分段式存储管理。

段式	页式
分段由用户设计划分，每段对应一个相应的的程序模块，有完整的逻辑意义	分页用户看不见，由操作系统为内存管理划分
段面是信息的逻辑单位 便于段的共享，执行时按需动态链接装入。 段长不等，可动态增长，有利于新数据增长。 二维地址空间：段名、段中地址；段号、段内单元号	页面是信息的物理单位 页一般不能共享 页面大小相同，位置不能动态增长。 一维地址空间
管理形式上象页式，但概念不同	往往需要多次缺页中断才能把所需信息完整地调入内存

实现页（段）的共享是指某些作业的逻辑页号（段号）对应同一物理页号（内存中该段的起始地址）。页（段）的保护往往需要对共享的页面（段）加上某种访问权限的限制，如不能修改等；或设置地址越界检查，对于页内地址（段内地址）大于页长（段长）的存取，产生保护中断。

二． 问答题

3．一个页式存储管理系统使用 FIFO、OPT 和 LRU 页面替换算法，如果一个作业的页面走向为：

(1) 2、3、2、1、5、2、4、5、3、2、5、2。

(2) 4、3、2、1、4、3、5、4、3、2、1、5。

(3)1、2、3、4、1、2、5、1、2、3、4、5。

当分配给该作业的物理块数分别为 3 和 4 时，试计算访问过程中发生的缺页中断次数和缺页中断率[2]。

（1）物理块数=3

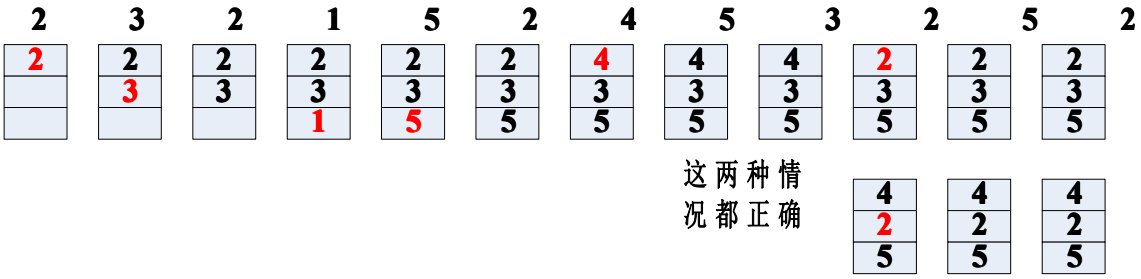
FIFO

2	3	2	1	5	2	4	5	3	2	5	2
2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	3	2	2	2	2	2	5	5
			1	1	1	4	4	4	4	4	2

缺页中断次数为 9 次，缺页中断率为 9/12 = 75.0%

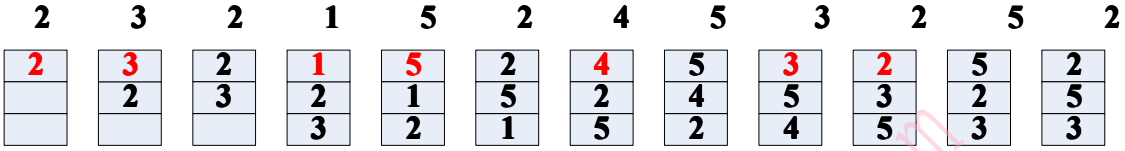
OPT

[2] 注意：刚开始的物理内存为空，此时每调入一个新页也要作为一次缺页中断进行计数。



缺页次数为 6 次，缺页中断率为 6/12 = 50.0%

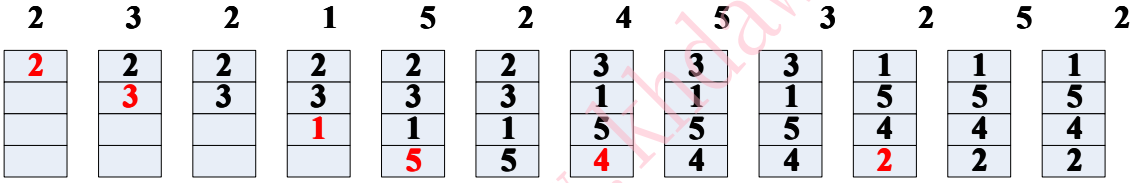
LRU



缺页次数为 7 次，缺页中断率为 7/12 = 58.3%

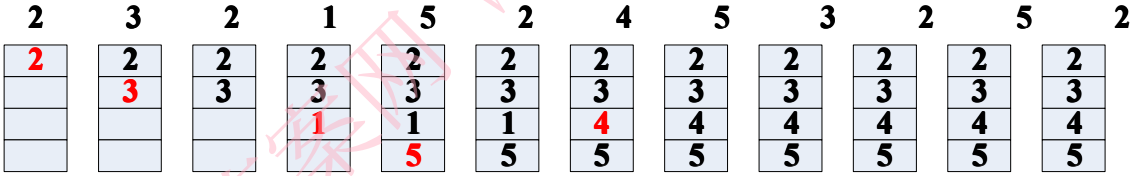
物理块数=4

FIFO



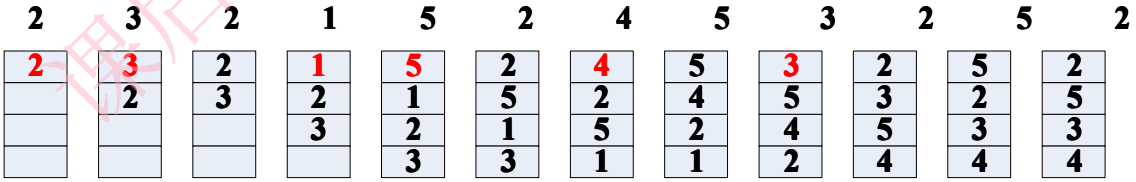
缺页次数为 6 次，缺页中断率为 6/12 = 50.0%

OPT



缺页次数为 5 次，缺页中断率为 5/12 = 41.7%

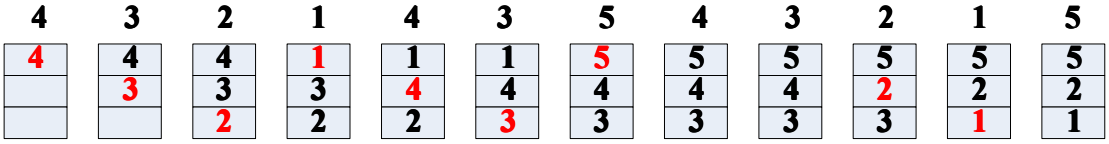
LRU



缺页次数为 6 次，缺页中断率为 6/12 = 50.0%

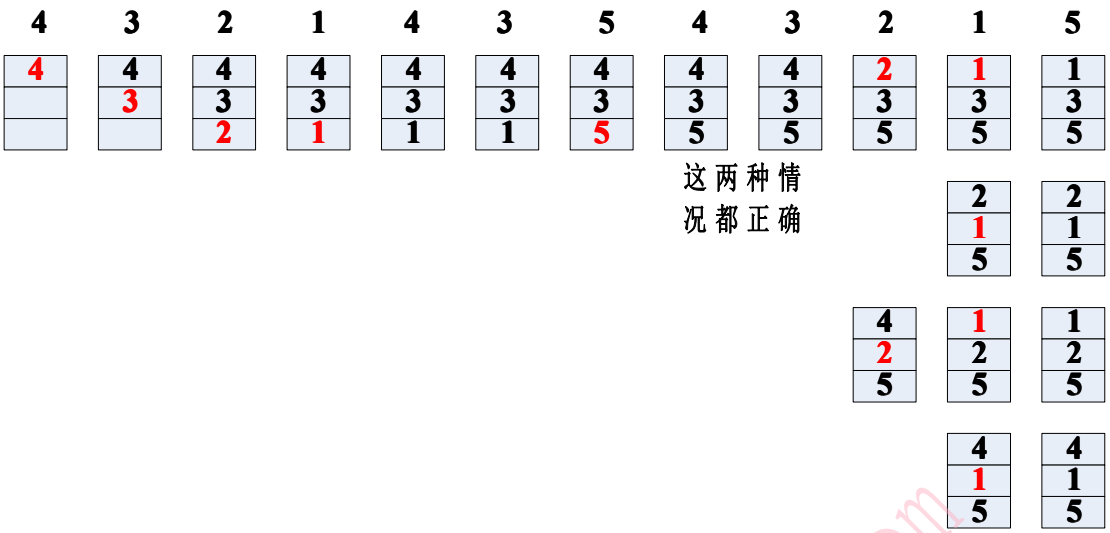
（2）物理块数=3

FIFO



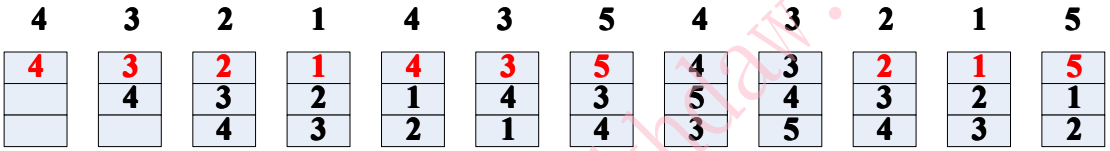
缺页次数为 9 次，缺页中断率为 9/12 = 75.0%

OPT



发生 7 次缺页中断，缺页中断率为 7%12 = 58.3%

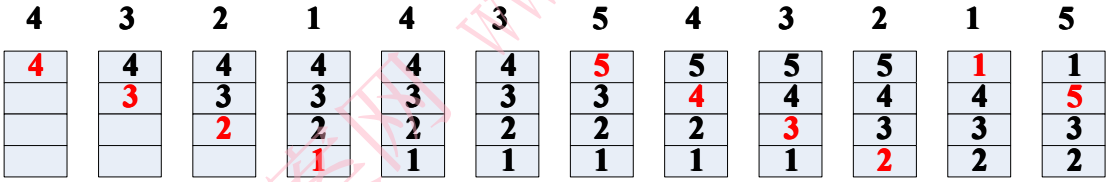
LRU



发生 10 次缺页中断，缺页中断率为 10/12 = 83.3%

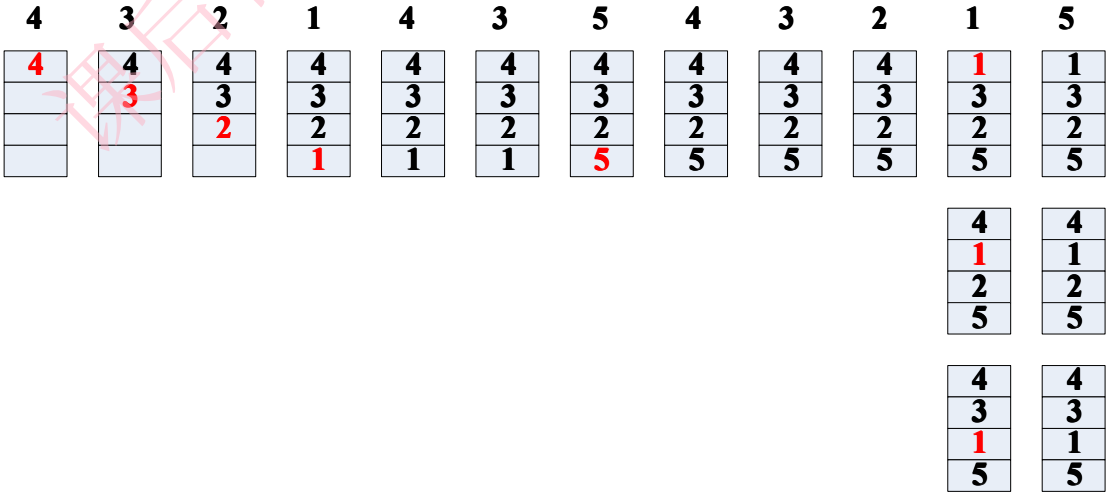
物理块数=4

FIFO



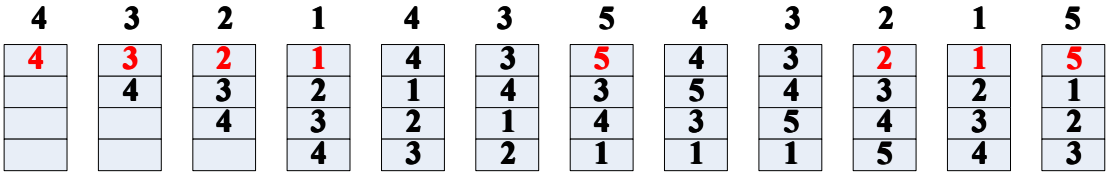
发生 10 次缺页中断，缺页中断率为 10/12 = 83.3%

OPT



发生 6 次缺页中断，缺页中断率为 6/12 = 50.0%

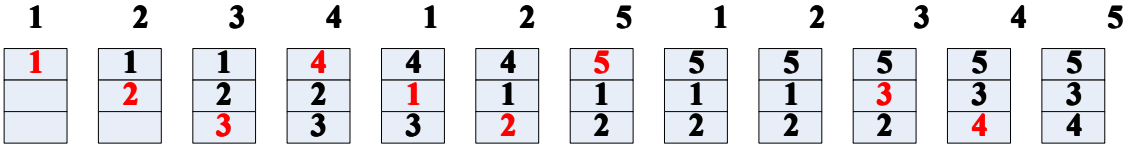
LRU



发生 8 次缺页中断，缺页中断率为 8/12 = 66.7%

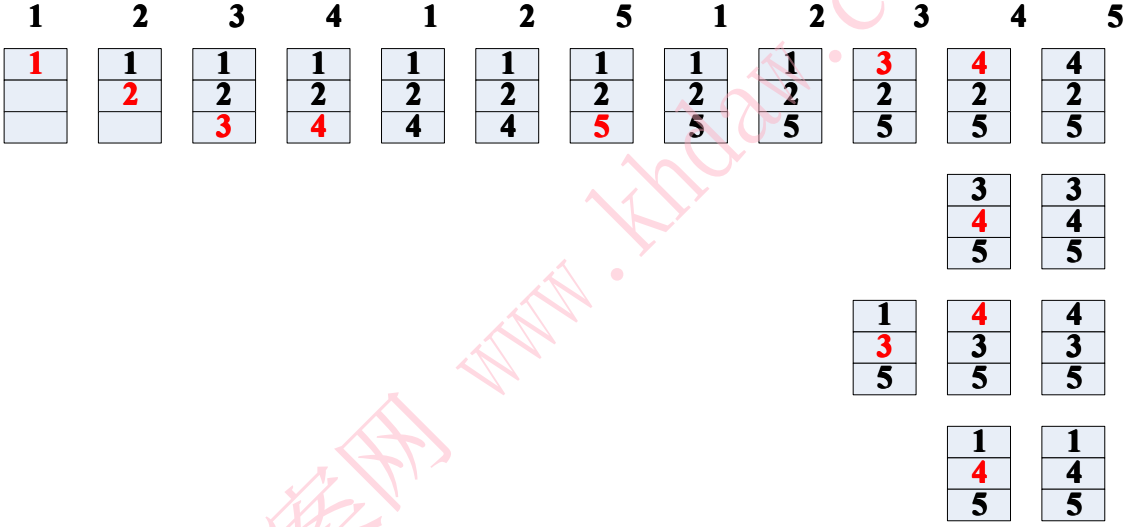
（3）物理块数 = 3

FIFO



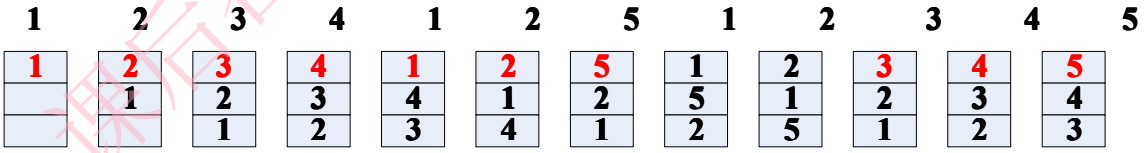
发生 9 次缺页中断，缺页中断率为 9/12 = 75.0%

OPT



发生 7 次缺页中断，缺页中断率为 7/12 = 58.3%

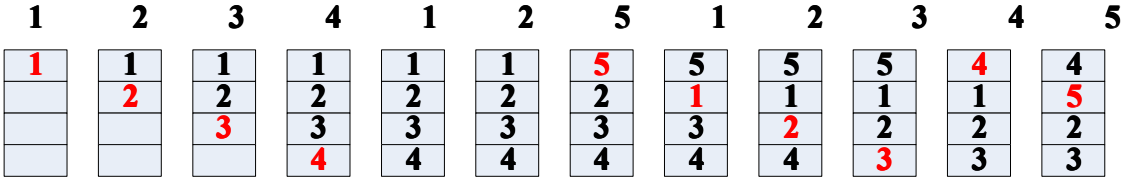
LRU



发生 10 次缺页中断，缺页中断率为 10/12 = 83.3%

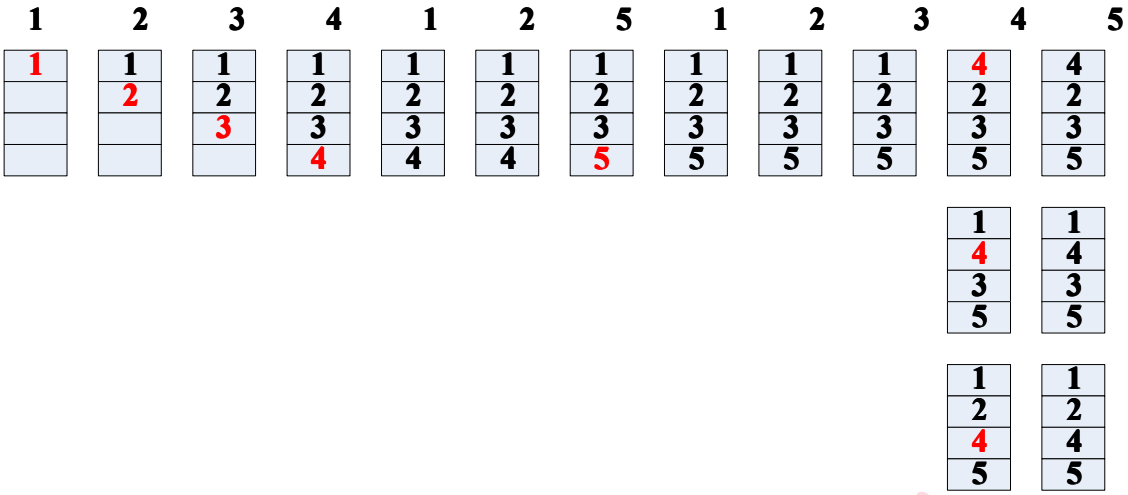
物理块数= 4

FIFO



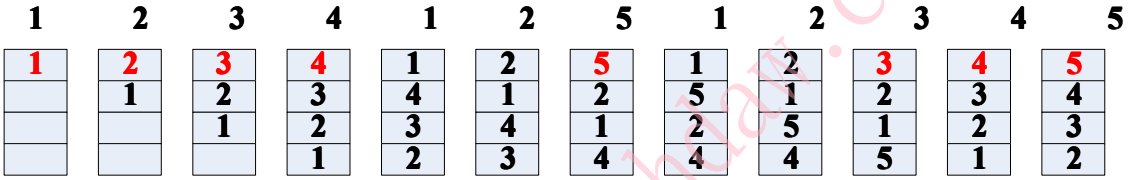
发生 10 次缺页中断，缺页中断率为 10/12 = 83.3%

OPT



发生 6 次缺页中断，缺页中断率为 6/12 = 50.0%

LRU



发生 8 次缺页中断，缺页中断率为 8/12 = 66.7%

答案汇总[3]：

	FIFO				OPT				LRU			
物理块数	3		4		3		4		3		4	
序列（1）	9	75.0%	6	50.0%	6	50.0%	5	41.7%	7	58.3%	6	50.0%
序列（2）	9	75.0%	10	83.3%	7	58.3%	6	50.0%	10	83.3%	8	66.7%
序列（3）	9	75.0%	10	83.3%	7	58.3%	6	50.0%	10	83.3%	8	66.7%

11．给定段表如下：

段　　号	段　首　址	段　　长
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

给定地址为段号和位移：1）[0，430]、2）[3，400]、3）[1，1]、4）[2，500]、5）[4，42]，试求出对应的内存物理地址。

（1）[0，400]　∵偏移量[400]<0 段段长[600]　∴0 段首址[219]+偏移量[400]=649

（2）[3，400]　∵偏移量[400]<3 段段长[580]　∴3 段首址[1327]+偏移量[400]=1727

[3] 注意：这里的答案格式为：　发生中断的次数　缺页中断率。

（3）[1，1]　∵偏移量[1]<1 段段长[14] ∴1 段首址[2300]+偏移量[1]=2301
（4）[2，500] ∵偏移量[500]>2 段段长[100] ∴地址越界
（5）[4，42]　∵偏移量[42]<4 段段长[96] ∴4 段首址[1952]+偏移量[42]=1994

14．设有一页式存储管理系统，向用户提供的逻辑地址空间最大为 16 页，每页 2048 字节，内存总共有 8 个存储块。试问逻辑地址至少应为多少位？内存空间有多大？

逻辑地址 2^{11} ×2^4 ，故为 15 位。内存大小为 $2^3\times2^{11}=2^{14}$B=16KB。

20．在一个分页虚存系统中，用户编程空间 32 个页，页长 1KB，主存为 16KB。如果用户程序有 10 页长，若已知虚页 0、1、2、3，已分到页框 8、7、4、10 ，试把虚地址 0AC5H 和 1AC5H 转换成对应的物理地址。

虚地址 0AC5H = （2757）$_{10}$=1024*2 （页号）+ 709（段内偏移量）
映射到物理页框第 4 页。
对应的物理地址为　4*1024 + 709 = （4805）$_{10}$=12C5H
虚地址 1AC5H=（6853）$_{10}$ = 1204*6（页号） + 709（段内偏移量）
页表中尚未有分配的页框，此时引发缺页中断，由系统另行分配页框。

习题五

一．思考题

2．简述各种 I/O 控制方式及其主要优缺点。

（1）询问方式。 又称程序直接控制方式。I/O 指令或询问指令测试一台设备的忙闲标志位，决定主存储器和外围设备是否交换一个字符或一个字。
优点：原理比较简单，实现时无需增加额外的硬件设备，成本较低。
缺点： ①一旦 CPU 启动 I/O 设备，便不断查询 I/O 的准备情况，终止了原程序的执行。
②CPU 在反复查询过程中，浪费了宝贵的 CPU 时间。
③I/O 准备就绪后，CPU 参与数据的传输工作，此时 CPU 也不能执行原程序。
总之，询问方式的主要缺点是运行效率不高。
（2）中断方式。CPU 启动 I/O 设备后，不必查询 I/O 设备是否就绪，而是继续执行现行程序，对设备是否就绪不加过问。
优点：不必忙式查询 I/O 准备情况，CPU 和 I/O 设备可实现部分并行，提高了 CPU 的利用率。
缺点：输入输出操作直接由中央处理器控制，每传送一个字符或一个字，都要发生一次中断，仍耗费大量中央处理器时间。
（3）DMA 方式。主存和 I/O 设备之间有一条数据通路，在主存和 I/O 设备之间成块地传送数据过程中，无需 CPU 干预，实际操作由 DMA 直接执行完成。
优点：线路比较简单，价格并不昂贵。
缺点：增加主存地址寄存器、数据移位寄存器等硬件逻辑，不仅有中断结构，还增加了 DMA 传输控制机构。增加了制造成本，但功能较差，不能满足复杂 I/O 要求。
（4）通道方式。通道能完成主存储器和外围设备之间的信息传送，与中央处理器并行地执行操作。
优点： ①自成独立体系，大大减少了外围设备和中央处理器的逻辑联系。把中央处理器从琐碎的输入输出操作中解放出来。
②外围和中央处理器能实现并行操作。
③通道和通道之间能实现并行操作。
④各通道上的外围设备也能实现并行操作。
提高整个系统的效率。
缺点： ①具有通道装置的计算机的主机、通道、控制器和设备之间采用四级连接，实施三级控制。设计技术比较复杂。
②价格较高，一般在大型机中使用。

7．叙述 I/O 系统的层次及其功能。

I/O 系统从底层开始分别是硬件、中断处理程序、设备驱动程序、设备无关软件、最上面是用户进程。
硬件：执行 I/O 操作。
中断处理程序：当 I/O 结束时，唤醒驱动程序。

设备驱动程序：置设备寄存器：检查状态。
设备无关软件：命名；保护；阻塞；缓冲；分配。
用户进程：进行 I/O 调用；格式化 I/O；假脱机。

12．为什么要引入缓冲技术？其实现的基本思想是什么？

引入缓冲技术的理由：

①改善中央处理器与外围设备之间速度不匹配的矛盾。

②协调逻辑记录大小与物理记录大小不一致的问题。

③提高 CPU 和 I/O 设备的并行性。

④减少 I/O 对 CPU 的中断次数和放宽对 CPU 中断响应时间的要求。

缓冲技术实现的基本思想：

当一个进程执行写操作输出数据时，先向系统申请一个输出缓冲区，将数据高速送到缓冲区。若为顺序写请求，则不断把数据填到缓冲区，直到它被装满为止。此后，进程可以继续它的计算，同时，系统将缓冲区内容写到 I/O 设备上。

当一个进程执行读操作输入数据时，先向系统申请一个输入缓冲区，系统将一个物理记录的内容读到缓冲区中，根据进程要求，把当前需要的逻辑记录从缓冲区中选出并传送给进程。

26．Spooling 如何把独占设备改造成共享设备的？

SPOOLing 是 Simultaneous Peripheral Operation On-Line（即外部设备联机并行操作）的缩写，它是关于慢速字符设备如何与计算机主机交换信息的一种技术，也称为假脱机技术。

SPOOLing 系统既不同于脱机方式，也不同于直接耦合方式。它在输入和输出之间增加了“输入井”和“输出井”的排队转储环节，以消除用户的“联机”等待时间。在系统输入模块收到作业输入请求信号后，输入管理模块中的读过程负责将信息从输入装置中读入输入井缓冲区。当缓冲区满时，由写过程将信息从缓冲区写到外存的输入井中，读过程和写过程反复循环，直到一个作业输入完毕。当读过程读到一个硬件结束标志之后，系统再次驱动写过程把最后一批信息写入外存输入井并调用中断处理程序结束该次输入。然后，系统为该作业建立作业控制块，从而使输入井中的作业进入作业等待队列，等待作业调度程序选中后进入内存运行。系统在管理输入井过程中可以“不断”读入输入的作业，直到输入结束或输入井满而暂停。SPOOLing 系统并没有为任何进程分配，而只是在输入井和输出井中为进程分配一存储区和建立一张 I/O 请求表。这样便把独占设备改造为共享设备。

二．问答题

7．假定磁盘有 200 个柱面，编号 0~199，当前存取臂的位置在 143 号柱面上，并刚刚完成了 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：为完成上述请求，下列算法存取臂移动的总量是多少？并算出存取臂移动的顺序。

（1）先来先服务算法 FCFS。　（2）最短查找时间优先算法 SSTF。

（3）扫描算法 SCAN。　（4）电梯调度。

由于当前存取臂的位置在 143 号柱面上，并刚刚完成 125 号柱面的服务请求，所以其存取臂的方向如图所示。

(1)先来先服务算法

移动次序依次为 143→86→147→91→177→94→150→102→175→130。

存取臂移动次数为

|86-143| + |147-86| + |91-147| + |177-91| + |94-177| + |150-94| + |102-150| + |175-102| + |130-175|

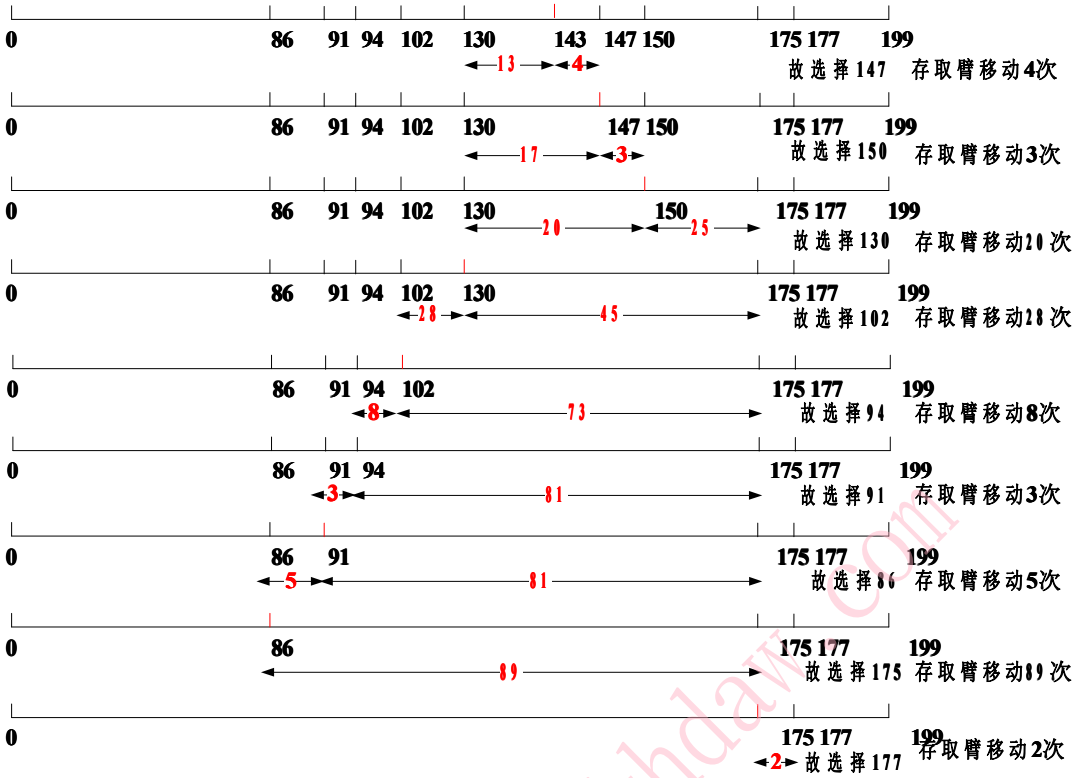
= 57 + 61 + 56 + 86 + 83 + 56 + 48 + 73 + 45

= 565（次）

（2）最短查找时间优先：总是先执行查找时间最短的那个磁盘请求。

移动次序依次为：

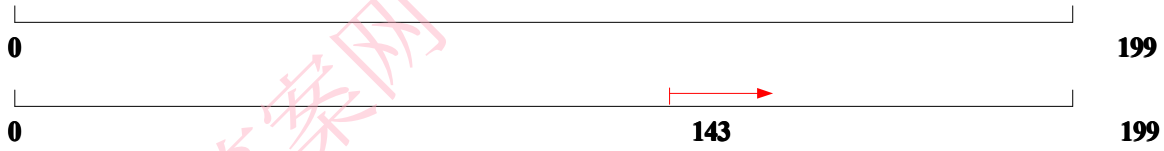
143→147→150→130→102→94→91→86→175→17



存取臂移动次数为

4 + 3 + 20 + 28 + 8 + 3 + 5 + 89 + 2 = 162（次）

（3）扫描算法 SCAN：磁盘臂每次沿一个方向移动，扫过所有的柱面，遇到最近的 I/O 请求便进行处理，直到最后一个柱面后，再向相反的方向移动回来。



移动次序依次为：143→147→150→175→177→199→130→102→94→91→86。

存取臂移动次数为|199-143| + |86-199| = 56 + 113 = 169（次）。

（4）电梯调度算法：每次总是选择沿臂的移动方向最近的那个柱面，如果同一柱面上有多个请求，还需进行旋转优化。

移动次序依次为：143→147→150→175→177→130→102→94→91→86。

存取臂移动次数为：|143-177| + |177-86| = 34 + 91 = 125（次）[4]

算法	移动次序	存取臂移动次数
FCFS	143→86→147→91→177→94→150→102→175→130	565 次
SSTF	143→147→150→130→102→94→91→86→175→17	162 次
SCAN	143→147→150→175→177→199→130→102→94→91→86	169 次
电梯调度	143→147→150→175→177→130→102→94→91→86	125 次

习题六

一．思考题

5．什么是文件的物理结构？它有哪几种组织方式？

文件的物理结构和组织是指逻辑文件在物理存储空间中的存放方法和组织关系。
组织方式
（1)顺序文件 将文件中逻辑上连续的信息存放到存储介质的依次向另的块中便形成顺序结构，这类文件叫顺序文件，又称连续文件。
（2）连接文件 使用指针来表示文件中各个记录之间的关系，文件信息存放在外存的若干个物理块中，第一块文件信息的物理地址由文件目录给出，而每一块的指针指出了文件的下一个物理块位置。通常，指针内容为 0 时，表示文件至本块结束。
（3）直接文件 在直接存取存储设备上，利用 hash 法把记录的关键字与其它地址之间建立某种对应关系，以便实现快速存取的文件叫直接文件或散列文件。
（4）索引文件 系统为每个文件建立了一张索引表，其中，每个表目包含一个记录的键（或逻辑记录号）及其记录数据的存储地址，存储地址可以是记录的物理地址，也可是记录的符号地址，这种类型的文件称索引文件。索引表的地址可由文件目录指出，查阅索引表先找到的是相应记录键（或逻辑记录号），然后，获得数据存储地址。

6．叙述各种文件物理组织方式的主要优缺点。

（1）顺序文件
优点：顺序存取记录时速度较快。批处理、系统文件用的最多。
缺点：建立文件前需要能预先确定文件长度，以便分配存储空间；修改、插入和增加文件记录有困难；对直接存储器做连续分配，会造成空现块的浪费。
（2）连接文件
优点：可以将文件的逻辑记录顺序与它所在存储空间的物理记录顺序完全独立开来、存放信息的物理块不必连续而借助于指针表达记录之间的逻辑关系；克服了顺序结构不适宜于增、删、改的缺点。
缺点：必须将指针与数据信息存放在一起，破坏了物理块的完整性；仅适用于顺序存储；整体性能较低。
（3）直接文件
优点：可用在不能采用顺序组织方法、次序较乱、又需在极短时间内存取的场合，对于实时处理文件、操作系统目录文件、存储管理的页表查找、编译程序变量名表等特别有效。
缺点：冲突问题，如何设计 Hash 函数使得冲突尽可能少发生。
（4）索引文件
优点：具备连接文件的优点；具有直接读写任意一个记录的能力；便于文件的增、删、改。
缺点：增加了索引表的空间开销和查找时间，大型文件的索引表的信息量甚至可能远远超过文件记录本身的信息量。

二．问答题

7．一个 UNIX 文件 F 的存取权限为：rwxr-x---,该文件的文件主 uid＝12，gid=1,另一个用户的 uid=6，gid=1，是否允许该用户执行文件 F？

F 的存取权限为：rwxr-x---,表示文件主可对 F 进行读、写及执行操作，同组用户可对 F 进行读及执行操作，但其他用户不能对 F 操作。因为另一用户的组标识符 gid 相同，故而允许该用户执行文件 F。

9．一个 UNIX/Linux 文件，如果一个盘块的大小为 1KB，每个盘块占 4 个字节，那么，若进程欲访问偏移为 263168 字节处的数据，需经过几次间接？

UNIX/Linux 文件系统中，直接寻址为 10 块，一次间接寻址为 256 块，二次间接寻址为 256^2 块，三次间接寻址为 256^3 块。
偏移为 263168 字节的逻辑块号是：263168/1024=257。块内偏移量=263168-257×1024=0。由于 10<257<256+10，故 263168 字节在一次间接寻址内。

16．如果一个索引节点为 128B，指针长 4B，状态信息占用 68B，而每块大小为 8KB，问在索引节点中有多大空间给指针？使用直接、一次间接、二次间接和三次间接指针分别可表示多大的文件？

由于索引节点为 128B，而状态信息占用 68B，故索引节点中用于磁盘指针的空间大小为：128-68=60 字节。
一次间接、二次间接和三次间接指针占用三个指针项，因而直接指针项数为：60/4-3=12 个。每块大小为 8KB。所以，直接指针时：12×8192=98304B。
一次间接指针时：8192/4=2048，即一个磁盘块可装 2048 个盘块指针，2048×8192=16MB。
二次间接指针时：2048×2048=4M，即二次间接可装 4M 个盘块指针，4M×8192=32GB。
三次间接指针时：2048×2048×2048=8G，即三次间接可装 8G 个盘块指针，[8G×8192=16TB][5]。

[5]应该是 8G×8K = 64G