操作系统作业

**第一章**

* 1. 什么是计算机系统？计算机系统是怎样构成的？了解PC二第组成情况，说明：（1）硬件组织的基本结构；（2）主要系统软件和应用软件（若有的话）及他们的作用。  
     答：计算机系统就是按人的要求接收和存储信息，自动进行数据处理和计算，并输出结果信息的机器系统。  
     计算机系统由硬件（子）系统和软件（子）系统组成，前者是系统赖以工作的实体，后者是和各种程序和文件，用以指挥全系统按指定的要求进行工作。  
     硬件结构主要包括：1）运算器，2）主存储器，3）控制器，4）输入设备，5）输出设备，6）辅助存储器，7）总线，8）转换设备，9）输入/输出控制系统，10）电源和场地设备。  
     软件主要包括：系统软件，支撑软件和应用软件。
  2. 什么是操作系统？请举例说明操作系统在计算机系统中的重要地位。  
     操作系统是计算机系统中的一个软件，是一些程序的集合，操作系统能有效的组织和管理计算机斯通中的硬件和软件资源，合理的组织计算机工作流程，控制程序的执行，并向用户提供各种服务功能，使得用户能够灵活方便有效的使用计算机，并使整个计算机系统能够高效的运行。  
     操作系统是硬件与其他软件系统之间的接口，任何计算机只有在操作系统的指挥控制下，各种计算机资源才能被合理的非陪给用户使用。
  3. 操作系统的基本特征是什么？说明他们之间的关系。  
     基本特征：1）并发性，2）共享性，3）随机性

操作系统要同时管理着多个程序的并发运行，同时这些程序会怎样运行对于操作系统来说也是随机的。

1-11. 现有以下应用计算机的场合，请为其选择适当的操作系统。  
 1）航空航天，核变研究；  
 2）国家统计局数据处理中心；

3）学校学生上机学习编程；

4）高炉炉温控制；

5）民航订票系统；

6）发送电子邮件（在两个地区之间）。

答：1）实时操作系统；

2）分布式操作系统；

3）个人计算机曹祖系统；

4）实时操作系统；

5）网络操作系统；

1. 网络操作系统

1-13. 什么是SPOOLing技术？它有什么作用？你认为未来先进的个人计算机会把SPOOLing技术作为一个关键特性吗？

答：（以下内容来自百度百科词条）SPOOLing技术是低速输入输出设备与主机交换的一种技术，通常也称为“假脱机真联机”，他的核心思想是以联机的方式得到脱机的效果。低速设备经通道和外设在主机内存的缓冲存储器与高速设备相联，该高速设备通常是辅存。为了存放从低速设备上输入的信息，或者存放将要输出到低速设备上的信息（来自内存），在辅存分别开辟一固定区域，叫“输出井”（对输出），或者“输入井”（对输入）。简单来说就是在内存中形成缓冲区，在高级设备形成输出井和输入井，传递的时候，从低速设备传入缓冲区，再传到高速设备的输入井，再从高速设备的输出井，传到缓冲区，再传到低速设备。  
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，通常称为“假脱机技术”。  
SPOOLing系统既不同于脱机方式，也不同于直接藕合方式,SPOOLing技术实际上是一种外围设备同时联机操作技术，又称为排队转储技术。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。若系统的某台台行式打印机采用了虚拟设备技术，那么若有进程要求对它打印输出时,SPOOLing系统并不是将这台打印机直接分配给进程，而是在共享设备（磁盘）上的输出,SPOOLing存储区中为其分配一块存储空间，进程的输出数据以文件形式表示的。各进程的数据输出文件形成了一个输出队列，由输出POOLing系统控制这台打印机进程，依次将队列中的输出文件实际打印输出。在SPOOLing系统中，实际上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章I/O请求表。这样，便把独占设备改造为共享设备。

我认为未来这个技术可能并不会在个人电脑上推行，现在的技术已经王权可以用CPU的高速寄存器所替代，而且有些硬件也是会由自己的嵌入式操作程序的。

1-14. 先了解以下UNIX系统的外壳程序（Shell），它是不是操作系统的一部分，为什么？

答：Shell是命令解释器，同时也是高级编程语言，它是操作系统的一部分，因为操作系统的涵盖功能里需要这一部分程序来协助操作系统来管理，同时也可以用它来扩充设备。

**第二章**

2-2.为了支持操作操作系统，现代处理器一般都提供哪两种工作状态，以隔离操作系统和普通程序？两种状态各有什么特点？

管态和目态。管态一般指操作系统管理程序运行的状态，具有较高的特权级别，又称为特权态、系统态；目态一般指用户程序时的状态，具有较低的特权级别，又称普通态、用户态。

2-3.什么是分级的存储体

容量、速度和成本三个目标不可能同时达到最优，要做权衡。存取速度快，每比特价格高，容量大，每比特价格越低。

解决方案：采用层次化的存储体系结构，当沿着层次下降时，每比特的价格将下降，容量将增大，速度将变慢，处理器的访问频率下降。

2-5.请简述程序局部性原理。这个原理在分级的存储体系结构中是怎样起作用的。

时间局部性，空间局部性。

作用：提高存储系统效能。

2-6.什么是存储保护？有哪些方法实现存储保护？

对主存中的信息加以严格的保护，使操作系统其他程序不被破坏，是操作系统及其他程序正确运行的必要条件之一。

方法：操作系统给每个进程分配一个存储区域，同时提供了界限地址寄存器和存储键两个存储保护机构。

2-10．中断的一般处理过程怎样的？若多个中断同时发生呢？

简单中断的处理过程：硬件设备产生一个中断🡪外理器结束当前执行的指令🡪处理器发送中断应答信号🡪处理器将PC和PSW压入栈🡪根据中断设置加载新的PC🡪中断处理程序处理剩余状态信息🡪中断处理程序处理中断🡪恢复被中断程序的上下文环境🡪恢复老的PSW和PC的值。

处理多个中断：方法一，当处理一个中断时禁止中断，此时系统将会对任何新发生的中断置之不理。在这期间大声的中断将保持挂起状态。当处理器再次允许中断时，这个新的中断信号会被处理器检测到，并作出处理。

方法二，中断按照优先度分级，允许级较高的中断打断优先级较低的中断处理过程。

2-13.常用的I/O控制技术有哪些？各有什么特点？

常用的I/O控制技术有：程序控制I/O技术，中断驱动I/O技术，DMA技术以及通道。

程序控制会耗费大量的时间轮询以获得这个信息。降低系统性能。

中断驱动中CPU速度较快，而外设速度很慢，如果中断处理器来回的在CPU与外设之间切换，效率不高还浪费CPU的时间。

DMA（直接存储器访问）：DMA自动管理数据的传送，这个过程完成后给处理器发一个中断，只在开始和结束时关注，提高了I／O的处理效能。会有总线竞争，处理器访问总线速度会变慢。

通道：使CPu从I/O处理中解脱出来，提高CPU与设备、设备与设备之间的并行度。

**第三章**

3.4 假设有三个作业，他们的进入时间及估计运行时间如下：

|  |  |  |
| --- | --- | --- |
| 作业号 | 进入时刻 | 估计运行时间（分钟） |
| 1 | 10：00 | 60 |
| 2 | 10：10 | 60 |
| 3 | 10：25 | 15 |

在单道批处理方式下，采用先来先服务算法和最短作业优先算法进行作业调度。请给出它们的调度顺序，并分别计算出作业平均周转时间和带权平均周转时间。请对计算结果进行解释。

答：先来先服务：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业号 | 进入时间 | 估计运行时间 | 开始时间 | 结束时间 | 周转时间 | 带权周转时间 |
| 1 | 10：00 | 60 | 10：00 | 11：00 | 60 | 60/60=1 |
| 2 | 10：10 | 60 | 11：00 | 12：00 | 110 | 110/60=11/6 |
| 3 | 10：25 | 15 | 12：00 | 12：15 | 110 | 110/15=22/3 |

调度顺序为：1,2,3

平均周转时间：T=(60+110+110)/3=280/3

带权平均周转时间：W=(1+11/6+22/3)/3=61/18

最短作业优先算法:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业号 | 进入时间 | 估计运行时间 | 开始时间 | 结束时间 | 周转时间 | 带权周转时间 |
| 1 | 10：00 | 60 | 10：00 | 11：00 | 60 | 60/60=1 |
| 2 | 10：10 | 60 | 11：15 | 12：15 | 125 | 125/60=25/12 |
| 3 | 10：25 | 15 | 11：00 | 11：15 | 50 | 50/15=10/3 |

调度顺序为:1,3,2

平均周转时间：T=(60+125+50)/3=235/3

带权平均周转时间：W=(1+25/12+10/3)/3= 77/36

3.5 有一个两道的批次处理操作系统，作业调度采用最短作业优先的调度算法，进程调度采用基于优先数的抢占式调度算法，有如下的作业序列：

|  |  |  |  |
| --- | --- | --- | --- |
| 作业 | 进入时间 | 估计运行时间（分钟） | 优先数 |
| JOB1 | 10：00 | 40 | 5 |
| JOB2 | 10：20 | 30 | 3 |
| JOB3 | 10：30 | 50 | 4 |
| JOB4 | 10：50 | 20 | 6 |

其中优先数数值越小优先级越高。

（1）列出所有作业进入内存时间及运行结束时间。

（2）计算作业平均周转时间和带权平均周转时间。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作业 | 进入时间 | 估计运行时间 | 运行时间段 | 周转时间 | 带权周转时间 |
| JOB1 | 10：00 | 40 | 10：00-10：20  10：50-11：10 | 70 | 70/40=7/4 |
| JOB2 | 10：20 | 30 | 10：20-10：50 | 30 | 30/30=1 |
| JOB3 | 10：30 | 50 | 11：10-12：00 | 90 | 90/50=9/5 |
| JOB4 | 10：50 | 20 | 12：00-12：20 | 90 | 90/20=9/2 |

平均周转时间：T=(70+30+90+90)/4=70

带权平均周转时间:W=(7/4+1+9/5+9/2)/4=181/80

3.6 某系统采用不能移动已在内存储器中作业的可变分区方式管理内存储器，现有供用户使用的内存空间100K，系统配有4台磁带机，有一批作业如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 作业 | 进入时间 | 估计运行时间（分钟） | 需要内存（K） | 需要磁带机（台） |
| JOB1 | 10：00 | 25 | 15 | 2 |
| JOB2 | 10：20 | 30 | 60 | 1 |
| JOB3 | 10：30 | 10 | 50 | 3 |
| JOB4 | 10：35 | 20 | 10 | 2 |
| JOB5 | 10：40 | 15 | 30 | 2 |

该系统采用多道程序设计技术，对磁带机采用静态分配，忽略设备工作时间和系统进行调度所共花的时间，请分别写出采用“先来先服务调度算法”和“最短作业优先算法”选中作业执行的次序以及作业平均周转时间。

若允许移动已在内存中的作业，则作业被选中的次序又是怎样的呢？计算出作业平均周转时间。

答：先来先服务：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作业号 | 进入时间 | 估计运行时间 | 开始时间 | 结束时间 | 周转时间 |
| 1 | 10：00 | 25 | 10：00 | 10：25 | 25 |
| 2 | 10：20 | 30 | 10：25 | 10：55 | 35 |
| 3 | 10：30 | 10 | 11：30 | 11：40 | 70 |
| 4 | 10：35 | 20 | 10：55 | 11：15 | 40 |
| 5 | 10：40 | 15 | 11：15 | 11：30 | 50 |

调度顺序为：1,2,4,5,3

（10：00作业1进入，剩余85k内存，2台磁带机；

10：20作业2进入，剩余25k内存，1台磁带机；

10：25作业1结束，剩余40k内存，3台磁带机；

10：30作业3到达，内存不够，无法执行；

10：35作业4进入，剩余30k内存，1台磁带机，此时3依旧无法进入；

10：40作业5到达，磁带机不够，无法进入；

10：55作业2结束，剩余90k内存，2台磁带机，作业5进入，3等待；

11：15作业4结束，剩余70k内存，2台磁带机，3等待；

11：30作业5结束，作业3进入；

11：40作业3结束）

平均周转时间：T=(25+35+70+40+50)/5=44

最短作业优先算法:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作业号 | 进入时间 | 估计运行时间 | 开始时间 | 结束时间 | 周转时间 |
| 1 | 10：00 | 25 | 10：00 | 10：25 | 25 |
| 2 | 10：20 | 30 | 10：25 | 10：55 | 35 |
| 3 | 10：30 | 10 | 11：30 | 11：40 | 70 |
| 4 | 10：35 | 20 | 11：10 | 11：30 | 55 |
| 5 | 10：40 | 15 | 10：55 | 11：10 | 30 |

调度顺序为：1,2,5,4,3

平均周转时间：T=(25+35+70+55+30)/5=43

3.7 给定一组作业J1,J2,…,Jn,其运行时间分别为T1,T2,…,Tn。假定这些作业是同时到达的，并且将在一台CPU上按单道批处理方式运行。试证明，若按最短作业优先算法运行这些作业，则作业平均周转时间最少。

答：不失一般性，假定按最短的作业优先调度算法，调度顺序分别为J1，J2..Jm，其运

行时间分别为：tl,t2...tn。则作业Ji的周转时间为：Ti=t1+t2+…+ti.所以，全部作业的平均周转时间为：T=[T1+T2+...+Tn]/n=[tl+(t1+t2)+(t1+t2+t3)+…+（tl+t2+…+tn)]/n,显然,当tl≤t2≤…≤tm时，每个T达到最小值（i=l,2.…n)。因此，T最小。

**第四章**

4.13 设有无穷多个信息，输入进程把信息逐个写入缓冲区，输出进程逐个地从缓冲区中取出信息。在下述两种情况下：缓冲区是环形的，最多可以容纳 n 个信息；缓冲区是无穷大的。试分别回答下列问题？

(1)输入、输出两进程读、写缓冲区需要什么条件？

(2)用 P,V 操作写出输入，输出两进程的同步算法，并给出信号量含义以及初值。

(3)指出信号量的值的变化范围和其值的含义。

答： 一：当缓冲区的大小为 n 时

（1）当缓冲区信息为空的时候，输出进程无法读，处于等待状态，当缓冲区信息为满的时候无法写，都某个缓冲区单位进行读写的时候，要互斥

（2）

1.空的信号量empty初值为n, 满的信号量为full初值为0, 对缓冲区单元的互斥信号量为mutex,j,k为缓冲区单位地址，初值为0

写进程 读进程

P(empty) P(full)

P(mutex) P(mutex)

向 Buffer[i]写入信息 从 Buffer[k]中读信息

V(mutex) V(mutex)

V(full) V(empty)

j:=(j+1)mod n k:=(k+1)mod n

(4)empty表示还有多少缓冲区单元为空，如果 empty=0,表示缓冲区满，系统调用写进程时，写进程处于等待态

full 表示缓冲区都多少有信心的单元，如果 full=0, 表示缓冲区空，系统调用写进程时，读进程处于等待态

mutex 表示对于缓冲区单元的互斥信号量,当mutex=1时，开锁，mutex=0时，闭锁

二．当缓冲区大小为无穷大时

（1）当缓冲区信息为空的时候，输出进程无法读，处于等待状态，当缓冲区信息为满的时候无法写，都某个缓冲区单位进行读写的时候，要互斥

（2）

1．空的信号量empty不用设, 满的信号量为full初值为0,对缓冲区单元的互斥信号量 为mutex,j,k为缓冲区单位地址，初值为0

写进程 读进程

P(full)

P(mutex) P(mutex)

向 Buffer[i]写入信息 从 Buffer[k]中读信息

V(mutex) V(mutex)

V(full)

j:=(j+1)mod n k:=(k+1)mod n

(4)full表示缓冲区都多少有信心的单元，如果 full=0, 表示缓冲区空，系统调用写进程时，读进程处于等待态

mutex表示对于缓冲区单元的互斥信号量,当mutex=1时，开锁，mutex=0时，闭锁

4.14假定一个阅览室最多可以容纳 100 人，读者进入和离开阅览室都必须在阅览室门口的一个登记表上进行登记（进入时登记，离开时去掉登记项）而且每次只允许一人登记或者去掉登记，问：

(1) 应编写几个进程完成这项工作，程序的主要动作是些什么？应该设置几个进程？进程和程序间的关系如何？

(2) 用 P,V 操作写出这些进程的同步通信关系。

答：编写两个进程，一个处理读者进入，一个处理读者离开，进程是程序的动态执行

设置信号量full为初值为0，空的信号量empty初值为100,互斥信号量mutex初值为1

进入 离开

P(empty) P(full)

P(mutex) P(mutex)

登记 取消登记

V(mutex) V(mutex)

V(full) V(empty)

进入 离开

4.18 爱睡觉的理发师问题，一个理发店有两间相连的屋子，一间是私室，里面有一把理发椅，另一个是等候室，有一个滑动门和 N 把椅子。理发师忙的时候，通向私室的门被关闭，新来的顾客找一把空椅子坐下，如果椅子都被占用了，则顾客只好离去，如果没有顾客，则理发师在理发椅上睡觉。并打开通向私室的门。理发师睡觉时，顾客可以叫醒他理发，请编写理发师和顾客的程序，正确实现同步和互斥问题。

解：VAR:

S1,S2 ：Semaphore;{初值为0,实现理发师与顾客的同步}

Mutex：Semaphore：{初值为1,实现对waiting的互斥}

waiting：INTEGER:{初值为0,等待的顾客数}

理发师进程

REPEAT

P(S1) {若无顾客，则睡觉 }

P(mutex)

Waiting:=waiting-1

V(S2); (唤醒一个等待的客户)

V(mutex)

理发

Until FALSE

顾客进程

P(mutex)

IF(waiting<N) (还有空的椅子)

THEN BEGIN

Waiting：=-waiting+1 ;(等待顾客数加 1)

V(mutex);

V(S1) {通知理发师}

P(S2) {若无理发师，挂起}

坐下理发

END

ELSE V(mutex)

4.20巴拿马运河建在太平洋和大西洋之间，由于太平洋和大西洋水面高度不同，有巨大落差，所以运河中建有 T(T≥2)级船闸，并且只能允许单向通行，船闸依次编号为 1,2,…,T，由大西洋来的船需要经过船闸 T,T-1.,,,2,1 通过运河到达太平洋，由太平洋来的船需要经由船闸 1,2,…,T-1,T 通过运河到达大西洋。

使用 P,V 操作正确解决大西洋和太平洋的船只通航问题。

答：Array: S1[T] of Semaphore {为每个船闸设置的信号量初值都为 1}

Array: count1[T] of INTEGER {为每个船闸设置通往大西洋的船的计数值,初值都为0}

Array: count2[T] of INTEGER {为每个船闸设置通往太平洋的船的计数值,初值都为0}

对count设置互斥信号量mutex

去大西洋的进程:

int j

for(j=0;j<T,j++){

P(mutex)

if(count1[j]==0)

P(S[j])

count[j]++

过第 j+1 个船闸

P(mutex)

count[j]--

if(count[j]==0)

V(S[j])

V(mutex)

}

去太平洋的进程：

int k

for(k=T-1;k≥T,k++){

P(mutex)

if(count2[k]==0)

P(S[k])

count[k]++

过第 k+1 个船闸

P(mutex)

count[k]--

if(count[k]==0)

V(S[k])

V(mutex)

}

4.22 设 A,B,C 三个进程共享一个存储资源F,A对F只读不写，B对F只写不读，C对F先读后写。（当一个进程写F时，其他进程既不能读F，也不能写F,但多个进程同时读F是允许的）试利用管程的方法或者P,V操作，写出A,B,C三个进程的框架，要求：（1）执行正确（2）正常运行时不产生死锁；(3)使用F的并发度高。

答：信号量和P、V操作。

var rmutex,wmutex:semaphore;  
 rmutex:=wmutex:=1;  
 count:integer;count:=0;  
cobegin  
{  
 process P1  
 begin  
 repeat  
 P(rmutex);   
 count:=count+1;  
 if count=1 then P(wmutex);  
 V(rmutex);  
 Read F;  
 P(rmutex);  
 count:=count-1;   
 if count=0 then V(wmutex);  
 V(rmutex);

P(rmutex);  
count:=count-1;  
if count=0 then V(wmutex);  
V(rmutex);  
untile false;

end

process P2  
begin  
 repeat  
 P(wmutex);  
 Write F;  
 V(wmutex);  
 untile false;

process P3  
begin  
 repeat  
 P(rmutex);  
 count:=count+1;  
 if count=1 then P(wmutex);  
 V(rmutex);  
 Read F;  
 P(rmutex);  
 count:=count-1;  
 if count=0 then V(wmutex);  
 V(rmutex);  
 P(wmutex);  
 Write F;  
 V(wmutex);  
 untile false;

End

}  
coend

4.24 试用P,V操作解决第二类读者写者问题。所谓第二类读者写者问题是指写者优先，条件为：

（1）多个读者可以同时进行读；

（2）写者必须互斥（只允许一个写者写，同时也不能写者、读者同时进行）；

（3）写者优先于读者（一旦有写者来，则后续读者必须等待，唤醒时优先考虑写者）。

答：设置两个信号量Wmutex、Rmutex和一个公共变量Rcount。其中：Wmutex表示“允许写”，初值是1；公共变量Rcount表示“正在读”的进程数，初值是0；Rmutex表示对Rcount的互斥操作，初值是1。

Writer Render

P(Rmutex);

if(Rcount=0)

P(Wmutex); P(Wmutex);

Write; ++Rcount;

V(Wmutex); V(Rmutex);

Read;

P(Rmutex);

--Rcount;

if(Rcount=0)

V(Wmutex);

V(Rmutex);

4.29 请用管程实现哲学家就餐问题。

答：对每个哲学家设置信号量。

test()函数有以下作用：   
a. 如果当前处理的哲学家处于饥饿状态且两侧哲学家不在吃饭状态，则当前哲学家通过   
test()函数试图进入吃饭状态。   
b. 如果通过test()进入吃饭状态不成功，那么当前哲学家就在该信号量阻塞等待，直到   
其他的哲学家进程通过test()将该哲学家的状态设置为EATING。   
c. 当一个哲学家进程调用put\_forks()放下筷子的时候，会通过test()测试它的邻居，   
如果邻居处于饥饿状态，且该邻居的邻居不在吃饭状态，则该邻居进入吃饭状态。    
伪码：   
#define N 5 /\* 哲学家人数\*/   
#define LEFT (i-1+N)%N /\* i的左邻号码 \*/   
#define RIGHT (i+1)%N /\* i的右邻号码 \*/   
typedef enum { THINKING, HUNGRY, EATING } phil\_state; /\*哲学家状态\*/   
monitor dp /\*管程\*/   
{   
phil\_state state[N];   
semaphore mutex =1;   
semaphore s[N]; /\*每个哲学家一个信号量，初始值为0\*/   
void test(int i)   
{   
if ( state[i] == HUNGRY &&state[LEFT(i)] != EATING && state[RIGHT(i)] != EATING )   
{   
state[i] = EATING;   
V(s[i]);   
}   
}   
void get\_forks(int i)   
{   
P(mutex);   
state[i] = HUNGRY;   
test(i); /\*试图得到两支筷子\*/   
V(mutex);   
P(s[i]); /\*得不到筷子则阻塞\*/   
}   
void put\_forks(int i)   
{   
P(mutex);   
state[i]= THINKING;   
test(LEFT(i)); /\*看左邻是否进餐\*/   
test(RIGHT(i)); /\*看右邻是否进餐\*/   
V(mutex);   
}   
}

4.35有5个批处理作业A到E几乎同时到达一计算中心。他们估计运行时间分别为10,6,2,4和8分钟，其优先数(由外部设定)分别为3，5，2，1，4其5级为最高优先级，对于下列每种调度算法，计算其平均周转时间，可忽略进程切换的开销。

（1） 时间片轮转法

（2） 优先级调度法

（3） 先来先服务法（按照次序 10,6,2,4,8 运行）

（4） 最短作业优先

答：(1) 和时间片的长短有关。

(2)运行顺序是(6,8,10,2,4) (6+14+24+26+30)/5=100/5=20分钟

(3)(10+16+18+22+30)/5=19.2分钟

(4) (2+(2+4)+(2+4+6)+(2+4+6+8)+( 2+4+6+8+10))/5=14分钟

# 习题五

4.用可变分区方式管理主存时,假定内存中按地址顺序依次有五个空闲区,空闲区的大小依次为32K、10k、5k、228k、100k。现有五个作业J1、J2、J3、J4和J5。它们各需内存1K、10K、108K、28K和115K。若采用最先适应分配算法能把这五个作业按J1~J5的次序全部装入内存吗？你认为按怎样的次序装入这五个作业可使内存空间利用率最高。

**答：（1）若采用最先适应分配法，无法将5个作业全部装入内存**

**（2）通过对最佳适应分配法和最差适应分配法的分析，其中最差适应分配法的内存空间利用率最高。**

10.有一个操作系统采用段式存储管理方案，用户区内存为512K，分配时截取空闲块的前半部分（小地址部分）。初始时内存全部空闲。系统执行如下申请、释放操作序列：

申请300K,申请100K，释放300K，申请150K，申请50K，申请90K

1. 若采用首先适应算法，空闲块表中有哪些空块（指出大小，地址）；
2. 若采用最佳适应算法，空闲块表中有哪些空块（指出大小，地址）；
3. 若随后又申请80K，针对上述两种情况说明结果？其结果说明了什么问题？

**答：（1）空闲块：0--90 , 200--300,400---512**

**空闲块: 0--90 , 150--300,450—512**

**（2）空闲块:80--90, 200--300,400---512**

**空闲块:80--90 , 150--300,450--512**

11.假如一个程序的段表如下：

段号 状态位 段起始地址 段长 存取控制

0 0 100 40 W

1 1 2010 20 W

2 0 1590 100 E

3 0 75 50 R

其中，状态位为“1”表示该段不在内存。存取控制：W表示可写，R表示可读，E表示可执行。对于以下的逻辑地址可能会发生什么情况：

1. STORE 1，[0，50]
2. STORE 1，[1，10]
3. LOAD 1，[2，77]
4. LOAD 1，[3，20]

答： **（1）逻辑地址为：150**

**（2）逻辑地址为：2020**

**（3）逻辑地址为：1667**

**（4）逻辑地址为：95**

26.有一个虚拟存储系统。分配给某进程3页内存，开始时内存为空，页面访问序列如下：

6，5，4，3，2，1，5，4，3，6，5，4，3，2，1，6，5

1. 若采用先进先出页面置换算法（FIFO），缺页次数为多少？
2. 若采用最近最少使用页面置换算法（LRU），缺页次数为多少？
3. 若采用最佳页面置换算法呢？

27．有一台计算机含有4个页面，每一页的装入时间，最后一次修改时间以及R与M位的值如下表（时间为时钟周期）：

页 装入时间 最后访问时间 R M

0 126 279 0 0

1 230 260 1 0

2 120 272 1 1

3 160 280 1 1

1. NRU应该淘汰哪一页？
2. FIFO应该淘汰哪一页？
3. LRU应该淘汰哪一页？
4. 第二次机会应该淘汰哪一页？

**答：（1）NRU应淘汰第0页**

**（2）FIFO应淘汰第2页**

**（3）LRU应淘汰第1页**

**（4）第二次机会应淘汰第0页**

31.有一个虚拟存储系统采用最近最少使用（LRU）页面置换算法，每个程序占3页内存，其中一页用来存放程序和变量i、j（不作它用）。每一页可存放150个整数变量。程序A和程序B如下：

程序A:

var C: array[1..150,1.. 100] of integer;

i,j:integer;

for i :=1 to 150 do

for j :=1 to 100 do

C[i,j]:=0;

程序B:

var C:array[1..150,1..100] of integer;

i,j;integer;

for j:=1 to 100 do

for i:=1 to 150 do

C[I,i] :=0;

设变量i、j放在程序页中，初始时，程序及变量i ,j已在内存，其余两页为空，矩防C按行序存放。

(1)试向当程序A和程序B执行完后,分别缺页多少次?

(2)最后留在内存中的各是矩阵C的哪一部分?

**答：（1）100次**

**（2）程序A运行完内存两个页面中分别为：**

**第一页：array[148,1]到array[148,100]和array[149,1]到array[149,50]**

**第二页：array[149,51] 到array[149,100]和array[150,1]到array[150,100]**

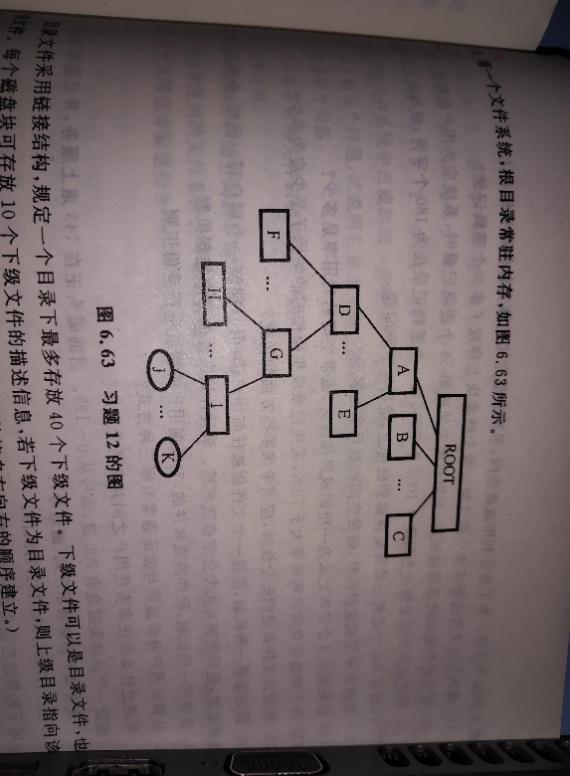
**程序B运行完内存两个页面中分别为：**

**第一页：array[148,1]到array[148,100]和array[149,1]到array[149,50]**

**第二页：array[149,51] 到array[149,100]和array[150,1]到array[150,100]**

# 习题六

12.有一个文件系统，根目录常往内存，如图所示。



目录文件采用链接结构，规定一个目录下最多存放40个下级文件。下级文件可以是目录文件，也可以是普通文件。每个磁盘块可存放10个下级文件的描述信息，若下级文件为目录文件，则上级目录指向该目录文件的第一块，否则指向普通文件的文件控制块。(假设文件按自左向右的顺序建立.)

(1)普通文件采用UNIX的三级索引结构，即文件控制块中给出13个磁盘地址，前10个磁盘地址指出文件前10块的物理地址,第11个磁盘地址指向一级索引表，一级索引表给出256个磁盘地址，即指出该文件第11块至第266块的物理地址;第12个磁盘地址指向二级索引表，二级索引表中指出256个一级索引表的地址；第13个磁盘地址指向三级索引表，三级索引表中指出256个二级索引表的地址。该文件系统中的普通文件最大可有多少块?假设主索引表放在FCB中,若要读文件\A\D\G\I\K中的某一块,最少要启动磁盘几次?最多要启动磁盘几次?若要减少启动磁盘的次数，可采用什么方法?

(2)普通文件采用链接结构。若要读\A\D\G\I\K的第75块，最少启动硬盘几次，最多几次?

20.假设某文件由100个逻辑记录组成，每个逻辑记录长度为80 个字符。磁盘空间被划分为若干块，块度为2048个字符。为了充分利用磁盘空间，采用成组方式把该文件保存在磁盘上。

(1)请回答该文件占用多少个磁盘块？每个磁盘块存放了多少个逻辑记录?

(2)若文件物理结构是链接结构，当用户要求处理第56个逻辑记录时，系统应为用户做哪些工作？

21.假定磁带的记录密度为每英寸1000个字符，每一个逻辑记录长为180个字符，块与块之间的间隙0.5英寸，现有800个逻辑记录需要存储到磁带上，请回答下列问题:

(1)在没有采用成组操作时，磁带空间的利用率是多少?

(2)在采用以8个逻辑记录为一组的成组操作时，磁带空间的利用率是多少?

(3)为了使磁带空间的利用率大于70%,采用记录成组操作时的块因子至少应为多少?

**答：（1）1000字符/英寸×0.5英寸=500个字符**

**采用不成组时操作的利用率：180/（180+50）=26.5%**

**（2）利用率：（180×8）/（500+180×8）=74.2%**

**（3）设记录成组块因子是Y时，磁带的利用率大于70%**

**则：180Y/（500+180Y）>=0.70 解得Y>=7**

**因此要使磁带空间的利用率>70%，采用记录成组时，其成组因子至少为7**

27. 假设一个活动头磁盘有200道，编号从0~199。当前磁头正在143道上服务，并且刚刚完成了125道的请求。现有如下访盘请求序列(磁道号):  
86，147，91，177,94, 150,102,175,130

试给出采用下列算法后磁头移动的顺序和移动总量(总磁道数)

（1）最短寻道时间优先(SSTF)磁盘调度算法。

（2）扫描法(SCAN)磁盘调度算法（假设沿磁头移动方向不再有访问请求时，磁头沿相反方向移动）。

**答：**

1. **移动顺序：147,150,130,102,94,91,86,175,130**

**移动总量：**

1. **移动顺序：147,177,94,91,86,75,63,22**

**移动总量：177—143+177—22=189**

28.假设磁盘的移动臂现在第8号柱面上，有6个访盘请求在等待，如下图所示。请给出最省时间的响应次序。

序号 柱面号 磁头号 扇区号

1. 9 6 3
2. 7 5 6
3. 15 20 6
4. 9 4 4
5. 20 9 5
6. 7 15 2

**答：最省时间相应次序是：2,6,4,1,3,5**

29.假定某磁盘的旋转速度是每圈20毫秒，格式化后每个盘面被分成10个扇区，现有10个逻辑记录存放在同一磁道上，安排如下所示。

扇区号 1 2 3 4 5 6 7 8 9 10

逻辑记录 A B C D E F G H I J

处理程序要顺序处理这些记录，每读出一个记录后处理程序要花54毫秒的时间进行处理，然后再顺序读下一个记录并处理，直到处理完这些记录，回答：

1. 顺序处理完这10个记录总共花费了多少时间？
2. 请给出一种记录有花分布的方案，使处理程序能在最短时间内处理完这10个记录，并计算优化分布时需要花费的时间。

**答：（1）磁盘旋转一个扇区所需时间=20/10=2ms**

**读出并处理第一条记录所需时间=2+4=6ms**

**处理完第一条记录磁头旋转到存放第四条逻辑记录所在的扇区了，需等到旋转到存放第二条逻辑记录的扇区时，才能读第二条逻辑记录，**

**读出并处理第二条记录所需时间=8\*2+2+4=22。**

**读出并处理其他几条记录所需与读出并处理第二条记录类似，故顺序处理10条记录所需时间=6+22\*9=204ms。**

**（2）一种记录优化分布的方案如图所示。**

**扇区号： 1 4 7 10 3 6 9 2 5 8**

**逻辑记录：A B C D E F G H I J**

**这种记录优化分使处理程序在处理完前一条逻辑记录时磁头正好旋转到下一条逻辑记录所在的扇区，处理所需的时间最短，处理完这10条记录需要花费的时间=10\*（2+4）=60ms。**

# 习题七

3.对于输入输出系统中的I/O请求队列和I/O完成队列，试：

（1）设计关于它们的同步结构；

（2）用类C语言描述请求者进程与设备驱动进程关于I/O请求队列的同步操作，以及设备驱动程序与请求者进程关于I/O完成队列的同步操作。

5.下列工作在四层I/O的哪一层上完成？

（1）对于读磁盘操作，计算磁道、扇区和磁头；

**中断处理层**

（2）维护最近使用的块而设的超高速缓存；

**设备独立性软件层**

（3）向设备寄存器写命令；

**设备驱动程序层**

（4）查看用户是否被允许使用设备；

**设备独立性软件层**

（5）为了打印，把二进制整数转化为ASCII码。

**用户空间软件层**

13.用类C语言描述设备驱动程序、中断控制器、设备控制表DCT，以实现以下操作：

（1）open(device);

(2)close(device)

(3)get\_block(device,buffer);

(4)put\_block(device,buffer)。

16.在进行设备管理时，必须要设计一些数据结构表格，对I/O设备的状态进行登记。在本章中给出了四类I/O设备的数据结构表格，它们分别是设备控制表DCT、系统设备表SDT、控制表COCT和通道控制表CHCT。能否为一个新的操作系统设备管理部分设计一套设备管理的数据结构，比上述四类I/O设备的数据结构更合理？

20.设备驱动程序的主要功能是什么？

**答：设备驱动程序是I/0进程与设备控制器之间的通信程序。**

**设备驱动程序的主要功能如下:**

**①将接收到的抽象要求转换为具体要求。**

**②检查用户I/O请求的合法性，了解I/O设备的状态，传递有关参数，设置设备的工作方式。③发出I/O命令，启动分配到的I/O设备，完成指定的I/0操作。**

**④及时响应由控制器或通道发来的中断请求，并根据其中断类型(正常、异常结束的中断或其他类型中断)调用相应的中断处理程序进行处理。**

**⑤对于设置有通道的计算机系统，驱动程序还应能够根据用户的I/O请求，自动地构成通道程序。**

# 习题八

3.考虑这样一种资源分配策略：对资源的申请和释放可以在任何时刻进行。如果一个进程的资源得不到满足，则考察所有由于等待资源而被阻塞的进程，如果他们有申请进程所需要的资源，则把这些资源取出分给申请进程。

例如，考虑一个有三类资源的系统，Available=（4，2，2）。进程A申请（2，2，1），可以满足；进程B申请（1，0，1），可以满足；若A再申请（0，0，1），则被阻塞（无资源可分）。此时，若C申请（2，0，0），它可以分得剩余资源（1，0，0），并从A已分得得资源中获得一个资源，于是，进程A的分配向量变成：Available=（1，2，1），而需求向量变成：Need=（1，0，1）。

（1）这种分配方式会导致死锁吗？若会，举一个例子；若不会，说明死锁的哪一个必要条件不成立。

（2）会导致某些进程的无限等待吗？

**答：（1）不会导致死锁，破坏了死锁的第三个请求和保持条件，由于等待资源而被阻塞的进程的资源，可以被其他需要的进程剥夺。**

**（2）会导致**

4，假定一个系统有4种资源，R={6，4，4，2}，当前系统状态如表所示。该状态安全吗？请阐述理由。

|  |  |  |  |
| --- | --- | --- | --- |
| 资源申请  进程 | | 目前占有量 | 最大需求量 |
| A B C D | A B C D |
| P1 | 2 0 1 1 | | 3 2 1 1 |
| P2 | 1 1 0 0 | | 1 2 0 2 |
| P3 | 1 1 0 0 | | 1 1 2 0 |
| P4 | 1 0 1 0 | | 3 2 1 0 |
| P5 | 0 1 0 1 | | 2 1 0 1 |

７．一个计算机系统有某种资源６个，供ｎ个进程使用，每个进程至少需要２个资源。当ｎ为何值时系统不会发生死锁？

**答：N ≤ 5时系统不会发生死锁。6个资源，每个进程至少需要2个资源，不发生死锁必须保证至少有1个进程可以得到所需的全部资源并执行完毕，因此N最大为5。**

15.某系统当前有同类资源１０个，进程P、Q、R所需资源总数分别为8、4、9。它们向系统申请资源的次序和数量如图所示。

（１）系统采用银行家算法分配资源，请给出系统完成第６次分配后各进程的状态及所占资源量。

（２）在以后各次的申请中，哪次的申请要求可先得到满足？

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 次序 | 进程 | 申请量 | 次序 | 进程 | 申请量 |
| １ | Ｒ | ２ | ６ | Ｑ | ２ |
| ２ | Ｐ | ４ | ７ | Ｒ | ３ |
| ３ | Ｑ | ２ | ８ | Ｐ | ２ |
| ４ | Ｐ | ２ | ９ | Ｒ | ３ |
| ５ | Ｒ | １ |  |  |  |

**答：（１）第6次分配后剩余资源2个**

**进程  占有资源 进程状态 完成任务尚需资源**

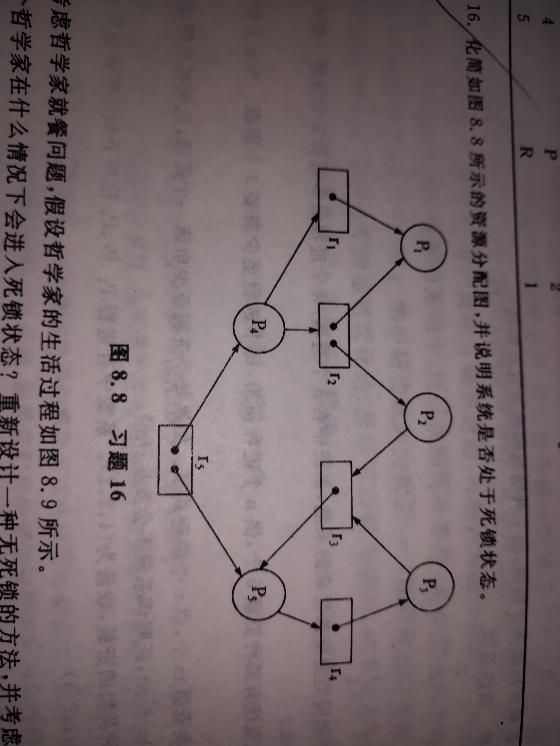
**R   2个   　　等待  　　7个**

**P   6个   　　等待  　　2个**

**Q   0个   　　等待  　　8个**

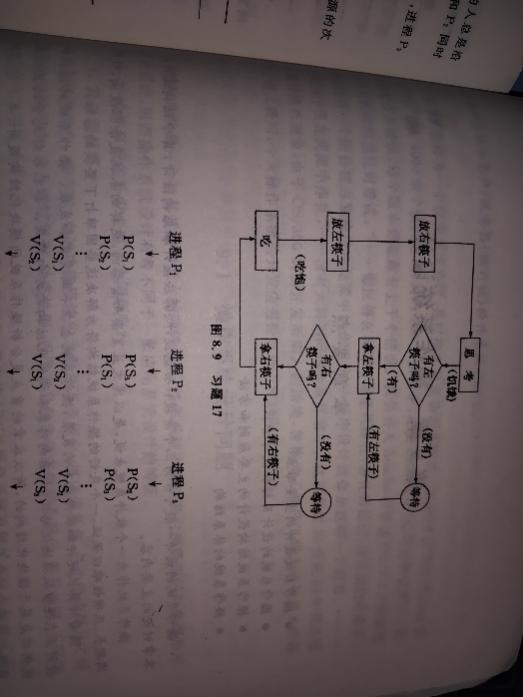
**（２）以后各次申请中，第8次进程P的申请要求可以先得到满足**

16.化简如图所示的资源分配图，并说明系统是否处于死锁状态。



**答：** **系统处于死锁状态**

19.设有三个进程P1、P2、P3，各自按下列顺序执行程序代码：



其中S1、S2、S3是信号量，且初值均为1.

在执行时是否会产生死锁？如果能产生死锁，请说明在什么情况下会产生死锁？并给出一个防止死锁产生的修改办法。

**答：会导致死锁发生，比如运行某时刻，P1占用了S1，请求S3；P2占用了s2，请求S1；P3占用了s3，请求S2。这样就会出现死锁问题。为了避免这种后果，可以采取如下方法(1)有序分配法，每个进程按一定顺序来进行资源请求；(2)预先分配法，一次性将某进程请求的资源全部分配给它；(3)银行家算法，在某进程申请资源时先进行安全性检查等方法都可以避免出现死锁。**