操作系统习题集

（2012版）

目 录

（＊ 所标的题目超出教科书范围，可不看 ）

第一章 操作系统引论

aa1．选择题

1．计算机操作系统的功能是_____。
A．把源程序代码转换为目标代码
B．实现计算机用户之间的相互交流
C．完成计算机硬件与软件之间的转换
D．控制、管理计算机系统的资源和程序的执行

2．操作系统是一组_____。
A．文件管理程序　　B．中断处理程序　　C．资源管理程序　　D．设备管理程序

3．操作系统的功能是进行处理机管理、_____管理、设备管理、文件管理和作业管理等。
A．进程　　B．存储器　　C．硬件　　D．软件

4．_____不是分时系统的特点。
A．多个用户是经过网络连接，同时使用计算机系统
B．各用户可同时请求系统服务
C．各用户的请求彼此独立，互不干扰
D．用户以会话方式控制自己的程序运行

5*．_____指令是非特权指令。
A．启动I／O　　B．设置中断屏敝　　C．传送PSW　　D．trap

6．“中断”的概念是指_____。
A．暂停处理机执行　　B．暂停处理机对现行程序的执行
C．停止整个系统运行　　D．使处理机空转

7．在____的控制下，计算机系统能及时处理由过程控制反馈的数据，并作出响应。
A．批处理操作系统　　B．实时操作系统
C．分时操作系统　　D．多处理机操作系统

8*．下列中断不属于强迫性中断的是____。
A．传输结束　　B．断电
C．运行的程序请求分配一块内存　　D．目态程序执行特权指令

9*．计算机系统中设置的访管指令，____执行。
A．只能在目态　　B．只能在管态
C．既可在目态又可在管态　　D．在目态和管态下都不能

10．操作系统为用户程序完成与____的工作。
A．硬件无关和应用无关　　B．硬件相关和应用无关
C．硬件无关和应用相关　　D．硬件相关和应用相关

11*．Windows NT Server是一种____。
A．单用户多进程系统　　B．多用户多进程系统
C．单用户单进程系统　　D．多用户单进程系统

12*．用户程序在目态下使用特权指令将引起的中断是属于____。

A．硬件故障中断　　B．程序中断　　C．外部中断　　D．访管中断

13．分时操作系统的主要目的是____。

A．计算机系统的交互性　　B．计算机系统的实时性

C．计算机系统的可靠性　　D．提高软件的运行速度

14．在操作系统中，用户界面指的是_____。

A．硬件接口、软件接口和操作环境　　B．命令接口、程序接口和操作环境

C．硬件接口、命令接口和操作环境　　D．硬件接口、命令接口和程序接口

15*．特权指令_____执行。

A．只能在目态下　　B．只能在管态下

C．在目态或管态下均能　　D．在目态或管态下均不能

16．下列管理功能中，_____不属于操作系统的功能。

A．处理器管理　　B．软件管理　　C．作业管理　　D．设备管理

17*．当CPU执行操作系统代码时，称处理机处于____。

A．执行态　　B．目态　　C．管态　　D．就绪态

18．以下描述与操作系统无关的是 _____ 。

A．方便用户的程序集合

B．控制和管理计算机系统的硬件和软件资源

C．计算机系统的硬件和软件资源的集合

D．合理地组织计算机工作流程

19．分时操作系统的特点是 _____ 。

A．交互性、同时性(多路性)、独立性、及时性

B．可靠性、交互性、独立性、及时性

C．可靠性、交互性、独立性、及时性

D．交互性、同时性(多路性)、独立性、动态性

20．下列各项中，_____ 不是现代操作系统的主要特征。

A．并发性　　B．共享性　　C．确定性　　D．虚拟性

21．以下关于操作系统作用的叙述中，不正确的是 _____ 。

A．管理系统资源　　B．控制程序执行

C．改善人机界面　　D．提高用户软件运行速度

22．从用户的观点看，操作系统是_____ 。

A．用户与计算机之间的接口

B．控制和管理计算机资源的软件

C．合理地组织计算机工作流程的软件

D．由若干层次的程序按一定的结构组成的有机体

23．_____操作系统允许在一台主机上同时连接多台终端，多个用户可以通过各自的终端同时交互地使用计算机。

A．网络　　B．分布式　　C．分时　　D．实时

24．若把操作系统看作计算机系统资源的管理者，下列的____不属于操作系统管理的资源。

A．程序　　B．内存　　C．CPU　　D．中断

25．在下列操作系统的各个功能组成部分中，_____不需要硬件的支持。

A．进程调度　　B．时钟管理　　C．地址影射　　D．中断系统

26．在下列操作系统中，对响应时间要求最高的是_____。

A．批处理系统　　B．分时系统　　C．实时系统　　D．网络操作系统

27．对出现的中断事件是由_____进行处理的。

A．硬件　　B．操作系统　　C．用户程序　　D．解释程序

28*．____命令应该只在核心态下执行。

A.读时钟日期　　B.计算圆周率π　　C.屏蔽所有中断　　D.调用过程(procedure)

29．有关原语的说法中，____是正确的。

A．原语是不可中断执行的用户过程　　B．原语是不可中断执行的操作系统过程

C．原语是可中断执行的用户过程　　D．原语是可中断执行的操作系统过程

30．原语应是____。

A．操作系统中的一个函数

B．操作系统中的一个过程

C．操作系统中的一个执行不可中断的过程

D．操作系统中的一个执行可中断的函数

31．下面哪一项不是引入操作系统的主要目的是_____。

A．方便用户使用　　B．更有效地利用软、硬件资源

C．及时响应用户请求　　D．改善系统性能

32*．只能在核心态下执行的指令是____。

A．读时钟日期　　B．屏蔽所有中断　　C．改变文件内容　　D．调用库函数

33*．Windows3.1 是一个____位的操作系统。

A．16　　B．32　　C．48　　D．64

34．多道批处理系统的主要缺点是_____ 。

A．CPU利用率低　　B．不能并发执行　　C．缺少交互性　　D．以上都不是

35*．分布式计算机系统具备的基本功能是 _____ 。

A．通信、并行计算、资源管理　　B．通信、并行计算、资源共享

C．并行计算、资源共享、存储器共享　　D．通信、并行计算、存储器共享

36*．在下列 4 个操作系统中，____具有多道程序设计特点，但不是分时系统(多用户系统)。

A．OS/2　　B．Windows 3.1　　C．UNIX　　D．Windows NT

37*．下列关于 Windows NT 的说法中，____是错误的。

A．Windows NT中的每一个进程都是对象，有些进程也是可以共享的资源

B．Windows NT中，进程是资源分配和处理机调度的基本单位

C．Windows NT 5.0就是Windows 2000

D．Windows NT的内核采用微内核的形式

38．多道程序设计是指____。

A．在多台处理机上同时执行多道程序　　B．在多台处理机上同一时刻执行多道程序

C．在一台处理机上同时执行多道程序　　D．在一台处理机上同一时刻执行多道程序

39．从用户的观点看，操作系统是____。

A．用户与计算机之间的接口　　B．控制和管理计算机系统的资源

C．合理组织计算机工作流程　　D．一个大型的工具软件

40．配置了操作系统的计算机是一台比原来的物理计算机功能更强大的计算机，这样的计算机只是一台逻辑上的计算机．称为____计算机。

A．虚拟　　B．物理　　C．并行　　D．共享

41．操作系统是对____进行管理的软件。

A．系统软件　　B．系统硬件　　C．计算机资源　　D．计算机程序

42*．多道批处理的发展是建立在____硬件支持上的。

A．集成电路　　B．高速缓存　　C．通道和中断机构　D．大容量硬盘

43．批处理系统的主要缺点是_____。

A．CPU利用率低　　D．不能并发执行　　C．缺少交互性　　D．以上都不是

44．如果分时系统的时间片一定，那么____，则响应时间越长。

A．内存越大　　B．内存越少　　C．用户数越少　　D．用户数越多

45 分时操作系统通常采用____策略为用户服务。

A．先来先服务　　B．短作业优先　　C．时间片轮转　　D．最高响应比

46．在下列性质中，哪一个不是分时系统的特征____。

A．多路性　　B．交互性　　C．独占性　　D．成批性

47．在____操作系统的控制下，计算机系统能及时处理由过程控制反馈的数据并作出响应。

A．批处理　　B．分时　　C．实时　　D．网络

48．设计实时操作系统时，首先要考虑系统的_____。

A．实时性和可靠性　　B．实时性和灵活性　　C．灵活性和可靠性　　D．灵活性和可移植性

49．UNIX 操作系统是一种多用户的、人机交互的_____ 。

A．多道批处理系统　　B．实时系统　　C．分时系统　　D．分布式系统

50*．主要由于____原因，使UNIX易于移植。

A、UNIX是由机器指令书写的　　B、UNIX大部分由汇编少部分用C语言编写

C、UNIX是用汇编语言编写的　　D、UNIX小部分由汇编大部分用C语言编写

51．操作系统在计算机系统中处于____之间的位置。

A．计算机硬件和软件　　B．计算机硬件和用户

C．处理机和用户　　D．外部设备和处理机

52．实时操作系统必须在____的时间内响应一个新任务。

A．一个机器周期　　B．被控对象规定　　C．任意周期　　D．时间片

53．在操作系统中，____部分属于微内核。

A．作业调度软件　　B．用户命令解释程序

C．磁盘文件目录管理软件　　D．进程通信服务例程

54．批处理系统的主要缺点是____。

A．CPU利用率低　　B．外部设备利用率低

C．不能并发执行　　D．缺少交互性

55．操作系统提供给用户程序的接口是____。

A．命令解释程序　　B．系统调用　　C．P、V操作　　D．对话框

56．分时系统响应时间与____有关。

A．每个应用进程分配的时间片长度　　B．进程大小

C．就绪进程数目　　D．就绪进程数目和时间片长度

57．下列选项中，____不属于操作系统提供给用户的可使用资源。

A．中断机制　　B．处理机　　C．存储器　　D．I/O设备

58．操作系统的最主要设计目标是_____。

A．方便性和有效性　　B．方便性和可扩展性

C．有效性和可扩展性　　D．有效性和开放性

59．操作系统的两个最主要的特征是____。

A．并发性和虚拟性　　B．并发性和共享性

C．共享性和异步性　　D．共享性和虚拟性

60．下面各项中，_____不是操作系统的基本特征。

A．并发和共享　　B．虚拟　　C．交互性　　D．异步

61．下列选项中，在用户态执行的是_____。(2011全国试题)

A．命令解释程序　　B．缺页处理程序

C．进程调度程序　　D．时钟中断处理程序

62．下列选项中，不可能在用户态发生的事件是_____。(2012全国试题)

A．系统调用　　B．外部中断　　C．进程切换　　D．缺页

63．中断处理和子程序调用都需要压栈以保护现场，中断处理一定要保存而子程序调用不需要保存其内容的是____。(2012全国试题)

A．程序计数器　　B．程序状态字寄存器

C．通用数据寄存器　　D．通用地址寄存器

第一章操作系统引论选择题参考答案：

1．D	2．C	3．B	4．A	5．D	6．B	7．B	8．C	9．C	10．B
11．B	12．B	13．A	14．B	15．B	16．B	17．C	18．C	19．A	20．C
21．D	22．A	23．C	24．D	25．A	26．C	27．B	28．C	29．B	30．C
31．C	32．AB	33．A	34．C	35．AB	36．B	37．B	38．C	39．A	40．A
41．C	42．C	43．C	44．D	45．C	46．D	47．C	48．A	49．C	50．D
51．B	52．B	53．D	54．D	55．B	56．D	57．A	58．A	59．B	60．C
61．A	62．C	63．B							

第二章 进程管理

1．选择题

1．有关进程的下列叙述中，_____是正确的。
A．进程是静态的文本　　B．进程与程序是一一对应的
C．进程与作业是一一对应的　　D．多个进程可以在单个CPU上同时执行
2．进程之间的制约关系可以归结为_____。
A．同步与互斥　　B．并发与异步　　C．同步与并发　　D．同步与异步
3．下列的进程状态变化中，_____的变化是不可能发生的。
A．运行→就绪　　B．运行→等待　　C．等待→运行　　D．等待→就绪
4．进程和程序的本质区别是_____。
A．存储在内存和外存　　B．顺序和非顺序执行机器指令
C．分时使用和独占使用计算机资源　　D．动态和静态特征
5．某进程所要求的一次打印输出结束，该进程被唤醒，其进程状态将从_____。
A．就绪状态到运行状态　　B．等待状态到就绪状态
C．运行状态到等待状态　　D．运行状态到就绪状态
6．进程调度是从_____选择一个进程投入运行。
A．就绪队列　　B．等待队列　　C．作业后备队列　　D．提交队列
7．下列叙述中，正确的叙述是_____。
A. 实现多道程序设计的目的是提高程序员编程的效率
B. 在有虚拟存储器的系统中，可以运行比主存容量还大的程序
C. 操作系统的目的是为了提高计算精度
D. 操作系统必须具备分时系统
8．已获得除CPU以外的所有所需资源的进程处于_____状态。
A．运行　　B．就绪　　C．自由　　D．等待
9．进程具有并发性和_____两大重要属性。
A．动态性　　B．静态性　　C．易用性　　D．封闭性
10．两个进程合作完成一个任务，在并发执行中，一个进程要等待其合作伙伴发来消息，或者建立某个条件后再向前执行，这种关系称为进程间的_____。
A．同步　　B．互斥　　C．竞争　　D．合作
11．在多道程序系统中，为了保证公共变量的完整性，各进程应互斥进入相关临界区。所谓临界区是指_____。
A．一个缓冲区　　B．一段数据区　　C．同步机制　　D．一段程序
12．一个进程是____ 。
A．由协处理器执行的一个程序　　B．一个独立的程序 + 数据集
C．PCB结构、程序和数据的集合　　D．一个独立的程序
13．多道程序系统中的操作系统分配资源以____为基本单位。
A．程序　　B．进程　　C．作业　　D．用户
14．进程从等待状态转到就绪状态的原因可能是_____ 。

A．请求I/O　　B．I/O完成
C．被进程调度程序选中　　D．另一个进程运行结束

15．采用多道程序设计能_____。
A．增加平均周转时间　　B．发挥并提高并行工作能力
C．缩短每道程序的执行时间　　D．降低对处理器调度的要求

16．某个进程从等待状态进入就绪状态可能是由于_____。
A．现运行进程执行了启动I/O指令　　B．现运行进程执行了P操作
C．现运行进程执行了V操作　　D．现运行进程时间片用完

17．在计算机系统中，允许多个程序同时进入内存并运行，这种方法称为_____。
A．SPOOLing技术　　B．虚拟存储技术
C．缓冲技术　　D．多道程序设计技术

18．多道程序的引入主要是为了_____。
A．提高CPU的速度　　B．提高内存的存取速度
C．提高计算机的使用效率　　D．减少CPU处理作业时间

19．多道程序系统中，当_____时，进程从执行状态转变为就绪状态。
A．进程被进程调度程序选中　　B．时间片到
C．等待某一事件　　D．等待的事件发生

20．并发进程相互之间_____。
A．必须通信　　B．可以有交往也可无关
C．一定会竞争共享资源　　D．相互间一定要同步

21．下列选项中，导致创建新进程的操作是_____。（2010全国试题）
I．用户登录成功　　II．设备分配　　III．启动程序执行
A．仅I和II　　B．仅II和III　　C．仅I和III　　D．I、II和III

22．若信号量S的初值为2，当前值为-1，则表示有___个等待进程。
A．0　　B．1　　C．2　　D．3

23．设与某资源关联的信号量初值为3，当前值为1。若M表示该资源的可用个数，N表示等待该资源的进程数，则M、N分别是_____。（2010全国试题）
A．0、1　　B．1、0　　C．1、2　　D．2、0

24．操作系统中，对信号量S的P原语操作定义中，使进程进入相应等待队列的条件是___。
A．S≠0　　B．S<0　　C．S=0　　D．S>0

25．为了使两个进程能同步运行，最少需要 ___ 个信号量。
A．1　　B．2　　C．3　　D．4

26．下面叙述中正确的是___ 。
A．操作系统的一个重要概念是进程，因此不同进程所执行的代码也一定不同
B．为了避免发生死锁，各进程只能逐个申请资源
C．操作系统用PCB管理进程，用户进程可以从PCB中读出与本身运行状态有关的信息
D．进程同步是指某些进程之间在逻辑上的相互制约关系

27．信箱通信是一种___通信方式。
A．直接　　B．间接　　C．低级　　D．信号量

28．进程控制块记录了进程执行时的情况，它的内容可由____进行修改。
A．操作系统　　B．进程自己　　C．中断装置　　D．用户

29．支持多道程序设计的操作系统在运行过程中，不断地选择新进程运行来实现 CPU 的共享，下列选项中，____不是引起操作系统选择新进程的直接原因。
A．运行进程的时间片用完　　B．运行进程出错
C．运行进程要等待某一时件发生　　D．有新进程进入就绪状态

30．并发性是指若干事件在____发生。

A．同一时刻　　B．同一时间间隔内　　C．不同时刻　　D．不同时间间隔内

31．有关 PV 操作的说法中____是错误的。

A.“PV操作不仅是进程互斥的有效工具，而且是简单方便的同步工具”

B.“PV操作不能实现进程间通信”

C.“进程调用P操作测试自己所需的消息是否到达”

D.“进程调用V操作向其它进程发送消息”

32．使若干并发进程共享一临界资源而不发生与进程推进速度有关错误，涉及相关临界区的错误说法是____。

A.“一次最多让一个进程在临界区执行”

B.“任何一个进入临界区执行的进程必须在有限时间内退出临界区”

C.“可以强迫一个进程无限地等待进入它的临界区”

D.“可能没有任何进程在临界区执行”

33．通常，用户进程被建立后，____。

A．便一直存在于系统中，直到被操作人员撤消

B．随着程序运行正常或异常结束而撤消

C．随着时间片轮转而撤消与建立

D．随着进程的阻塞或唤醒而撤消与建立

34．有关并发进程相互之间的关系，正确的说法是____。

A．肯定是无关的　　B．肯定是有交往的

C．可能是无关的，也可能是有交往的　　D．一定要互斥执行

35．当一个进程____就要退出等待队列而进入就绪队列。

A．启动了外设　　B．用完了规定的时间片

C．获得了所等待的资源　　D．能得到所等待的处理器

36．有n个并发进程竞争必须互斥使用的共享资源时，若某进程调用P操作后成为第一个等待使用该资源者，则这时信号量的值为____。

A．0　　B．1　　C．-1　　D．n-1

37．在同一系统中，假设同时存在为两个相互独立的C++源程序进行编译的两个进程(它们使用同一个编译程序)，它们之间的关系正确的是：_____。

A．它们可以并发执行，两者逻辑上有依赖关系

B．它们可以并发执行，两者逻辑上无依赖关系

C．它们不可以并发执行，但两者逻辑上有依赖关系

D．它们不可以并发执行，因为两个进程运行的是同一个编译程序

38．S.queue、S.value是信号量S的两个组成部分，当S.queue为空时，S.value的值是____。

A、S.value≤0　　B、S.value=0　　C、S.value=1　　D、Svalue≥0

39．设有三个进程共享一个资源，如果每次只允许一个进程使用该资源，则用PV操作管理时信号量S的可能取值是____。

A、1,0,-1,-2　　B、2,0,-1,-2　　C、1,0,-1　　D、3,2,1,0

40．临界区是指并发进程中访问共享变量的____段。

A、管理信息　　B、信息存储　　C、数据　　D、程序

41．如下参数中，不能用于进程间通信的是____。

A．消息　　B．信件　　C．信号量　　D．口令

42．当输入输出操作正常结束时，操作系统将请求该操作的进程的状态设置成____。

A．等待状态　　B．运行状态　　C．就绪状态　　D．挂起状态

43．对具有相关临界区的 n 个并发进程采用 P、V 操作实现进程互斥时，信号量的初值应定义为____。

A．0　　B．1　　C．n　　D．n-1

44．多个进程间可通过P、V操作交换信息实现进程同步和互斥，因此信号量机制是进程间的一种_____通信方式。

A．高级　　B．低级　　C．消息缓冲　　D．间接

45．属于进程通信原语的有____。

A．P操作原语　　B．V操作原语　　C．创建进程原语　　D．send原语

46．涉及PV操作的正确说法是____。

A．PV操作只能解决进程互斥问题

B．PV操作只能解决进程同步问题

C．PV操作能用于解决进程互斥问题，也能解决进程同步问题

D．PV操作是一种高级通信方式

47．并发进程执行时可能会出现与时间有关的错误，这种错误是与____无关的。

A．使用共享资源　　B．进程被打断的时间

C．进程占用处理器的总时间　　D．进程交替执行的次序

48．设有12个同类资源可供4个进程共享，资源分配情况如下表所示。

进程	已占用资源数	最大需求数
P1	2	4
P2	3	6
P3	4	7
P4	1	4

当进程P1，P2，P3，P4又都相继提出申请要求，为使系统不致死锁，应满足____的要求。

A．P1　　B．P2　　C．P3　　D．P4

49．进程控制块中的现场信息是在____保存的。

A．创建进程时　　B．处理器执行指令时

C．中断源申请中断时　　D．中断处理程序处理中断前

50．采用____的手段可以防止系统出现死锁。

A．PV操作管理临界资源　　B．限制进程互斥使用临界资源

C．资源静态分配策略　　D．定时运行死锁检测程序

51．进程所请求的一次打印输出结束后，将使该进程状态从____。

A．运行态变为就绪态　　B．运行态变为等待态

C．就绪态变为运行态　　D．等待态变为就绪态

52*．线程是操作系统的重要概念，不具有线程管理的操作系统有____。

A．Windows 3.2　　B．Linux　　C．Windows NT　　D．Windows XP

53．进程从就绪状态进入运行状态的原因可能是____。

A．等待某一事件　　B．被选中占有处理器

C．时间片用完　　D．等待的事件已发生

54．操作系统中，资源分配的基本单位是____。

A．进程　　B．线程　　C．作业　　D．程序

55*．构成网络操作系统通信机制的是____。

A．进程　　B．线程　　C．通信原语　　D．对象

56．某计算机系统中若同时存在 5 个进程，则处于等待状态的进程最多可有____个。

A．0　　B．1　　C．4　　D．5

57．若系统中有5个并发进程涉及某个相同的变量A，则变量A的相关临界区是由____临界区构成。

A．2个　　B．3个　　C．4个　　D．5个

58．在下述进程状态的转换中，_____是不可能的。

A．运行态→就绪态　　B．运行态→等待态

C．等待态→就绪态　　D．就绪态→等待态

59．若 P、V 操作的信号量 S 的初值为 3，当前值为-1，则表示在 S 上有____个等待进程。

A．0　　B．1　　C．2　　D．3

60．以下叙述中，正确的是____ 。

A．进程调度原语主要是按一定的算法，从阻塞队列中选择一个进程，将处理机分配给它。

B．预防死锁发生可通过破坏死锁的四个必要条件之一来实现，但破坏互斥条件的可能性不大。

C．采用信号量同步机制的系统，进程进入临界区时要执行 V 原语

D．既考虑作业的等待时间，又考虑作业执行时间的调度算法称为电梯调度算法。

61．设有n个进程使用同一个共享变量，如果最多允许m（m＜n）个进程同时进入相关临界区，则信号量的变化范围是_____ 。

A．n,n-1,...,n-m　　B．m,m-1,...1,0,-1,...m-n

C．m,m-1,...1,0,-1,...m-n-1　　D．m,m-1,...1,0,-1,...m-n+1

62．对于有两个并发进程的系统，设互斥信号量为mutex，若mutex=0，则_____。

A．表示没有进程进入与mutex相关的临界区

B．表示有一个进程进入与mutex相关的临界区

C．表示有一个进程进入与mutex相关的临界区，另一个进程等待进入

D．表示有两个进程进入与mutex相关的临界区

63．在进程管理中，当____时，进程从运行状态变为就绪状态。

A．时间片用完　　B．被进程调度程序选中

C．等待某一事件发生　　D．等待的事件发生

64．下列因素中，____不一定是引起进程调度的因素。

A．一个进程运行完毕　　B．运行进程被阻塞

C．一个高优先级进程被创建　　D．实时调度中，一个紧迫的任务到来

65．当一个进程正等待着____时，称其为等待状态。

A．合作进程的一个消息　　B．分配给它一个时间片

C．调度程序选中它　　D．进入内存

66．若进程P一旦被唤醒就能投入运行，则系统可能是____。

A．非抢占式调度方式，进程P的优先级最高

B．抢占式调度方式，就绪队列上的所有进程的优先级皆比P低

C．就绪队列为空队列

D．抢占式调度方式，P的优先级高于当前运行的进程

67．单CPU系统中，关于进程的叙述正确的是____。

A．一个处于等待状态的进程一旦分配了 CPU，即进入运行状态

B．只能有一个进程处于就绪状态

C．一个进程可以同时处于就绪状态和等待状态

D．最多只有一个进程处于运行状态

68．下列有关PV操作和死锁的叙述中，正确的是____。

A．V操作可能引起死锁　　B．P操作不会引起死锁

C．使用PV操作不会引起死锁　　D．以上说法均不正确

69．在分时系统中，下列描述中，_____不属于响应时间的一部分。

A．处理机对请求信息进行处理的时间

B．从键盘输入的请求信息传送到处理机的时间

C．请求信息在外存队列上排队等待的时间

D．所形成的响应回送到终端显示器的时间

70．在具有挂起状态的系统中，若当前内存空间高度吃紧，系统将使一个正在等待I/O的进程进入_____状态。

A．活动就绪　　B．静止就绪　　C．活动阻塞　　D．静止阻塞

71．下列说法中，正确的是____。

A．一般来说，用户进程的PCB存放在用户区，系统进程的PCB存放在系统区

B．某进程的一个线程处于阻塞状态，则该进程必然处于阻塞状态

C．在多道程序设计环境中，为了提高CPU效率，内存中的进程越多越好

D．同步是指并发进程之间存在的一种制约关系

72．在下述关于父进程和子进程的叙述中，正确的是____。

A．父进程创建了子进程，因此父进程执行完了，子进程才能运行

B．子进程执行完了，父进程才能运行

C．撤消子进程时，应该同时撤消父进程

D．撤消父进程时，应该同时撤消子进程

73．多道程序设计能充分发挥_____之间的并行工作能力。

A．CPU与外设　　B．进程与进程　　C．内存与进程　　D．内存与外设

74．在有m个进程的系统中出现死锁时，死锁进程的个数k应满足的条件是_____ 。

A．k≥2　　B．1＜k＜m　　C．1＜k≤m　　D．k≥1

75．在一个单处理机系统中，若有 4 个用户进程，且假设当前时刻为用户态，则处于就绪状态的用户进程至少有___个。

A．0　　B．1　　C．2　　D．3

76．有甲、乙两道算题，每道需执行1小时（其中处理器的工作时间为12分钟）。若它们在多道系统中执行，甲、乙两道题总共需执行80分钟，则处理器的利用率为 _____ 。

A．50%　　B．40%　　C．30%　　D．20%

77．下面的描述中，_____是错误的。

A．进程执行的相对速度不能有进程自己来控制

B．P、V操作是原语操作

C．利用信号量的P、V操作可以交换大量信息

D．同步是指并发进程之间次年在的一种制约关系

78．当输入输出操作正常结束时，操作系统将请求该操作的进程的状态设置成_____。

A．等待状态　　B．运行状态　　C．就绪状态　　D．挂起状态

79．如果单CPU系统中有n个并发进程，则就绪队列中进程个数最多可达_____个。

A．n　　B．n-1　　C．n-2　　D．1

80．一个进程的基本状态可以从其它两种基本状态转变过去，这个基本状态一定是_____。

A．执行状态　　B．阻塞状态　　C．就绪状态　　D．完成状态

81．当进程A使用磁带机时，进程B又申请磁带机，这种情况_____。

A．是不可能出现的　　B．是没法解决的　　C．就是死锁　　D．以上均不正确

82．进程具有的特性包括：_____。

①动态性　②共享性　③并发性　④相互制约性　⑤独立性　⑥静态性

A．①③④⑤　　B．①②④⑤　　C．②④⑤⑥　　D．①②④⑥

83．在引入线程的操作系统中，把_____作为调度和分派的基本单位，而把_____作为资源拥有的基本单位。

A．进程 线程　　B．程序 线程　　C．程序 进程　　D．线程 进程

84．S为死锁状态的充要条件是_____，该充要条件称为死锁定理。

A．当且仅当S状态的资源分配图是可完全简化的

B．当且仅当S状态的资源转换图是不可完全简化的

C．当且仅当S状态的资源分配图是不可完全简化的

D．当且仅当S状态的资源转换图是可完全简化的

85．现有3个同时到达的作业J1、J2、J3，它们的执行时间分别为T1、T2和T3，且T1<T2<T3。系统按单道方式运行且采用短作业优先算法，则平均周转时间为_____。

A．T1+T2+T3　　B．(T1+T2+T3)/3　　C．(3T1+2T2+T3)/3　　D．(T1+2T2+3T3)/3

86．进程P0和P1的共享变量定义及其初值为：

boolean flag[2];

int turn=0;

flag[0]=FALASE; flag[1]=FALSE;

若进程P0和P1访问临界资源的类C伪代码实现如下：

```
void P0( ) //进程P0
{ while(TRUE) {
  flag[0]=TRUE; turn=1;
  while(flag[1] && (turn==1)) ;
  临界区;
  flag[0]=FALSE;
  }
}
```

```
void P1( ) //进程P1
{ while(TRUE) {
  flag[1]=TRUE; turn=0;
  while(flag[0] && (turn==0)) ;
  临界区;
  flag[1]=FALSE;
  }
}
```

则并发执行进程P0和P1时产生的情形是_____。（2010全国试题）

A．不能保证进程互斥进入临界区，会出现“饿死”现象

B．不能保证进程互斥进入临界区，不会出现“饿死”现象

C．能保证进程互斥进入临界区，会出现“饿死”现象

D．能保证进程互斥进入临界区，不会出现“饿死”现象

87．在支持多线程的系统中，进程P创建的若干线程不能共享的是____。（2011全国试题）

A．进程P的代码段　　B．进程P中打开的文件

C．进程P的全局变量　　D．进程P中某线程的栈指针

88．有两个并发进程P1和P2，共享初值为1的变量x。P1对x加1，P2对x减1。加1和减1操作的指令序列分别如下所示。

```
//加1操作                          //减1操作
load  R1, x    //取x到寄存器R1中    load  R2, x
inc  R1                           dec  R2
store  x, R1    //将R1的内容存入x   store  x, R2
```

两个操作完成后，x的值____。（2011全国试题）

A．可能为-1或3　　B．只能为1

C．可能为0、1或2　　D．可能为-1、0、1或2

89．下列关于进程和线程的叙述中，正确的是_____。（2012全国试题）

A．不管系统是否支持线程，进程都是资源分配的基本单位

B．线程是资源分配的基本单位，进程是调度的基本单位

C．系统级线程和用户级线程的切换都需要内核的支持

D．同一进程的各个线程拥有各自不同的地址空间

第二章进程管理选择题参考答案：

1．D	2．A	3．C	4．D	5．B	6．A	7．B	8．B	9．A	10．A
11．D	12．C	13．B	14．B	15．B	16．C	17．D	18．C	19．B	20．B
21．C	22．B	23．B	24．B	25．B	26．D	27．B	28．A	29．D	30．B
31．B	32．C	33．B	34．C	35．C	36．C	37．B	38．D	39．A	40．D
41．D	42．C	43．B	44．B	45．C	46．C	47．C	48．A	49．D	50．C
51．D	52．A	53．B	54．A	55．C	56．C	57．D	58．D	59．B	60．B
61．B	62．B	63．A	64．C	65．A	66．D	67．D	68．D	69．C	70．D
71．D	72．D	73．A	74．B	75．A	76．C	77．C	78．C	79．B	80．C
81．D	82．A	83．D	84．C	85．C	86．D	87．D	88．C	89．A	

2．应用题

1．若进程Pa、Pb和Pc单独执行时间分别是1小时、1.5小时和2小时，其中处理机工作时间分别为10分钟 、15分钟和35分钟。如果采用多道程序设计方法，让Pa、Pb和Pc并行工作，假定处理机利用率达到50%，请问系统效率能提高百分之几？

答：Ta、Tb和Tc并行工作共用CPU时间为：

(10+15+35)/50%=120 (分钟)

单道方式执行时总时间为60+90+120=270分钟

故系统效率提高：(270-120)/270*100%=55.6%

进程同步问题

A. 生产者-消费者问题类

1. (西北工大2000年试题)由三个进程get，copy和put以及两个缓冲区buffer1和buffer2完成一项输入/输出操作。进程get的功能是把一张卡片上的信息从读卡机上读进buffer1；进程copy的功能是把buffer1中的信息复制到buffer2；进程put的功能是取出buffer2中的信息并从打印机上打印输出。试用P、V操作完成这三个进程间的尽可能并发正确执行的关系(用程序或框图表示)，并指明信号量的作用和初值。

解：可设置6个信号量mutex1，mutex2，empty1，empty2，full1，full2。其中：

mutex1和mutex2是互斥信号量，初值为1，分别用于对buffer1和buffer2的互斥访问；

empty1和empty2为同步信号量，初值为1，分别表示buffer1和buffer2是否空闲，1表示空闲，0表示不空闲；

full1和full2为同步信号量，初值为0，分别表示buffer1和buffer2中是否有可取用的信息，1表示有可取用的信息，0表示无可取用的信息。

```
semaphore mutex1, mutex2, empty1, empty2, full1, full2 ;
mutex1=mutex2=1;        //互斥信号量
empty1=empty2=1;        //生产者进程的同步信号量
full1=full2=0;          //消费者进程的同步信号量
parbegin
   process get( ) //读进程(生产者进程)
   {
      while (1) {
```

```
            从读卡机读入一张卡片的信息;
            P(empty1)          //看看buffer1是否空闲
            P(mutex1);   //互斥访问buffer1
            将信息放入buffer1;
            V(mutex1);
            V(full1);      //通知进程copy，buffer1中已有信息科取(若copy正在等待，则唤醒之)
        }
    }
    process copy( ) //复制进程(既是消费者又是生产者进程)
    {
        while (1) {
            P(full1)       //看看buffer1是否有信息可取
            P(mutex1);   //互斥访问buffer1
            从buffer1中复制出信息;
            V(mutex1);
            V(emtpy1);   //通知get，buffer1中的信息已取走(可能唤醒get)
            P(empty2);   //看看buffer2是否空闲
            P(mutex2);   //互斥访问buffer2
            将复制的信息放入buffer2;
            V(mutex2);
            V(full2);      //通知put，buffer2中已有信息
        }
    }
    process put( )         //输出进程(消费者进程)
    {
        while (1) {
            P(full2);      //测试buffer2中是否有信息
            P(mutex2);   //互斥访问buffer2
            从buffer2中取出信息;
            V(mutex2);
            V(empty2);   //通知copy，buffer2中的信息已取走
        }
    }
parend
```

【讨论】由于本题中对于两个缓冲区buffer1和buffer2来说，都只有一个生产者和一个消费者，因此互斥信号量mutex1和mutex2实际上是可以省去的。

以下第2、3、4题实际上与本题是同一道题。

2. (北京大学1990年试题)有三个进程PA、PB和PC协作解决文件打印问题：PA将文件记录从磁盘读入主存的缓冲区1，每执行一次读一个记录；PB将缓冲区1的记录复制到缓冲区2，每执行一次复制一个记录；PC将缓冲区2的内容打印出来，每执行一次打印一个记录。缓冲区的大小和一个记录大小一样。试用P、V操作来保证文件的正确打印。

解：BEGIN

```
    semaphore mutex1，mutex2，avail1，avail2，full1，full2；
    mutex1 ：= 1；mutex2 ：= 1；     {实际上mutex1,mutex2可以省去}
```

```
avail1 ： = 1；avail2 ： = 1；
full1 ： = 0；full2 ： = 0；
PARBEGIN
    PA：BEGIN
        L1：read from disk；
            P（avail1）；
            P（mutex1）；
            put to buffer 1；
            V（full1）；
            V（mutex1）；
            goto L1；
        END
    PB：BEGIN
        L2：P（full1）；
            P（mutex1）；
            get from buffer 1；
            V（avail1）；
            V（mutex1）；

            P（avail2）；
            P（mutex2）；
            put to buffer 2；
            V（full2）；
            V（mutex2）；
            goto L2 ；
        END
    PC：BEGIN
        L3：P（full2）；
            P（mutex2）；
            get from buffer 2；
            V（avail2）；
            V（mutex2）；
            print RECORD
            goto L3 ；
        END
  PAREND
END
```

3. 有三个进程，Reader进程读入数据number1，将其放入缓冲器B_1，Executor进程将B_1中数据取出，处理成数据number2，将其放入缓冲器B_2，Printer进程将number2数据取出打印，假设B_1 和B_2只能存放一个数据，用P、V操作管理这三个进程的执行。

解：解：采用P、V操作的同步算法如下：

```
BEGIN
    semaphore  empty1, full1, empty2, full2 ;
    empty1.vale = empty2.value = 1 ;
```

```
ful2.value = full2.value = 0 ;
PARBEGIN
Reader：BEGIN
        L1：read number1 ;
            P(empty1) ;
            B1=number1 ;
            V(full1) ;
            goto L1；
        END
Executor：BEGIN
        L2：P(full1) ;
            take number1 from B1 ;
            V(empty1) ;
            Process number1-->number2 ;
            P(empty2) ;
            B2=number2 ;
            V(full2) ;
            goto L2；
        END
    Printer：BEGIN
            L3：P（full2）；
                take  number2 from B2 ;
                V(empty2) ;
                Print(number2) ;
                goto L3；
            END
  COEND
END
```

4. 假定系统有三个并发进程read, move和print共享缓冲器B1和B2。进程read负责从输入设备上读信息，每读出一个记录后把它存放到缓冲器B1中。进程move从缓冲器B1中取出一记录，加工后存入缓冲器B2。进程print将B2中的记录取出打印输出。缓冲器B1和B2每次只能存放一个记录。要求三个进程协调完成任务，使打印出来的与读入的记录的个数，次序完全一样。请用PV操作，写出它们的并发程序。（注：本题与第3题是同一个题，与第5题类似）

解：参考程序如下：

```
begin
  SR, SM1, SM2, SP: semaphore;
  B1, B2 : record;
  SR:=1; SM1:=0; SM2=1; SP:=0;
 cobegin
  process read
  X : recoed;
  begin
    R: X:=从输入设备上读入的一个记录;
      P(SR);
```

```
        B1:=X;
        V(SM1);
        goto R;
    end;
    process move
    Y : record;
    begin
     M:  P(SM1);
        Y :=B1;
        V(SR);
        加工Y中的记录;
        P(SM2);
        B2 := Y;
        V(SP);
        goto M;
    end;
    process print
    Z : record;
    begin
      P: P(SP);
        Z := B2;
        V(SM2);
        打印Z中的记录;
        goto P;
    end;
  coend;
end;
```

5. 今有3个并发进程R、M、P，它们共享一个缓冲器B。进程R负责从输入设备读入信息，每读一个记录后把它存放在缓冲器B中。进程M在缓冲器B中加工进程R存入的记录。进程P把加工后的记录打印出来。缓冲器B中每次只能存放一个记录，当记录被加工输出后，缓冲器B中又可以存放一个新的记录。为协调它们的工作，采用PV操作进行管理。

解：

```
semaphore SR,SM,SP;
SR=1; SM=0; SP=0;
parbegin
Process R
{
    while (1) {
        从输入设备读入信息X;
        P(SR);          //看看缓冲区B是否是空的
        B=X;            //信息存入缓冲区B
        V(SM);          //通知M，缓冲区B中已有记录
    }
}
Process M
```

```
{
    while (1) {
        P(SM);          //测试R是否已在B中存放信息
        在缓冲器B中加工进程R存入的记录;
        V(SP);          //通知P缓冲区B中的信息已可打印
    }
}
Process P
{
    while (1) {
        P(SP);     //测试M是否已将信息加工好
        从B中取M加工后的信息Y;
        V(SR);     //通知R，缓冲区B已可房信息
        Print(Y);  //打印信息Y
    }
}
parend
```

6. 若一只盘子一次只能放一个水果，A只往盘中放苹果，B只往盘中放梨子，C只从盘中取苹果，D只从盘中取梨子。试用：(1) 信号量和P、V操作；(2) 管程，写出同步算法。

解：(1) 采用P、V操作的同步算法如下：

```
semaphore SAB=1;  //A、B的资源信号量，同时又是它们的互斥信号量
semaphore SC=0;   //C的资源信号量(用于与A同步)
semaphore SD=0;   //D的资源信号量(用于与B同步)
begin
  parbegin
  process A:    //进程A的算法描述
  {
    while(true) {
      取一个苹果;
      wait(SAB);   //测试盘子是否为空
      将一苹果放入盘中;
      signal(SC)   //通知C盘中已有苹果(可能唤醒C)
    }
  }
  process C:
  {
    while(true) {
      wait(SC);   //测试盘子是否有苹果
      从盘中取出苹果;
      signal(SAB);   //通知A(或B)盘子一空(可能唤醒A或B)
      消费该苹果;
    }
  }
  process B:    //进程B的算法描述
```

```
  {
    while(true) {
      取一个梨子；
      wait(SAB);   //测试盘子是否为空
      将一梨子放入盘中；
      signal(SD)   //通知D盘中已有梨子(可能唤醒D)
    }
  }
  process D:
  {
    while(true) {
      wait(SD);   //测试盘子是否有梨子
      从盘中取出梨子;
      signal(SAB);   //通知A(或B)盘子一空(可能唤醒A或B)
      消费该梨子;
    }
  }
  parend
end
```

(2) 采用管程的同步算法如下：

首先定义管程MPC，该管程可描述如下：

```
type MPC=monitor
  var flag：integer;//flag=0:盘中无水果；=1盘中有苹果；=2盘中有梨子
  empty：condition;     //用于A或B等待空盘子
  W: array[1..2] of condition  //W[1]用于等待苹果，W[2]用于等待梨子
  procedure entry put(integer k)
   begin
         if flag>0 then empty.wait; //生产者A或B进程阻塞
         flag=k;
         放一k号水果入盘中;     //设1号水果为苹果，2号水果为梨子
         if W[k].queue then full.signal;  //若有等待k号水果者，则唤醒之
   end
 procedure entry get(integer k)
   begin
     if flag<>k then W[k].wait;    //消费者C或D进程阻塞
     从盘中取k号水果;
     flag := 0;
     if empty.queue then empty.signal;  //若等待队列非空，则唤醒队首的一个生产者进程
  end
begin
  flag :=0;  //初始化内部数据
end
```

A、B、C、D四个进程的同步算法可描述如下：

```
parbegin
  Process A
```

```
    begin
        任取一个苹果;
        MPC.put(1);
    end
    Process B
    begin
        任取一个梨子;
        MPC.put(2);
    end
    Process C
    begin
        MPC.get(1);
        吃苹果;
    end
    Process D
    begin
        MPC.get(2);
        吃梨子;
    end
parend
```

7. 设自行车生产车间有两个货架，货架A可以存放8个车架，货架B可以存放20个车轮；又设有4个工人，他们的活动是重复劳动，分别为：工人1 加工一个车架放入货架A中；工人2、3分别加工车轮放入货架B中（每人每次放入1个车轮）；工人4从货架A中取一个车架，再从货架B中取两个车轮，组装成一辆自行车。试用PV操作实现四个工人的合作。

【分析】信号量Aempty表示货架A的空位数，其初值为8；信号量Afull表示货架A上存放的车架数，其初值为0；信号量Bempty表示货架B的空位数，其初值为20；信号量Bfull表示货架B上存放的车轮数，其初值为0；信号量mutex用于互斥（初值为1）。

解：BEGIN

```
    semaphore Aempty，Bempty，Afull，Bfull，mutex；
    Aempty：=8；Bempty：=20；Afull：=0；Bfull：=0；mutex：=1；
    PARBEGIN
    Worker1：BEGIN
                L1：生产1个车架；
                    P（Aempty）；          //测试货架A是否有空位置
                    P（mutex）；           //互斥使用货架A
                    车架放到货架A；
                    V（Afull）；           //货架A上的车架数增1，必要时唤醒等待的进程
                    V（mutex）；
                    goto L1；
                END
    Worker2、3：BEGIN
                  L2：生产1个车轮；
                      P（Bempty）；        //测试货架B是否有空位置
                      P（mutex）；         //互斥使用货架B
```

```
                车轮放到货架B；
                V（Bfull）；          //货架B上的车轮数增1，必要时唤醒等待的进程
                V（mutex）；
                goto L2；
            END
    Worker4：BEGIN
            L3：P（Afull）；                  //测试货架A上是否有车架
                P（Bfull）；P（Bfull）；      //测试货架B上是否有2个车轮
                P（mutex）；
                取1个车架；取2个车轮；
                V（Aempty）；                 //货架A空位置增1
                V（Bempty）；V（Bempty）；//货架B空位置增2
                V（mutex）；
                组装成一辆自行车；
                goto L3；
            END
        PAREND
    END
```

8. 假定有一个成品仓库，总共能存放8台成品，生产者进程把生产成品放入仓库，消费者进程从仓库中取出成品消费。为了防止积压，仓库满时就停止生产。由于仓库搬运设备只有一套，故成品的存入和取出只能分别进行，试用P、V操作来实现该方案。

解：

```
semaphore mutex, empty, full ;
mutex=1;       //互斥信号量
empty=8;       //生产者进程的同步信号量
full=0;        //消费者进程的同步信号量
parbegin
  process Pi  //生产者进程
  {
    while (1) {
      生产一个成品x;
      P(empty)      //看看仓库是否还有空间可放成品
      P(mutex);     //互斥使用搬运设备
      用搬运设备将成品放入仓库;
      V(full);      //仓库中成品数增1(可能唤醒一个消费者)
      V(mutex);
    }
  }
  process Cj  //消费者进程
  {
    while (1) {
      P(full)       //看看仓库是否有成品
      P(mutex);     //互斥使用搬运设备
      用搬运设备将成品从仓库取出;
      V(emtpy);     //仓库中可放成品数增1(可能唤醒一个生产者)
      V(mutex);
```

```
        }
    }
parend
```

9. 有三个进程 R、W1、W2 共享一个缓冲器 B，而 B 中每次只能存放一个数。当 B 中无数时，进程 R 可将从输入设备上读入的数存放到缓冲器 B 中；若存放到 B 中的是奇数，则允许进程 W1 将其取出打印；若存放到 B 中的是偶数，则允许进程 W 2 将其取出打印；同时规定：进程 R 必须等缓冲器中的数被取出后才能再存放下一个数；进程 W1 或 W2 对每次存入缓冲器的数只能打印一次；W 1 和 W 2 都不能从空的缓冲器中取数。用 P、V 操作作为同步机制写出三个并发进程的同步算法。(动作部分可用文字描述)

解：

```
semaphore S,S1,S2;
S=1; S1=S2=0;
parbegin
Process R
{
    while (1) {
        从输入设备上读入的数 x;
        P(S);
        B=x;
        if (x%2==1)  V(S1);     //若是奇数，则通知 W1
        else  V(S2);            //若是偶数，则通知 W2
    }
}
Process W1
{
    while (1) {
        P(S1);    //看看缓冲器 B 中是否有奇数
        y=B;      //从缓冲器 B 中取奇数存于 y
        V(S);     //通知 R，缓冲器已空，可以在往里存数了
        Print(y); //打印
    }
}
Process W2
{
    while (1) {
        P(S2);    //看看缓冲器 B 中是否有偶数
        y=B;      //从缓冲器 B 中取偶数存于 y
        V(S);     //通知 R，缓冲器已空，可以在往里存数了
        Print(y);
    }
}
parend
```

10. 进程 P1 使用缓冲区 buffer 向进程 P2，P3，P4 发送消息（如图 2-1 所示），要求每当 Pl 向 buffer 中发消息时，只有当 P2，P3，P4 进程都读取这条消息后 P1 才可向 buffer 中发送新的消息。试用信号量机制描述如下图所示进程的动作过程。（本题是下题的特例）

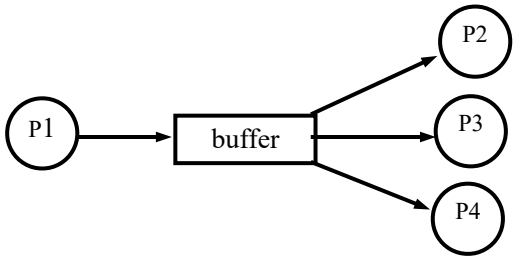


图 2-1

解：设P1、P2、P3、P4的资源信号量分别为S1、S2、S3、S4

```
semaphore S1,S2,S3,S4;
S1.value=3;S2.vale=S3.vale=S4.value=0;    （3分）
parbegin
process P1
{
  while (condition)
  {
    P1生成一个消息;
    P（S1）; P（S1）; P（S1）;
    P1将消息存入缓冲区buffer;
    V（S2）; V（S3）; V（S4）;
  }
}
process Pi(i=2,3,4)
{
  while （condition）
  {
    P（Si）;
    Pi从buffer中取出消息;
    V（S1）;
    Pi消费（使用）该消息;
  }
}
parend
```

另一解法如下（此法更一般化，可用来解下题）：

```
semphore S1[3], S[3];
for (i=0;i<3;i++) {
     S1[i]=1;
     S[i]=0;
}
parbegin
process P1
{
    while (1) {
        P1 生成一个消息;
        for (i=0;i<3;i++)  P(S1[i]); //看看 P2～P4 是否将消息取走
```

```
            P1 将消息存入缓冲区 buffer;
            for (i=0;i<3;i++) V(S[i]);  //通知 P2～P4 缓冲区 buffer 中已有消息
        }
    }
    Process P2
    {
        while(1) {
            P(S[0]);     //看看 buffer 中是否已有消息
            从 buffer 中取出消息;
            V(S1[0]);    //通知 P1, P2 已从缓冲区 buffer 中取走消息
            消费（使用）该消息;
        }
    }
    Process P3
    {
        while(1) {
            P(S[1]);     //看看 buffer 中是否已有消息
            从 buffer 中取出消息;
            V(S1[1]);    //通知 P1, P3 已从缓冲区 buffer 中取走消息
            消费（使用）该消息;
        }
    }
    Process P4
    {
        while(1) {
            P(S[2]);     //看看 buffer 中是否已有消息
            从 buffer 中取出消息;
            V(S1[2]);    //通知 P1, P4 已从缓冲区 buffer 中取走消息
            消费（使用）该消息;
        }
    }
    parend
```

11. (北大 1994 年试题)进程 A1，A2，...，An1 通过 m 个缓冲区向进程 B1，B2，...，Bn2 不断地发送消息。发送和接收工作遵循如下规则：

    1　每个发送进程一次发送一个消息，写入一个缓冲区，缓冲区大小与消息长度一样；

    2　对每个消息，B1，B2，...，Bn2都需各接收一次，读入各自的数据区内；

    3　m个缓冲区都满时，发送进程等待，没有可读的消息时，接收进程等待。

    试用P、V操作组织正确的发送和接收操作。（上题是本题的特例）

【分析】这是P-C问题变形。把这一组缓冲区看成n2组缓冲区。

解：设置一个信号量mutex实现诸进程对缓冲区的互斥访问；两个信号量数组empty[n2]和full[n2]描述n2组缓冲区的使用情况。mutex初值为1，数组empty的元素初值为m，数组full的元素初值为0。

```
var mutex: semaphore :=1;
empty,full: array[0..n2-1] of semaphore;
i: integer;
```

```
for (i=0;i<n2;i++)
{
    empty[i]=m;full[i]=0;
}
Aj ( )      //j=1,2,...,n1
{
   while (1)  {
                . . . . . .
        for (int i=0;i<n2;i++)
             P(empty[i]);
        P(mutex);
        将消息放入缓冲区；
         V(mutex);
         for (i=0;i<n2;i++)
              V(full[i]);
                . . . . . .
     }
}
Bi ( )      //i=1,2,...,n2
{
   while (1)
   {
                . . . . . .
        P(full[i]);
        P(mutex);
        将消息从缓冲区取出；
         V(mutex);
         V(empty[i]);
                . . . . . .
     }
}
parbegin
   A1( );
   A2( );
      . . . . . .
   An1( );
   B1( );
   B2( );
      . . . . . .
   Bn2( );
parend
```

12. 进程 P1 使用缓冲区 buffer 向进程 P2，P3，P4 发送消息（如第 10 题的图 2-1 所示），要求每当 Pl 向 buffer 中发消息时，只有当 P2，P3，P4 进程都读取这条消息后 P1 才可向 buffer 中发送新的消息。试用管程机制描述四个进程的动作过程。

解：定义管程如下(用伪代码)：

```
type P_C=monitor
  var flag : array[2..4] of integer;
  cc: array[1..4] of condition;
procedure put( )  //P1调用此过程将消息放入缓冲区
   begin
        if flag[2]+flag[3]+flag[4]>0 then cc[1].wait;
        将消息放入缓冲区buffer;
        for i := 2 to 4 do
            begin
                flag[i] := 1;
                if cc[i].queue then cc[i].signal;
             end
   end
procedure get(integer i)  //P2、P3、P4调用此过程从buffer中取消息
   begin
         if flag[i]=0 then cc[i].wait;        //若buffer中无消息则等待
         从buffer中取消息;
         flag[i] := 0;
         if flag[2]+flag[3]+flag[4] = 0 and cc[1].queue then cc[1],signal;
    end
begin
     flag[2] := flag[3] := flag[4] := 0;            //初始化数据
end
采用管程描述P1、P2、P3、P4四个进程的活动如下：
parbegin
    P1( )
    begin
       repeat
            生成一个消息;
            P_C.put( );
       until false
    end
    P2( )
    begin
       repeat
            P_C.get(2);
            消费从缓冲区中取到的消息;
       until false
    end
    P3( )
    begin
       repeat
            P_C.get(3);
            消费从缓冲区中取到的消息;
```

```
        until false
    end
    P4( )
    begin
        repeat
            P_C.get(4);
            消费从缓冲区中取到的消息;
        until false
    end
parend
```

13. 某自动质量检测系统有三个进程Q、A、B组成。进程Q每次取一件产品检测，把检测后的产品存放在货架F上，F的容量为每次只能存放一件产品。若货架上存放的是合格产品则让进程A取出，并在产品上贴标签后包装；若货架上存放的是不合格产品则让进程B取出后，将其丢入废物箱。回答下列问题：

（1）写出用PV操作管理时应定义的信号量及初值：　①　。

（2）完成下列算法中的填空，使它们能按上述要求正确地并发执行。

进程Q:	进程A:	进程B：
取一件产品检测;	⑤;	⑦;
②;	y:=F中产品；	z:=F中产品;
F:=检测后的产品	⑥;	⑧;
If F=合格产品 then ③	对产品贴标签且包装;	把产品丢入废物箱;
else ④		

解：(1) ①定义信号量empty，fulla，fullb，初值分别为1，0，0。

(2) ② P(empty);　③ V(fulla);　④ V(fullb);

⑤ P(fulla);　⑥ V(empty);

⑦ P(fullb);　⑧ V(empty);

14. 系统有三个进程Read,Write1,Write2共享一个整数缓冲器B，B中每次只能存放一个整数。Read进程每次启动输入设备输入一个整数到n。若n中是奇数，则由进程Write1将其取出打印；若n中是偶数，则由进程Write2将其取出打印。规定输入与打印整数的个数和次序完全一致。

要求：（1）完善如下程序，在下列A、B空白处填入有关语句，并说明物理意义。

```
begin S, SO, SE: semaphore;
        n: integer;
        S:=1;
        SO:=0;
        SE:=0;
    Parbegin
    process Read
    Begin
     L1:从输入设备读一整数到n;
        P(S);
        B:=n;
        if n mod 2=1 then V(SO)
        else V(SE);
    goto L1
    end;
```

```
process Write1
begin
  L2: P(SO);
    Y:=B;
    ____(A)____;
    print Y;
    goto L2
end;
process Write2
begin
    L3:____(B)____;
      Z:=B;
      V(S);
      Print Z;
      goto L3
end;
Parend;
end;
```

（2）说明信号量S，SO，SE作用及它们的初值的物理意义。

（3）Read进程中V(SO)与V(SE)对调，程序功能将发生什么变化。

解：(1) (A) V(S)　　(B) P(SE)

(2) 信号量S，SO，SE作用是实现进程Read、Write1和Write2之间的同步。信号量S的初值为1，表示开始时缓冲器B是空的，可以存放一个整数，当缓冲器B中存有整数时，其值变为0；信号量SO的初值为0，表示开始时缓冲器B中没有奇数，当缓冲器B中存有一个奇数时，SO的值变为1；信号量SE的初值为0，表示开始时缓冲器B中没有偶数，当缓冲器B中存有一个偶数时，SE的值变为1。

(3) Read进程中V(SO)与V(SE)对调，程序功能将变为：若缓冲器中存放的整数n为奇数时，则由进程Write2将其取出打印；若n中是偶数，则由进程Write1将其取出打印。

15. (2009全国试题)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( )送入缓冲区某个单元中；P2每次用getodd( )从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。（本题与第9题、第14题实际上是同一题）

解：定义P1的资源信号量empty来表示缓冲区中空单元个数，用于P1与P2、P3的同步；定义P2的资源信号量S1来表示缓冲区中奇数的个数，用于P2和P1的同步；定义P3的资源信号量S2来表示缓冲区中偶数的个数，用于P3和P1的同步；定义互斥信号量mutex，用于三个进程互斥访问缓冲区。算法描述如下：

```
var empty,s1,s2,mutex: semaphore := N,0,0,1;
Parbegin
P1: begin
      x := produce( );          /* 生成一个数 */
      P(empty);          /* 判断缓冲区是否有空单元 */
      P(mutex);          /* 是否有进程访问缓冲区 */
      put( );            /* 将生成的数送入缓冲区的某个单元 */
```

```
        if x mod 2=0 then
            V(S2);          /* 如果是偶数，向P3发出信号 */
        else
            V(S1);          /*如果是奇数，向P2发出信号 */
        V(mutex);
    end
P2：begin
        P(S1);              /* 缓冲区中是否有奇数 */
        P(mutex);
        getodd( );
        V(empty);           /* 向P1发出信号 */
        V(mutex);
        countodd( );        /* 原答案将此行放在临界区中，因只有P2调用，不需互斥，故移至此 */
    end
P3：begin
        P(S2);
        P(mutex);
        geteven( );
        V(empty);
        V(mutex);
        counteven( );
    end
Parend
```

16. 设有n个缓冲区构成的循环缓冲区，每个缓冲区能容纳一个整数。写进程Writer把整数逐个存入缓冲区，读进程Reader则逐个从缓冲区中读出并打印输出，要求打印的与输入的完全一样，即个数、次序、数值一样。试问：

（1）写进程与读进程间具体的制约关系如何？

（2）用PV操作写出这两个进程的同步算法程序。

解：(1) 写进程与读进程间具体的制约关系是同步和互斥关系。

(2) 采用PV操作的同步算法程序如下：

```
semaphore mutex, empty, full;
mutex=1;        //互斥信号量，用于两个进程互斥访问缓冲区
empty=n;        //同步信号量，表示空闲缓冲区的数量
full=0;         //同步信号量，表示放有整数的缓冲区个数
parbegin
    process Writer( )
    {
        while (1) {
            produce_an_integer( );
            P(empty);
            P(mutex);
            write_an_integer_to_buffer( );
            V(mutex);
            V(full);
```

```
        }
    }
    process Reader( )
    {
        while (1) {
            P(full);
            P(mutex);
            get_an_integer_from_buffer( );
            V(mutex);
            V(empty);
            print_an_integer( );
        }
    }
parend
```

17. 一组生产者进程和一组消费者进程共享9个缓冲区，每个缓冲区可以存放一个整数。生产者进程每次一次性向3个缓冲区写入整数，消费者进程每次从缓冲区取出一个整数。请用：(1)信号量和P、V操作；(2)管程，写出并发进程能正确执行的算法程序。

解：(1) 信号量和P、V操作的算法描述如下：

```
semaphore empty=9;          //生产者的资源信号量，表示空缓冲区个数
seamphore full=0;           //消费者的资源信号量，表示满缓冲区个数
seamphore mutex=1;          //互斥信号量，用于互斥访问共享变量i或j
int i=0;                    //生产者使用是缓冲区下标
int j=0;                    //消费者使用是缓冲区下标
parbegin
process producer_m  (m=1, 2, 3, …)
begin
  int a,b,c;                          //该生产者使用的局部变量
  repeat
    生产者获得3个整数a, b, c ;
    P(empty); P(empty); P(empty);     //得到3个空缓冲区
    P(mutex);                         //生产者互斥访问共享变量i
    buffer[i]=a;                      //整数a存入缓冲区
    i=(i+1) mod 9;
    buffer[i]=b;                      //整数b存入缓冲区
    i=(i+1) mod 9;
    buffer[i]=c;                      //整数c存入缓冲区
    i=(i+1) mod 9;
    V(mutex);
    V(full); V(full); V(full);        //通知消费者缓冲区已有数据（满缓冲区数增3）
  until false
end
process consumer_k  (k=1, 2, 3, …)
begin
  int x;
```

```
  repeat
    P(full);                          //看看缓冲区是否有整数
    P(mutex);                         //消费者互斥访问共享变量j
    x=buffer[j];                      //从缓冲区取整数到x中
    j=(j+1) mod 9
    V(mutex);
    V(empty);                         //空缓冲区数增1
    consume  x;
  until false
end
parend
```

（2）采用管程机制解决上述问题。

首先定义管程PC，该管程可描述如下：

```
type PC=monitor
  var i, in, out, count：integer;
  buffer：array[0..8] of integer;
  full, empty：condition;
  procedure entry put(integer a[ ])
   begin
        P: if count+3>n then {
             empty.wait; //生产者进程阻塞
             goto P;
             }
        for (i=0; i<3; i++) {        //一次连续放3个整数
      buffer[in] := a[i];
      in := (in+1) mod 9;
      count := count+1;
         }
         for (i=0;i<3;i++)
         if full.queue then full.signal;  //若等待队列非空，则唤醒一个消费者(最多唤醒3个)
   end
 procedure entry get(int x)
   begin
      if count≤0 then full.wait;     //消费者进程阻塞
      x := buffer[out];
      out := (out+1) mod 9;
      count := count-1;
      if empty.queue then empty.signal;  //若等待队列非空，则唤醒队首的一个生产者进程
  end
begin
  in :=out :=0; count :=0;  //初始化内部数据
end
```

生产者和消费者进程可描述如下：

```
process producer ( )              //生产者进程
```

```
begin
  a: array[0..2] of integer;
    repeat
       生成3个整数存于数组a中;
       PC.put(a);
  until false;
end
process consumer ( )            //消费者进程
begin
  var x:integer;
  repeat
       PC.get(x);
       consume x;
  until false;
end
```

18. 有n个输入进程、m个计算进程和p个输出进程，通过循环缓冲区A和循环缓冲区B进行数据传送，如下图2-2所示。

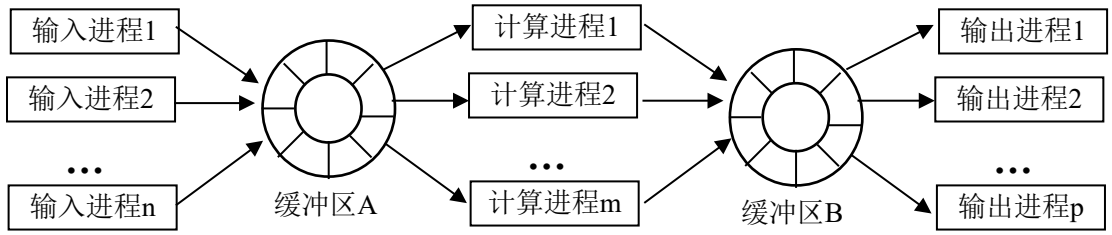


图2-2

已知缓冲区A有N个缓冲块，缓冲区B有M个缓冲块。输入进程每次输入1个数据块存入缓冲区A的1个缓冲块中；计算进程每次从缓冲区A取出1个数据块，处理后的数据块存入缓冲区B的1个缓冲块中；输出进程每次从缓冲区B中取出1个数据块进行输出操作。试用P、V操作实现进程间的同步与互斥。

【说明】n个输入进程和m个计算进程构成一对生产者-消费者问题；m个计算进程和p个输出进程构成另一对生产者-消费者问题。计算进程担当双重较色，即既是消费者，又是生产者。

解：

```
semaphore mutex1, mutex2, empty1, full1, empty2, full2;
int in1, out1, in2, out2;
mutex1=1;            //互斥信号量，用于互斥访问共享变量in1和out1
mutex2=1;            //互斥信号量，用于互斥访问共享变量in2和out2
empty1=N;            //同步信号量，表示缓冲区A的空缓冲区个数
empty2=M;            //同步信号量，表示缓冲区B的空缓冲区个数
full1=0;             //同步信号量，表示缓冲区A的满缓冲区个数
full2=0;             //同步信号量，表示缓冲区B的满缓冲区个数
in1=out1=in2=out2=0;  //共享变量，表示缓冲区的下标变量
parbegin
    process inputi( )      //n个输入进程，i=1, 2, . . . , n
    {
        while (1) {
```

```
            读入一个数据块X;
            P(empty1);          //判断缓冲区A是否有空闲位置放数据块，若无则等待
            P(mutex1);          //互斥访问共享变量in1和缓冲区A
            bufferA[in1]=X;
            in1=(in1+1)%M;
            V(mutex1);
            V(full1);           //通知计算进程，缓冲区A中已有数据(可能唤醒计算进程)
        }
    }
    process computej( )         //m个计算进程，j=1, 2, . . . , m
    {
        while (1) {
            P(full1);           //判断缓冲区A中是否有可取的数据块，若无则等待
            P(mutex1);          //互斥访问out1和缓冲区A
            Y=bufferA[out1];
            out1=(out1+1)%M;
            V(mutex1);
            V(empty1);          //通知输入进程，缓冲区A中已被取走一个数据块
            处理数据块Y;
            P(empty2);          //判断缓冲区B是否有空位置存放数据块，若无则等待
            P(mutex2);          //互斥访问共享变量in2和缓冲区B
            bufferB[in2]=处理后的数据块Y;
            in2=(in2+1)%N;
            V(mutex2);
            V(full2);       //通知输出进程，缓冲区B中已有数据(可能唤醒输出进程)
        }
    }
    process outputk( )          //p个输出进程，k=1, 2, . . . , p
    {
        while (1) {
            P(full2);           //判断缓冲区B中是否有可取的数据块，若无则等
            P(mutex2);          //互斥访问共享变量out2和缓冲区B
            Z=bufferB[out2];
            out2=(out2+1)%N;
            V(mutex2);
            V(empty2);          //通知计算进程，缓冲区B中已被取走一个数据块
        }
    }
parend
```

消费者角色

生产者角色

19. 假定有一个信箱可存放N封信。当信箱不满时，发信者可把信件送入信箱；当信箱中有信时，收信者可以从信箱中取信。用指针r、k分别刻存信和取信的位置，请用管程(Monitor)来管理这个信箱，使发信者和收信者能正确工作。

解：引入条件变量empty和full，当信箱满时，发信者在empty上阻塞(调用empty.wait)；当信箱空时，收信者在full上阻塞。

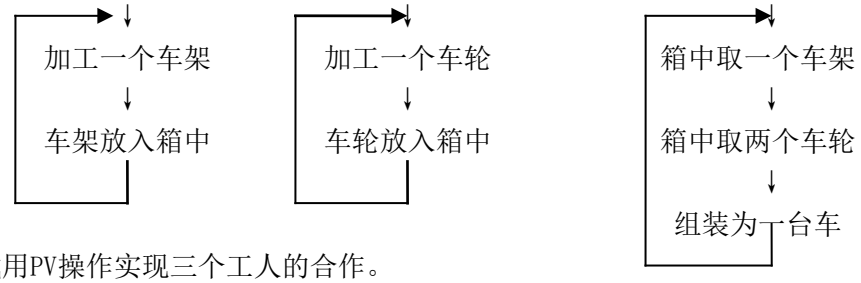
发信者将一信件送入信箱后，若有收信者阻塞，则唤醒一个收信者(调用full.signal)；收信者取走一

信件后，若有法信者阻塞，则唤醒一个发信者(调用empty.signal)。
引入整型变量count，用于对信箱中信件计数。
//管程定义如下：

```
type MailBoxManager=monitor
  var r,k,count:integer;
  mailbox: array[0..N-1] of message;
  full,empty: condition;
  procedure entry get( meg )
  begin
     if (count=0) then full.wait; //收信者阻塞
     meg:=mailbox[r];
     r:= (r+1) mod N;
     count:=count-1;
     if empty.gueue then empty.signal;//唤醒一个发信者
 end
  procedure entry put( mg )
  begin
     if (count=N) then empty.wait;//发信者阻塞
     mailbox[k]:=mg;
     k:= (k+1) mod N;
     count:=count+1;
     if full.queue then full.signal; //唤醒一个收信者
  end
  begin  //初始化部分
     r:=0; k:=0; count:=0;
  end
//发信者和收信者可描述为：
 Sender:  //发信者进程
 begin
  repeat
       write a message in mg; //写信
       MailBoxManager.put(mg); //发信
  until false;
 end
 Receiver: //收信者进程
 begin
  repeat
       MailBoxManager.get(msg); //收信
       process the message in msg;//处理信件
  until false;
 end
```

20. 设自行车生产线上有一只箱子，其中有N个位置(N≥3)，每个位置可存放一个车架或一个车轮；又设有三个工人，其活动分别为：

工人1活动　　工人2活动　　工人3活动



试用PV操作实现三个工人的合作。

【分析】工人1和工人2向箱子中放车架或车轮的活动，除了受到箱子容量的限制外，还应考虑车架和车轮的数量差，即不允许只放车架或只放车轮，因为那样可能会引起死锁(学生思考：为什么？)。显然，工人1放车架时至少应留出2个放车轮的空位；同样，工人2放车轮时至少应留出1个空位放车架。若count1和count2分别代表车架数和车轮数，则不妨规定-(N-1)≤count1－count2≤N-2

解：semaphore mutex, avail, availa, availb, fulla, fullb;

```
mutex=1;      //互斥信号量，用于访问共享变量count1和count2
avail=N;      //同步信号量，用于表示箱子中空闲位置数
availa=N-2;   //同步信号量，用于表示箱子中可放车架的数量
availb=N-1;   //同步信号量，用于表示箱子中可放车轮的数量
fulla=0;      //同步信号量，用于表示箱子中已放的车架数量
fullb=0;      //同步信号量，用于表示箱子中已放的车轮数量
parbegin
  process worker1( )
  {
     while (1) {
          加工一个车架;
          P(availa);
          P(avail);       //看看箱子中是否还有放车架的位置
          P(mutex);
          车架放入箱中;
          V(mutex);
          V(fulla);       //通知工人3，箱子中多了个车架
          V(availb);
     }
  }
  process worker2( )
  {
     while (1) {
          加工一个车轮;
          P(availb);
          P(avail);       //看看箱子中是否还有放车架的位置
          P(mutex);
          车轮放入箱中;
          V(mutex);
          V(fullb);       //通知工人3，箱子中多了个车架
          V(availa);
     }
  }
```

```
process worker3( )
{
    while (1) {
        P(availa);
        P(mutex);
        从箱中取一个车架;
        V(mutex);
        V(avail);
        P(availb);
        P(mutex);
        从箱中取一个车轮;
        P(mutex);
        V(avail);
        P(availb);
        P(mutex);
        从箱中再取一个车轮;
        V(mutex);
        V(avail);
        组装为一台车;
    }
}
parend
```

21. 利用管程解决教科书上的生产者-消费者问题。

利用管程解决生产者-消费者问题，首先要为它们建立一个管程，命名为Produer_Consumer，简称为PC。其中包括两个过程：

① put(item)过程。生产者利用该过程将自己生产的产品投放到缓冲池中，并用整型变量count来表示在缓冲池中已有的产品数目，当count≥n时，表示缓冲池已满，生产者等待(阻塞)。

② get(item)过程。消费者利用该过程从缓冲池中取出一个产品，当count≤0时，表示缓冲池已无可取的产品，消费者应等待。

PC管程可描述如下：

```
type PC=monitor
  var in,out,count: integer;
  buffer: array[0..n-1] of item;
  full,empty: condition;
  procedure entry put(item)
    begin
      if count≥n then empty.wait; //生产者进程阻塞
      buffer[in] := nextp;
      in := (in+1) mod n;
      count := count+1;
      if full.queue then full.signal; //若等待队列非空，则唤醒队首的一个消费者进程
    end
  procedure entry get(item)
    begin
      if count≤0 then full.wait;
```

```
        nextc := buffer[out];
        out := (out+10 mod n;
        count := count-1;
        if empty.queue then empty.signal; //若等待队列非空，则唤醒队首的一个生产者进程
    end
begin
    in :=out :=0; count :=0; //初始化内部数据
end
```

生产者和消费者可描述如下：

```
process producer ( )              //生产者进程
begin
  repeat
      produce an item in nextp;
      PC.put(item);
  until false;
end
process consumer ( )              //消费者进程
begin
  repeat
      PC.get(item);
      consume the item in nextc;
  until false;
end
```

22. (北邮1998年试题)某庙寺有小和尚、老和尚若干。有一水井和一个水缸，由小和尚提水入缸供老和尚饮用。水缸可容纳10桶水，水取自同一井中。水井很窄，每次只能容一个水桶打水。水桶总数为3个。每次入水、取水仅为1桶水，且不可同时进行。试用一种同步机制，写出小和尚和老和尚入水、取水的活动过程。

解：采用信号量机制。

```
semaphore mutex, empty, full, S;
mutex=1;       //互斥信号量
empty=10;      //小和尚的资源信号量，用于与老和尚同步，假设开始时水缸为空
full=0;        //老和尚的资源信号量，用于与小和尚同步
S=3;       //水桶资源信号量
parbegin
   小和尚i  (i=1, 2, . . . , m)        //m个小和尚进程
   {
        while (1) {
            P(S);          //取一水桶，准备入水
            P(empty);  //看看水缸是否还有空间入水
            P(mutex);    //互斥
            从水井取水，倒入水缸中;
            V(mutex);
            V(full);       //通知老和尚，水缸中已增加了一桶水
            V(S);          //释放水桶
```

```
        }
    }
    老和尚j  (J-1, 2, ... , n)          //n个老和尚进程
    {
        while (1) {
            P(S);          //取一水桶，准备取水
            P(full);       //看看水缸中是否有水
            P(mutex);
            从水缸中取一桶水;
            V(mutex);
            V(empty);   //水缸增加一个桶空间
            V(S);          //释放水桶
            饮用水;
        }
    }
parend
```

23. 有如图2-3所示的工作模型：

三个进程P_0、P_1、P_2和三个缓冲区B_0、B_1、B_2，进程间借助相邻缓冲区传递消息：P_0每次从B_0中取出一条消息经加工后送入B_1中，P_1每次从B_1中取出一条消息经加工后送入B_2中，P_2每次从B_2中取出一条消息经加工后送入B_0中。B_0，B_1，B_2分别可存放3，2，2个消息。初始时B_0中有2个消息，B_1，B_2中各有1个消息。用P、V操作写出P_0，P_1，P_2的同步及互斥流程。

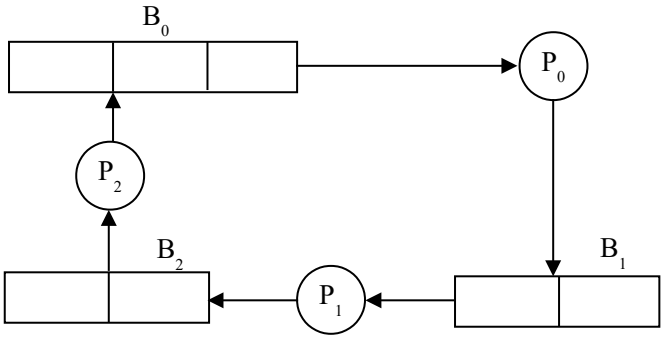


图2-3

解：
```
semaphore empty0,full0,empty1,full1,empty2,full2,mutex;
empty0=1;full0=2; //冲区B0有2个消息，还可放1个消息
empty1=1; full1=1;       //冲区B1有1个消息，还可放1个消息
empty2=1; full2=1;       //冲区B2有1个消息，还可放1个消息
mutex=1;       //互斥信号量
parbegin
    Process P0
    {
        while (1) {
            P(full0);       //看看B0中是否有消息
            P(mutex);     //互斥访问B0
            从缓冲区B0中取一个消息x;
            V(mutex);
```

```
            V(empty0);  //B0中空出一个存放消息的位置
            加工消息x;
            P(empty1);  //看看B1中是否可放一个消息
            P(mutex);   //互斥访问B1
            将加工后的x存入缓冲区B1;
            V(mutex);
            V(full1);   //B1中增加一个消息
        }
    }
    Process P1
    {
        while (1) {
            P(full1);           //看看B1中是否有消息
            P(mutex);           //互斥访问B1
            从缓冲区B1中取一个消息y;
            V(mutex);
            V(empty1);          //B1中空出一个存放消息的位置
            加工消息y;
            P(empty2);          //看看B2中是否可放一个消息
            P(mutex);           //互斥访问B2
            将加工后的x存入缓冲区B2;
            V(mutex);
            V(full2);           //B2中增加一个消息
        }

    }
    Process P2
    {
        while (1) {
            P(full2);   //看看B2中是否有消息
            P(mutex);   //互斥访问B2
            从缓冲区B2中取一个消息z;
            V(mutex);
            V(empty2);  //B2中空出一个存放消息的位置
            加工消息z;
            P(empty0);  //看看B0中是否可放一个消息
            P(mutex);   //互斥访问B0
            将加工后的z存入缓冲区B0;
            V(mutex);
            V(full0);   //B0中增加一个消息
        }
    }
parend
```

24. 现有3个生产者P1、P2、P3，他们都要生产桔子水，每个生产者都已分别购得两种不同原料，待购得

第三种原料后就可配制成桔子水，装瓶出售。有一供应商能源源不断地供应糖、水、桔子精，但每次只拿出一种原料放入容器中供给生产者。当容器中有原料时需要该原料的生产者可取走，当容器空时供应商又可以放入一种原料。假定：

生产者P1已购得糖和水；

生产者P2已购得水和桔子精；

生产者P3已购得糖和桔子精。

试用：1）管程；2）信号量和P、V操作，写出供应商和3个生产者之间能正确同步的算法。

解：(1) 采用管程机制

```
type PC=monitor          //定义管程，命名为PC
var flag : integer;
C : array[0. . 3] of condition;
procedure entry put ( )        //供应商调用此过程向容器中放一种原料
begin
   if (flag>0) then C[0]. wait;          //若容器不空，供应商阻塞
   随机地向容器中放一种原料x;
   if (x是桔子精) then begin
      flag := 1;                  //置容器中是桔子精的标志
      if C[1]. queue then C[1]. signal       //若生产者P1正在等待桔子精，则唤醒之
   end
   else if (x是糖) then begin
      flag := 2;                  //置容器中是糖的标志
      if C[2]. queue then C[2]. signal       //若生产者P2正在等待糖，则唤醒之
   end
   else begin
       flag := 3;                 //置容器中是水的标志
       if C[3]. queue then C[3]. signal       //若生产者P3正在等待水，则唤醒之
   end
end
procedure entry get (int k)   //生产者调用此过程从容器中取一原料
begin
   if (flag≠k) then C[k]. wait;   //若容器中没有自己想要的原料则等待
   从容器中取出自己所需的原料;
   flag := 0;        //置容器已空的标志
   if C[0]. squeue then C[0]. signal;   //若供应商正在等待，则唤醒之
end
begin
   flag := 0;        //初始化，开始时容器为空
end
```

采用管程机制时，同步的算法描述如下：

```
process 供应商( )
begin repeat
     PC. put ( );          //向容器中放入一种原料
   until false
end
process P1( )
```

```
        begin repeat
            PC.get (1);          //从容器中取桔子精
            用三种原料配制成桔子水，装瓶出售;
          until false;
        end
        process P2( )
        begin repeat
            PC.get (2);          //从容器中取糖
            用三种原料配制成桔子水，装瓶出售;
          until false;
        end
        process P3 ( )
        begin repeat
            PC.get (3);          //从容器中取水
            用三种原料配制成桔子水，装瓶出售;
          until false;
        end
        parbegin
            供应商( );
            P1 ( );
            P2 ( );
            P3 ( );
        parend
```

(2) 采用信号量机制

```
semaphore empty, ful1a, fullb, fullc;
empty=1;      //开始时容器是空的，可以放一种原料
fulla=0;      //开始时容器中无桔子精，用于阻塞P1
fullb=0;      //开始时容器中无糖，用于阻塞P2
fullc=0;      //开始时容器中无水，用于阻塞P3
parbegin
   process 供应商
   {
      while (true) {
         随机地取一种原料x;
         P(empty);          //看看容器是否空，不空则等待
         将x放入容器中;
         if (x是桔子精) V(fulla);//通知（或唤醒）P1
         else if (x是糖) V(fullb);      //通知（或唤醒）P2
         else V(fullc);  //通知（或唤醒）P3
      }
   }
   process P1
   {
      while (true) {
         P(fulla);      //看看容器中是否有桔子精，若无则阻塞
```

```
            从容器中取出桔子精;
            V(empty);     //通知供应商，容器空了。若供应商因容器不空而阻塞，则唤醒之
            用三种原料配制成桔子水，装瓶出售;
        }
    }
    process P2
    {
        while (true) {
            P(fullb);       //看看容器中是否有糖，若无则阻塞
            从容器中取出糖;
            V(empty);     //通知供应商，容器空了。若供应商因容器不空而阻塞，则唤醒之
            用三种原料配制成桔子水，装瓶出售;
        }
    }
    process P3
    {
        while (true) {
            P(fullc);       //看看容器中是否有水，若无则阻塞
            从容器中取出糖;
            V(empty);     //通知供应商，容器空了。若供应商因容器不空而阻塞，则唤醒之
            用三种原料配制成桔子水，装瓶出售;
        }
    }
parend
```

25. 有一材料保管员，他保管笔和纸若干。有A、B两组学生，A组学生每人都备有纸，B组学生每人备有笔。任一学生只要能得到另一种材料就可以写信。有一个可以放一张纸或一支笔的小盒，当小盒中无物品时，保管员就可任意放一张纸或一支笔共学生取用，每次允许一个学生从中取出自己所需要的材料，当学生从盒子中取走材料后允许保管员再放一件材料，请用：1）信号量和P、V操作；2）管程，写出他们并发执行时能正确工作的算法程序。

解：(1) 采用信号量和P、V操作

```
semaphore empty, fulla, fullb;
empty=1; fulla=fullb=0;
parbegin
  process 保管员
  {
      while (true) {
          任取一种物品x;
          P(empty);       //准备往小盒中放物品，若小盒不空则阻塞
          将x放入小盒中;
          if (x是笔) V(fulla);       //通知A组学生小盒中放了笔(可能唤醒一个A组学生)
          else V(fullb);         //通知B组学生小盒中放了纸(可能唤醒一个B组学生)
      }
  }
  process Ai  ( i = 1, 2, . . . )       //A组学生进程
```

```
    {
        P(fulla);       //看看小盒中是否有笔，若无则阻塞
        从小盒中取出笔;
        V(empty);    //通知保管员，小盒已空。若保管员因小盒不空而阻塞，则唤醒之
        用笔盒纸写信;
    }
    process Bj  ( j = 1, 2, . . . )        //A组学生进程
    {
        P(fullb);       //看看小盒中是否有纸，若无则阻塞
        从小盒中取出纸;
        V(empty);    //通知保管员，小盒已空。若保管员因小盒不空而阻塞，则唤醒之
        用笔盒纸写信;
    }
parend
```

(2) 采用管程

```
type PC=monitor           //定义管程，命名为PC
var flag : integer;
C : array[0. . 2] of condition;
procedure put ( )           //保管员调用此过程向小盒中放笔或纸
begin
    if (flag>0) then C[0]. wait;              //若小盒不空，保管员阻塞
    随机地向容器中放一种物品x(笔或纸);
    if (x是笔) then begin
        flag := 1;                    //置小盒中是笔的标志
        if C[1]. queue then C[1]. signal        //若有A组学生正在等待笔，则唤醒之
    end
    else begin                        //若x是纸
        flag := 2;                    //置小盒中是纸的标志
        if C[2]. queue then C[2]. signal       //若有A组学生正在等待纸，则唤醒之
    end
end
procedure get (int k)       //A、B组学生调用此过程从小盒中取k号物品
begin
    if (flag≠k) then C[k]. wait;    //若小盒中没有自己想要的物品，则等待
    从小盒中取出自己所需的k号物品;
    flag := 0;          //置容器已空的标志
    if C[0]. squeue then C[0]. signal;    //若供应商正在等待，则唤醒之
end
begin
    flag := 0;          //初始化，开始时小盒为空
end
```

采用管程机制时，同步的算法描述如下：

```
process 保管员( )
    begin repeat
        PC. put ( );        //向小盒中放入一种物品(笔或纸)
```

```
    until false
end
process Ai ( ) ( i = 1, 2, . . . )      //A组学生进程
begin
    PC.get (1);      //从小盒中取笔
    用笔盒纸写信;
end
process Bj ( ) ( j = 1, 2, . . . )      //A组学生进程
begin
    PC.get (2);      //从小盒中取纸
    用笔盒纸写信;
end
parbegin
    保管员( );
    A1 ( );
    A2 ( );
    . . . . . .
    B1 ( );
    B2 ( );
    . . . . . .
parend
```

26. 另一个经典同步问题(patil, 1971)：三个吸烟者在一个房间内，还有一个香烟供应者。为了制造和抽掉香烟，每个吸烟者需要三样东西：烟草、纸和火柴，供应者有丰富的货物提供。三个吸烟者中，第一个有自己的烟草，第二个有自己的纸，第三个有自己的火柴。供应者随机地将两样东西放在桌子上，允许一个吸烟者进行对健康不利的吸烟。当吸烟者完成吸烟后唤醒供应者，供应者再把两样东西放在桌子上，唤醒一个吸烟者。试采用：(1) 信号量和P、V操作；(2) 管程，编写他们同步工作的算法程序。

解：(1) 定义4个信号量用于供应者与3个吸烟者的同步：S0为供应者的同步信号量；S1、S2和S3分别为第一个吸烟者、第二个吸烟者和第三个吸烟者的资源信号量。供应者随机取的两样东西是纸和火柴，则执行V(S1)通知第一个吸烟者；两样东西是烟草和火柴，则执行V(S2)通知第二个吸烟者；两样东西是烟草和纸，则执行V(S3)通知第三个吸烟者。算法描述如下：

```
semaphore S0, S1, S2, S3;
S0=S1=S2=S3=0;
parbegin
供应者：begin
            L0:
                随机地取两样东西x, y放在桌子上;
                if (x, y是纸和火柴) V(S1);       //通知第一个吸烟者
                else if (x, y是烟草和火柴) Vl(S2);   //通知第二个吸烟者
                else V(S3);          //通知第三个吸烟者
                P(S0) ;              //供应者阻塞，等待吸烟者唤醒
                goto L0;
        end
第一个吸烟者：begin
                  L1: P(S1);       //看看供应者是否在桌上放了纸和火柴
```

```
                从桌上取纸和火柴;
                加工香烟;
                抽烟;
                V(S0);      // 唤醒供应者
                goto L1;
            end
第二个吸烟者：begin
              L2: P(S2);      //看看供应者是否在桌上放了烟草和火柴
                从桌上取烟草和火柴;
                加工香烟;
                抽烟;
                V(S0);      // 唤醒供应者
                goto L2;
            end
第三个吸烟者：begin
              L3: P(S1);      //看看供应者是否在桌上放了烟草和纸
                从桌上取烟草和纸;
                加工香烟;
                抽烟;
                V(S0);      // 唤醒供应者
                goto L3;
            end
parend
```

(2) 定义管程SMook如下：

```
type SMook=monitor
var flag : integer;
var C : array [1.. 3] of condition;
var C1 : condition;
procedure entry put( )
begin
    随机地取两样东西x, y放在桌子上;
    if (x, y是纸和火柴) then begin
          flag :=1;          //通知第一个吸烟者
          if C[1].squeue then C[1].signal;      //唤醒第一个吸烟者
          end
    else if (x, y是烟草和火柴) then begin
          flag := 2;          //通知第二个吸烟者
          if C[2].squeue then C[2].signal;      //唤醒第一个吸烟者
          end
    else begin
          flag := 3;          //通知第三个吸烟者
          if C[3].squeue then C[3].signal;      //唤醒第一个吸烟者
          end
    C.signal ;                //供应者阻塞，等待吸烟者唤醒
end
```

```
procedure entry get(int k)
begin
    if flag<>k then C[k].wait;
    从桌上取两样东西;
    flag := 0;
    if C.squeue then C.signal;
end
begin
    flag := 0;
end
```

采用管程机制时，4个进程的行为描述如下：

```
parbegin
  供应者：begin
            repeat
              SMook.put( );
            until false
          end
  吸烟者k (k=1, 2, 3)
    begin
      repeat
        SMook.get(k);
        抽烟;
      until false
    end
parend
```

27. 设有三组进程P_i、Q_j、R_k，其中P_i、Q_j构成一对生产者-消费者，共享一个由M1个缓冲区构成的循环缓冲buf1。Q_j、R_k构成另一对生产者-消费者，共享一个由M2个缓冲区构成的循环缓冲buf2。如果P_i每次生产一个产品投入buf1，Q_j每次从buf1中取两个产品组装成一个后投入buf2，R_k每次从buf2中取出三个产品包装出厂。试采用信号量和P、V操作编写他们同步工作的算法程序。

解：对于P_i、Q_j，设置资源信号量empty1、full1用于它们的同步，互斥信号量mutex1用于它们对下标变量的互斥访问；对于Q_j、R_k，设置资源信号量empty2、full2用于它们的同步，互斥信号量mutex2用于它们对下标变量的互斥访问。同步算法描述如下：

```
semaphore empty1, full1, mutex1, empty2, full2, mutex2;
int in, out, i, j ;
empty1=M1; empty2=M2; mutex1=mutex2=1; full1=full2=0;
in=out=i=j=0;
parbegin
  process Pi (i=1, 2, 3, . . .  )
  {
    while(1) {
      生产一个产品x;
      P(empty1);
      P(mutex1);
      buf1[in]=x;
```

```
            in=(in+1) mod M1;
            V(mutex1);
            V(full1);
        }
    }
    process Qj ( j=1, 2, 3, . . .  )
    {
        while (1) {
            for (m=0; m<2; m++) {
                P(full1);
                P(mutex1);
                y[m]=buf1[out];
                out=(out+1) mod M1;
                V(mutex1);
                V(empty1);
            }
            把y[0]和y[1]组装成一个产品y;
            P(empty2);
            P(mutex2);
            buf2[i]=y;
            i=(i+1) mod M2;
            V(mutex2);
            V(full2);
        }
    }
    process Rk ( k=1, 2, 3, . . .  )
    {
        while (1) {
            for (n=0; n<3; n++) {
                P(full2);
                P(mutex2)
                z[n]=buf2[j];
                j=(j+1) mod M2;
                V(mutex2);
                V(empty2);
            }
            将z[0], z[1], z[2]包装准备出厂;
        }
    }
parend
```

28. 在一个实时系统中，有两个进程P和Q，它们循环工作。P每隔1秒由脉冲寄存器获得输入，并把它累加到整型变量W上，同时清除脉冲寄存器。Q每隔1小时输出整型变量W的值并把它复位。系统提供了标准例程INPUT和OUTPUT供I/O，提供了延时系统调用Delay(seconds)。试采用信号量和P、V操作写出两个并发进程循环工作的算法。

解：semaphore mutex = 1;
int W = 0;
parbegin
process P
{
while (1) {
Delay(1);
x = INPUT( );
P(mutex);
W = W+x;
V(mutex);
清除脉冲寄存器;
}
}
process Q
{
while (1) {
Delay(3600);
P(mutex);
OUTPUT(W);
W = 0;
V(mutex);
}
}
parend

29. 桌上有一只盘子，最多可容纳两个水果，每次仅能放入或取出一个水果。爸爸向盘中放苹果，妈妈向盘中放桔子，两个儿子专门等吃盘子中的桔子，两个女儿专门等吃盘子中的苹果。试用：(1)信号量和P、V操作；(2)管程，来实现爸爸、妈妈、儿子、女儿间的同步与互斥关系。

解：(1) 采用信号量机制

```
semaphore mutex, empty, fulla, fullo;
mutex=1;        //互斥信号量，用于对盘子的互斥使用
empty=2;        //同步信号量，表示盘子中可容纳水果的数量
fulla=0;        //同步信号量，表示盘子中的苹果数
fullo=0;        //同步信号量，表示盘子中的桔子数
process father( )     //父亲进程
{
    while (1) {
        取一个苹果;
        P(empty);        //测试盘子中是否还能往里放水果
        P(mutex);        //互斥使用盘子
        将苹果放入盘子中;
        V(mutex);
        V(fulla);        //通知女儿盘子中的苹果数已增1(可能会唤醒女儿)
    }
```

```
}
process mother( )   //母亲进程
{
    while (1) {
        取一个桔子;
        P(empty);       //测试盘子中是否还能往里放水果
        P(mutex);       //互斥使用盘子
        将桔子放入盘子中;
        V(mutex);
        V(fullo);       //通知儿子盘子中的桔子数已增1(可能会唤醒儿子)
    }
}
process son_i( )        //儿子进程，i=1或2
{
    while (1) {
        P(fullo);       //测试盘子中是否有桔子
        P(mutex);       //互斥使用盘子
        取一个桔子;
        V(mutex);
        V(empty);       //盘子中可放水果的位置数已增1(可能会唤醒父亲或母亲进程)
        吃桔子;
    }
}
process daughter_j( )       //女儿进程，j=1或2
{
    while (1) {
        P(fulla);       //测试盘子中是否有苹果
        P(mutex);       //互斥使用盘子
        取一个苹果;
        V(mutex);
        V(empty);       //盘子中可放水果的位置数已增1(可能会唤醒父亲或母亲进程)
        吃苹果;
    }
}
parbegin
   father( );
   mother( );
   son1( );
   son2( );
   daughter1( );
   daughter2( );
parend
```

(2) 采用管程来解决本问题。管程定义如下：

```
type PC=monitor         //管程名为PC
var empty, apple, orange : integer
```

```
cf, cm, cs, cd : condition;
procedure puta( )          //父亲进程调用此过程向盘子中放苹果
  begin
    if empty=0 then cf.wait;        //若盘子不空，父亲进程等待
    将苹果放入盘子;
    apple := apple+1;               //盘子中苹果数增1
    empty := empty-1;               //盘子中可放水果的位置数减1
    if cd.queue then cd.signal;     //若有女儿在等待苹果，则唤醒她
  end
procedure puto( )          //母亲进程调用此过程向盘子中放桔子
  begin
    if empty=0 then cf.wait;        //若盘子不空，母亲进程等待
    将苹果放入桔子;
    orange := orange+1;                 //盘子中苹果数增1
    empty := empty-1;               //盘子中可放水果的位置数减1
    if cs.queue then cs.signal;     //若有儿子在等待桔子，则唤醒他
  end
procedure geta( )          //女儿进程调用此过程从盘子中取苹果
  begin
    if apple=0 then cd.wait;        //若盘子无苹果，女儿进程等待
    从盘子中取苹果;
    apple := apple-1;               //盘子中苹果数减1
    empty := empty+1;               //盘子中可放水果的位置数增1
    if cf.queue then cf.signal;     //若父亲在等待放苹果，则唤醒他
    else if cm.queue then cm.signal; //否则，若母亲在等待放桔子，则唤醒她
  end
procedure geto( )          //儿子进程调用此过程从盘子中取桔子
  begin
    if orange=0 then cs.wait;       //若盘子无桔子，儿子进程等待
    从盘子中取桔子;
    orange := orange -1;            //盘子中桔子数减1
    empty := empty+1;               //盘子中可放水果的位置数增1
    if cm.queue then cm.signal;    //若母亲在等待放桔子，则唤醒她
    else if cf.queue then cf.signal; //否则，父亲在等待放苹果，则唤醒他
  end
begin
   empty := 2; apple := 0; orange := 0;        //初始化管程的内部数据
end
```

采用上述管程后，本问题可描述如下：

```
parbegin
    process father( )      //父亲进程
    begin
      repeat
         取一个苹果;
         PC.puta( );      //放苹果
```

```
    until false
  end
  process mother( )    //母亲进程
  begin
    repeat
      取一个桔子;
      PC.puto( );    //放桔子
    until false
  end
  process daughter1( )      //女儿1进程
  begin
    repeat
      PC.geta( );    //取一个苹果
      吃苹果;
    until false
  end
  process daughter2( )      //女儿2进程
  begin
    repeat
      PC.geta( );    //取一个苹果
      吃苹果;
    until false
  end
  process son1( )      //儿子1进程
  begin
    repeat
      PC.geto( );    //取一个桔子
      吃桔子;
    until false
  end
  process son2( )      //儿子2进程
  begin
    repeat
      PC.geto( );    //取一个桔子
      吃桔子;
    until false
  end
parend
```

30. 某工厂有两个生产车间和一个装配车间，两个生产车间分别生产A、B两种零件，装配车间的任务是把A、B两种零件装配成产品。两个生产车间每生产一个零件后都要分别把它们送到装配车间的货架F1、F2上，F1存放零件A，F2存放零件B，F1与F2的容量均为可以存放10个零件。装配工人每次从货架上取一个A零件和一个B零件，然后组装成产品。请用：(1) 信号量和P、V操作，(2) 管程，进行正确的管理。

解：(1) 采用信号量和P、V操作

```
semaphore mutex, empty1, empty2, full1, full2;
mutex=1;      //互斥信号量，用于互斥使用货架
empty1=10;   //同步信号量，表示货架F1可容纳的零件数
empty2=10;   //同步信号量，表示货架F2可容纳的零件数
full1=0;  //同步信号量，表示货架F1已放的零件数
full2=0;  //同步信号量，表示货架F1已放的零件数
parbegin
    process workerAi ( )        //第一个生产车间的工人进程，i=1, 2, . . . , n
    {
        while (1) {
            生产一个零件A;
            P(empty1);       //测试货架F1是否可放零件A
            P(mutex);
            将一个零件A放到货架F1上;
            V(mutex);
            V(full1);
        }
    }
    process workerBj ( )        //第二个生产车间的工人进程，j=1, 2, . . . , m
    {
        while (1) {
            生产一个零件B;
            P(empty2);       //测试货架F2是否可放零件B
            P(mutex);
            将一个零件B放到货架F2上
            V(mutex);
            V(full2);
        }
    }
    process workerCi ( )        //装配车间的工人进程，i=1, 2, . . . , k
    {
        while (1) {
            P(full1);          //测试货架F1是否可放零件A
            P(mutex);
            从货架F1上取一个零件A
            V(mutex);
            V(empty);
            P(full2);
            P(mutex);
            从货架F2上取一个零件B
            V(mutex);
            V(empty);
            用零件A、B组装成一个产品;
        }
    }
```

```
    parend
```

(2) 采用管程

```
    type PC=monitor
    var countA, countB : integer;
    CA, CB, CCA, CCB : condition;
    procedure putA( )          //往货架F1上放一个零件A
      begin
        if countA=10 then CA.wait;
        将一个零件A放到货架F1上;
        countA := countA+1;
        if CCA.queue then CCA.signal;
      end
    procedure putB( )          //往货架F2上放一个零件B
      begin
        if countB=10 then CB.wait;
        将一个零件B放到货架F2上;
        countB := countBA+1;
        if CCB.queue then CCB.signal;
      end
    procedure getA( )          //从货架F1上取一个零件A
      begin
        if countA=0 then CCA.wait;
        从货架F1上取一个零件A
        countA := countA-1;
        if CA.queue then CA.signal;
      end
    procedure getB( )          //从货架F2上取一个零件B
      begin
        if countB=0 then CCB.wait;
        从货架F2上取一个零件B
        countB := countB-1;
        if CB.queue then CB.signal;
      end
    begin
       countA := 0; countB := 0;
    end
```

采用上述管程时，本问题的算法可描述如下：

```
    parbegin
      process workerAi ( )          //第一个生产车间的工人进程，i=1, 2, . . . , n
      begin
        repeat
            生产一个零件A;
            PC.putA( );             //将一个零件A放到货架F1上
        until false
      end
```

```
    process workerBj ( )          //第二个生产车间的工人进程，j=1, 2, . . . , m
    begin
      repeat
          生产一个零件B;
          PC.putB( );             //将一个零件B放到货架F2上
      until false
    end
    process workerCi ( )          //装配车间的工人进程，i=1, 2, . . . , k
    begin
      repeat
          PC.getA( );             //从货架F1上取一个零件A
          PC.getB( );             //从货架F2上取一个零件B
          用零件A、B组装成一个产品;
      until false
    end
  parend
```

31. (大连理工2000年试题)用信号量和P、V操作解决进程之间的同步互斥问题。有n(n>1)个进程将字符读入到一个容量为80的缓冲区中，当缓冲区满后，有另一个进程P_b负责一次取走这80个字符。这种过程循环往复，请写出n个读入进程(P_1，P_2，P_3，. . . ，P_n)和P_b的动作序列。（可用文字或伪代码来描述动作序列，设每个读入进程每次读入1个字符到缓冲区）

解：

```
semaphore mutex, empty, full;
mutex=1;      //互斥信号量，用于互斥访问共享变量count
empty=80;     //同步信号量，表示当前缓冲区可容纳的字符个数
full=0;       //同步信号量，1表示缓冲区满，0表示缓冲区未满
int count=0;  //计数变量
parbegin
  process Pi ( )    // i=1, 2, 3, . . . , n
  {
     while (1) {
        读入一个字符;
        P(empty);
        P(mutex);
        将读入的字符存入缓冲区;
        count++;
        if (count==80)  V(full);
        V(mutex);
     }
  }
  process Pb ( )
  {
     while (1) {
        P(full);
        P(mutex);
        从缓冲区取走80个字符;
```

```
            count=0;
            V(mutex);
            for (int i=0; i<80; i++)
                V(empty);
        }
    }
parend
```

32．某银行提供1个服务窗口和10个顾客等待座位。顾客到达银行时，若有空座位，则到取号机领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下：（2011全国试题）

```
cobegin
{
    process 顾客i
    {
        从取号机获得一个号码;
        等待叫号;
        获得服务;
    }
    process 营业员
    {
        while (TRUE)
        {
            叫号;
            为顾客服务;
        }
    }
}
coend
```

请添加必要的信号量和P、V（或wait( )、signal( )）操作实现上述过程的互斥和同步。要求写出完整的过程，说明信号量的含义并赋初值。

解：(说明：此参考答案不是标准答案)

```
begin
semaphore mutex=1;      //用于顾客取号的互斥信号量
semaphore seat=10;      //顾客等待座位的资源信号量，当没有空座位时顾客在其上阻塞
semaphore S1=0;         //营业员与顾客的同步信号量，当没有顾客时营业员在其上阻塞
semaphore S2=0;         //顾客与营业员的同步信号量，等待叫号时顾客在其上阻塞
cobegin
{
    process 顾客i
    {
        P(seat);        //若没有空座位，顾客等待
        P(mutex);       //取号互斥
        从取号机获得一个号码;
        V(mutex);
        V(S1);          //通知营业员，已有顾客
```

```
            P(S2);
            等待叫号;
            获得服务;
        }
        process 营业员
        {
            while (TRUE)
            {
                P(S1);      //若无顾客则等待
                V(S2);      //唤醒等待叫号的顾客
                叫号;
                V(seat);    //空出一个座位(此行放在顾客进程的等待叫号后面也可)
                为顾客服务;
            }
        }
    }
    coend
    end
```

37．有三个并发进程A、B和C，共享一个缓冲器F。F中每次只能存放一个数。进程A每次产生一个随机数R，将其存入F中。若存放到F中的数是5的倍数，则由进程B将其取出并打印，否则由进程C将被5除后的余数打印出来。为防止数的丢失和重复取同一个数，现用PV操作进行管理。请在下面程序的空格中填上合适的操作，以达到上述要求。

```
begin S1,S2,S3:semaphore;
F : integer;
S1:=1; S2:=0; S3:=0;
cobegin
process A
begin
L1:  产生随机数R;
    P(S1) ;
    F:= R;
    if R mod 5=0 then
      V(S2);
    else  V(S3);
    goto L1;
end;
process B
begin
  L2: P(S2);
     x := F;
     V(S1);
     print x;
     goto L2;
end;
process C
```

```
begin
  L3: P(S3);
    y := F;
    V(S1) ;
    y := y mod 5;
    print y;
    goto L3;
end;
coend;
end;
```

B. 读者-写者问题类

1. 修改教科书上的读者-写者同步算法，使得它对写者优先，即一旦有写者到达，后续的读者都必须等待，而无论是否有读者在读文件。用信号量机制解决本问题。

解法1：可以增加一个信号量W，用于在写进程到达时封锁后续的读者进程。

```
BEGIN
    semaphore  mutex1, mutex2, w, rc ;
    mutex1 = 1 ; mutex2 = 1 ; w = 1 ;
    rc = 0 ;
    parbegin
        process Reader_i(i=1,2,3, . . .  )
        {
            P(w) ;
            P(mutex1) ;
            rc = rc + 1 ;
            IF (rc == 1)  P(mutex2) ;
            V(mutex1) ;
            V(w) ;
            Reading the file ;
            P(mutex1) ;
            rc = rc – 1 ;
            IF (rc == 0)  V(mutex2) ;
            V(mutex1) ;
        }
        process Writer_j  (j=1,2,3, . . .  )
        {
            P(w) ;
            P(mutex2) ;
            Writing the file ;
            V(mutex2) ;
            V(w) ;
        }
    parend
END
```

【讨论】上述算法执行时，当一个写者进程写完时，可能唤醒后一个写者，也可能唤醒第一个在W上阻塞是读者进程。要使一个写者写完离开临界区时，若有别的写者，则唤醒一个写者；若无写者等待时，才唤醒一个读者，可以采用如下的算法：

解法2：设置5个互斥信号量和2个共享计数变量：

- 互斥信号量Rsem1：当有写进程在写时，第一个读者在Rsem1上阻塞；第一个写者执行wait(Rsem1)操作，最后一个写进程执行signal(Rsem1)操作。
- 互斥信号量Rsem2：其它读进程在Rsem2上阻塞。
- 互斥信号量Rmutex：用于读进程互斥访问共享变量Rcounter。
- 变量Rcounter：用于读进程计数。
- 互斥信号量Wsem：写进程在Wsem上排队阻塞。
- 互斥信号量Wmutex：用于写进程互斥访问共享变量Wcounter。
- 变量Wcounter：用于写进程计数。

“优先写者”算法描述如下：

```
semaphore Wmutex, Wsem, Rmutex, Rsem1,Rsem2;
int  Rcounter = Wconuter = 0;
Wmutex=Wsem=Rmutex=Rsem1=Rsem2=1;
parbegin
process Reader_i (i = 1，2，…)
{
   wait(Rsem2);
   wait(Rsem1);
   wait(Rmutex);
   Rcounter = Rcounter + 1;
   if(Rcounter==1) wait(Wsem);
   signal(Rmutex);
   signal(Rsem1);
   signal(Rsem2);
   Reading( );
   wait(Rmutex);
   Rcounter = Rcounter - 1;
   if(Rcounter==0)signal(Wsem);
   signal(Rmutex);
}
process Writer_j (j=1，2，…)
{
   wait(Wmutex);
   Wcounter++;
   if (Wcounter==1) wait(Rsem1);
   signal(Wmutex);
   wait(Wsem);
   Writing( );
   signal(Wsem);
   wait(Wmutex);
   Wcounter=Wcounter-1;
```

```
    if (Wcounter==0) signal(Rsem1);
    signal(Wmutex);
  }
  parend
```

2. 修改教科书上的读者-写者同步算法，使得它对写者优先，即一旦有写者到达，后续的读者都必须等待，而无论是否有读者在读文件。用管程机制解决本问题。

解：利用管程解决读者-写者问题（优先写者的算法）如下：

利用管程解决读者-写者问题，首先要为它们建立一个管程，命名为Reader_Writer，简称为R_W。其中包括4个过程：

1) Rin( )过程。读者利用该过程进入读文件。用整型变量RN来表示读者数，WN表示写者数。若WN≥1时，表示已有写者要/在写，读者等待(阻塞)；否则RN增1。读者读完离开时，RN减1，当RN=0时，唤醒1个写者。
2) Rout( )过程。读者利用该过程退出读文件状态。读者读完离开时，RN减1，当RN=0时，若有写者等待，唤醒1个写者。
3) Win( )过程。写者利用该过程进入写文件状态。首先WN增1，当RN≥1或WN>1时，表示有读者在读或别的写者在写，写者等待。
4) Wout( )过程。写者写完离开时WN减1。若WN>0，则唤醒1个写者；否则，若有读者等待，则唤醒所有等待的读者。

R_W管程可描述如下(用伪代码)：

```
type R_W=monitor
  var RN,WN: integer; //读者、写者计数变量
  R,W: condition;
  procedure Rin( ) //读者进入读的过程
   begin
      if WN>0 then R.wait; //读者进程阻塞
      RN := RN+1;//读者数增1
   end
  procedure Rout( )//读者离开
   begin
      RN := RN-1;
      if RN=0 and W.queue then W.signal; //若写者等待，则唤醒队首的一个写者
   end
procedure Win( ) //写者进入写的过程
   begin
     WN := WN+1;
     if RN>0 or WN>1 then W.wait; //写者等待
  end
procedure Wout( ) //写者离开
  begin
      WN := WN-1;
      if WN>0 then W.signal; //若有写者等待，则唤醒队首的一个写者进程
     else while (R.queue) R.signal; //否则若有读者等待，则逐个唤醒所有等待的读者
  end
begin
```

```
    RN :=0; WN :=0;  //初始化内部数据
  end
```

利用管程解决读者-写者问题时，读者和写者进程可描述为：

```
  parbegin
   reader: begin
       R_W.Rin( );
         Read File;
       R_W.Rout( );
     end
   writer: begin
       R_W.Win( );
         Write File;
       R_W.Wout( );
     end
  parend
```

3. 今有一个文件F供进程共享，现把这些进程分成A、B两组，规定同组的进程可以同时读文件F；但当有A组（或B组）的进程在读文件F时就不允许B组（或A组）的进程读文件F。试用P、V操作来进行管理。

解：

```
begin
    S1，S2，SAB：semaphore；
    C1，C2：integer；
    S1 := 1；S2 := 1；SAB := 1；C1 := 0；C2 := 0；
  parbegin
    Process Ai （i = 1，2，3，...）
    begin
        P（S1）；
        C1 := C1 + 1 ；
        if C1 = 1 then P（SAB）；
        V（S1）；
        Read file F ；
        P（S1）；
        C1 := C1 - 1 ；
        if C1 = 0 then V（SAB） ；
        V（S1）；
    end；
    Process Bi （i = 1，2，3，...）
    begin
        P（S2）；
        C2 := C2 + 1 ；
        if C2 = 1 then P（SAB）；
        V（S2）；
        Read file F ；
        P（S2）；
        C2 := C2 - 1 ；
```

```
            if C2 = 0 then V（SAB）；
            V（S2）；
        end；
    parend；
end；
```

【说明】本题与第4题、第5题、第6题实际上是同一题。

4. (武汉理工大学2002年试题)过独木桥问题：一条小河上有一座独木桥，假设河东、河西都有人要过桥，为了保证安全，规定只要桥上无人，则允许一方的人过桥，待一方的人全部过完后，另一方的人才允许过桥。如果把每个过桥者看做一个进程，请用P、V操作实现正确管理。

解：

```
semaphore mutexAB, mutexA, mutexB;
mutexAB=1;          //互斥信号量，用于两个方向过桥者之间的互斥
mutexA=1;           //互斥信号量，用于访问共享变量countA的互斥
mutexB=1;           //互斥信号量，用于访问共享变量countB的互斥
int countA, countB;
countA=0;           //用于自东向西过桥者计数
countB=0;           //用于自西向东过桥者计数
parbegin
  process E_to_Wi ( )   //自东向西过桥者进程，i=1, 2, . . . , n
  {
      P(mutexA);
      if (countA==0) P(mutexAB);
      countA++;
      V(mutexA);
      通过独木桥;
      P(mutexA);
      countA--;
      if (countA==0) V(mutexAB);
      V(mutexA);
  }
  process W_to_Ej ( )   //自西向东过桥者进程，j=1, 2, . . . , m
  {
      P(mutexB);
      if (countB==0) P(mutexAB);
      countB++;
      V(mutexB);
      通过独木桥;
      P(mutexB);
      countB--;
      if (countB==0) V(mutexAB);
      V(mutexB);
  }
parend
```

5. 图2-4中车站A与车站B之间是一段单行铁路，把每列通过AB段的列车看做一个进程，为使列车在AB段安全行驶避免撞车，请用P、V操作实现管理。

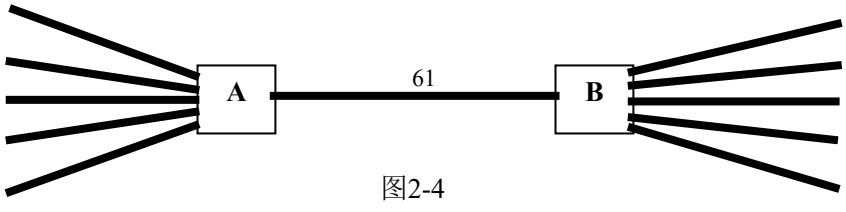


图2-4

6. 图2-5是一条运河和一条公路，运河上有2座间隔80米的吊桥A和B，一条弯曲的公路为绕过一片沼泽地从这两座桥上通过。已知运河上的运输由100米长的驳船担负，公路运输由普通的汽车担负。运河和公路都是单向行驶的（如图2-5中箭头所示）。一条驳船距离吊桥A约20米时先拉响汽笛警告，若桥上无车辆则吊桥被吊起，直到驳船通过A后吊桥才被放下，对吊桥B也按此方法处理。对于汽车，只要前方的吊桥落下就可以通行。试用P、V操作实现车船运输机制且不产生死锁。

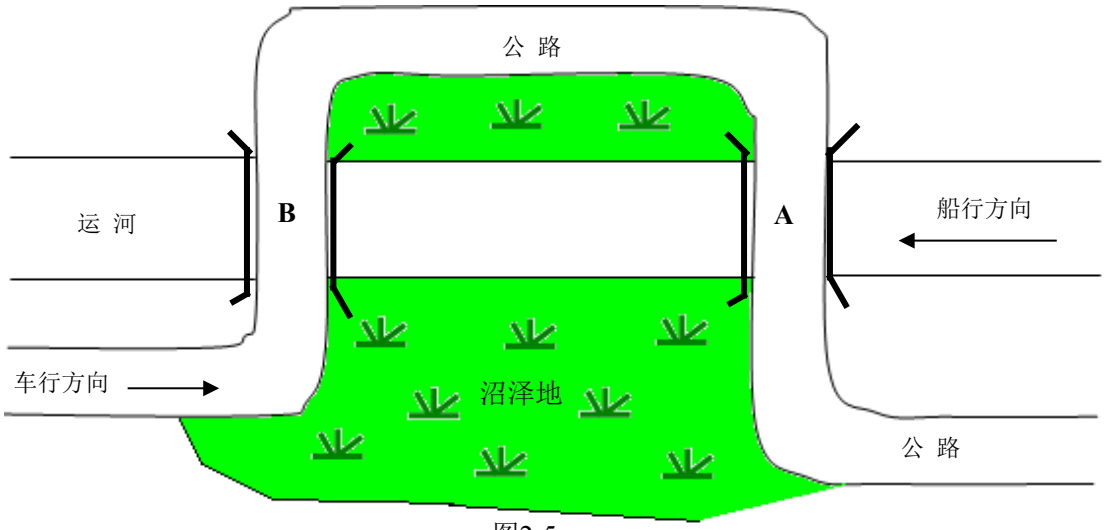


图2-5

【分析】

因为吊桥A和B之间的长度小于驳船的长度，因此当驳船过桥时，应请求两个吊桥抬起来，否则可能产生死锁。

当前面的驳船请求并占有吊桥后，后续的驳船将不必再请求吊桥，鱼贯式地过桥即可。当这批船队的最后一条驳船离开吊桥B后，再释放两个吊桥A和B。汽车过桥的原理与驳船类似。

因此，本题实际上是“单行铁路问题”、“过独木桥问题”、“A、B两组读进程共享一个文件”等问题的变形。

解：semaphore Ship_mutex, Car_mutex, S_C_mutex;

```
Ship_mutex=1;      //互斥信号量，用于Ship互斥访问共享变量Ship_count
Car_mutex=1;       //互斥信号量，用于Car互斥访问共享变量Car_count
S_C_mutex=1;       //互斥信号量，用于Ship与Car互斥过桥A、B
Pargegin
   process ship( )
   {
      P(Ship_mutex);
      if (Ship_count==0) P(S_C_mutex);
      Ship_count++;
```

```
        V(Ship_mutex);
        Ship_go_on( );          //船通过AB段
        P(Ship_mutex);
        Ship_count--;
        if (Ship_count==0) V(S_C+mutex);
        V(Ship_mutex);
    }
    process car( )
    {
        P(Car_mutex);
        if (Car_count==0) P(S_C_mutex);
        Car_count++;
        V(Car_mutex);
        Car_go_on( );           //汽车通过BA段
        P(Car_mutex);
        Car_count--;
        if (Car_count==0) V(S_C+mutex);
        V(Car_mutex);
    }
parend
```

7. 用PV操作解决读者-写者问题的正确程序如下：

```
begin
    S, Sr: Semaphore;
    rc: integer;
    S:=1; Sr:=1; rc:=0;
parbegin
    PROCESS Reader i ( i = 1, 2…)
                begin
        P(Sr) ;
        rc:=rc+1;
        if rc=1 then P(S);
        V(Sr）;
        read file;
        P(Sr）;
        rc:=rc-1
        if rc=0 thenV(S);
        V(Sr);
    end ;
    PROCESS Writer j ( j = 1, 2 … )
        begin
            P(S);
            Write file;
            V(S)
        end;
```

```
    parend ;
end;
```

请回答：（1）信号量S_r的作用；（2）程序中什么语句用于读写互斥，写写互斥；（3）若规定仅允许5个进程同时读怎样修改程序？

解：(1) 信号量S_r的作用的作用是保证读者互斥访问共享变量rc。

(2) 程序中语句if rc=1 then P(S);用于读写互斥；语句P(S); 用于写写互斥。

(3) 若规定仅允许5个进程同时读，修改程序如下(蓝色部分是新增加的内容)：

```
begin
    S, Sr: Semaphore;
    rc: integer;
    S:=1; Sr:=1; rc:=0;
    S1 : semaphore := 5;
parbegin
  PROCESS Reader i ( i = 1, 2…)
                  begin
        P(S1);
        P(Sr) ;
        rc:=rc+1;
        if rc=1 then P(S);
        V(Sr）;
        read file;
        P(Sr）;
        rc:=rc-1
        if rc=0 thenV(S);
        V(Sr);
        V(S1);
     end ;
  PROCESS Writer j ( j = 1, 2 … )
     begin
        P(S);
        Write file;
        V(S)
     end;
  parend ;
end;
```

8. 有P1、P2、P3三个进程共享一个表格F，P1对F只读不写，P2对F只写不读，P3对F先读后写。进程可同时读F，但有进程写时，其他进程不能读和写。请用：(1) 信号量和P、V操作；(2) 管程，编写三个进程能正确工作的算法程序。

解：(1) 采用P、V操作，算法描述如下：

```
semaphore Rmutex, mutex;
Rmutex=mutex=1; //前者用于互斥访问共享变量Rcount，后者用于写互斥
int Rcount=0;       //读者计数变量
parbegin
  process P1
```

```
{
    P(Rmutex);          //互斥访问共享变量Rcount
    Rcount=Rcount+1;    //读者数增1
    if (Rcount==1) P(mutex);   //第一个读者应与写者互斥
    V(Rmutex);
    read F;
    P(Rmutex);
    Rcount=Rcount-1;    //读者离开，读者数减1
    if (Rcount==0) V(mutex);   //最后一个写者离开时应允许写者写
    V(Rmutex);
}
process P2
{
    P(mutex);
    write F;
    V(mutex);
}
process P3
{
    P(Rmutex);          //前半部分与P1相同
    Rcount=Rcount+1;
    if (Rcount==1) P(mutex);
    V(Rmutex);
    read F;
    P(Rmutex);
    Rcount=Rcount-1;
    if (Rcount==0) V(mutex);
    V(Rmutex);
    P(mutex);           //后半部分与P2相同
    write F;
    V(mutex);
}
parend
```

(2) 采用管程，算法描述如下：

建立一个管程，命名为R_W。其中包括4个过程：

(1) Rin( )过程。读者利用该过程进入读文件。用整型变量RN来表示读者数，WN表示写者数。若WN≥1时，表示已有写者要/在写，读者等待(阻塞)；否则RN增1。读者读完离开时，RN减1，当RN=0时，唤醒1个写者。

(2) Rout( )过程。读者利用该过程退出读文件状态。读者读完离开时，RN减1，当RN=0时，若有写者等待，唤醒1个写者。

(3) Win( )过程。写者利用该过程进入写文件状态。首先WN增1，当RN≥1或WN>1时，表示有读者在读或别的写者在写，写者等待。

(4) Wout( )过程。写者写完离开时WN减1。若WN>0，则唤醒1个写者；否则，若有读者等待，则唤醒所有等待的读者。

```
type R_W=monitor
```

```
  var RN,WN: integer; //读者、写者计数变量
  R,W: condition;
  procedure Rin( )  //读者进入读的过程
   begin
       if WN>0 then R.wait; //读者进程阻塞
       RN := RN+1;//读者数增1
   end
  procedure Rout( ) //读者离开
   begin
       RN := RN-1;
       if RN=0 and W.queue then W.signal;  //若写者等待，则唤醒队首的一个写者
   end
    procedure Win( )  //写者进入写的过程
   begin
      WN := WN+1;
      if RN>0 or WN>1 then W.wait; //写者等待
  end
   procedure Wout( )      //写者离开
   begin
       WN := WN-1;
       if WN>0 then W.signal;  //若有写者等待，则唤醒队首的一个写者进程
      else while (R.queue) R.signal; //否则若有读者等待，则逐个唤醒所有等待的读者
  end
begin
  RN :=0; WN :=0;   //初始化内部数据
end
```

利用管程解决本题的读者-写者问题时，进程同步可描述为：

```
parbegin
 process P1:   //只读不写
    begin
      R_W.Rin( );
  Read File;
      R_W.Rout( );
 end
 process P2:   //只写不读
    begin
      R_W.Win( );
  Write File;
      R_W.Wout( );
 end
 process P3:          //先读后写
    begin
      R_W.Rin( );
      Read File;
      R_W.Rout( );
```

```
            R_W.Win( );
            Write File;
            R_W.Wout( );
          end
      parend
```

9．用PV操作解决读者写者问题的程序如下：（此题与第7题重复）

```
begin S, Sr : semaphore; rc : integer;
    S:=1; Sr:=1; rc := 0;
cobegin
  process Reader i  (i=1, 2, …)
  begin
     P(Sr);
     rc:=rc+1;
     if rc=1 then P(S);
     V(Sr);
     read file;
     P(Sr);
          rc:=rc-1;
     if rc=0 then V(S);
     V(Sr);
  end;
  process Writer j   (j=1, 2, …)
  begin
     P(S);
     write file;
     V(S);
  end;
 coend;
end;
```

请回答：(1) 信号量Sr的作用；(2) 程序中什么语句用于读写互斥，写写互斥；(3) 若规定仅允许5个进程同时读，该怎样修改程序？

答：

(1)信号量Sr用于读者互斥访问共享计数变量rc。

(2)if rc=1 then P(S)中的P(S)用于读写互斥，写者进程中的P(S)用于写写互斥、读写互斥。

(3)增加一个信号量S5，初值为5，P(S5)语句加在读进程开头的P(Sr)之前，V(S5)加在读进程程序的最后。

C. 哲学家进餐问题类

1.　利用管程解决教科书上的哲学家进餐问题。

解：利用管程解决哲学家进餐问题，首先要为它们建立一个管程，命名为Philosopher，简称为PH。其中包括2个过程：

(1) get(i)过程。第i号哲学家取筷子的过程。用整型数组chop[0..4]来表示筷子，chop[i]=1表示第i号筷子空闲，chop[i]=0表示第i号筷子已被取走。当chop[i]=1且chop [(i+1) mod 5]=1时，第i号哲学家才可取筷子(chop[i]=0，chop [(i+1) mod 5]=0)；否则阻塞。

(2) put(i)过程。第i号哲学家放下筷子的过程。对左右筷子对应的变量分别置1，并且唤醒等待该筷子的哲学家进程。

PH管程可描述如下(用伪代码)：

```
type PH=monitor
  var chop:array[0..4] of integer;
  cc: array[0..4] of condition;
procedure entry get(integer i)  //第i号哲学家取筷子过程
   begin
      j:=(i+1) mod 5;
    L1: if chop[i]=0 or chop[j]=0 then
            begin
               if chop[i]=0 then cc[i]. wait;
               else cc[j]. wait;
               goto L1;       //被唤醒后必须回到if语句开头（L1处）
            end
            chop[i]:=0;              //拿起左边的筷子
            chop[j]:=0;       //拿起右边的筷子
   end
procedure entry put(integer i) //第i号哲学家放下筷子过程
   begin
        chop[i]:=1; //放下左边筷子
        if cc[i].queue then cc[i].signal;
        chop[j]:=1; //放下右边筷子
        if cc[j].queue then cc[j].signal;
    end
begin
    chop[0]:=:chop[1]:=chop[2]:=1;          //初始化数据
    chop[3]:=chop[4]:=1;
end
```

利用管程解决哲学家进餐问题时，第i个哲学家进程可描述为：

```
Pi ( )  (i=0. . 4)
begin
  repeat
     PH. get(i); //取筷子
           吃面条;
           PH.put(i);    //放下筷子
           思考问题;
  until false
end
 parbegin
     P0( );
     P1( );
     P2( );
     P3( );
     P4( );
```

parend

2. (清华1997年试题)设有5个哲学家，共享一张方有5把椅子的桌子，每人分得一把椅子。但是桌子上总共只有5支筷子，在每个人两边各放一支。哲学家中有饥饿时才试图分两次从两边拾起筷子就餐。就餐的条件是：

(1) 哲学家想吃饭时，先提出吃饭的要求；

(2) 提出吃饭要求，并得到2支筷子后，方可吃饭；

(3) 如果筷子已被他人获得，则必须等待该人吃完饭后才能获得筷子；

(4) 任何哲学家在自己为拿到2支筷子吃饭之前，决不放下手中的筷子；

(5) 刚开始就餐时，只允许2个哲学家请求吃饭。

试问：

(1) 描述一个保证不会出现两个邻座同时要求吃饭的算法；

(2) 描述一个既没有两邻座同时吃饭，又没有人饿死的算法；

(3) 在什么情况下，5个哲学家全部吃不上饭？

解：(1)

```
semaphore S[5]={0,0,0,0,0};              //同步信号量，用于阻塞对应的哲学家
semaphore mutex=1;                       //互斥信号量，用于互斥访问标志变量
int flag[5]={0,0,0,0,0};                 //标志变量
Pi ( )          //第i号哲学家的活动过程，i=0,1,2,3,4
{
      m=(i+1)%5;
      n=(i-1+5)%5;
  L1: P(mutex);
      if (flag[m]==1 || flag[n]==1)
      {
          V(mutex);
          flag[i] = −1;
          P(S[i]);
          goto L1;
      }
      flag[i]=1;                 //flag[i]=1表示第i号哲学家要求吃饭
      V(mutex);
      拿起左边筷子;
      拿起右边筷子;
      吃饭；
      放下左边筷子;
      放下右边筷子;
      P(mutex);
      flag[i]=0;
      if (flag[m]== -1)          //若右边哲学家正在等待，则唤醒之
      { flag[m]=0; V(S[m]); }
      if (flag[n]== -1)          //若左边哲学家正在等待，则唤醒之
      { flag[m]=0; V(S[m]); }
      V(mutex);
}
```

```
parbegin
    P0( );
    P1( );
    P2( );
    P3( );
    P4( );
parend
```

【说明】上述算法由于能保证不会出现两个邻座同时要求吃饭的情况，因此也就能保证只有2个哲学家请求吃饭。还能保证不可能有两人取同一根筷子(从而取筷子也不需要考虑互斥问题)。但是上述算法可能出现饿死现象。例如，1、3号哲学家吃饭时，2号哲学家不可能吃饭。假设1号哲学家先吃完，但在3号哲学家吃完之前，1号哲学家又吃饭；3号哲学家吃完后，在1号哲学家吃完之前又要求吃饭，如此1、3号哲学家轮流交替吃饭，2号哲学家将饿死。

(2)

```
semaphore S[5]={0,0,0,0,0};          //同步信号量，用于阻塞对应的哲学家
semaphore mutex=1;                   //互斥信号量，用于互斥访问标志变量
int flag[5]={0,0,0,0,0};             //标志变量
Pi ( )          //第i号哲学家的活动过程，i=0,1,2,3,4
{
    m=(i+1)%5;
    n=(i-1+5)%5;
  L1: P(mutex);
    if (flag[m]==1 || flag[n]==1 || flag[m]==-2 || flag[n]==-2)
    {
        V(mutex);
        flag[i] = -1;
        P(S[i]);
        goto L1;
    }
    flag[i]=1;                  //flag[i]=1表示第i号哲学家要求吃饭
    V(mutex);
    拿起左边筷子;
    拿起右边筷子;
    吃饭；
    放下左边筷子;
    放下右边筷子;
    P(mutex);
    flag[i]=0;
    if (flag[m]== -1)           //若右边哲学家正在等待，则唤醒之
    { flag[m]= -2; V(S[m]); }
    if (flag[n]== -1)           //若左边哲学家正在等待，则唤醒之
    { flag[m]= -2 ; V(S[m]); }
    V(mutex);
}
parbegin
    P0( );
```

```
    P1( );
    P2( );
    P3( );
    P4( );
parend
```

(3) 去掉“只允许2个哲学家请求吃饭”和“不允许两个邻座同时要求吃饭”的条件，按下面给出的算法：

```
semaphore chopstick[5]={1,1,1,1,1};   //互斥信号量，用于互斥拿起5根筷子
Pi ( )              //第i号哲学家的活动过程，i=0,1,2,3,4
{
    j=(i+1)%5;
    while (1) {
        P(chopstick[i]);      //拿起左边的筷子
        P(chopstick[j]);      //拿起右边的筷子
        吃饭;
        V(chopstick[i]);      //放下左边的筷子
        V(chopstick[j]);      //放下右边的筷子
    }
}
parbegin
    P0( );
    P1( );
    P2( );
    P3( );
    P4( );
parend
```

上述算法中，当5个哲学家同时拿起左边的筷子，再拿右边筷子时都将阻塞，此时系统将出现死锁，这种情况下，5个哲学家全部吃不上饭。

3. (上海交通大学1996年试题)哲学家甲请哲学家乙、丙、丁到某处讨论问题，约定全体到齐后开始讨论；在讨论的间隙四位哲学家进餐，每人进餐都需使用刀、叉各一把(假设每个哲学家进餐时都是先拿刀，再拿叉)，餐桌上的布置如图2-6所示。请用信号量及P、V操作说明这四位哲学家的同步、互斥过程。

解：设置四个互斥信号量fork1、fork2、knife1、knife2，其初值均为1，分别表示资源叉1、叉2、刀1、刀2是

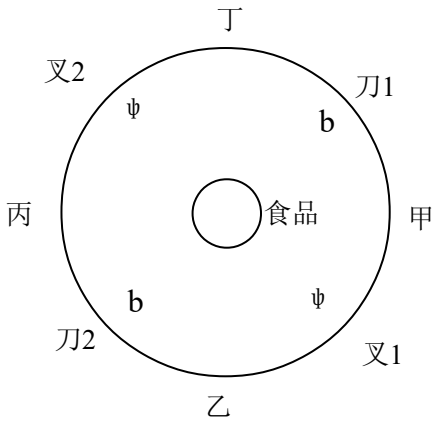


图2-6 哲学家进餐问题(b表示刀，ψ表示叉)

否可用。同步算法描述如下：

```
semaphore fork1, fork2, knife1, knife2;
fork1=fork2=knife1=knife2=1;
  process Pa( )      //哲学家甲
  {
    while (1)
    {
        讨论问题;
        P(knife1);      //拿起右边的刀子
        P(fork1);       //拿起左边的叉子
        进餐;
        V(knife1);      //放下右边的刀子
        V(firk1);       //放下左边的叉子
    }
  }
  process Pb( )      //哲学家乙
  {
    while (1)
    {
        讨论问题;
        P(knife2);      //拿起左边的刀子
        P(fork1);       //拿起右边的叉子
        进餐;
        V(knife2);      //放下左边的刀子
        V(firk1);       //放下右边的叉子
    }
  }
  process Pc( )      //哲学家丙
  {
    while (1)
    {
        讨论问题;
        P(knife2);      //拿起右边的刀子
        P(fork2);       //拿起左边的叉子
        进餐;
        V(knife2);      //放下右边的刀子
        V(firk2);       //放下左边的叉子
    }
  }
  process Pd( )      //哲学家丁
  {
    while (1)
    {
        讨论问题;
        P(knife1);      //拿起左边的刀子
```

```
            P(fork2);       //拿起右边的叉子
            进餐;
            V(knife1);      //放下左边的刀子
            V(firk2);       //放下右边的叉子
        }
    }
parbegin
    Pa( );
    Pb( );
    Pc();
    Pd( );
parend
```

【注意】此问题与教科书上的五个哲学家进餐问题不同。本算法中，由于甲、丙哲学家先拿右边的刀子后拿左边的叉子，而乙、丁哲学家先拿左边的刀子，后拿右边的叉子，故不会发生死锁问题。

D. 其它互斥同步问题

1. 某由西向东的单行车道有一卡脖子的路段AB(如图2-7所示)，为保证行车的安全需设计一个自动管理系统，管理原则如下：
   当AB段之间无车行驶时，可让到达A点的一辆车进入AB段行驶；
   当AB段有车行驶时，让到达A点的车等待；
   当在AB段行驶的车驶出B点后，可让等待在A点的一辆车进入AB段。

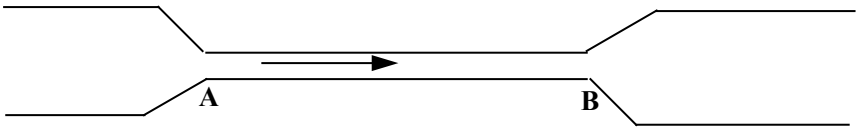


图2-7

   请回答下列问题：
   (1) 把每一辆需经过AB段的车辆看作是一个进程，则这些进程在AB段执行时，它们之间的关系应是同步还是互斥？
   (2) 用PV操作管理AB段时，应怎样定义信号量，给出信号量的初值以及信号量可能取值的含义。
   (3) 若每个进程的程序如下，请在方框中填上适当的PV操作，以保证行车的安全。

```
Parbegin
Process (A→B)i (i=1, 2, . . . )
  begin
      到达A点；
      [          ]
      在AB段行驶；
      到达B点；
      [          ]
      驶出B点；
  end;
Parend
```

解：(1) 是互斥问题。

(2) 定义一个互斥信号量mutex，其初值为1。信号量mutex的可能取值为1、0、-1、-2 . . . 。mutex的值为1，表示AB段没有车行驶，也没有车在A点等待；mutex=0，表示有一辆车载AB段行驶，但没有在A点等待；mutex=-1，表示有一辆车载AB段行驶，同时有一辆车在A点等待；mutex=-2，表示有一辆车载AB段行驶，同时有两辆车在A点等待；mutex=-3，表示有一辆车载AB段行驶，同时有三辆车在A点等待。以此类推。

(3) 第一个方框内填 P(mutex);，第二个方框内填V(mutex);

2. 假定一个阅览室可供50个人同时阅读。读者进入和离开阅览室时都必须在阅览室入口处的一个登记表上登记，阅览室有50个座位，规定每次只允许一个人登记或注销登记。

要求：（1）用PV操作描述读者进程的同步算法（可用流程图表示，登记、注销可用自然语言描述）；

（2）指出算法(流程图)中所用信号量的名称、作用及初值。

解：（1）用 PV 操作描述读者进程的同步算法如下：

```
semaphore mutex=1, seat=50;
parbigin
process reader_i  (i=1, 2, 3, …)
  begin
    读者来到阅览室;
    P(seat);              //看看阅览室是否有空位置
    P(mutex);                 //等级需互斥
    在登记表上登记;
    V(mutex);
    在阅览室阅览资料;
    P(mutex);
    在登记表上注销;
    V(mutex);
    V(seat);              //阅览室空座位数增 1，必要时唤醒等待进入阅览室的读者
    该读者离开;
  end
parend
```

（2）互斥信号量 mutex，其初值为 1，用于互斥使用登记表；资源信号量 seat，其初值为 50，用于表示阅览室空座位的数量。

3. 在一个盒子里，混装了数量相等的黑白围棋子。现在用自动分拣系统把黑子、白子分开，设分拣系统有两个进程P1和P2，其中P1拣白子，P2拣黑子。规定每个进程每次拣一子；当进程在拣时，不允许另一个进程取拣；当一个进程拣了一子时，必须让另一进程去捡。假设从拣白子开始。试写出量进程能正确并发执行的算法程序。

解：

```
semaphore mutex, S1, S2;
mutex=1; //互斥信号量
S1=1;    //P1的同步信号量，用于与P2同步，从P1拣第一个白子开始
S2=0;    // P2的同步信号量，用于与P1同步
parbegin
  Process P1
  {
     while (1) {
```

```
        P(S1);
        P(mutex);    //拣子必须互斥
        拣一白子;
        V(mutex);
        V(S2);       //通知P2可以拣一黑子了
        }
    }
    Process P2
    {
        while (1) {
        P(S2);       //看看P1是否已捡完一个白子
        P(mutex);    //拣子必须互斥
        拣一黑子;
        V(mutex);
        V(S1);       //通知P1可以拣一白子了
        }
    }
parend
```

4. 某小型超级市场，可容纳 50 个人同时购物。入口处备有篮子，每个购物者可持一只篮子入内购物。出口处结账，并归还篮子(出入口仅允许一个人通过)。（此题与第 2 题相同）

要求：(1)试用 PV 操作描述购物者进程的同步算法(可用流程图描述)

(2)说明算法(流程图)中使用的信号量的名称，作用和初值。

解：(1) 算法描述如下：

```
semaphore mutex, S;
mutex=1; S=50;
parbegin
  Process Pi ( i=1,2,3, . . . )        //顾客进程
  {
      P(S);          //超市可容纳人数减 1，若超市已满员则阻塞
      P(mutex);      //入口互斥
      拿一个篮子并通过入口;
      V(mutex);      //允许后边顾客进入超市
      在超市挑选商品;
      P(metex);      //出口互斥
      在出口处结账;
      归还篮子;
      V(mutex);
      V(S);          //超市可容纳人数增 1，必要时唤醒一个等待进入的顾客进程
      离开超市;
  }
parend
```

(2) 设置资源信号量 S，用于表示超市可容纳顾客的人数，其初值为 50；互斥信号量 metex，用于出入口处的互斥，其初值为 1。

5. 两个并发进程的程序如下：

```
begin
    N:integer;
    N:=1;
parbegin
    process A
        begin
        L1:  N:=N+1;
             go to L1;
        end;
    process B
        begin
        L2:  print(N);
             N:=0;
             go to L2;
        end;
parend;
end;
```

请回答：

(1)指出这两个并发进程的临界区。

(2)指出它们并发执行时可能出现的“与时间有关的错误”。

(3)用PV操作进行管理，写出使它们能正确并发执行的程序。

解：(1) 进程A的临界区是“N:=N+1;”，进程B的临界区是“print(N); N:=0;”

(2) 为叙述方便，不妨假设某时刻共享变量N=10。可能出现的“与时间有关的错误”如下：

① 进程B执行了print（N）打印10后被中断；系统切换到进程A，进程A执行N:=N+1，将N的值增至11后，又切换到进程B，进程B执行语句N:=0; 将N清零，最终N的值为0，而此时N正确的值应该是1，即出现“与时间有关的错误”。

② 进程A执行N := N+1时，求出N+1的值为11，刚想赋值给N时，正好发生进程切换到B，B执行print(N);语句，打印的值是10，再执行N:=0;，将N清零。此后进程又切换到A，A完成被打断的赋值操作，将N的值赋值为11，而此时N正确的值应该是1。

(3) 定义一个互斥信号量mutex，其初值为1，采用PV操作使它们能正确并发执行的伪代码程序如下：

```
begin
    mutex : semaphore;
    N:integer;
    N:=1;
  mutex := 1;
   parbegin
     process A
        begin
        L1:
             P(mutex);
             N:=N+1;
             V(mutex);
             go to L1;
        end;
```

```
process B
    begin
    L2:
        P(mutex);
        print(N);
        N:=0;
        V(mutex);
        go to L2;
    end;
parend;
end;
```

6. 四个进程A、B、C、D都要读一个共享文件F，系统允许多个进程同时读文件F，但限制是进程A和进程C不能同时读文件F，进程B和进程D也不能同时读文件F，为了使这四个进程并发执行时能按系统要求使用文件，现用PV操作进行管理，请回答下面的问题：

（1）应定义的信号量及初值：_____。

（2）在下列的程序中填上适当的P、V操作，以保证它们能正确并发工作。

process A	process B	process C	process D
begin	begin	begin	begin
①；	③；	⑤；	⑦；
read F；	read F；	read F；	read F；
②；	④；	⑥；	⑧；
end；	end；	end；	end；

解：(1) 应定义的信号量是S1，S2，初值为1，1。

(2) ① P(S1);　② V(S1);

③ P(S2);　④ V(S2);

⑤ P(S1);　⑥ V(S1);

⑦ P(S2);　⑧ V(S2);

7. A，B 两组学生进行投球比赛，规定 A 组(或 B 组)的一个学生投了一个球后应让 B 组(或 A 组)的一个学生投一个球。假定让 A 组的学生先开始投球，用 PV 操作控制时，回答如下问题：

(1)应定义的信号量的个数和初值：_____

(2)在两组工作流程的方框位置填上适当的 P、V 操作，使其能按规定进行。

A 组：

①

投一个球

②

B 组：

③

投一个球

④

解：(1) 应定义2个信号量SA和SB，初值分别为1和0。

(2) ① P(SA);　　② V(SB);　　③ P(SB);④ V(SA);

8. (华中理工大学1999年试题)设公共汽车上，司机和售票员的活动分别是：

司机的活动：　启动车辆；
　　正常行驶；
　　到达停车；
售票员的活动：关车门；
　　售票；
　　开车门；

在汽车不断地到站、停车、行驶过程中，这两个活动有什么同步关系？用信号量和P、V操作实现它们的同步。

解：司机和售票员活动的同步关系为：售票员关门后，向司机发开车信号，司机接到开车信号后启动车辆，在汽车行驶过程中售票员售票，到站时司机停车，售票员在停车后开车门让乘客上下车。因此，司机启动车辆的动作必须与售票员关门的动作取得同步；售票员开门的动作也必须与司机停车的动作同步。

设置2个信号量S1、S2，信号量S1表示是否允许司机启动汽车，其初值为0；信号量S2表示是否允许售票员开门，其初值为0。用P、V操作描述的同步算法如下：

```
BEGIN
  semaphore S1, S2;
  S1=S2=0;
  parbegin
    process driver ( )              //司机进程
      begin
        repeat
          P(S1);                    //司机判断是否可以启动汽车
          启动车辆;
          正常行驶;
          到站停车;
          V(S2);                    //告诉售票员可以开门了
        until false
      end
    begin
      repeat
        关车门;
        V(S1);                      //告诉司机可以启动汽车了
        售票;
        P(S2);                      //看看是否可以开车门
        开车门;
        上下乘客;
      until false
    end
  parend
END
```

9. (北大1991年试题)有一个仓库，可以存放A和B两种产品，仓库的存储空间足够大，但要求：

（1）每次只能存入一种产品（A或B）；

（2）-N < A产品数量 - B产品数量 < M 。

其中，N，M是正整数。试用“存放A”和“存放B”和P、V操作描述产品A与产品B的入库过程。

【分析】若不放A产品，B产品放最多能放N-1个，此后“存放B”应该阻塞；若不放B产品，A产品放最多能放M-1个，此后“存放A”应该阻塞。故“存放A”和“存放B”的资源信号量SA与SB的初始值分别为M-1和N-1。mutex为互斥信号量。

解：引入3个信号量：互斥信号量mutex；存放A的资源信号量sa，初值为M-1；存放B的资源信号量sb，其初值为N-1。

```
semaphore mutex,sa,sb;
mutex.value=1;sa.value=M-1;sb.value=N-1;
parbegin
存放A：
{
  while (true)
  {
     get product A；
    P(sa);
    P(mutex);
    put product A;
    V(mutex);
    V(sb);
  }
}
存放B：
 {
   while (true) {
    get product B;
    P(sb);
    P(mutex);
    put product B;
    V(sa);
    V(mutex);
   }
 }
parend
```

10. 假设缓冲区buf1和buf2都无限大，进程P1向buf1写数据，进程P2向buf2写数据。现要求buf1数据个数与buf2数据个数的差保持在[m,n]之间(m,n皆为整数，m小于n)，请用信号量描述此同步关系。(浙江大学2001年试题)——本题与上一题类似但又有区别

【分析】本题没有限制m≤0和n＞0，故不能像上题那样用两个资源信号量来处理，而采用计数变量count1和count2来处理(学生应知道为什么)。count1和count2应满足m<=count1-count2<=n。同步算法描述如下：

```
begin
   var mutex, S1,S2 : semaphore := 1,0,0 ;
   count1, count2 : integer := 0, 0 ;
 parbegin
```

```
process P1( )
 begin
  repeat
   computing_a_data( );
   P(mutex) ;
   if (count1-count2 > n) then begin
      V(mutex);
      P(S1);          //P1阻塞
   else
      V(mutex) ;
   end
   write_a_data_to_buf1( ) ;
   P(mutex);
   count1 := count1+1 ;
   V(S2);
   V(mutex);
  until false;
 end
process P2( )
 begin
  repeat
   get_a_data( );
   P(mutex);
   if (count1-count2< m) then begin
       V(mutex);
       P(S2);  //P2阻塞
   else
       V(mutex);
   end
   write_a_data_to_buf2( );
   P(mutex);
   count2 := count2+1;
   V(S1);
   V(mutex);
  until false;
 end
parend
end
```

11. 有一个仓库存放两种零件A和B，最大库容量为可存放1000个零件A或B。有一车间不断地取A和B进行装配，每次各取一个。有两组供应商分别不断地供应A和B（每次一个）。为保证齐套和合理库存，当某种零件的数量比另一种数量超过100个时，暂停对数量大的零件的进货，集中补充数量少的零件。试用P、V操作正确地实现之。

解：采用P、V操作的同步算法如下：

```
 BEGIN
```

```
semaphore mutex，availab，full1，full2，sa，sb；
mutex.vale = 1；
availab.value = 1000；fulla.value = fullb.value = 0；
sa.value = 100；sb.value = 100；
PARBEGIN
store_A：BEGIN
          L1：P（availab）；
              P（sa）；
              P（mutex）：
              store A；
              V（fulla）；
              V（sb）；
              V（mutex）；
              goto L1；
          END
store_B：BEGIN（2分）
          L2：P（availab）；
              P（sb）；
              P（mutex）；
              store B；
              V（fullb）；
              V（sa）；
              V（mutex）；
              goto L2；
            END
      take：BEGIN
             L3：P（fulla）；
                 P（fullb）；
                 P（mutex）；
                 take  A and B；
                 V（availab）；
                 V（availab）；
                 V（mutex）；
                 goto L3；
            END
   PAREND
END
```

12. (北大 1995 年试题)有一个仓库存放两种零件 A 和 B，最大库容量为 m 个。有一车间不断地取 A 和 B 进行装配，每次各取一个。为避免零件锈蚀，遵循先入库者先出库的原则。有两组供应商分别不断地供应 A 和 B（每次一个）。为保证齐套和合理库存，当某种零件的数量比另一种数量超过 n（n<m）个时，暂停对数量大的零件的进货，集中补充数量少的零件。试用 P、V 操作正确地实现之。

【注】此题与上题类同

解：BEGIN

```
semaphore mutex，availa，full1，availb，full2，sa，sb；
mutex ： = 1；
availa ： = m；fulla ： = 0；
availb ： = m；fullb ： = 0
sa ： = n；sb ： = n；
PARBEGIN
store_A：BEGIN
          L1：P（availa）；
              P（sa）；
              P（mutex）：
              store A；
              V（fulla）；
              V（sb）；
              V（mutex）；
              goto L1；
          END
store_B：BEGIN
          L2：P（availb）；
              P（sb）；
              P（mutex）；
              store B；
              V（fullb）；
              V（sa）；
              V（mutex）；
              goto L2；
            END
     take：BEGIN
             L3：P（fulla）；
                 P（fullb）；
                 P（mutex）；
                 take  A and B；
                 V（availa）；
                 V（availb）；
                 V（mutex）；
                 goto L3；
            END
  PAREND
END
```

13. 某高校计算机系开设网络课并安排上机实习，假设机房共有 2m 台机器，有 2n 名学生选该课，规定：
（1）每两个学生组成一组，各占一台机器，协同完成上机实习；
（2）只有一组学生到齐，并且此时机房有空闲机器时，该组学生才能进入机房；
（3）上机实习由一名教师检查。检查完毕，一组学生同时离开机房。
试用P、V操作模拟上机实习过程。（北京大学，1997年）

【分析】在本题中，为了保证系统的控制流程，增加了Monitor进程，用于控制学生的进入和计算机分配。

从题目本身来看，虽然没有明确写出这一进程，但实际上这一进程是存在的。因此，在解决这类问题时，需要对题目加以认真分析，找出其隐蔽的控制机制。学生、教师、Monitor的活动过程可以描述如下：

学生的活动过程：

学生到达——V(student);
等待允许进入——P(enten);
用Monitor指定的计算机上机实习；
实习完成——V(finish);
等待老师检查——P(check);
释放计算机——V(computer);

老师的活动过程：

Repeat
等待学生完成实习——P(finish);
等待另一学生完成——P(finish);
检查学生实习；
检查完成一学生的实习——V(check);
检查完另一学生的实习——V(check);
Until false

Monitor的活动过程：

Repeat
等待学生到达——P(student);
等待另一学生到达——P(student);
获取一台计算机——P(computer);
获取一台计算机——P(computer);
允许学生进入并给其指定一台计算机——V(enter);
允许学生进入并给其指定一台计算机——V(enter);
Until false

解：根据上面的分析，同步算法表述如下：

```
BEGIN
    semaphore  computer, student, enter, finish, check;
    student := 0；enter := 0；
    computer :=  2m；finish := 0；check := 0；
PARBEGIN
process Student( )
    BEGIN
        V(student); //表示有学生到达
        P(enter);  //等待允许进入
        用Monitor指定的计算机上机实习；
        V(finish);  //表示实习完成
        P(check);  //等待教师检查
        V(computer); //释放计算机资源
    END
 process Teacher( )
  BEGIN
   L1：P(finish);  //等待学生实习完成
```

```
        P(finish); //等待另一个学生实习完成
        检查2个学生的实习;
        V(check); //表示检查完成
        V(check); //每组有两个学生
        goto L1;
      END
    process Monitor( )
     BEGIN
      L2：P(student); //等待学生到达
        P(student); //等待另一个学生到达
          P(computer); //获取一台计算机
              P(computer); //获取另一台计算机
        V(enter); //允许学生进入并给其指定一台计算机
        V(enter); //允许学生进入并给其指定一台计算机
        goto L2;
       END
    PAREND
    END
```

以下算法稍有不同(摘自某参考书)，但也是可以的：

```
     BEGIN
        integer student，computer，enter，finish，check;
        student ： = 0；enter ： = 0;
        computer ： = 2m；finish ： = 0；check ： = 0;
        PARBEGIN
            Student：BEGIN
                        V（student）;          {表示有学生到达}
                        P（computer）;         {获取一台计算机}
                        P（enter）;            {等待允许进入}
                        Do it with partner;
                        V（finish）;           {表示实习完成}
                        P（check）;            { 等待教师检查}
                        V（computer）;         {释放计算机资源}
                    END
            Teacher：BEGIN
                    L1：P（finish）;           {等待学生实习完成}
                         P（finish）;          {等待另一个学生实习完成}
                         Check the work;
                         V（check）;           {表示检查完成}
                         V（check）;
                         goto L1;
                     END
            Monitor：BEGIN
                     L2：P（student）;          {等待学生到达}
                          P（student）;         {等待另一个学生到达}
                          V（enter）;           {允许学生进入}
```

```
                V（enter）；          {允许学生进入}
                goto L2；
            END
        PAREND
    END
```

14. 理发师问题描述如下：理发店包含一间接待室和一间工作室，接待室内有n(n≥1)把椅子，而工作室只有1把椅子。如果没有顾客，理发师就去睡觉；如果来时所有椅子都有人，那么顾客离去；如果理发师在忙且接待室有空闲椅子，那么此顾客会坐在其中1把空闲的椅子上等待；如果理发师在睡觉；则顾客会唤醒他。请采用信号量机制解决该理发师问题(可用伪代码描述)。

解：引入5个信号量和1个控制变量：

- 控制变量count用来记录等待理发的顾客数，初值为0；当count=n时，新来的顾客离去。
- 信号量customers用来记录等候理发的顾客数，并用作阻塞理发师进程，初值为0；
- 信号量barbers用来记录等候顾客的理发师数，并用作阻塞顾客进程，初值为0；
- 信号量mutex用于对共享变量count的互斥访问，初值为1；
- 信号量cut用来告知理发师可以开始理发，并用作阻塞理发师进程，初值为0；
- 信号量finish用来告知顾客理发已完成，并用作阻塞顾客进程，初值为0；

信号量机制解决理发师问题算法描述如下：

```
semaphore customers,barbers,mutex:=0,0,1;       //定义信号量并初始化
semaphore cut,finish:=0,0;
int count=0;
Parbegin
Process Customers( )            //定义并发进程
{
   P(mutex);
   if (count>=n)
   {
      V(mutex);
      exit();  //没有空椅子，离开
   }
   else
   {
       count=count+1;
       V(mutex);
       V(customers);  //唤醒理发师
       SIT_ON_chair();  //坐在椅子上等候
       P(barbers);  //等待理发师召唤
       Stand_up(); //从椅子上起身
       P(mutex);
       count=count-1;  //坐椅子等待的顾客数减1
       V(mutex);
       SIT_ON_cut_chair();//坐在理发椅子上
       V(cut);  //告诉理发师可以开始理发
       P(finish);  //等待理发完成
   }
```

```
} //顾客进程结束
Process barber( )          //理发师进程
{
   while (true)
   {
        P(customers); //等待顾客到来
        Clear_cut_chair(); //整理一下理发椅子
        V(barbers); //召唤一个顾客
        P(cut); //等待顾客就座
        CUT_hair(); //理发
        V(finish); //告诉顾客已理完
   }
}
Parend //并发进程的定义结束
```

说明：1．坐椅子等待的顾客数减1的语句也可以放到理发师进程中。

2．本程序也没有考虑顾客理完发后的付款问题。

15. (青岛大学2002年试题)青岛崂山有一处景点称作上清宫，游客在宫内游玩后可以在宫门口免费搭乘轿车游览崂山的风景区，游览完毕再返回宫门口（如下图2-8所示）。

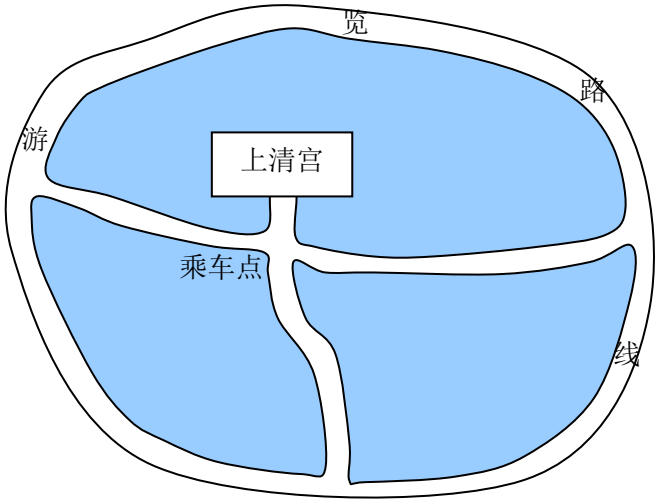


图2-8 崂山风景区

已知风景区内游览轿车总量有M辆，游客总数为N，约定：

a) 每辆轿车限乘一位游客；
b) 如果有空闲的轿车，应当允许想游览的游客乘坐；
c) 无空闲的轿车时，游客应该等待；
d) 若没有想游览的乘客，轿车也应等待。

试利用P、V操作实现N个游客进程和M辆轿车进程的同步操作过程。

解：定义4个信号量car_avail、car_taken、finished和take_off。car_avail初值为M，它表示空闲轿车数量；car_taken表示是否有游客乘车，用于阻塞轿车进程；finished用于游览是否结束；take_off用于同步游客是否已下车，后3个信号量初值都为0。

```
semaphore car_avail=M,car_taken=0,finished=0,take_off=0;
parbegin
```

```
process passenger ( )                    //游客进程
begin
    游上清宫;
    P(car_avail);              //看看是否有轿车可乘，若无则等待
    V(car_taken);              //唤醒一等待的轿车进程
    游客上车;
    P(finished);               //游客等待游览结束
    游客下车;
    V(take_off);               //通知轿车进程，游客已下车
end
process car ( )
begin
  repeat
    P(car_taken);              //若无游客，轿车进程等待
    带一游客游览风景区;
    V(finished);               //通知游客游览已结束
    P(take_off);               //等待游客下车
    V(car_avail);              //空闲轿车数增1：若有等待的游客则唤醒之
  until false
end
parend
```

16. (南开1997年试题)在南开大学和天津大学之间有一条弯曲的小路，其中从S到T一段路每次只允许一辆自行车通过，但其中间有一个小的安全岛M（同时允许两辆自行车停留），可供两辆自行车已从两端进入小路的情况下错车使用，如图2-9所示。试设计一个算法使来往的自行车均可顺利通过。

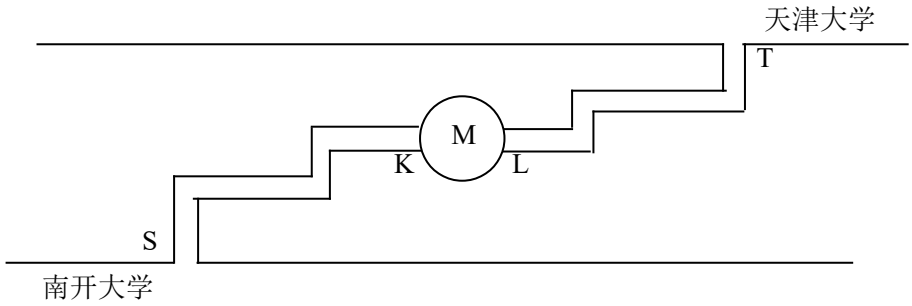


图2-9　小路示意图

【分析】路段T到L及路段S到K同时各只允许一辆自行车通过，而安全岛M允许两辆自行车使用，因此可以用3个信号量来管理它们：信号量SK表示是否允许自行车通过路段S到K，其初值为1；信号量TL表示是否允许自行车通过路段T到L，其初值为1；信号量M表示安全岛上还可以停放自行车的数目，其初值为2。另外，同一方向上的自行车最多只能有一辆通过S与T之间的路段(两个方向上有两辆)，因此还应使用2个信号量来控制：信号量ST表示是否允许自行车从南开大学去天津大学，其初值为1；信号量TS表示是否允许自行车从天津大学去南开大学，其初值为1；

解：用P、V操作的算法描述如下：

```
BEGIN
semaphore SK, TL, M, ST, TS;
SK=TL=ST=TS=1; M=2;
parbegin
```

```
process Nan_to_Tian ( )          //从南开大学去天津大学
  begin
    P(ST);          //只允许一辆自行从南开大学去天津大学（互斥）
    P(SK);          //路段S到K只允许一辆自行车通过（互斥）
    从S走到K;
    P(M);
    进入安全岛M;
    V(SK);          //允许其它自行车通过S与K之间的路段
    P(TL);          //T与L之间路段只允许一辆自行车通过（互斥）
    V(M);           //离开安全岛
    从L走到T;
    V(TL);          //允许其它自行车通过T与L之间的路段
    V(ST);          //允许其它自行车从南开大学去天津大学
  end
process Tian_to_Nan ( )          //从天津大学去南开大学
  begin
    P(TS);          //只允许一辆自行从天津大学去南开大学
    P(TL);          //路段T到L只允许一辆自行车通过
    从T走到L;
    P(M);
    进入安全岛M;
    V(TL);          //允许其它自行车通过S与K之间的路段
    P(SK);          //S与K之间路段只允许一辆自行车通过
    V(M);           //离开安全岛
    从K走到S;
    V(SK);          //允许其它自行车通过S与K之间的路段
    V(TS);          //允许其它自行车从天津大学去南开大学
  end
parend
END
```

【讨论】由于控制了同一方向上的自行车最多只能有一辆通过S与T之间的路段，因此安全岛M上的自行车数目永远不会超过2辆，所以信号量M其实是可以省掉的。

17. (北交大1997年试题)讨论图2-10所示的交通管理例子（各方向上的汽车是单行、直线行驶），试用P、V操作实现各方向上汽车行驶的同步。假设开始时路口A、B之间的马路上无车辆。

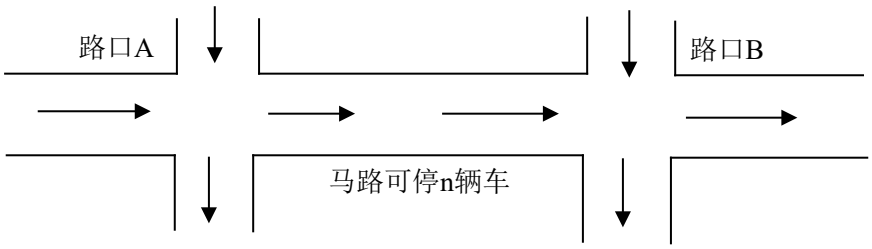


图2-10

解：假设自西向东进入路口A的汽车为Xi，自北向南进入路口A的汽车为Yj，自北向南进入路口B的汽车为Z_k。显然，同方向的汽车可共享路口A(或路口B)，非同方向的汽车应互斥经过路口。对于自西

向东的汽车，除了考虑2个路口与另一方向汽车的互斥问题外，还应考虑路口A、B之间路面可停n辆车的问题，当该段路面满时，自西向东的汽车不再允许进入路口A。

```
semaphore mutex1,mutex2,S1,S2,S3,S4,count;
int Cx1, Cy, Cz, Cx2;
mutex1=1;     //用于Xi和Yj互斥进入路口A
mutex2=1;     //用于Xi和Zk互斥进入路口B
S1=1;     //用于Xi访问共享计数变量Cx1的互斥
S2=1;     //用于Yj访问共享计数变量Cy的互斥
S3=1;     //用于Xi访问共享计数变量Cx2的互斥
S4=1;     //用于Zk访问共享计数变量Cz的互斥
count=9; //路口A、B之间可容纳的汽车数
Cx1=0;   //用于Xi通过路口A的计数
Cx2=0;   //用于Xi通过路口B的计数
Cy=0;     //用于Yj通过路口A的计数
Cz=0;     //用于Zk通过路口B的计数
parbegin
process Xi  (i=1,2,3,  . . .  )
{
    P(count);  //看看路口A、B之间是否还有空间
    P(S1);      //准备进入路口A
    Cx1=Cx1+1;
    if (Cx1==1) P(mutex1);   //与Yj互斥通过路口A
    V(S1);      //允许后续Xi进入路口A
    通过路口A;
    P(S1);
    Cx1=Cx1-1;
    if (Cx1==0) V(mutex1);          //允许Yj通过路口A
    V(S1);
    在A、B段路面行驶;
    P(S3);              //准备进入路口B。互斥访问Cx2
    Cx2=Cx2+1;
    if (Cx2==1) P(mutex2);
    V(S3);
    V(count);         //AB段可容纳汽车数增1(唤醒因A、B段满员而阻塞的Xi)
    通过路口B;
    P(S3);      //互斥使用Cx2
    Cx2=Cx2-1;
    if (Cx2==0) V(mutex2);          //允许Zk通过路口B
    V(S3);
}
process Yj  (j=1,2,3, . . . )
{
    P(S2);
    Cy=Cy+1;
    if (Cy==1) P(mutex1);
```

```
    V(S2)
    通过路口A;
    P(S2);
    Cy=Cy-1;
    if (Cy==0) V(mutex1);
    V(S2);
}
process Zk  (k=1,2,3, . . .  )
{
    P(S4);
    Cz=Cz+1;
    if (Cz==1) P(mutex2);
    V(S4)
    通过路口B;
    P(S4);
    Cz=Cz-1;
    if (Cz==0) V(mutex2);
    V(S4);
}
parend
```

18. 仅有k个进程，它们的标号依次为1，2，…，k，允许它们同时读文件file，但必须满足条件：参加同时读文件的进程的标号之和需小于或等于k，请使用：(1) 信号量与P、V操作；(2) 管程，编写协调多进程读文件的算法程序。

解：(1) 采用信号量机制的算法描述如下：

```
semaphore mutex=1;     //互斥信号量，用于互斥访问共享变量sum
semaphore S=0;         //用于阻塞不能读文件的进程
int sum=0;             //用于记录和的值，若和大于k，则在S上阻塞
int count=0;           //用于记录在S上阻塞的进程数
Parbegin
process Pi (i=1,2,3, . . . , k)
{
L1: P(mutex);
    if (i+sum>k)
    {
        count ++;      //阻塞进程数增1
        V(mutex);      //允许其它进程读
        P(S);          //本进程阻塞
        goto L1;       //进程被唤醒后从L1重新开始执行
    }
    else
    {
        sum=sum+i;
        V(mutex);
    }
```

```
        Read File;
        P(mutex);
        sum=sum-i;
        while (count>0)
        {
            V(S);            //唤醒所有在S上阻塞的进程
            count --;        //也可只唤醒一个阻塞进程，但并发度降低
        }
        V(mutex);
    }
    Parend
```

（2）采用管程（名字为RF）的算法描述如下：

```
type RF=monitor
    var count : integer;
    C : condition;
    procedure entry Rin(i)
    begin
        L1:
        if count+i>k then
        begin
            C.wait;
            goto L1;
        end
        count:=count+i;
    end;
    procedure entry Rout(i)
    begin
        count := count-i;
        while C.queue do C.signal;
    end
    begin
        count := 0;
    end
process Pi (i=1,2,3, ... , k)
begin
    RF. Rin(i);
    Read File;
    RF.Rout(i)
end;
```

19. 有一个阅览室，读者进入时必须先在一张登记表上登记，该表位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。假如阅览室有100个座位。试用：（1）信号量和P、V操作；（2）管程，来实现读者进程的同步算法。

解：（1）

```
semaphore seat, mutex ;
    seat = 100;          //座位资源信号量
```

```
mutex = 1;          //用于互斥访问登记表
parbegin
  process STi  (i=1, 2, . . . )      //学生进出
  {
      P(seat);      //学生到达后，若已无座位，则等待
      P(mutex);   //登记互斥
      在登记表上登记;
      V(mutex);
      在阅览室阅览;
      P(mutex);
      注销登记信息;
      V(mutex);
      V(seat);            //空出一个座位，可能唤醒一个等待座位的学生
  }
parend
```

（2）使用管程的解法如下：

```
type Read=monitor
var seat : integer;
C : condition;
procedure entry in ( )          //进入
begin
    if (seat=0) then C.wait;          //若无座位，等待
    seat := seat-1;                        //座位数减1
    在登记表上登记;   //唤醒第一个吸烟者
end
procedure entry out (int k)   //离开阅览室
begin
    seat := seat-1;
    注销登记信息;
    if C.squeue then C.signal;       //若有等待者，则唤醒之
end
begin
    seat := 100;            //初始化
end
```

采用管程机制时，学生进程的行为描述如下：

```
process STi ( )            (i=1, 2, . . .  )
begin
     来到阅览室;
     Read. in ( );          //进入阅览室
     阅览;
     Read. out ( );         //走出阅览室
     离开阅览室;
end
parbegin
     ST1 ( );
```

```
        ST2 ( );
        ST3 ( );
        . . .
    parend
```

20. 一个快餐厅有4类职员：(1) 领班：接受顾客点菜；(2) 厨师：准备顾客的饭菜；(3) 打包工：将做好的饭菜打包；(4) 出纳员：收款并提交食品。每个职员和顾客可看做一个进程，试用一种同步机制写出能让4类职员正确并发运行的程序。

解：采用信号量和P、V操作来解决此同步问题

```
semaphore S1, S2, S3, S4, S5;
S1 = 0 ;  //顾客数，初值为0，用于阻塞领班
S2 = 0 ;  //厨师的资源信号量，初值为0，用于阻塞厨师
S3 = 0 ;  //打包工的资源信号量，初值为0，用于阻塞打包工
S4 = 0 ;  //出纳员的资源信号量，初值为0，用于阻塞出纳员
S5 = 0 ;  //顾客的资源信号量，初值为0，用于出纳员通知顾客来付款
S6 = 0 ;  //顾客的资源信号量，初值为0，用于出纳员通知顾客取食品
parbegin
  process 顾客
  {
      V(S1);        //顾客到达，唤醒领班
      完成向领班的点菜;
      P(S5);        //等待出纳员召唤
      付款;
      P(S6);        //等待取食品
      拿着食品离开;
  }
  process 领班
  {
      while(!) {
          P(S1);        //等待顾客到来
          接收顾客的点菜;
          V(S2);        //通知厨师准备顾客的饭菜
      }
  }
  process 厨师
  {
      while (1) {
          P(S2);        //等待领班的召唤
          准备顾客的饭菜;
          V(S3);        //通知打包工
      }
  }
  process 打包工
  {
      while (1) {
```

```
            P(S3);          //等待厨师的召唤
            将做好的饭菜打包;
            V(S4);          //通知出纳员
        }
    }
    process 出纳员
    {
        while (1) {
            P(S4);          //等待打包工的召唤
            V(S5);          //通知顾客来付款
            收款;
            V(S6);          //通知顾客并提交食品
        }
    }
parend
```

21. Jurassic公园有一个恐龙博物馆和一个花园，有m个旅客和n辆旅行车，每辆车仅能乘一个旅客。旅客在博物馆逛了一会儿，然后排队乘坐旅行车，当一辆车可用时，它载入一个旅客，再绕花园行驶任意长时间。若n辆车都已被旅客乘坐游玩，则想坐车的旅客要等待。如果一辆车已经空闲，但没有旅客了，那么。车辆要等待。试用信号量和P、V操作同步m个旅客和n辆车子。

解：

```
semaphore S1, S2, up, down;
S1=0;     //开始时没有旅客乘车
S2=n;     //有n辆旅行车
up=0;
down=0;
parbegin
    process 旅客i  (i=1, 2, . . . , m)
    {
        逛恐龙博物馆;
        P(S2);        //看看是否有旅行车可乘
        V(S1);        //唤醒等待的旅行车
        V(up);        //旅客上车
        看花园景色;
        P(down);      //等待下车
        下车;
        V(S2);        //空旅行车数增1，若有旅客等待则唤醒之
        离开公园;
    }
    process 旅行车j       (j=1, 2, . . . , n)
    {
        while (true) {
            P(S1);    //等待旅客到来
            P(up);    //等待旅客上好车
            带着旅客绕花园行驶;
            行驶完毕，回到发车点。
```

```
            停车，开车门;
            V(down);     //告诉哦旅客可以下车了
        }
    }
parend
```

22. (中科院2000年试题)现有100名毕业生去甲、乙两公司求职，两公司合用一间接待室，其中甲公司准备招收20人，乙公司准备招收10人，招完为止。两公司各有一位人事主管在接待学生，每位主管每次只可接待一人，其他毕业生在接待室外排成一个队伍等待。使用信号量和P、V操作实现对此过程的管理，写出需要的假设，以及所设计的数据结构和算法。

解：

```
semaphore mutex, S1, mutex1;
int count, count1, count2;
mutex=1;       //人事主管互斥访问变量count(学生人数)
S1=0;          //人事主管人数，用于阻塞学生进程
mutex1=1;      //用于互斥访问共享变量count1和count2
count=100;     //未面试学生人数
count1=20;     //甲公司准备招收的人数
count2=10;     //乙公司准备招收的人数
parbegin
  process 学生i     ( i=1, 2, . . . , 100 )
  {
      来到招聘处准备排队;
      P(mutex1);  //准备看看两公司是否还有招聘名额
      if (count1+count2==0)       //两个公司都已招满
      {  V(mutex1); exit( ); }       //离开
      else V(mutex1);
      P(S1);  //若人事主管都在忙于接待，则排队等待
      参加某一人事主管的招聘面试;
      面试结束，离开;
  }
  process 甲公司人事主管
  {
      while (true) {
          P(mutex);
          if (count==0) {V(mutex); break; } //若已无学生，结束
          V(mutex);
          P(mutex1);
          if (count1==0) {V(mutex1); break; }     //若已招满，离开
          V(mutex1);
          V(S1);    //招呼一学生进来面试
          与学生交谈、面试;
          if (录用该生) {
              记下该生的相关信息;
              P(mutex1);
              count1=count1-1;     //需招聘人数减1
```

```
            V(mutex1);
        }
        请学生离开;
        P(mutex);
        count=count-1;      //未面试学生人数减1
        V(mutex);
    }
    招聘结束，离开;
}
process 乙公司人事主管
{
    while (true) {
        P(mutex);
        if (count==0) {V(mutex); break; } //若已无学生，结束
        V(mutex);
        P(mutex1);
        if (count2==0) {V(mutex1); break; }    //若已招满，离开
        V(mutex1);
        V(S1);   //招呼一学生进来面试
        与学生交谈、面试;
        if (录用该生) {
            记下该生的相关信息;
            P(mutex1);
            count2=count2-1;   //需招聘人数减1
            V(mutex1);
        }
        请学生离开;
        P(mutex);
        count=count-1;      //未面试学生人数减1
        V(mutex);
    }
    招聘结束，离开;
}
parend
```

23. 有一个电子转账系统共管理10000个帐户，为了向客户提供快速转账业务，有许多并发执行的资金转账进程，每个进程读取一行输入，含有：贷方帐号、借方帐号、借贷的款项数。然后，把一款项从贷方帐号划转到借方帐号上，这样便完成了一笔转账交易。写出进程调用Monitor，以及Monitor控制电子资金转账系统的程序。

解：

```
type B=monitor      //建立管程B
var count, flag1, flag2 : integer;    //帐户计数
C1, C2, C3 : condition;
function in1( ) : integer          //读一行的进入区代码
begin
  if flag=1 then C1. wait;      //若有别的进程在读，等待
```

```
    if count>0 then flag1 := 1; //置正在读标志
    return count;              //返回未处理帐号数
  end
  procedure out1( )            //退出读一行状态
  begin
    count := count-1;          //未处理帐号数减1
    if C1.queue then C1.signal;       //若有进程等待读，则唤醒之
    else flag1 := 0;   //置无进程在读的标志
  end
  procedure in2 ( )    //转账操作的进入区代码
  begin
    if flag2=1 then C2.wait;
    flag2 := 1;
  end
  procedure out2 (    )
  begin
    if C2.queue then C2.signal;
    else flag2 := 0;
  end
  begin
     count := 1000; flag1 := 0; flag2 :=1;        //内部数据初始化
  end
  parbegin
    process Pi  ( i=1, 2, ... n )    //n个转账进程并发执行
    begin
      repeat
          x := B. in1( );            //调用管程的函数in1 ( )
          if x=0 then exit();      //若1000个帐号已处理完毕，进程结束
          读取一行输入R(含有：贷方帐号、借方帐号、借贷的款项数);
          B.out1( );
          B.in2( );
          按照输入行R，把一款项从贷方帐号划转到借方帐号上
          B.out2( );
       until flase
    end
  parend
```

24. 在一个分页存储管理系统中，用free[index]数组记录每个页框状态，共有n个页框(index=0, . . . , n-1)。当free[index]=true时，表示第index个页框空闲，free[index]=false时，表示第index个页框被占用。试设计一个管程，它有两个过程acquire和release分别负责分配和回收一个页框。

解：设计如下管程，命名为AM。

```
type AM=monitor
intvar flag : integer;
var C1, C2 : condition;
procedure  acquire ( )          //分配一个页框的过程
```

```
 begin
    L1: if flag=1 then begin
          C1.wait;   //若有别的进程在访问数组free[index]，阻塞
          goto L1;
    end
    flag := 1;
    L2: var index : integer :=0;
    while index<n do
    begin
        if free[index] = true then
        begin
           free[index] := flase;     //标记该页框已被占用
           分配index号页框给请求的进程;
           break;   //退出循环
        end
    end
    if index = n then begin
          flag := 0;
          if C1.queue then C1.signal;
          C2.wait;          //若无空闲页框可分配，阻塞
          goto L2;
    end
    flag := 0;
    if C1.queue then C1.signal;
 end
procedure  release (integer index)        //回收index号页框的过程
  begin
    L1: if flag=1 then begin
           C1.wait; //若有别的进程在访问数组free[index]，阻塞
           goto L1;
    end
    flag := 1;
    free[index] := flase;        //释放index号页框
    if C2.queue then C2.signal;    //若有进程因无空闲页框而阻塞，则唤醒之
    flag := 0;
    if C1.queue then C1.signal;    //若有进程等待访问free[ ]，则唤醒之
  end
begin
    flag := 0;        //内部数据初始化
end
```

25. (北交大1999)有一阅览室，读者进入阅览室必须先在一张登记表(TB)上登记，该表为每一座位设一个表目，读者离开时要消掉其登记信息。阅览室共有100个座位，为了描述读者的动作，请用类PASCAL语言和P、V操作写出进程间的同步算法。

约定：

1） flag的值：0——座位空闲；1——座位被占用。

2） 用语句i=getflag(0)可搜索到一个空座位i。用语句i.flag=0或1，可给标志位赋值。

3） 用i=getname(reader_name)可搜索到某读者所登记的座位号i。用i.name=0或i.name=reader_name可给姓名字段赋值，0表示清除读者姓名。

4） 计数信号量用count，互斥信号量用mutex。

座号 (i)	姓名(name)	标志(flag)
1	Wang	1
2	Zhang	1
⋮	⋮	⋮
99	0	0
100	0	0

解：var count, mutex : semaphore := 100, 1;

```
Reader_i ( )              { ( i=1, 2, . . . , n ) }
  begin
    P(count);          { 读者到来，先看看有无空座位 }
    P(mutex);          { 互斥访问登记表 }
    i := getflag(0);   { 在TB中搜索到空座位i }
    i.flag := 1;       { 置座位被占用的标志 }
    i.name=reader_name_i;     { 登记姓名 }
    V(mutex);
    在阅览室阅览;
    P(mutex);          { 准备消掉登记信息 }
    i := getname(reader_name_i);          { 在登记表中搜索自己的名字 }
    i.flag := 0;       { 注销登记信息 }
    i.name := 0;
    V(mutex);
    V(count);          { 座位数增1。若有读者等待，则唤醒之 }
  end
parbegin
  Reader_1 ( );
  Reader_2 ( );
  ……
  Reader_n ( );
parend
```

26．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？请写出能正确管理的程序。

解：参考程序如下：

```
begin
  S1, S2 : semaphore;
  S1:=1; S2:=0;
cobegin
  process A_to_Bi (i=1,2, …, n)
  begin
```

```
            P(S1);
        {表演};
        V(S2);
    end;
    process B_to_Aj  (j=1, 2, …, n)
    begin
        P(S2);
        {表演};
        V(S1);
    end;
coend;
end;
```

第三章 处理机调度与死锁

1．选择题

1．下列算法中，操作系统用于作业调度的算法是_____。

A．先来先服务算法　　B．先进先出算法

C．最先适应算法　　D．时间片轮转算法

2．在批处理系统中，周转时间是指_____。

A．作业运行时间　　B．作业等待时间和运行时间之和

C．作业的相对等待时间　　D．作业被调度进入内存到运行完毕的时间

3．在作业调度中，排队等待时间最长的作业被优先调度，这是指_____调度算法。

A．先来先服务　　B．短作业优先

C．响应比高优先　　D．优先级

4．下列算法中，用于进程调度的算法是_____。

A．最先适应　　B．最高响应比优先

C．均衡资源调度　　D．优先数调度

5．两个进程争夺同一个资源_____。

A．一定死锁　　B．不一定死锁

C．只要互斥就不会死锁　　D．以上说法都不对

6．下列各项中，不是进程调度时机的是 _____ 。

A．现运行的进程正常结束或异常结束　　B．现运行的进程从运行态进入就绪态

C．现运行的进程从运行态进入等待态　　D．有一进程从等待态进入就绪态

7．进程调度算法有多种，____ 不是进程调度算法。

A．先来先服务调度算法　　B．最短查找时间优先调度算法

C．静态优先数调度算法　　D．时间片轮转调度算法

8．作业调度程序从____状态的队列中选取适当的作业投入运行。

A．就绪　　B．提交　　C．等待　　D．后备

9．在实时操作系统中，经常采用____调度算法来分配处理器。

A.先来先服务　　B.时间片轮转　　C.最高优先级　　D.可抢占的优先级

10．采用时间片轮转调度算法主要是为了_____。

A．多个终端都能得到系统的及时响应

B．先来先服务

C．优先权高的进程及时得到调度

D．需要CPU时间最短的进程先做

11．下面关于优先权大小的论述中，不正确的论述是____。

A．计算型作业的优先权，应低于I/O型作业的优先权

B．系统进程的优先权应高于用户进程的优先权

C．资源要求多的作业，其优先权应高于资源要求少的作业

D．在动态优先权时，随着进程运行时间的增加，其优先权降低

12．产生死锁的原因是____有关。

A．与多个进程竞争CPU

B．与多个进程释放资源

C．仅由于并发进程的执行速度不当

D．除资源分配策略不当外，也与并发进程执行速度不当

13．有关产生死锁的叙述中，正确的是____。

A．V操作可能引起死锁　　B．P操作不会引起死锁

C．PV操作使用得当不会引起死锁　　D．以上说法均不正确

14．有关死锁的论述中，_____是正确的。

A．“系统中仅有一个进程进入了死锁状态”

B．“多个进程由于竞争CPU而进入死锁”

C．“多个进程由于竞争互斥使用的资源又互不相让而进入死锁”

D．“由于进程调用V操作而造成死锁”

15．有关资源分配图中存在环路和死锁关系，正确的说法是____。

A．图中无环路则系统可能存在死锁

B．图中无环路则系统可能存在死锁，也可能不存在死锁

C．图中有环路则系统肯定存在死锁

D．图中有环路则系统可能存在死锁，也可能不存在死锁

16．“死锁”问题的讨论是针对_____的。

A．某个进程申请系统中不存在的资源

B．某个进程申请资源数超过了系统拥有的最大资源数

C．硬件故障

D．多个并发进程竞争独占型资源

17．考虑到公平对待进程和提高系统资源工作的并行度，操作系统会经常调整进程的优先级，通常应提高____的进程优先级。

A．需计算时间长　　B．很少使用外设

C．使用CPU时间长　　D．启动外设次数多

18．实时系统中的进程调度，通常采用____算法。

A．响应比高者优先　　B．短作业优先

C．时间片轮转　　D．抢占式的优先数高者优先

19．UNIX操作系统采用的进程调度算法为____。

A、不可强占处理机的动态化先数调度算法

B、可强占处理机的动态化先数调度算法

C、不可强占处理机的静态优先数调度算法

D、可强占处理机的静态化先数调度算法

20．当进程调度采用最高优先级调度算法时，从保证系统效率的角度来看，应提高____进程的优先级。

A．连续占用处理器时间长的　　B．在就绪队列中等待时间长的

C．以计算为主的　　D．用户

21．产生系统死锁的原因可能是由于____。

A．进程释放资源　　B．一个进程进入死循环

C．多个进程竞争资源出现了循环等待　　D．多个进程竞争共享型设备

22．采用时间片轮转调度算法时，对不同的进程可以规定不同的时间片。一般来说，对_____进程给一个较小的时间片比较合适。

A．需运算时间长的　　B．需经常启动外设的

C．不需使用外设的　　D．排在就绪队列末尾的

23．对资源采用按序分配策略能达到_____的目的。

A．防止死锁　　B．避免死锁　　C．检测死锁　　D．解除死锁

24．一种既有利于短小作业又兼顾到长作业的作业调度算法是____。

A．先来先服务　　B．轮转　　C．最高响应比优先　　D．均衡调度

25．在单处理器的多进程系统中，进程什么时候占用处理器和能占用多长时间，取决于__

A．进程相应的程序段的长度　　B．进程总共需要运行时间多少

C．进程自身和进程调度策略　　D．进程完成什么功能

26．在解决死锁问题的方法中，属于“死锁避免”策略的是_____。

A．银行家算法　　B．死锁检测算法

C．资源有序分配法　　D．资源分配图化简法

27．系统出现死锁的原因是_____。

A．计算机系统出现了重大故障

B．有多个等待态的进程同时存在

C．若干进程因竞争资源而无休止地等待着它方释放已占有的资源

D．资源数大大少于进程数或进程同时申请的资源数大大超过资源总数

28．在操作系统中，所谓“死锁”是指_____。

A．程序死循环　　B．多个进程彼此等待资源而不能前进的状态

C．硬件故障　　D．时间片太短，进程的调进调出太频繁而效率太低

29．假设有三个进程竞争同类资源，如果每个进程需要2个该类资源，则至少需要提供该类资源_____个，才能保证不会发生死锁。

A．3　　B．4　　C．5　　D．6

30．以下_____不属于死锁的必要条件。

A．互斥使用资源　　B．占有并等待资源

C．不可抢夺资源　　D．静态分配资源

31．在为多个进程所提供的可共享的系统资源不足时，可能出现死锁。但是，不适当的___也可能产生死锁。

A．进程优先权　　B．资源的静态分配

C．进程的推进顺序　　D．分配队列优先权

32．采用资源剥夺法可以解除死锁，还可以采用___方法解除死锁。

A．执行并行操作　　B．撤消进程

C．拒绝分配新资源　　D．修改信号量

33．系统中有 4 个并发进程，都需要某类资源 3 个。试问该类资源最少为___个时，不会因竞争该资源而发生死锁。

A．9　　B．10　　C．11　　D．12

34．在下列解决死锁的方法中，不属于死锁预防策略的是____ 。

A．资源的有序分配法　　B．资源的静态分配法

C．分配的资源可剥夺法　　D．银行家算法

35．分时系统中进程调度算法通常采用____。

A．响应比高者优先　　B．时间片轮转法

C．先来先服务　　D．短作业优先

36．设有三个作业J1、J2、J3，它们的到达时间和执行时间如下表：

作业名	到达时间	执行时间
J1	8:00	2小时
J2	8:45	1小时

J3　　9:30　　0.25小时

它们在一台处理器上按单道运行，若采用短作业优先调度算法，则此三作业的执行次序是____。

A．J3,J2,J1　　B．J1,J2,J3

C．J1,J3,J2　　D．J3,J1,J2

37．在下列作业调度算法中，可能引起作业长时间不能被装入执行的算法是____ 。

A．FCFS算法　　B．计算时间短的作业优先算法

C．最高响应比优先算法　　D．动态优先数调度算法

38．windows98提供了多任务运行环境，允许占用处理器的进程运行一个规定的时间片，对处理器的分配采用了____算法。

A．先来先服务　　B．时间片轮转

C．优先数　　D．动态调整优先数

39．在非抢占调度方式下，运行进程执行 V 原语后，其状态_____。

A．不变　　B．要变　　C．可能要变　　D．可能不变

40．在多进程的并发系统中，肯定不会因竞争____而产生死锁。

A．打印机　　B．磁带机　　C．磁盘　　D．CPU

41．通常不采用____方法来解除死锁。(蓝色选项与教科书不同，但更合理)

A．终止一个死锁进程　　B．终止所有死锁进程

C．从死锁进程处抢夺资源　　D．从非死锁进程处抢夺资源

42*．UNIX System V 的进程调度原理基于____算法。

A．先来先服务　　B．短作业优先

C．时间片轮转　　D．时间片+优先级

43．设系统中有P1、P2、P3三个进程，并按P1、P2、P3的优先次序调度运行，它们的内部计算和I/O操作时间如下：

P1：计算60 ms—I/O 80 ms—计算20 ms

P2：计算120 ms—I/O 40ms—计算40ms

P3：计算40 ms—I/O 80ms—计算40ms

设调度程序执行时间忽略不计，完成这三个进程比单道运行节省的时间是_____。

A．140ms　　B．160ms　　C．170ms　　D．180ms

44．有三个作业A、B、C，它们的到达时间和执行时间依次为(8:50和1.5小时)、(9:00和0.4小时)、(9:30和1小时)。当作业全部到达后，批处理单道系统按响应比高者优先算法进行调度，则作业被选中的次序为_____。

A．(ABC)　　B．(BAC)　　C．(BCA)　　D．(CAB)

45．设系统中有n个并发进程，竞争资源R，且每个进程都需要m个R类资源，为使该系统不会因竞争该类资源而死锁，资源R至少要有____个。

A．n*m+1　　B．n*m+n　　C．n*m+1-n　　D．无法预计

46．下列选项中，降低进程优先级的合理时机是_____。（2010全国试题）

A．进程的时间片用完　　B．进程刚完成I/O，进入就绪队列

C．进程长期处于就绪队列中　　D．进程从就绪队列转为运行状态

47．下列进程调度算法中，综合考虑进程等待时间和执行时间的是_____。（2009 全国试题）

A．时间片轮转调度算法　　B．短进程优先调度算法

C．先来先服务调度算法　　D．高响应比优先调度算法

48．某计算机系统中有 8 台打印机，有 k 个进程竞争使用，每个进程最多需要 3 台打印机。该系统可能会发生死锁的 k 的最小值是_____。（2009 全国试题）

A．2　　B．3　　C．4　　D．5

49．进程调度的关键问题是_____。

A．内存的分配　　B．时间片的确定　　C．调度算法的确定　　D．I/O设备的分配

50．下列选项中，满足短任务优先且不会发生饥饿现象的调度算法是____。（2011全国试题）

A．先来先服务　　B．高响应比优先　　C．时间片轮转　　D．非抢占式短任务优先

51．某时刻进程的资源使用情况如下表所示。

进程	已分配资源			尚需资源			可用资源		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	2	0	0	0	0	1	0	2	1
P2	1	2	0	1	3	2			
P3	0	1	1	1	3	1			
P4	0	0	1	2	0	0			

此时的安全序列是____。

A．P1，P2，P3，P4　B．P1，P3，P2，P4　C．P1，P4，P3，P2　D．不存在

52．设有五个进程P0、P1、P2、P3、P4共享三类资源R1、R2、R3，这些资源总数分别为18、6、22，T0时刻的资源分配情况如下表所示，此时存在的一个安全序列是_____。（2012全国试题）

进程	已分配资源			资源最大需求		
	R1	R2	R3	R1	R2	R3
P0	3	2	3	5	5	10
P1	4	0	3	5	3	6
P2	4	0	5	4	0	11
P3	2	0	4	4	2	5
P4	3	1	4	4	2	4

A．P0，P2，P4，P1，P3　　B．P1，P0，P3，P4，P2

C．P2，P3，P4，P1，P0　　D．P3，P4，P2，P1，P0

53．一个多道批处理系统中仅有P1和P2两个作业，P2比P1晚5ms到达，它们的计算和I/O操作顺序如下：

P1：计算60ms，I/O80ms，计算20ms

P2：计算120ms，I/O40ms，计算40ms

若不考虑调度和切换时间，则完成两个作业需要的时间最少是_____。（2012全国试题）

A．240ms　　B．260ms　　C．340ms　　D．360ms

54．某单处理器多进程系统中有多个就绪进程，则下列关于处理机调度的叙述中，错误的是_____。

A．在进程结束时能进行处理机调度

B．创建新进程后能进行处理机调度

C．在进程处于临界区时不能进行处理机调度

D．在系统调用完成并返回用户态时能进行处理机调度

第三章处理机调度与死锁选择题参考答案：

1．A	2．B	3．A	4．D	5．B	6．D	7．B	8．D	9．D	10．A
11．C	12．D	13．D	14．C	15．D	16．D	17．D	18．D	19．A	20．B
21．C	22．B	23．A	24．C	25．C	26．A	27．C	28．B	29．B	30．D
31．C	32．B	33．A	34．D	35．B	36．C	37．B	38．B	39．A	40．D
41．C	42．D	43．B	44．B	45．C	46．A	47．D	48．C	49．C	50．B
51．D	52．D	53．B	54．C						

2．应用题

1. 有两个作业A和B，分别在7:00和8:30到达系统，它们估计的计算时间分别为0.8小时和0.1小时，系统在9:00开始以响应比高者优先算法进行调度。在单道系统中该两个作业被选中时的响应比各为多少？

解：9:00时，作业A的响应比=1+2/0.8=3.5
作业B的响应比=1+0.5/0.1=6
所以9:00时作业调度程序选中作业B
9:06作业B结束，调度作业A，此时作业A的响应比=1+2.1/0.8=3.625
综上可知，在单道系统中A、B两个作业被选中时的响应比分别为3.625和6

2. 有一个具有两道作业的批处理系统（最多可有两道作业同时装入内存执行），作业调度采用计算时间短的作业优先调度算法，进程调度采用以优先数为基础的抢占式调度算法，今有如下作业序列，作业优先数即为进程优先数，优先数越小优先级越高：

作业名	到达时间	估计运行时间	优先数
J1	10 : 10	20分钟	5
J2	10 : 20	30分钟	3
J3	10 : 30	25分钟	4
J4	10 : 50	20分钟	6

列出所有作业进入内存时间及结束时间。
（1）计算平均周转时间。

解：先作必要的分析（可在草稿纸上完成，分析过程不计分）：

10：10　　J1被调入，开始运行
10：20　　J2进入内存，因优先级高，开始运行
　　J1运行了10分钟，还剩10分钟，因优先级低，运行态变就绪态
10：30　　J1继续就绪
　　J2运行了10分钟，还剩20分钟
　　J3到达，但不能被调入
10：50　　J2运行结束，J4到达
　　调入短作业J4，但因J4优先级比J1低，J1开始继续运行
11：00　　J1运行结束
　　J3被调入，因优先级高，开始运行
　　J4因优先级低，仍就绪
11：25　　J3运行结束，J4开始运行
11：45　　J4运行结束

（1）各个作业进入主存时间、结束时间和周转时间如下表所示：（6分）

作业名	提交时间	进入时间	结束时间	周转时间
J1	10：10	10：10	11：00	50
J2	10：20	10：20	10：50	30
J3	10：30	11：00	11：25	55
J4	10：50	10：50	11：45	55

（2）平均周转时间：（50+30+55+55）/4=47.5（min）

3. 有一个多道程序设计系统，采用不可移动的可变分区方式管理主存空间，设主存空间为100K，采用最先适应分配算法分配主存，作业调度采用响应比高者优先算法，进程调度采用时间片轮转算法（即内存中的作业均分CPU时间），今有如下作业序列：

作业名	提交时间	需要执行时间	要求主存量
J1	10 : 00	40分钟	25K
J2	10 : 15	30分钟	60K
J3	10 : 30	20分钟	50K
J4	10 : 35	25分钟	18K
J5	10 : 40	15分钟	20K

假定所有作业都是计算型作业且忽略系统调度时间。回答下列问题：

(1) 列表说明各个作业被装入主存的时间、完成时间和周转时间；

(2) 写出各作业被调入主存的顺序；

(3) 计算5个作业的平均周转时间。

解：先作必要的分析（可在草稿纸上完成，分析过程不计分）：

10：00　J1到达，被调入内存并开始运行，内存剩75KB

10：15　J2到达，被调入内存并开始与J1一起运行。J1运行了15分钟，还需执行25分钟。内存剩15KB

10：30　J3到达，因内存不能满足，不被调入。J1、J2继续运行。

10：35　J4到达，因内存不能满足，不被调入。此时J1相当于已运行了15+10=25分钟，还需执行15分钟；J2相当于已运行了10分钟，还需执行20分钟。

10：40　J5到达，因内存不能满足，不被调入。J1、J2继续运行。

11：05　J1运行结束，J2还需执行5分钟。内存中的2个空闲分区分别是25K和15K，能满足J4或J5需求，但不能满足J3的需要。J4的响应比=1+30/25 =2. 2；J5的响应比=1+25/15=2.67，J5被调入内存并与J2一起运行，内存中的空闲分区大小是5KB和15K。

11：15　J2运行结束，内存中的空闲分区是65KB和15K。J3 的响应比=1+45/20=3.25，J4的响应比=1+40/25=2.6，J3被调入内存与J5一起运行。内存中的空闲分区大小是15KB和15K。J3需执行20分钟，J5还需执行10分钟。

11：35　J5运行结束，J4调入内存运行。J3还需执行10分钟，J4还需执行25分钟。

11：55　J3运行结束

12：10　J4运行结束

答：（1）各个作业被装入主存的时间、完成时间和周转时间如下表所示：

作业名	装入主存时间	作业完成时间	周转时间
J1	10：00	11：05	65
J2	10：15	11：15	60
J3	11：15	11：55	85
J4	11：35	12：10	95
J5	11：05	11：35	55

（2）作业被调入主存的顺序为J1，J2，J5，J3，J4。

（3）平均周转时间=（65+60+85+95+55）/5=72（分钟）。

4. (北京大学1997年试题)某系统有A,B,C三类资源（数量分别为17，5，20）和P1~P5五个进程，在T_0时刻系统状态如下表所示：

进程	最大资源需求量			已分配资源数量		
	A	B	C	A	B	C
P1	5	5	9	2	1	2
P2	5	3	6	4	0	2
P3	4	0	11	4	0	5
P4	4	2	5	2	0	4
P5	4	2	4	3	1	4

系统采用银行家算法实施死锁避免策略,请回答下列问题：

① T_0时刻是否为安全状态？若是，请给出安全序列。

②在T_0时刻若进程P2请求资源（0，3，4），是否能实施资源分配？为什么？

③在②的基础上，若进程P4请求资源（2，0，1），是否能实施资源分配？为什么？

解：① 由已知条件可得尚需矩阵Need和可用资源向量Avalable如下：

	Need			Avalable		
	A	B	C	A	B	C
P1	3	4	7	2	3	3
P2	1	3	4			
P3	0	0	6			
P4	2	2	1			
P5	1	1	0			

利用银行家算法对此时刻的资源分配情况进行分析如下表：

进程	Work	Need	Allocation	Work+Allocation	Finish
P4	2 3 3	2 2 1	2 0 4	4 3 7	true
P2	4 3 7	1 3 4	4 0 2	8 3 9	true
P3	8 3 9	0 0 6	4 0 5	12 3 14	true
P5	12 3 14	1 1 0	3 1 4	15 4 18	true
P1	15 4 18	3 4 7	2 1 2	17 5 20	true

从上述分析可知，存在一个安全序列P4，P2，P3，P5，P1，故T_0时刻系统是否安全的。

② 在T_0时刻若进程P2请求资源(0，3，4)，不能实施资源分配。因为当前C类资源剩余3个而P2请求4个，客观条件无法满足它的请求，因此不能实施资源分配，P2阻塞。

③ 在②的基础上，若进程P4请求资源（2，0，1），可以实施资源分配。因为由①可知，P4是安全序列中的第一个进程，只要P4的请求量没有超出它的尚需量，系统满足它的请求后仍处于安全状态，即仍然存在安全序列P4，P2，P3，P5，P1。

5. 某计算机系统有9台磁带机，它们供N个进程竞争使用，每个进程可能需要3台磁带机。请问N为多少时，系统没有死锁的危险，并说明其原因。

解：最坏的情况是N个进程每个进程都分得了2台磁带机，若在这种情况下仍有富余的磁带机，可供某一个进程使用，则该进程可得到所需的全部磁带机，从而可运行完成。该进程运行完成后释放的磁带机又可共其他进程使用，从而使得到磁带机的进程运行完成。它们释放的磁带机又可共其他没有完成的进程使用，如此下去，最终可使所有进程得到所需的全部磁带机，从而运行到底。这种情况就没有因竞争磁带机而发生死锁的危险。由上分析，只要满足下式

$N(3-1)+1 \leqslant 9$

即 $N \leqslant 4$ 时，系统没有死锁的危险。

6. 某系统有同类资源m个供n个进程共享，如果每个进程最多需要x个资源（$1 \leqslant x \leqslant m$）且各进程的最大需求量之和$\Sigma Need_i$小于（m + n）。证明系统没有因申请该类资源而发生死锁的危险。

证明：在最坏情况下，每个进程都已占有(x-1)个该类资源，各进程最多再申请 1 个资源就可以运行完毕，进而释放它所占有的全部资源。

在此情况下，系统剩余的资源数为：m-n*(x-1)。当 m-n*(x-1)≥1 时，即 n*x≤m+n-1 时，至少有 1 个进程可以获得全部资源，从而能运行完成，释放资源供别的进程使用，因此系统不会出现死锁。

因此得出，系统中所有进程的最大需求之和ΣNeed$_i$满足下式时不会死锁：

ΣNeed$_i$=n*x≤m+n-1 或 ΣNeed$_i$ <m+n

证毕。

7. 用银行家算法考虑下列系统状态：

进程　分配矩阵　　最大需求矩阵　　资源总数向量

A　3 0 1 1　　4 1 1 1　　6 3 4 2

B　0 1 0 0　　0 2 1 2

C　1 1 1 0　　4 2 1 0

D　1 1 0 1　　1 1 1 1

E　0 0 0 0　　2 1 1 0

问：(1) 系统是否安全？（应说明理由）

(2) 若进程B请求(0,0,1,0)，可否立即分配？请分析说明。

(3) 此后进程E也请求(0,0,1,0)，可否分配给它？请分析说明。

解：(1) 由已知条件可得Need和Avaiable矩阵如下：

进程　分配矩阵　　尚需矩阵(Need)　可用资源数向量(Avaiable)

A　3 0 1 1　　1 1 0 0　　1 0 2 0

B　0 1 0 0　　0 1 1 2

C　1 1 1 0　　3 1 0 0

D　1 1 0 1　　0 0 1 0

E　0 0 0 0　　2 1 1 0

利用银行家算法对此时刻的资源分配情况进行分析如下表：

进程	Work	Need	Allocation	Work+Allocation	Finish
D	1 0 2 0	0 0 1 0	1 1 0 1	2 1 2 1	true
A	2 1 2 1	1 1 0 0	3 0 1 1	5 1 3 2	true
B	5 1 3 2	0 1 1 2	0 1 0 0	5 2 3 2	true
C	5 2 3 2	3 1 0 0	1 1 1 0	6 3 4 2	true
E	6 3 4 2	2 1 1 0	0 0 0 0	6 3 4 2	true

从上述分析可知，存在一个安全序列D，A，B，C，E，故当前系统是否安全的。

（2）若进程B请求(0,0,1,0)，试分配并修改相应的数据结构，则系统状态变为：

进程　分配矩阵　　尚需矩阵(Need)　可用资源数向量(Avaiable)

A　3 0 1 1　　1 1 0 0　　1 0 1 0

B　0 1 1 0　　0 1 0 2

C　1 1 1 0　　3 1 0 0

D　1 1 0 1　　0 0 1 0

E　0 0 0 0　　2 1 1 0

利用银行家算法对此时刻的资源分配情况进行分析如下表：

进程	Work	Need	Allocation	Work+Allocation	Finish
D	1 0 1 0	0 0 1 0	1 1 0 1	2 1 1 1	true

A	2 1 1 1	1 1 0 0	3 0 1 1	5 1 2 2	true
B	5 1 2 2	0 1 0 2	0 1 1 0	5 2 3 2	true
C	5 2 3 2	3 1 0 0	1 1 1 0	6 3 4 2	true
E	6 3 4 2	2 1 1 0	0 0 0 0	6 3 4 2	true

从上述分析可知，存在安全序列D，A，B，C，E，故系统仍是否安全的，因此可以立即分配。

（3）此后进程E也请求(0,0,1,0)，则系统状态变为：

进程　分配矩阵　　尚需矩阵(Need)　可用资源数向量(Avaiable)

A　3 0 1 1　　1 1 0 0　　1 0 0 0

B　0 1 1 0　　0 1 0 2

C　1 1 1 0　　3 1 0 0

D　1 1 0 1　　0 0 1 0

E　0 0 1 0　　2 1 0 0

此时系统剩余资源（1，0，0，0）已不能满足任何进程的需求，即已找不到一个安全序列，系统状态将变为不安全，故不能分配给E。

8. 某系统有A、B、C、D这4类资源供5个进程共享，进程对资源的需求和分配情况如下表所示。现在系统中A、B、C、D类资源分别还剩1、5、2、0个，请按银行家算法回答下列问题：

进程	已占资源				最大需求数			
	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	2
P2	1	0	0	0	1	7	5	0
P3	1	3	5	4	2	3	5	6
P4	0	6	3	2	0	6	5	2
P5	0	0	1	4	0	6	5	6

现在系统是否处于安全状态? 为什么？

(1) 如果现在进程P2提出需要(0，4，2，0)个资源的请求，系统能否满足它的请求？为什么？

解：(1) 由已知条件可得Need矩阵如下：

进程　分配矩阵　　尚需矩阵(Need)　可用资源数向量(Avaiable)

P1　0 0 1 2　　0 0 0 0　　1 5 2 0

P2　1 0 0 0　　0 7 5 0

P3　1 3 5 4　　1 0 0 2

P4　0 6 3 2　　0 0 2 0

P5　0 0 1 4　　0 6 4 2

利用银行家算法对此时刻的资源分配情况进行分析如下表：

进程	Work	Need	Allocation	Work+Allocation	Finish
P1	1 5 2 0	0 0 0 0	0 0 1 2	1 5 3 2	true
P3	1 5 3 2	1 0 0 2	1 3 5 4	2 8 8 6	true
P2	2 8 8 6	0 7 5 0	1 0 0 0	3 8 8 6	true
P4	3 8 8 6	0 0 2 0	0 6 3 2	3 14 11 8	true
P5	3 14 11 8	0 6 4 2	0 0 1 4	3 14 12 12	true

从上述分析可知，存在安全序列P1，P3，P2，P4，P5，故系统状态是否安全的。（注：安全序列不唯一）

（2）若进程P2请求(0,4,2,0)，试分配并修改相应的数据结构，则系统状态变为：

进程	分配矩阵	尚需矩阵(Need)	可用资源数向量(Avaiable)
P1	0 0 1 2	0 0 0 0	1 1 0 0
P2	1 4 2 0	0 3 3 0	
P3	1 3 5 4	1 0 0 2	
P4	0 6 3 2	0 0 2 0	
P5	0 0 1 4	0 6 4 2	

进程P1已获得全部资源，可运行完成。P1结束归还资源后，系统剩余资源为(1, 1, 1, 2)，能满足P3的需求，P3可运行完成。P3结束释放资源后，系统剩余资源为(2, 4, 6, 6)，能满足P2的需求，P2可运行完成。P2结束释放资源后，系统剩余资源为(3, 8, 8, 6)。类似地，P4、P5也能获得所需资源而运行完成。因此存在安全序列P1, P3, P2, P4, P5，即系统仍然是否安全的，因此系统能满足P2的请求。

9. 对于页面大小为512字节的计算机，编制了下列汇编语言程序，程序的内存起始地址为1020，其堆栈指针指向逻辑地址8192（向低地址扩展），每条指令占据4个字节，给出程序执行的页面访问序列。

1020：（1）装入6144单元的字到0号寄存器；
1024：（2）压寄存器0到堆栈；
1028：（3）调用5120处的子程序，（压入返回地址到堆栈）；
5120：（4）堆栈指针减去立即数16；
5124：（5）比较堆栈指针所指单元内容和立即数4；
5128：（6）若相等转到5152。

解：取指令(1)，访问1#页，执行指令(1)时访问6144单元，因6144/512=12，故访问12#页；
取指令(2)，访问 2#页，执行指令(2)时访问堆栈(入栈)。入栈过程是：首先堆栈指针减 1，指向 8191，然后 0#寄存器中的内容(1 个字)入栈，假设字长为 4 个字节，则入栈后堆栈指针指向 8188。8191/512=15.9，因此入栈时访问的是15#页；
取指令(3)，访问 2#页，执行指令(3)时，因是调用子程序语句，故(3)的下一条指令(本题未给出)的地址 1032 需入栈，即需访问 15#页；
取指令(4)，访问 10#页，指令执行中不访问任何地址；
取指令(5)，访问 10#页，执行指令(5)时访问堆栈，即访问 15#页；
取指令(6)，访问 10#页，指令执行中不访问任何地址。
综上所述，题目所给的汇编程序段对应的页面访问序列为 1，12，2，15，2，15，10，10，15，10。

10. 某系统中有 10 台打印机，有三个进程 P_1，P_2，P_3 分别需要 8 台，7 台和 4 台。若 P_1，P_2，P_3 已申请到 4 台，2 台和 2 台。试问：按银行家算法能安全分配吗？请说明分配过程。

解：由题目所给条件，可得如下有关数据结构：

进程	Max	Allocation	Need	Available
P1	8	4	4	2
P2	7	2	5	
P3	4	2	2	

故按银行家算法能安全分配。分配过程是：首先将当前剩余的 2 台打印机全部分配给 P3，使 P3 得到所需的全部打印机数，从而可运行到完成。P3 完成后，释放的 4 台打印机全部分配给 P1，使 P1 也能运行完成；P1 完成后释放的打印机可供 P2 使用，使 P2 也能运行结束。即系统按 P3、P1、P2 的顺序分配打印机，就能保证系统状态是安全的。

11. 假如有四道作业，它们的提交时间及运行时间由下表给出：

作业号	提交时刻(时)	运行时间(小时)

1	8. 00	2. 00
2	8. 50	0. 50
3	9. 00	0. 10
4	9. 50	0. 20

采用单道运行，试问用先到先服务(FCFS)作业调度算法

(1)计算平均周转时间。(注：提交时刻小数点后的计时单位是小时即 8. 50 相当于 8:30)

(2)写出调度作业的顺序。

12. 有5个批处理作业（A，B，C，D，E）几乎同时到达一个计算中心，估计的运行时间分别为10，6，2，4，8分钟，他们的优先数分别为1，2，3，4，5（1为最低优先数）。对下面的各种调度算法，分别计算作业的平均周期时间。

（1）最高优先级优先

（2）短作业优先

解：(1) 采用最高优先级优先调度算法，各进程开始运行的时间、完成时间以及周转时间如下表：

进程	开始运行时间	完成时间	周转时间
A	20	30	30
B	14	20	20
C	12	14	14
D	8	12	12
E	0	8	8

平均周转时间为(30+20+14+12+8)/5=84/5=16.8(ms)

(2) 采用短作业优先调度算法，各进程开始运行的时间、完成时间以及周转时间如下表：

进程	开始运行时间	完成时间	周转时间
A	20	30	30
B	6	12	12
C	0	2	2
D	2	6	6
E	12	20	20

平均周转时间为(30+12+2+6+20)/5=70/5=14 (ms)

13. 假设某系统有某类资源12个，有P1、P2、P3三个进程来共享，已知P1、P2、P3所需该类资源总数分别为8、6、9，它们申请资源的次序和数量如下表所示：

序号	进程	申请量
1	P3	4
2	P1	2
3	P2	4
4	P1	1
5	P3	2
6	P2	2
:	:	:

系统采用银行家算法分配资源，问：

（1） 哪几次申请被满足会使系统进入不安全状态？请应说明理由。

（2） 执行完序号为6的申请后，各进程的状态和各进程占用的资源数如何？

答：(1) 显然，前3次申请满足后，系统状态仍是安全的，因为，此时系统中剩余资源数为2，P1、P2、P3

尚需资源数分别为6、2、5，存在安全序列{P2，P1，P3}；

此后，P1再申请1个资源(序号4)，若被满足，则系统中剩余的可用资源数为1，而P1、P2、P3尚需资源数分别为5、2、5，剩余资源已不能满足任何进程的尚需量，即已不存在安全序列，系统进入不安全状态。故系统不能满足P1的申请，P1阻塞。

同样，P3申请2个资源(序号5)，若被满足，则系统中剩余的可用资源数为0，而P1、P2、P3尚需资源数分别为6、2、3，剩余资源已不能满足任何进程的尚需量，即已不存在安全序列，系统进入不安全状态。故系统不能满足P3的申请，P3阻塞。

P2申请2个资源(序号6)，可以被满足。

(2) 执行完序号为6的申请后，P1、P3处于阻塞状态，P2处于运行状态或就绪状态。P1、P2、P3占用的资源数分别为2、6、4。

14. 当前系统中总共有10个资源，系统采用银行家算法分配资源。现有P、Q、R三个进程，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为：

序号	进程	申请数
1	Q	1
2	P	3
3	R	4
4	P	1
5	Q	2
6	R	2

回答(1)把系统处理完上述诸请求后，各进程的状态及所占资源量填入下表：

进程	所占资源量	进程状态
P		
Q		
R		

(2)若进程继续申请资源，你估计系统是否会出现死锁？为什么？

解：(1) 系统处理完上述诸请求后，各进程的状态及所占资源量如下表所示。

进程	所占资源量	进程状态
P	4	运行
Q	1	阻塞
R	4	阻塞

(2) 进程Q、R已阻塞，若进程P继续申请1个资源，可以得到满足，但系统中已无资源可分配，而进程P还需要3个资源，此后进程P再申请资源将会阻塞。最终，P、Q、R全部等待资源而阻塞，即发生了死锁。

15. 在单道批处理系统中，有四个作业到达输入井和需要的计算时间如表所示，现采用响应比最高者优先算法，忽略作业调度所化的时间。当第一个作业进入系统后就可开始调度。

作业	到达输入井时间	需计算时间	开始时间	完成时间	周转时间
1	8：00	2小时			
2	8：30	30分钟			
3	9：00	6分钟			
4	9：30	12分钟			

（1）填充表中空白处

（2）四个作业的执行次序为_____。

（3）四个作业的平均周转时间为_____。

16. 在单道批处理系统中，有下列四个作业，采用计算时间短的作业优先的调度算法，当第一个作业进入系统后就可以开始调度，忽略调度及I/O所化的时间。

(1) 按上述要求填充表中空白处

作业号	进入系统时间	需计算时间	开始时间	完成时间	周转时间
1	10:00	24 分钟			
2	10:06	1 小时			
3	10:12	36 分钟			
4	10:18	12 分钟			

(2) 四个作业的平均周转时间为_____。

解：（1）

作业号	进入系统时间	需计算时间	开始时间	完成时间	周转时间
1	10:00	24 分钟	10:00	10:24	24 分钟
2	10:06	1 小时	11:12	12:12	126 分钟
3	10:12	36 分钟	10:36	11:12	60 分钟
4	10:18	12 分钟	10:24	10:36	18 分钟

（2）(24+126+60+18)/4=57（分钟）。

17. 某系统中进程调度采用“时间片轮转”的策略，每个进程得到的时间片可随进程执行情况而变化。若进程经常产生中断，则给它分配较短的时间片，若进程被中断的次数很少，则分给一个较长的时间片，请解释为什么要这样做？

解：经常产生中断的进程连续占用处理器的时间较短，即使给它较长的时间片，它也可能在时间片未用完之前主动让出处理器，故只需较短时间片。中断次数很少的进程需要较长时间的连续运行，给它较长的时间片可减少进程调度次数从而减少系统开销。

18. 在一个单道程序系统，采用响应比高者优先算法管理作业。今有如下所示的作业序列，请列出各个作业的开始时间、完成时间和周转时间。注意：忽略系统开销。

作业名	进入输入井时间	需计算时间
JOB1	8.0时	1小时
JOB2	8.2时	0.8小时
JOB3	8.4时	0.4小时
JOB4	8.6时	0.6小时

19. 在一个多道程序系统中，设用户空间为 100K，主存空间管理采用最先适应分配算法，并采用先来先服务算法管理作业和进程。今有如下所示的作业序列，请列出各个作业开始执行时间、完成时间和周转时间。注意：忽略系统开销，时间用 10 进制。

作业名	进入输入井时间	需计算时间	主存需求量
JOB1	8. 0时	1小时	20K
JOB2	8. 2时	0. 6小时	60K
JOB3	8. 4时	0. 5小时	25K
JOB4	8. 6时	1小时	20K

20. 某系统中有 10 台打印机，有三个进程 P_1，P_2，P_3 分别需要 8 台，7 台和 4 台。若 P_1，P_2，P_3 已申请到 4 台，2 台和 2 台。试问：按银行家算法能安全分配吗？请说明分配过程。

21．某系统有如图 3-1 的状态变化图：

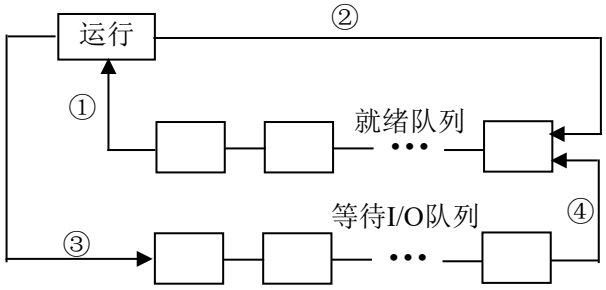


图3-1 系统状态变化图示

请回答下列问题：

(1)你认为该系统采用了怎样的进程调度策略?说出理由。

(2)把图中发生①～④的状态变化的具体原因填入下表的相应栏内。

变化	变 化 原 因
①	
②	
③	
④	

答：（1）该系统采用时间片轮转调度策略。因为进程会从运行态转换为就绪态的情况，故不可能是先来先服务调度策略，只可能是时间片轮转或抢占式动态优先级调度策略，但所有就绪进程都是排到就绪队列的尾部，因此不是抢占式动态优先级调度策略，而是时间片轮转策略。

（2）

变化	变 化 原 因
①	进程被调度程序选中，由就绪态变为运行态，占用处理机执行
②	进程时间片用完，由运行状态变为就绪状态，并排到就绪队列尾部
③	进程因等待I/O而由执行态变为阻塞态
④	进程等待的I/O已完成，由阻塞态变为就绪态，并排到就绪队列尾部

22．假定某系统当时的资源分配图如图3-2所示：

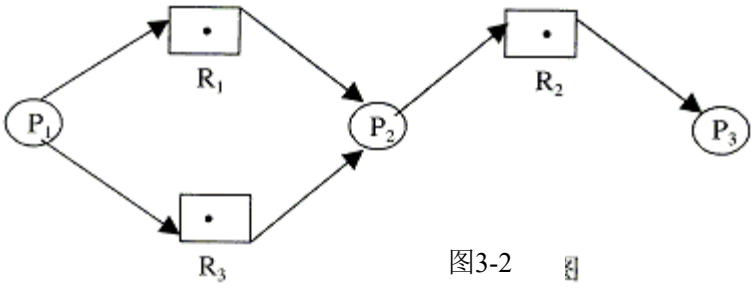


图3-2

（1）分析当时系统是否存在死锁。

（2）若进程P_3再申请R_3时，系统将发生什么变化，说明原因。

解：(1) 因为当时系统的资源分配图中不存在环路．所以不存在死锁。

(2) 当进程P_3申请资源R_3后，资源分配图中形成环路$P_2 \rightarrow R_2 \rightarrow P_3 \rightarrow R_3 \rightarrow P_2$，而$R_2$,$R_3$都是单个资源的类，该环路无法消除，所以进程$P_2$，$P_3$永远处于等待状态．从而引起死锁。

23．在一个多道程序设计系统中，不采用移动技术的可变分区方式管理主存。设用户空间为 100K，主存

空间采用最先适应分配算法，采用计算时间短的作业优先算法管理作业。今有如下所示的作业序列，请分别列出各个作业的开始执行时间、完成时间和周转时间。(注意：忽略系统开销)

作业名	进入输入井时间	需计算时间	主存需求存量
JOB1	8.0时	1小时	20K
JOB2	8.2时	0.6时	60K
JOB3	8.4时	0.5时	25k
JOB4	8.6时	0.4时	20K

解题时可在草稿上完成如下的分析过程：

8.0 时—　JOB1 到达，调入内存执行，占用内存 20KB，内存剩余空闲分区大小为 80KB

8.2 时—　JOB2 到达，调入内存成为就绪进程(非抢占式)，内存剩余空闲分区大小为 20KB

8.4 时—　JOB3 到达，因内存不够，不能调入

8.6 时—　JOB4 到达，调入内存成为就绪进程，内存全部用完

9.0 时—　JOB1 运行结束，空出 20KB 内存，JOB3 仍不能调入。JOB4 开始运行(因短作业优先调度常伴随短进程优先调度)

9.4 时—　JOB4 运行结束，空出 20KB 内存，内存中有 2 个大小皆为 20KB 的空闲分区，因不采用移动技术，故不能合并，JOB3 仍不能调入。JOB2 开始运行。

10.0 时—　JOB2 运行结束，空出 60KB 内存，3 个空闲分区合并成 1 个 100KB 的空闲分区，JOB3 调入内存开始运行

10.5 时—　JOB3 运行结束

答：各个作业的开始执行时间、完成时间和周转时间如下表所示。

作业名	开始执行时间	完成时间	周转时间
JOB1	8.0时	9.0时	1小时
JOB2	9.4时	10.0时	1.8小时
JOB3	10.0时	10.5时	2.1小时
JOB4	9.0时	9.4时	0.8小时

24．某多道程序设计系统供用户使用的主存为 100KB，磁带机 2 台，打印机 1 台。采用可变分区内存管理，采用静态方式分配外围设备，忽略用户作业的 I/O 时间。现有如下作业序列：

作业名	提交时间	需运行时间	主存需求量	磁带机需求	打印机需求
J1	8:00	25分钟	15KB	1	1
J2	8:20	10分钟	30KB	0	1
J3	8:20	20分钟	60KB	1	0
J4	8:30	20分钟	20KB	1	0
J5	8:35	15分钟	10KB	1	1

作业调度采用 FCFS 策略，优先分配主存低地址区域且不准移动已在主存中的作业，在主存中的作业均分 CPU 时间。现求：

(1) 作业被调度的先后次序；

(2) 全部作业运行结束的时间；

(3) 作业的平均周转时间；

(4) 最大作业周转时间。

先在草稿上分析如下：

8:00—　J1 到达，分配它所需资源(15KB 内存、 1 台磁带机、1 台打印机后，调入内存运行。余内存

85KB、磁带机 1 台。

8:20— J2 到达，因无打印机，不调入。同时 J3 到达，分配它内存 60KB，磁带机 1 台，调入内存，与 J1 均分 CPU 时间运行。余内存 25KB、磁带机和打印机都已分完(余 0 台)。

8:30— J1 结束，释放内存 15KB、磁带机 1 台、打印机 1 台。虽有打印机但内存不够，J2 仍不能调入；J4 到达，因低端内存 15KB 不够，分配高端内存 20KB 和磁带机 1 台，调入内存与 J3 一起运行。剩下内存空闲块是 15KB、5KB，打印机 1 台

8:35— J5 到达，因无磁带机，不能调入。

9:00— J3 结束。释放资源后，系统有内存 75KB，5KB、打印机和磁带机个 1 台。J2 调入，内存余 45KB，5KB、磁带机剩 1 台、打印机 0 台。J5 仍不能进入(无打印机)。将 J2、J4 运行。J4 还需运行 5 分钟。

9:05— J4 结束，释放资源后，内存空余 70KB、磁带机空 2 台、打印机 0 台。J5 仍不能进入。J2 单独运行(还需 5 分钟)。

9:15— J2 结束，释放资源后，内存有 100KB、磁带机有 2 台、打印机有 1 台。J5 调入运行。

9:30— J5 结束。

答：(1) 作业被调度的先后次序为 J1, J3, J4, J2, J5

(2) 全部作业运行结束的时间为 9:30

(3) 作业的平均周转时间为(30+55+40+40+55)÷5=44 (分钟)

(4) 最大作业周转时间为 55 分钟。

25．多道批处理系统中配有一个处理器和 2 台外设(D1 和 D2)，用户存储空间为 100MB。已知系统采用可抢占式的高优先数调度算法(优先数越大优先级越高)和不允许移动的可变分区分配策略，设备分配按照动态分配原则。今有 4 个作业同时提交给系统，如下表所示。

作业名	优先数	运行时间	内存需求
A	6	5分钟	50M
B	3	4分钟	10M
C	8	7分钟	60M
D	4	6分钟	20M

作业运行时间和I/O时间按下述顺序进行：

A． CPU (1分钟)，D1(2分钟)，D2(2分钟)

B． CPU (3分钟)，D1(1分钟)

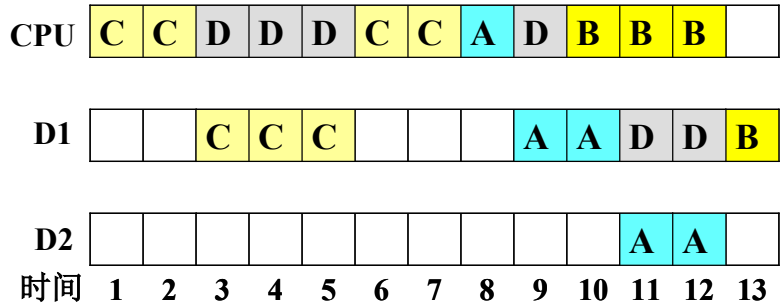
C． CPU (2分钟)，D1(3分钟)，CPU(2分钟)

D． CPU (4分钟)，D1(2分钟)

忽略其他辅助操作，求4个作业的平均周转时间是多少分钟。

解：先作必要的分析（可在草稿纸上完成，分析过程不计分）：

可用画图分析法：



也可用文字分析如下：

8：00　　4个作业同时到达，C被调入内存并开始运行，内存剩40MB。A因内存不够，不能调入，D、B也先后被调入内存，但不运行。剩余10MB。

8：02　　C让出CPU，开始使用外设D1。D运行。
8：05　　C用完外设D1，抢占CPU运行，D进入就绪状态。
8：07　　C运行结束。空出的60MB内存可使A调入并开始运行。
8：08　　A因使用外设D1而让出CPU，D继续运行。
8：09　　D因等待外设D1而阻塞，B开始运行。
8：10　　A用完外设D1，又开始用外设D2。D开始使用外设D1。
8：12　　A用完外设D2，结束；D用完外设D1，也结束。B开始使用外设D1。
8：13　　B用完外设D1而结束

由上述分析可知，各个作业的完成时间和周转时间如下表所示：

作业名	完成时间	周转时间
A	8：12	12分钟
B	8：13	13分钟
C	8：07	7分钟
D	8：12	12分钟

故平均周转时间 = (12+13+7+12) / 5 = 11（分钟）。

26．某多道程序设计系统采用可变分区内存管理，供用户使用的主存为 200KB，磁带机 5 台。采用静态方式分配外围设备，且不能够移动在主存中的作业，忽略用户作业的 I/O 时间、调度时间和移动作业时间。现有如下作业序列：

作业名	进入后备队列时间	运行时间	主存需求量	磁带机需求
A	8:30	40分钟	30KB	3
B	8:50	25分钟	120KB	1
C	9:00	35分钟	100KB	2
D	9:05	20分钟	20KB	3
E	9:10	10分钟	60KB	1

作业调度采用最高响应比优先算法、进程调度采用 SPF 算法时，求作业调度选中作业的次序及作业平均周转时间。

解题时可做如下分析，以便帮助解题：

8:30 —　　A 到达，调入内存运行。主存空闲分区为 170KB，磁带机还剩 2 台。
8:50 —　　B 到达，调入内存但不运行(SJF 算法属非抢占式)。主存空闲区为 50KB，磁带机还剩 1 台。A 还需运行 20 分钟
9:00 —　　C 到达，因内存和磁带机均不够，不能调入内存运行。A 还需运行 10 分钟
9:05 —　　D 到达，因磁带机不够，不能调入内存运行。
9:10 —　　E 到达，A 结束。内存中 2 个空闲分区大小分别为 30KB 和 50KB，磁带机还剩 4 台。C 的响应比=1+10/35=1.286，D 的响应比=1+5/20=1.25，E 的响应比=1+0/10=1.0。因 C 需内存 100KB，不够，故调入 D。内存中 2 个空闲分区大小分别为 10KB 和 50KB，磁带机还剩 1 台。因内存不够，E 也不能调入。因进程调度采用 SPF 算法，D 开始运行。
9:30 —　　D 结束，内存空闲区为 30KB 与 50KB，可用磁带机 4 台。因内存不够，C 和 E 仍不能调入。B 开始运行。

9:55 —　B 结束，释放内存和磁带机后，系统中有内存 200KB 和 5 台磁带机。C 的响应比=1+55/35=2.57，E 的响应比=1+45/10=5.5。E 先调入，随后 C 调入。短进程 E 开始运行。

10:05 —　E 结束，C 开始运行。

10:40—　C 结束

解：(1) 作业调度选中作业的次序为 A、B、D、E、C。（4 分）

(2) 作业 A 在 9:10 结束，其周转时间为 40 分钟；

作业 B 在 9:55 结束，其周转时间为 65 分钟；

作业 C 在 10:40 结束，其周转时间为 100 分钟；

作业 D 在 9:30 结束，其周转时间为 25 分钟；

作业 E 在 10:05 结束，其周转时间为 55 分钟；

故平均周转时间为(40+65+100+25+55)/5=57（分钟）（6 分）

27．某多道程序设计系统采用可重定位分区内存管理(即允许移动在主存中的作业)，供用户使用的主存为 200KB，磁带机 5 台。采用静态方式分配外围设备，忽略用户作业的 I/O 时间、调度时间和移动作业时间。现有如下作业序列：

作业名	进入后备队列时间	运行时间	主存需求量	磁带机需求
A	8:30	40分钟	30KB	3
B	8:50	25分钟	120KB	1
C	9:00	35分钟	100KB	2
D	9:05	20分钟	20KB	3
E	9:10	10分钟	60KB	1

假设作业调度采用最高响应比优先算法，进程调度采用时间片轮转算法 (均分 CPU 时间)。请回答下列问题：（此题是上题的变形）

(1) 写出作业调度选中作业的次序。

(2) 作业平均周转时间是多少分钟？

解题时可做如下分析，以便帮助解题：

8:30 —　A 到达，调入内存运行。主存空闲分区为 170KB，磁带机还剩 2 台。

8:50 —　B 到达，调入内存与 A 轮转运行。主存空闲区为 50KB，磁带机还剩 1 台。A 还需运行 20 分钟

9:00 —　C 到达，因内存和磁带机均不够，不能调入内存运行。A 还需运行 15 分钟，B 还需 20 分钟

9:05 —　D 到达，因磁带机不够，不能调入内存运行。

9:10 —　E 到达，因内存不够，不能调入内存运行。A 还需运行 10 分钟，B 还需 15 分钟

9:30 —　A 结束，内存空闲区为 30KB 与 50KB，可用磁带机 4 台。C 的响应比=1+30/35=1.86，D 的响应比=1+25/20=2.25，E 的响应比=1+20/10=3。内存紧凑后合并为一个 80KB 的空闲区，调入 E 运行，内存还有 20KB，磁带机还有 3 台。可再调入 D。此时 B、D、E 轮转运行。此时 B 还需再运行 5 分钟（即再过 15 分钟 B 将结束）。

9:45 —　B 结束，释放 120KB 内存和 1 台磁带机。因磁带机不够，C 仍不能调入内存运行。D 还需运行 15 分钟，E 还需 5 分钟

9:55 —　E 结束，释放 60KB 内存和 1 台磁带机。C 调入与 D 轮转运行。D 还需运行 10 分钟

10:15—　D 结束，C 还需运行 25 分钟。C 开始单独运行。

10:40—　C 结束

解：(1) 作业调度选中作业的次序为 A、B、E、D、C。

(2) 作业 A 在 9:30 结束，其周转时间为 60 分钟；

作业 B 在 9:45 结束，其周转时间为 55 分钟；

作业 C 在 10:40 结束，其周转时间为 100 分钟；

作业 D 在 10:15 结束，其周转时间为 70 分钟；

作业 E 在 9:55 结束，其周转时间为 45 分钟；

故平均周转时间为(60+55+100+70+45)/5=66（分钟）

28．若系统有同类资源 m 个，供 n 个进程共享，试问：当 m>n 和 m≤n 时，每个进程最多可以申请多少个这类资源而使系统一定不会发生死锁？

解：设每个进程申请该类资源的最大量为 x 个，则只要不等式 n(x-1)+1≤m 成立，系统一定不会发生死锁。因为最坏情况下，每个进程都已获得 x-1 各资源，则 n 个进程共获得 n(x-1)个资源，而不等式 n(x-1)+1≤m 表示每个进程都已获得 x-1 各资源后，系统仍有课分配的资源，这样，至少有一个进程可以得到全部资源，从而能执行完成，它完成后释放的资源又可使其它进程执行完成。

解不等式 n(x-1)+1≤m，可得　x≤1+(m-1)/n

于是可得

$$x=\begin{cases}1 & \text{当}m\leq n\\ 1+[(m-1)/n] & \text{当 } m>n\end{cases}$$

29．证明：若系统中所有作业同时到达，则采用短作业优先算法能得到最短的平均周转时间。

证：假设现有 n 个作业 J1、J2、J3、…、Jn，其运行时间依次为 t1、t2、t3、…、tn，满足 t1≤t2≤t3≤…≤tn，则短作业优先调度算法的平均周转时间为：

T1=(t1+(t1+t2)+(t1+t2+t3)+ … +(t1+t2+t3+ … +tn))/n

=(n×t1+(n-1)×t2+(n-2)×t3+…+tn)/n

设在任何其他调度算法中作业的调度顺序为 J_{i1}、J_{i2}、J_{i3}、… 、J_{in}，其运行时间依次为 ti1，ti2，ti3，…，tin，其中 i1、i2、i3、…、in 是 1、2、3、…、n 的一个排列，其平均周转时间为

T2=(ti1+(ti1+ti2)+(ti1+ti2+ti3)+ … +(ti1+ti2+ti3+ … +tin))/n

=(n×ti1+(n-1)×ti2+(n-2)×ti3+…+tin)/n

由于有 n>n-1>n-2>…>1，t1≤t2≤t3≤…≤tn

所以有 (n×t1+(n-1)×t2+(n-2)×t3+…+tn)/n≤(n×ti1+(n-1)×ti2+(n-2)×ti3+…+tin)/n

即有　T1≤T2。证毕

30．某个采用多道程序设计的计算机系统配有输入机和打印机各一台，现有程序 A 和程序 B 并行执行，且程序 A 先开始 50ms。假定程序 A 的执行过程为：计算 50ms，打印 100ms，再计算 50ms，打印 100ms，结束；程序 B 的执行过程为：计算 50ms，输入数据 60ms，再计算 50ms，打印 100ms，结束。当忽略调度和启动外设等所花费的时间时，回答下列问题：

(1) 把程序 A 和程序 B 并发执行时各自使用 CPU 与外设的时间用实线画在下图中

时间　　0　50　100　150　200　250　300　350　400

CPU　　ms

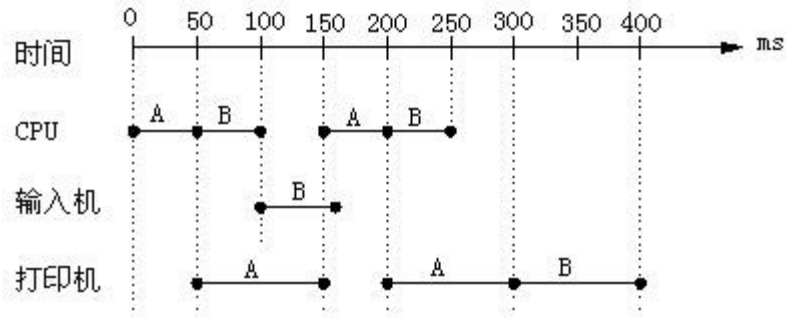
输入机

打印机

(2) 在程序开始执行直到两道程序都执行结束时，处理器的利用率是_____。

(3) 程序B从开始执行直到结束实际花费的时间是_____。

解：（1）



（2）处理器的利用率是(50+50+50+50)/400=50%

（3）程序B从开始执行直到结束实际花费的时间是400-50=350ms

31．若某计算问题的执行情况如图3-3：

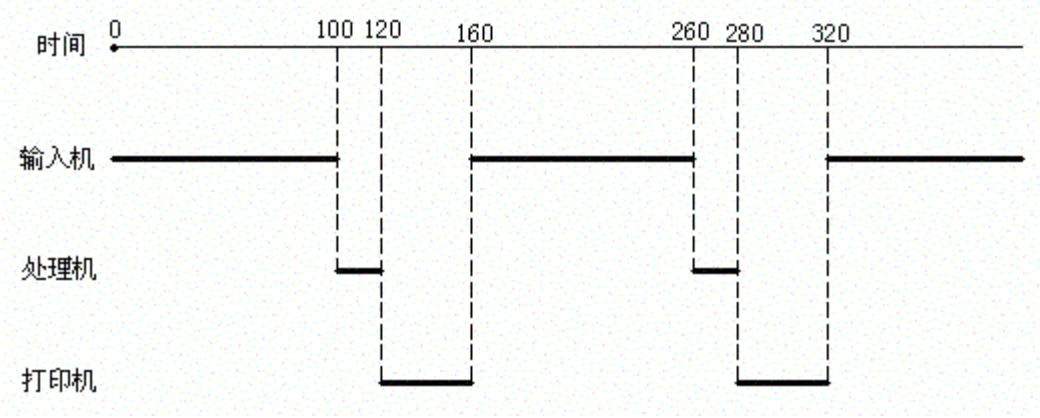


图3-3

请回答下列问题：

（1）叙述该计算问题中处理器、输入机和打印机是如何协同工作的。

（2）按图示的执行情况处理器的利用率为_____。

（3）处理器利用率不高的原因是_____。

（4）请画出能提高处理器利用率的执行方案。

解：(1) 上图中，处理器、输入机和打印机是串行(顺序)工作的。

(2) 按图示的执行情况处理器的利用率为20/160=12.5%。

(3) 处理器利用率不高的原因是处理器、输入机和打印机之间没有并行工作。

(4) 提高处理器利用率的执行方案如图3-4所示。

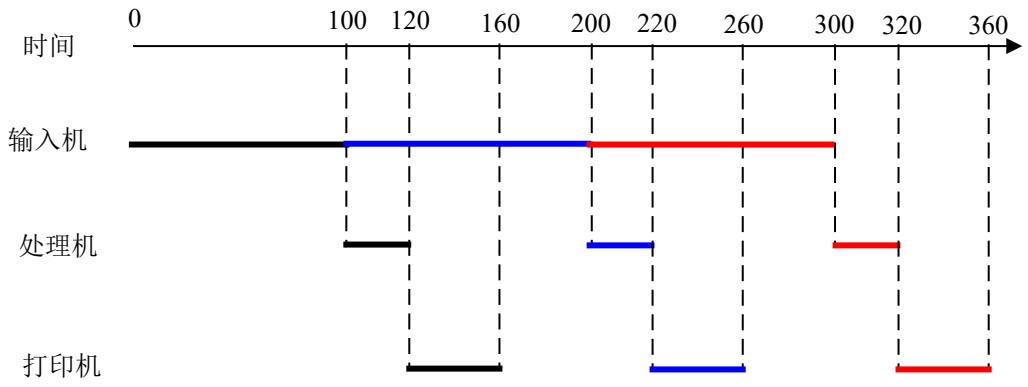


图3-4

32．设系统中仅有一类数量为 M 的独占资源，系统中 N 个进程竞争该类资源，其中各进程对该类资源

的最大需求量为W。当M、N、W分别取下列值时，试判断哪些情况可能会发生死锁？哪些情况不可能发生死锁？为什么？

①M=2, N=2, W=1　②M=3, N=2, W=2　③M=3, N=2, W=3

④M=5, N=3,W=2　⑤M=6, N=3, W=3

解：M、N、W满足关系式N(W-1)<M（或N(W-1)+1≤M）时，不会发生死锁。

用上述关系式判断，可知①、②、④三种情况不会发生死锁；而③、⑤两种情况可能会发生死锁。

33．设某计算机系统有1台输入机，1台打印机。现有2道程序同时投入运行，且程序A先开始运行，程序B后运行。程序A的运行轨迹为：计算50ms，打印100ms，再计算50ms，打印信息100ms，结束。程序B的运行轨迹为：计算50ms，输入数据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，在哪段时间等待？为什么？

（2）程序A、B运行时有无等待现象？若有，在什么时候发生等待现象？

答：（1）我们可以画出CPU、输入机和打印机的运行轨迹图如图3-5粗线所示。

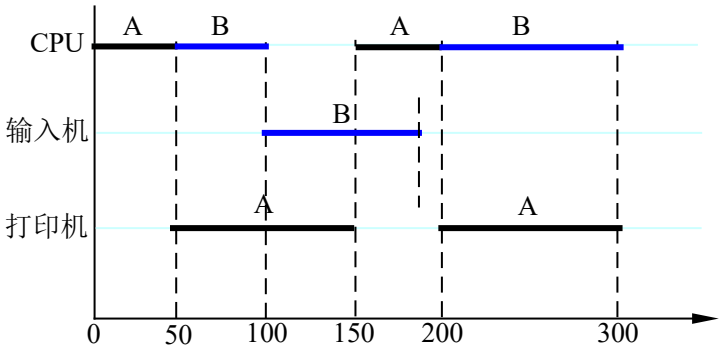


图3-5

从上图可以看出，CPU有等待时间，在100ms到150ms之间，共空闲50ms。在这段时间，A等待打印完成，B等待输入数据，CPU无事可做。

（2）程序A无等待现象；程序B有等待现象，在0～50ms之间以及180ms～200ms之间等待CPU。

34．一个单处理机多道分时系统收到了3个作业，作业提交情况如下表所示。

Job	作业提交时间	运行时间	其中	
			I/O时间	CPU时间
A	10.0	0.36小时	0.18小时	0.18小时
B	10.2	0.32小时	0.16小时	0.16小时
C	10.4	0.36小时	0.18小时	0.18小时

现假设：

（1）在单CPU上分时运行两道作业，若每道作业的I/O等待时间皆占各自总运行时间的50%，则CPU将有20%的时间空闲。

（2）系统有相当充足的可用资源（CPU除外）供用户使用。

请写出各个作业的结束时间。

解：由题目条件，可列出下表所示的运行情况。

时间	事件	道数	CPU等待	各占CPU	经过时间	作业	进度	还需
10.0-10.2	A提交并运行	1	50%	50%	0.2	A	0.1	0.08
10.2-10.4	B提交并运行	2	20%	40%	0.2	A B	0.08 0.08	0(结束) 0.08
10.4-10.6	C提交并运行	2	20%	40%	0.2	B	0.08	0(结束)

						C	0.08	0.1
10.6-10.8		1	50%	50%	0.2	C	0.1	0(结束)

因此，三个作业的完成时间为：A为10.4时，B为10.6时，C为10.8时。

35．假设有一台计算机，它有 1MB 内存，操作系统占用 200KB，每个用户也各占用 200KB 内存。用户进程等待 I/O 的时间占 80%，若增加 1MB 内存，则 CPU 的利用率将提高多少？

【相关知识】多道程序设计技术使 CPU 的利用率大为提高。设内存中有 N 个进程，每个进程等待 I/O 的百分比是 P，那么 N 个进程同时等待 I/O 的概率是 P^n(此时 CPU 空闲)，即 CPU 的利用率为 $1-P^n$，由此可见，CPU 的利用率是 N 的函数，称为多道程序度。

解：由题目所给条件可知，当前 1MB 内存可支持 4 个用户进程，由于每个用户进程等待 I/O 的时间为 80%，故 CPU 的利用率为

$1-(80\%)^4=1-(0.8)^4=0.5904=59.04\%$　　(59%)

若增加 1MB 内存，则系统中可同时运行 9 个进程，则 CPU 的利用率为

$1-(0.8)^9=86.58\%$　　(87%)

$87\%\div 59\%=146.65\%$　　(147%)

$146.65\%-100\%=46.65\%$　　(47%)

所以增加1MB内存使CPU的利用率提高了46.65%。(47%)

【说明】若计算中百分数都取整数，则结果如右边括号中所示，也是可以的。

36．设有两个处理机 P1 和 P2，它们各有一个硬件高速缓冲存储器 C1、C2，且各有一个主存储器 M1、M2，其性能如下所示：

	C1	C2	M1	M2
存储容量	4KB	4KB	2MB	2MB
存取时间	60ns	80ns	1μs	0.9μs

假定两个处理机的指令系统相同，它们的执行时间与存储器的平均存取时间成正比。如果执行某个程序时，所需的指令或数据在缓冲存储器中存取到的概率 P 是 0.7，试问这两个处理机的处理速度哪个快？当 P=0.9 时，处理机的处理速度哪个快？

解：由题目分析可知，处理机的平均处理时间T为：

$T = T1*P + (1-P)\,T2$

其中，T1 为高速缓冲存储器的存取时间，T2 为主存储器的存取时间，P 为指令或数据在高速缓冲存储器中存取的概率。（1μs = 1000ns）

(1) 当P=0.7时，P1的平均存取时间为：

$60*0.7+(1-0.7)\times 1000=342\ (ns)$

P2的平均存取时间为：

$80*0.7+(1-0.7)\times 0.9\times 1000=326\ (ns)$

根据题意，指令的执行时间与存储器的平均存取时间成正比，因此当P=0.7时，P2比P1的处理速度快。

(2) 当P=0.9时，P1的平均存取时间为：

$60+0.9+(1-0.9)\times 1000=154\ (ns)$

P2的平均存取时间为：

$80*0.9+(1-0.9)\times 0.9\times 1000=162\ (ns)$

因此当P=0.9时，P1比P2的处理速度快。

37．(中科院软件所 1999 年试题)某系统有 R1、R2 和 R3 共 3 种资源，在 T0 时刻 P1、P2、P3 和 P4 这 4 个进程对资源的占用和需求情况见表 1，此时系统的可用资源向量为(2, 1, 2)，试问：

(1) 将系统中各种资源总数和此刻各进程对各资源的需求数目用向量或矩阵表示出来；

(2) 如果此时P1和P2均发出资源请求向量(1, 0, 1)，为了保证系统的安全性，应该如何分配资源给这两个进程？说明你所采用策略的原因。

(3) 如果(2)中两个请求立即得到满足后，系统此时是否处于死锁状态？

表1 T0时刻4个进程对资源的占用和需求情况

	最大资源需求量Max			已分配资源量Allocation		
	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0
P2	6	1	3	4	1	1
P3	3	1	4	2	1	1
P4	4	2	2	0	0	2

解：(1) 系统中资源总数是可用资源数与各进程已分配资源数之和，即

(2, 1, 2) + (1, 0, 0) + (4, 1, 1) + (2, 1, 1) + (0, 0, 2) = (9, 3, 6)

各进程对各资源的需求量为Max与Allocation之差，即

$$\begin{bmatrix}3&2&2\\6&1&3\\3&1&4\\4&2&2\end{bmatrix}-\begin{bmatrix}1&0&0\\4&1&1\\2&1&1\\0&0&2\end{bmatrix}=\begin{bmatrix}2&2&2\\2&0&2\\1&0&3\\4&2&0\end{bmatrix}$$

(2) 若此时P1发出资源请求$Request_1$ (1, 0, 1)，按银行家算法进行检查：

$Request_1$ (1, 0, 1)≤$Need_1$ (2, 2, 2)

$Request_1$ (1, 0, 1)≤Available (2, 1, 2)

试分配并修改相应的数据结构，资源分配情况如下：

进程	Allocation			Need			Available		
P1	2	0	1	1	2	1	1	1	1
P2	4	1	1	2	0	2			
P3	2	1	1	1	0	3			
P4	0	0	2	4	2	0			

利用安全性检查算法检查，可知可用资源向量(1, 1, 1)已不能满足任何进程的需求，故若分配给P1，系统将进入不安全状态，因此此时不能将资源分配给P1。

若此时P2发出资源请求$Request_2$ (1, 0, 1)，按银行家算法进行检查：

$Request_2$ (1, 0, 1)≤$Need_2$ (2, 0, 2)

$Request_2$ (1, 0, 1)≤Available (2, 1, 2)

试分配并修改相应的数据结构，资源分配情况如下：

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

利用安全性检查算法，可得此时刻的安全性分析情况如下：

进程	Work	Need	Allocation	Work+Allocation	Finish
P2	1 1 1	1 0 1	5 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

从上面分析可知，存在一个安全序列{P2, P3, P4, P1}，故该状态时安全的，可以立即将 P2 所申请的资源分配给它。

(3) 如果(2)中两个请求立即得到满足后，系统此时并没有立即进入死锁状态，因为此时所有进程没有提出新的资源请求，全部进程都没有因资源请求没有得到满足而进入阻塞状态。只有当进程提出新的资源请求且全部进程(指P1-P4)都进入阻塞状态时，系统才处于死锁状态。

38．假设苗圃系统中有以下几个进程，每个进程的执行时间(单位：ms)和优先数如下(优先数越小，其优先级越高)：

进程	执行时间	优先数
P1	10	3
P2	1	1
P3	2	5
P4	1	4
P5	5	2

如果在 0 时刻，各进程按 P1、P2、P3、P4、P5 的顺序同时到达，忽略进程调度切换等辅助时间，试回答下列问题：当系统分别采用

(1)先来先服务调度算法；

(2)抢占式优先级调度算法；

(3)时间片轮转算法(时间片为1ms)。

在使用以上各种算法的情况下，分别求各进程的开始运行时间、完成时间以及平均周转时间。

解：(1)采用先来先服务调度算法，各进程开始运行的时间、完成时间以及周转时间如下表：

进程	开始运行时间	完成时间	周转时间
P1	0	10	10
P2	10	11	11
P3	11	13	13
P4	13	14	14
P5	14	19	19

平均周转时间为(10+11+13+14+19)/5=67/5=13.5(ms)

(2) 采用抢占式优先级调度算法，各进程开始运行的时间、完成时间以及周转时间如下表：

进程	开始运行时间	完成时间	周转时间
P1	6	16	16
P2	0	1	1
P3	17	19	19
P4	16	17	17
P5	1	6	6

平均周转时间为(16+1+19+17+6)/5=59/5=11.8(ms)

(3) 采用时间片轮转算法，各进程开始运行的时间、完成时间以及周转时间如下表：

进程	开始运行时间	完成时间	周转时间
P1	0	19	19
P2	1	2	2
P3	2	7	7
P4	3	4	4

P5	4	14	14

平均周转时间为(19+2+7+4+14)/5=46/5=9.2(ms)

39．有3个程序A、B、C，它们分别单独运行时的CPU和I/O占用时间如图3-6所示：

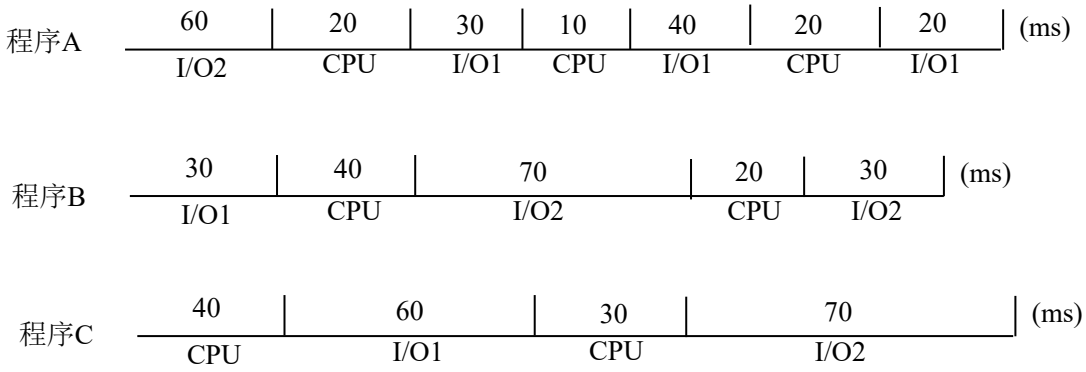


图3-6

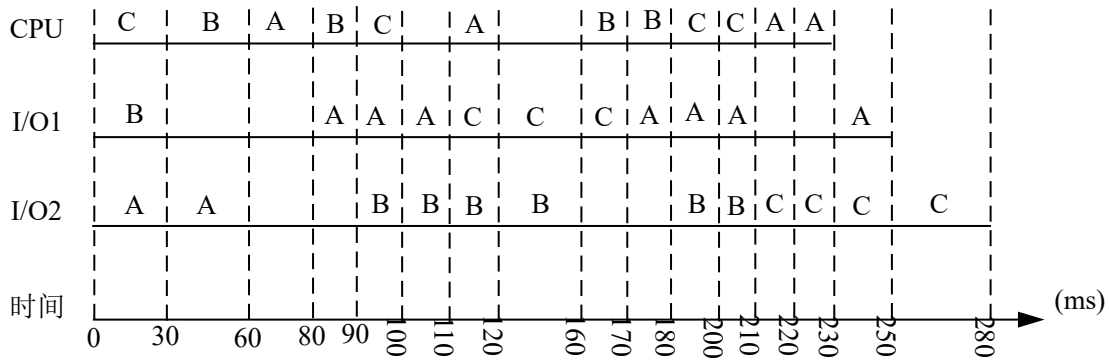
现在考虑3个程序同时运行。系统中的资源有一个CPU和两台输入输出设备(I/O1和I/O2)同时运行。三个程序的优先级为A最高，B次之，C最低，优先级高的程序可以中断优先级低的程序，但优先级与输入输出设备无关。请回答下列问题：

(1) 最早结束的程序是哪个？

(2) 最后结束的程序是哪个？

(3) 三个程序执行到结束分别用了多长时间？

(4) 计算这段时间的CPU利用率(三个程序完全结束为止)。（北工大2000）



解：先用图示分析三个程序的执行过程（如图3-7所示）：图3-7从上述分析可知：

(1) 最早结束的程序是B；

(2) 最后结束的程序是C；

(3) A、B、C三个程序执行到结束分别用了250ms、210ms、280ms；

(4) 这段时间的CPU利用率=(230-10-40)/280=180/280=0.643=64.3%。

40．若在后备作业队列中有同时到达等待运行的作业A、B、C，已知它们各自的运行时间为a、b、c，且满足关系a<b<c，试证明采用最短作业优先调度算法能获得最小平均周转时间。

解：对于最短作业优先调度算法而言，三个作业的总周转时间为

T1=a+(a+b)+(a+b+c)=3a+2b+c　　①

若不按短作业优先调度算法来调度者三个作业，不失一般性，假定调度顺序为B、A、C，则总周转时间为

T2=b+(b+a)+(b+a+c)=3b+2a+c　　②

②-①式得：

T2-T1=b-a>0

由此可见，短作业优先调度算法能获得最小平均周转时间。

41. (北大1995年试题)有一个具有两道作业的批处理系统（最多可有两道作业同时装入内存执行），作业调度采用计算时间短的作业优先调度算法，进程调度采用以优先数为基础的抢占式调度算法，今有如下作业序列，作业优先数即为进程优先数，优先数越小优先级越高：

作业名	到达时间	估计运行时间	优先数
A	10 : 00	40分钟	5
B	10 : 20	30分钟	3
C	10 : 30	50分钟	4
D	10 : 50	20分钟	6

列出所有作业进入内存时间及结束时间。

（2）计算平均周转时间。

【分析】

10:00　A作业到达，调入内存运行

10:20　B作业到达，调入内存，因其优先级高于A，故抢占CPU运行。A变为就绪，已运行20分钟，A今后还需运行20分钟。

10:30　C作业到达，因内存中已有两道作业，故C不能调入。作业B继续运行。

10:50　B运行结束。D到达。后备队列中有C、D两道作业，按短作业优先算法，调入D。但因D的优先级低于A，故A运行。

11:10　A结束。C调入。因C进程优先级高于D，故C运行。

12:00　C结束。D开始运行。

12:20　D结束。

解：（1）由上分析，可得这四道作业进入内存时间和结束时间如下表：

作业名(进程名)	进入内存时间	结束时间
A	10:00	11:10
B	10:20	10:50
C	11:10	12:00
D	10:50	12:20

（2）各作业的周转时间为：

作业A：11:10-10:00=70分钟

作业B：10:50-10:20=30分钟

作业C：12:00-10:30=90分钟

作业D：12:20-10:50=90分钟

平均周转时间为 (70+30+90+90) ÷4 = 70 (分钟)

42. 在单CPU盒两台输入/输出设备(I1、I2)的多道程序设计环境下，同时投入三个作业Job1、Job2、Job3运行。这三个作业队CPU和输入/输出设备的使用顺序和时间如下所示：

Job1：I2(30ms)；CPU(10ms)；I1(30ms)；CPU(10ms)；I2(20ms)

Job2：I1(20ms)；CPU(20ms)；I2(40ms)

Job3：CPU(30ms)；I1((20ms)；CPU(10ms)；I1(10ms)

假定 CPU、I1、I2 都能并行工作，Job1 优先级最高，Job2 次之，Job3 优先级最低，优先级高的作业进程可以抢占优先级低的作业进程的 CPU，但不抢占 I1 和 I2。试求：

(1)　三个作业投入到完成分别需要的时间。

(2)　从投入到完成的CPU利用率。

(3)　I/O设备I1和I2的利用率各是多少。

解：三个作业并发执行时的工作情况如图3-8所示。

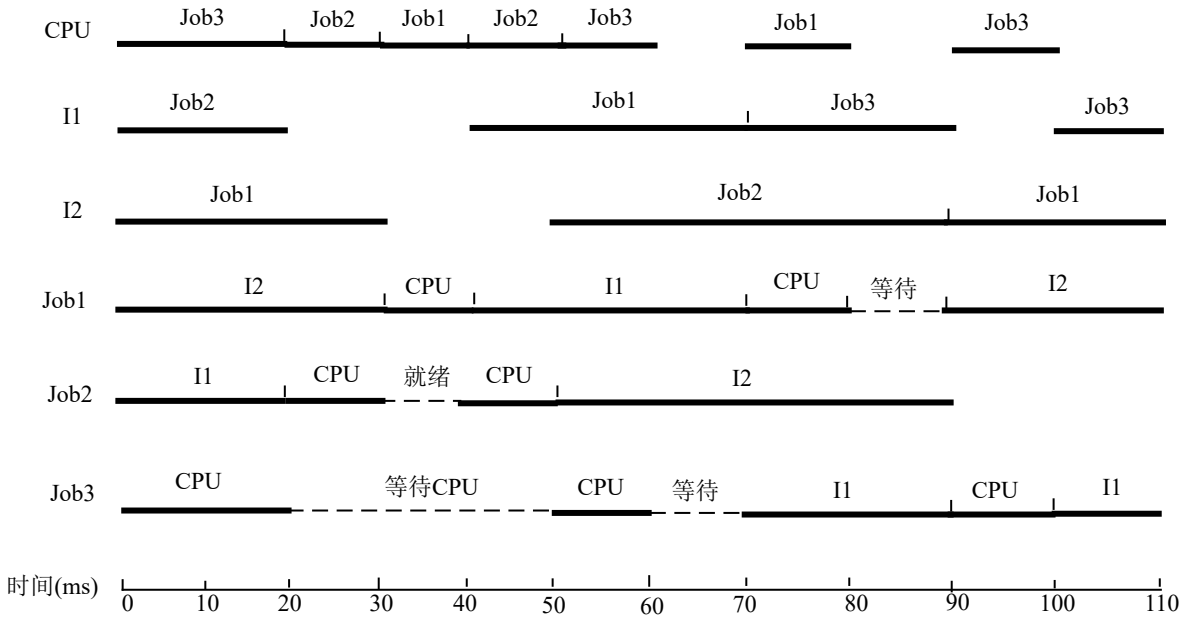


图 3-8

（1）由图 3-8 可知，Job1 从投入到运行结束需要 110ms，Job2 从投入到运行结束需要 90ms，Job3 从投入到运行结束需要 110ms

（2）CPU在时间段60ms到70ms，80ms到90ms，100ms到110ms期间空闲，所以CPU的利用率为：(110-30)/110=72.7%

（3）设备I1在时间段20ms到40ms，90ms到100ms期间空闲，所以设备I1的利用率为：(110-30) /110=72.7%

设备I2在时间段30ms到50ms期间空闲，所以设备I1的利用率为：(110-20) /110=81.8%

43. 假设一个计算机系统具有如下性能特征：

- 处理一次中断，平均耗时1ms；
- 一次进城调度，平均需要2ms；
- 将CPU分配给选中的进程，又需要平均1ms。

再假设其定时芯片每秒产生100次中断。请回答：

(1)　操作系统将百分之几的时间用于时钟中断处理？

(2)　如果操作系统采用时间片轮转法调度进程，10个时钟中断为1个时间片。那么，操作系统将百分之几的CPU时间用于进程调度(包括调度、分配CPU和引起调度的时钟中断处理时间)？

(3) 该系统CPU的最大利用率是多少？

解：(1) 因为 1 秒钟中断 100 次，所以中断周期为 10ms。每个周期要花费 1ms 进行中断处理，所以系统将 CPU 的 10%时间用于时钟中断处理。

(2)　由题目所给条件可知，时间片长度为10个时钟周期，即100ms。

(1ms+2ms+1ms)÷100ms=4%

即操作系统将4%的CPU时间用于进程调度

(3) 从(2)可知，在100ms时间片内，系统用于调度的时间为4ms，用于另外9次时钟中断9ms，系统开销总共为13ms。因此，若没有因用户进程等待I/O等原因而出现CPU空闲的情况，CPU的最大利用率为：(100-13)÷100=87%。

44. (西安交大2002年试题)有5个任务A，B，C，D，E，它们几乎同时到达，预计它们的运行时间为10，6，2，4，8min。其优先级分别为3，5，2，1和4，这里5为最高优先级。对于下列每一种调度算法，计算器平均进程周转时间（进程切换开销不考虑）。

(1) 先来先服务(按A,B,C,D,E顺序)算法；

(2) 优先级调度算法；

(3) 时间片轮转算法(设时间片为1min——原题无此假设)。

解：(1) 采用先来先服务算法时，5个任务在系统中的执行顺序、完成时间及周转时间如下表所示。

执行顺序	预计运行时间	开始执行时间	完成时间	周转时间
A	10	0	10	10
B	6	10	16	16
C	2	16	18	18
D	4	18	22	22
E	8	22	30	30

5个进程的平均周转时间为：(10+16+18+22+30)/5=19.2 (min)

(2) 采用最高优先级调度算法时，5个任务在系统中的执行顺序、完成时间及周转时间如下表所示。

执行顺序	预计运行时间	优先数	开始执行时间	完成时间	周转时间
B	6	5	0	6	6
E	8	4	6	14	14
A	10	3	14	24	24
C	2	2	24	26	26
D	4	1	26	30	30

5个进程的平均周转时间为：(6+14+24+26+30)/5=20 (min)

(2) 采用时间片轮转调度算法时，可以认为5个进程均分CPU时间，它们在系统中的执行情况如下：

第一轮：A，B，C，D，E （5min）

第二轮：A，B，C(C完成，即C的周转时间为8min)，D，E （5min）

第三轮：A，B，D，E （4min）

第四轮：A，B，D(D完成，即D的周转时间为17min)，E （4min）

第五轮：A，B，E （3min）

第六轮：A，B(B完成，即B的周转时间为23min)，E （3min）

第七轮：A，E （2min）

第八轮：A，E(E完成，即E的周转时间为28min) （2min）

第九轮：A （1min）

第十轮：A(A完成，即A的周转时间为30min) （1min）

所以，5 个进程的平均周转时间为：(8+17+23+28+30)/5=21.2 (min)

45. (上题的变形) 有5个任务A，B，C，D，E，它们几乎同时到达，预计它们的运行时间为10，6，2，4，8min。其初始优先数P_0分别为3，5，2，1和4，这里5为最高优先级。对于下列每一种调度算法，计算器平均进程周转时间（进程切换开销不考虑）。

(1) 抢占式优先级调度算法。运行状态进程的优先数P按$P_n=P_{n-1}+\alpha$变化，就绪状态进程的优先数P按Pn=Pn-1+β变化。其中n=0,1,2, . . . ，α= −0.5，β= 0.5，系统每30秒钟计算一次各进程的优先数。

进程从运行态转变为就绪态时，优先数=就绪队列中进程的最小优先数-1。

(2) 时间片轮转算法(设时间片为100ms)。

解：(1) 采用抢占式优先级调度算法时，由于 B 的初始优先级最高，故 B 最先开始运行。因系统每 10 秒钟计算一次各进程的优先数，

B运行60s(即1min)后，优先数变为P=5-0.5*2=4，A、C、D、E的优先数分别为4、3、2、5，E抢占CPU运行。B的优先数变为1。B还需5min。

E运行60s后，优先数变为P=5.0-0.5*2=4，A、B、C、D的优先数变为5、2、4、3，A抢占CPU运行。E的优先数变为1。E还需7min。

A运行60s后，其优先数变为4，B、C、D、E的优先数变为3、5、4、2。C抢占CPU运行。A的优先数变为1。A还需9min。

C运行60s后，其优先数变为4，A、B、D、E的优先数变为2、4、5、3。D抢占CPU运行。C的优先数变为1。C还需1min。

D运行60s后，其优先数变为4，A、B、C、E的优先数变为3、5、2、4。B抢占CPU运行。D的优先数变为1。D还需3min。

B运行60s后，其优先数变为4.0，A、C、D、E的优先数变为4、3、2、5。E抢占CPU运行。B的优先数变为1。B还需4min。

E运行60s后，其优先数变为4.0，A、B、C、D的优先数变为5、2、4、3。A抢占CPU运行。E的优先数变为1。E还需6min。

A运行60s后，其优先数变为4.0，B、C、D、E的优先数变为3、5、4、2。C抢占CPU运行。A的优先数变为1。A还需8min。

C运行60s后，其优先数变为4，A、B、D、E的优先数变为2、4、5、3。D抢占CPU运行。C的优先数变为1。C运行结束，周转时间为9min。

D运行60s后，其优先数变为4，A、B、E的优先数变为3、5、4。B抢占CPU运行。D的优先数变为2。D还需2min。

B运行60s后，其优先数变为4.0，A、D、E的优先数变为4、3、5。E抢占CPU运行。B的优先数变为2。B还需3min。

E运行60s后，其优先数变为4.0，A、B、D的优先数变为5、3、4。A抢占CPU运行。E的优先数变为2。E还需5min。

A运行60s后，其优先数变为4，B、D、E的优先数变为4、5、3。D抢占CPU运行。A的优先数变为2。A还需7min。

D运行60s后，其优先数变为4，A、B、E的优先数变为3、5、4。B抢占CPU运行。D的优先数变为2。D还需1min。

B运行60s后，其优先数变为4，A、D、E的优先数变为4、3、5。E抢占CPU运行。B的优先数变为2。B还需2min。

E运行60s后，其优先数变为4，A、B、D的优先数变为5、3、4。A抢占CPU运行。E的优先数变为2。E还需4min。

A运行60s后，其优先数变为4，B、D、E的优先数变为4、5、3。D抢占CPU运行。A的优先数变为2。A还需6min。

D运行60s后，其优先数变为4，A、B、E的优先数变为3、5、4。B抢占CPU运行。D的优先数变为2。D运行结束，周转时间为18min。

B运行60s后，其优先数变为4，A、E的优先数变为4、5。E抢占CPU运行。B的优先数变为3。B还需1min。

E运行60s后，其优先数变为4，A、B的优先数变为5、4。A抢占CPU运行。E的优先数变为3。E还需3min。

A运行60s后，其优先数变为4，B、E的优先数变为5、4。B抢占CPU运行。A的优先数变为3。A还需

5min。

B运行60s后，其优先数变为4，A、E的优先数变为4、5。E抢占CPU运行。B的优先数变为3。B运行结束，周转时间为22min。

E运行60s后，其优先数变为4，A的优先数变为5，A抢占CPU运行。E的优先数变为4。E还需2min。

A运行60s后，其优先数变为4，E的优先数变为5，E抢占CPU运行。A的优先数变为4。A还需4min。

E运行60s后，其优先数变为4，A的优先数变为5，A抢占CPU运行。E的优先数变为4。E还需1min。

A运行60s后，其优先数变为4，E的优先数变为5，E抢占CPU运行。A的优先数变为4。A还需3min。

E运行60s后，其优先数变为4，A的优先数变为5，A抢占CPU运行。E运行结束，周转时间为27min。

A 单独运行 3min 后结束，周转时间为 30min。

所以，5 个进程的平均周转时间为：(9+18+22+27+30)/5=21.2 (min)

【注】从分析解题过程可知，本题的抢占式优先级调度算法实际上是时间片为1min的轮转调度算法。

(2) 时间片轮转算法(设时间片为100ms)。由于时间片为0.1s，故可以看做各进程均分CPU时间。

5个进程轮转运行10min后，C结束，故C的周转时间为10min。

A、B、D、E4个进程再运行8min后，D结束，故D的周转时间为18min。

A、B、E3个进程再运行6min后，B结束，故B的周转时间为24min。

A、E2个进程再运行4min后，E结束，故E的周转时间为28min。

A 再单独运行 2min 后，A 结束，故 A 的周转时间为 30min。

所以，5 个进程的平均周转时间为：(10+18+24+28+30)/5=22 (min)

46. 考虑下面的进程集合，给出 FCFS、RR q=1、SPN 和 SRT 调度策略的完成时间、周转时间和 Tr/Ts(带权周转时间)比较表。

进程名	到达时间	处理时间
A	0	3
B	1	5
C	3	2
D	9	5
E	12	5

【说明】

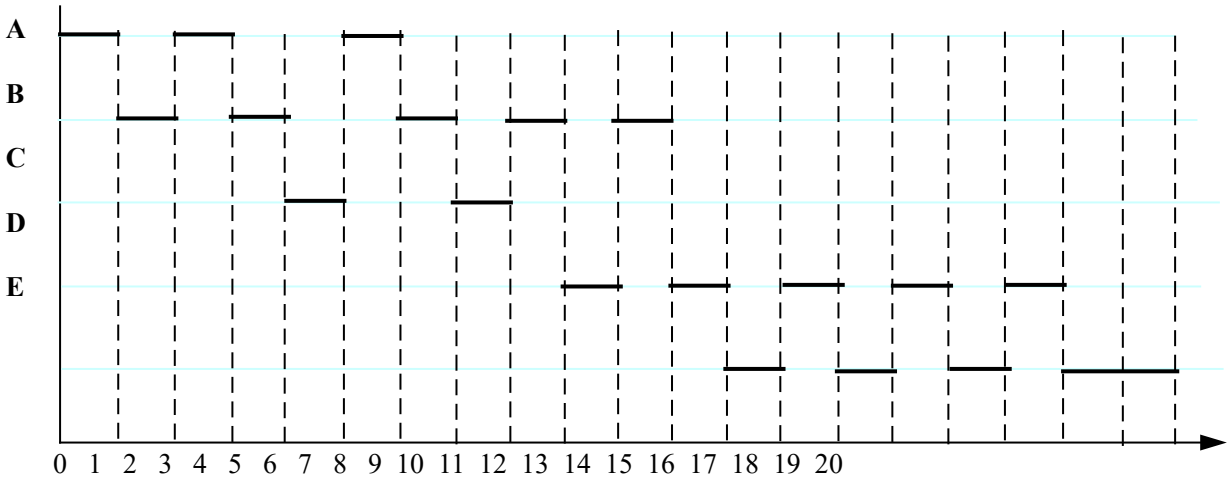
- RR q=1：时间片为 1 的轮转算法。RR 是 Round Robin 的缩写。
- SPN(Shortest Process Next)：短进程优先，非抢占式。它就是教科书上的最短作业优先(Shortest Job First, SJF)算法。
- SRT(Shortest Remaining Time)：最短剩余时间优先，是 SPN 增加了剥夺机制的版本。教科书上没介绍此算法。

答：调度策略的比较表：

	进程	A	B	C	D	E	平均
	到达时间	0	1	3	9	12	
	服务时间	3	5	2	5	5	
FCFS	完成时间	3	8	10	15	20	
	周转时间	3	7	7	6	8	6.2
	Tr/Ts	1	1.4	3.5	1.2	1.6	1.74
RR q=1 时间片=1	完成时间	6	11	8	18	20	
	周转时间	6	10	5	9	8	7.6
	Tr/Ts	2	2	2.5	1.8	1.6	1.98

SPN (SPF)	完成时间	3	10	5	15	20	
	周转时间	3	9	2	6	8	5.6
	Tr/Ts	1	1.8	1	1.2	1.6	1.32
SRT 短剩余时间	完成时间	3	10	5	15	20	
	周转时间	3	9	2	6	8	5.6
	Tr/Ts	1	1.8	1	1.2	1.6	1.32

关于 RR q=1 的分析图示如下图所示。



由图示分析可知：

A 完成时间：6

B 完成时间：11

C 完成时间：8

D 完成时间：18

E 完成时间：20

47．在单处理机系统中，有多个进程运行：一些以计算为主，一些以输入/输出为主。如何赋予进程占有处理器的优先级才能提高系统的效率，使系统的平均周转时间减少？

答：若计算型进程的优先级高于 I/O 型进程的优先级，计算型进程一旦占有了 CPU 便忙于计算，使 I/O 型进程得不到运行的机会，导致 I/O 设备空闲，达不到 CPU 与 I/O 操作并行的目的；多个 I/O 型进程在系统中停留时间增加，系统的平均周转时间增加。

若 I/O 型进程的优先级高于计算型进程的优先级，当它完成一项 I/O 操作后，便能立即获得 CPU，为下次 I/O 作准备工作，并启动外设。当设备被启动后，它便主动交出 CPU，由系统将 CPU 分配给计算型进程，从而使 CPU 与 I/O 设备并行操作，获得较好的运行效率。

因此，应赋予以 I/O 为主的进程更高的优先级。

第四章 存储器管理

1．选择题

1*．固定分区存储管理中，处理器设置的地址转换机构是____ 。
A．界限寄存器　　B．上限寄存器
C．下限寄存器　　D．上限寄存器和下限寄存器

2．存储保护的工作通常由_____实现。
A．软件　　B．硬件　　C．文件　　D．硬件和软件配合

3．段页式存储管理中，访问快表失败时，每访问一条指令或存取一个操作数都要_____次访问主存。
A．1　　B．2　　C．3　　D．4

4．在虚拟存储系统中，若进程在内存中占3块（开始时为空）采用先进先出页面淘汰算法，当执行访问页号序列为1、2、3、4、1、2、5、1、2、3、4、5、6时，将产生_____次缺页中断。
A．7　　B．8　　C．9　　D．10

5．采用段页式存储管理，在CPU中应设置_____寄存器。
A．段表和页表控制　　B．段表控制　　C．页表控制　　D．界地址

6．采用段页式存储管理时，内存地址分成段号、段内页号和页内地址三部分，____地址。
A．但仍是线性　　B．但仍是二维
C．故是三维　　D．从而成为四维

7．用户程序的逻辑地址可以不连续的存储管理方式是____。
A．固定分区　　B．可变分区　　C．页式　　D．段页

8．在可变分区分配方案中，为了实现主存的空间分配，采用____进行管理。
A．页表　　B．段表
C．段表＋页表　　D．分区分配表＋空闲区表

9．动态重定位是在____完成的。
A．作业执行前集中一次　　B．作业执行过程中集中一次
C．作业执行过程中　　D．作业执行过程中由用户

10．在以下的存储管理方案中，能扩充主存容量的是____ 。
A．固定式分区分配　　B．可变式分区分配
C．页式存储管理　　D．分页虚拟存储管理

11．在可变分区分配方案中，在空闲区表中以空闲区长度按递减顺序排列适合于_____算法。
A．最坏适应算法　　B．最先适应算法
C．最优适应算法　　D．首次循环适应算法

12．在存储管理中，提高内存利用率主要是通过____功能实现的。
A．存储分配　　B．存储保护　　C．存储扩充　　D．存储共享

13．在页式虚拟存储管理中，为实现地址变换，应建立____。
A．空闲区表　　B．分区分配表　　C．页表　　D．段表

14．在下述存储管理方案中，____管理方式要求作业的逻辑地址与占有主存的存储区域都是连续的。
A．段页式　　B．页式　　C．段式　　D．可变分区

15．将主存空闲区按地址顺序从小到大登记在空闲区表中，每次分配时总是顺序查找空闲区表，此种分

配算法称为____分配算法。

A．最先适应　　B．最优适应　　C．最坏适应　　D．随机适应

16．页式存储管理中，每次从主存中取指令或取操作数，当读快表失败时，要读____次主存。

A．1　　B．2　　C．3　　D．4

17．采用动态重定位方式装入的作业，在执行中允许____将其移动。

A．用户有条件地　　B．用户无条件地

C．操作系统有条件地　　D．操作系统无条件地

18．段式和页式存储管理的地址结构很类似，但是它们之间有实质上的不同。以下说法中，错误的是___。

A．页式的逻辑地址是连续的，段式的逻辑地址可以不连续

B．页式的地址是一维的，段式的地址是二维的

C．分页是操作系统进行的，分段是用户确定的

D．页式采用动态重定位方式，段式采用静态重定位方式

19．主存的地址空间常称为_____。

A．逻辑地址空间　　B．程序地址空间

C．物理地址空间　　D．相对地址空间

20．段页式存储管理中，每次从主存中取指令或取操作数，当读快表失败时，至少要_____次访问主存。

A．0　　B．1　　C．2　　D．3

21．支持程序浮动的地址转换机制是____。

A．页式地址转换　　B．段式地址转换

C．静态重定位　　D．动态重定位

22．在可变分区存储管理中，最优适应分配算法要求对空闲区表项按____进行排列。

A．地址从大到小　　B．地址从小到大

C．尺寸从大到小　　D．尺寸从小到大

23．在请求页式存储管理中，当查找的页不在_____中时，要产生缺页中断。

A．外存　　B．虚存　　C．内存　　D．地址空间

24．在段页式系统中（无快表），为获得一条指令或数据，必须_____访问内存。

A．1次　　B．2次　　C．3次　　D．4次

25．在一虚拟存储系统中，设主存的容量为32MB，辅存（硬盘）的容量为2GB，而地址寄存器的位数是32位，在这样的系统中，虚存的最大容量是_____。

A．1GB　　B．16MB　　C．1GB+16MB　　D．4GB

26．在段式存储管理的地址转换时，若段内地址大于段表中该段的长度，则发生 ____。

A．缺页中断　　B．溢出中断

C．硬件故障中断　　D．地址越界中断

27．在下列存储管理方式中，不要求将作业全部装入并不要求一个连续存储空间的管理方式是_____。

A．固定分区存储管理　　B．可变分区存储管理

C．页式存储管理　　D．请求页式存储管理

28．采用页式存储管理使处理器执行指令的速度_____。

A．提高　　B．降低　　C．有时提高有时降低　　D．不受影响

29．在段式存储管理中，_____。

A．以段为单位分配，每一段是一个连续存储区

B．段与段之间必定不连续

C．段与段之间必定连续

D．每段是等长的

30．页式虚拟存储管理中，当访问的页不在_____时，产生缺页中断。

A．内存　　B．外存　　C．虚存　　D．缓存

31．在虚拟存储的实现中，需要页面淘汰的原因是_____。

A．产生缺页中断时内存中没有空闲块　　B．内存空间太大

C．页面换出、换入太频繁　　D．进程要被封锁

32．以下说法中，____是错误的。

A．可变分区存储管理采用静态重定位　　B．分页存储管理采用动态重定位

C．动态重定位支持程序浮动　　D．段式存储管理静态重定位

33．以下_____不是影响缺页中断率的因素。

A．页面调度算法　　B．分配给作业的主存块数

C．程序的编制方法　　D．存储管理方式

34．分页式存储管理中，地址转换工作是由_____完成的。

A．硬件　　B．操作系统　　C．用户程序　　D．装入程序

35．把目标程序中的逻辑地址转换成主存空间的物理地址称为_____。

A．存储分配　　B．地址重定位　　C．地址保护　　D．程序移动

36．在操作系统的存储管理中，页式分配（分页）是_____。

A．把程序的逻辑空间和内存的物理空间按同样的尺寸分成若干页

B．把作业按其所需空间分成若干页

C．将内存的空闲空间分成若干页

D．随机地将每个作业的地址空间分成大小相同的若干页

37．在系统运行时，对于固定分区的存储管理方式，内存中能并发执行的作业的最大数量是

_____。

A．用户确定的　　B．可变的　　C．不受限制的　　D．固定的

38．在以下的存储管理方案中，允许动态扩充主存容量的是_____方式。

A．固定分区分配　　B．可变分区分配

C．页式存储管理　　D．请求分页存储管理

39．在分页虚拟存储管理中，对缺页中断率没有影响的因素是_____。

A．作业在输入井的等待时间　　B．页面调度算法

C．作业得到的主存块数　　D．程序的编制质量

40．某系统采用页式存储管理，页的大小为512B，设内存容量为16MB，内存的分配使用情况采用“位示图”表示，则位示图需要_____字节。

A．4K　　B．8K　　C．16K　　D．32K

41．内存分配的最佳适应算法的空闲区表是____。

A．按大小递减顺序排列　　B．按大小递增顺序排列

C．按地址由小到大排列　　D．按地址由大到小排列

42．虚拟存储器的最大容量____ 。

A．为内外存容量之和　　B．由计算机的地址结构决定

C．是任意的　　D．由作业的地址空间决定

43．很好地解决了“零头”（碎片）问题的存储管理方法是____ 。

A．页式存储管理　　B．段式存储管理

C．可变分区存储管理　　D．可重定位分区存储管理

44．系统“抖动”现象的发生是由____引起的。

A．页面置换算法选择不当　　B．交换的信息量过大

C．内存容量不足　　D．请求页式管理方案

45．采用段页式存储管理的系统中，若地址用32位表示，其中10位表示段号，页的大小为4KB，则允许每段的最大页号是____。

A．1024　　B．1023　　C．4096　　D．4095

46．进程在执行中发生了缺页中断，经操作系统处理后，应让其执行____指令。

A．被中断的前一条　　B．被中断的

C．被中断的后一条　　D．启动时的第一条指令

47．虚拟存储管理系统的理论基础是程序的____原理。

A．局部性　　B．全局性　　C．动态性　　D．虚拟性

48．在操作系统中，_____是以时间换取空间的技术。

A．假脱机技术　　B．虚拟存储器　　C．中断技术　　D．通道技术

49．设有3个起始地址都是0的目标模块A、B、C，长度依次为L、M、N，这3个模块按A、B、C顺序采用静态连接方式连接在一起后，模块C的起始地址变为____。

A．L+M+N　　B．L+M　　C．L+M-1　　D．L+M+1

50．下列页面置换算法中，会产生所谓Belady异常现象的是____。

A．最佳页面置换算法（OPT）　　B．先进先出页面置换算法（FIFO）

C．最近最久未使用算法（LRU）　　D．时钟页面置换算法（Clock）

51．操作系统中，具有虚拟存储管理功能的管理方法包括____存储管理。

A．动态分区　　B．分页式　　C．请求分段　　D．段页式

52．Windows 2000采用二级页表，其逻辑地址结构如下：

页目录索引dir(10位)	页表页索引page(10位)	页内偏移offset(12位)

则其页目录的表项数和页的大小分别是____。

A．10和12　　B．20和12　　C．1M（1兆）和4K　　D．1024和4096

53．下列对重定位的叙述中，正确的选项是____。

A．经过静态重定位后，指令代码并不发生变化

B．经过静态重定位后，数据地址和指令地址发生了变化

C．经过动态重定位后，数据地址和指令地址都发生了变化

D．经过动态重定位后，数据地址发生了变化而指令地址没有发生变化

54．假设某计算机系统的内存大小为2560KB，采用可变分区管理内存，在某一时刻内存的使用情况如下表所示：

始址	0K	200K	500K	1000K	1050K	1350K	1600K	1750K	1950K	2350K
状态	已用	未用	已用	未用	已用	未用	已用	未用	已用	未用
容量	200K	300K	500K	50K	300K	250K	150K	200K	400K	210K

此时若进程顺序请求200K、100K和50K的存储空间，系统采用某种内存分配算法为进程依次分配内存，分配后的内存使用情况如下表所示：

始址	0K	400K	500K	1000K	1050K	1450K	1600K	1750K	1950K	2400K
状态	已用	未用	已用	未用	已用	未用	已用	未用	已用	未用
容量	400K	100K	500K	50K	400K	150K	150K	200K	450K	160K

则该系统采用的内存分配算法是____适应算法。

A．首次　　B．循环首次　　C．最佳　　D．最坏

55．某动态分区分配存储管理系统，系统刚把始址为230K的的一小块内存分配出去后，内存中的空闲分区情况如下表所示：

序号	分区大小(KB)	分区始址(K)
1	80	50

2	75	250
3	55	450
4	90	550

有一个作业申请50KB内存，系统把第2个空闲区分配给了该作业50KB，则该系统采用的分区分配算法是____适应算法。

A．首次　　B．最佳　　C．循环首次　　D．最坏

56．下列选项中，对分段存储管理叙述正确的是____。

A．每个段必须是大小相等的　　B．每一段必须是连续的存储区

C．每一段不必是连续的存储区　　D．段之间的存储区必须是连续的

57．在一个分页虚存系统中，设页长2KB，某用户程序有30页。若该程序的虚页0、1、2、3、4、5、6、7已分别装入内存块4、17、18、20、25、26、30、32中，则该程序中的虚地址0AC5H和3AC5H对应的物理地址分别是____。

A．4AC5H和14AC5H　　B．4AC5H和20AC5H

C．8AC5H和20AC5H　　D．8AC5H和102C5H

58．某进程页面访问序列为 4,3,2, 1,4,3,5,4,3,2, 1,5，且开始执行时，内存中没有页面，分配给该进程的物理块数是 3，则采用 FIFO 页面置换算法和 LRU 页面置换算法时缺页率分别是____。

A．83%和75%　　B．85%和70%　　C．75%和83%　　D．84%和75%

59．在请求分页系统中，假如一个作业的页面走向是 1,2,1,3,1,2,4,2,1,3,4，分配给该作业的该作业的物理块数 M 为 2（初始为空），当用 FIFO 页面置换算法时，所发生的缺页次数是____次。

A．10　　B．9　　C．8　　D．7

60．在请求分页系统中，假如一个作业的页面走向是 1,2,1,3,1,2,4,2,1,3,4，分配给该作业的该作业的物理块数 M 为 2（初始为空），当用 LRU 页面置换算法时，所发生的缺页次数是____次。

A．10　　B．9　　C．8　　D．7

61．某基于动态分区存储管理的计算机，其主存容量为55MB(初始为空闲)，采用最佳适配(Best Fit)算法，分配和释放的顺序为：分配15MB、分配30MB、释放15MB、分配8MB、分配6MB，此时主存中最大空闲分区的大小是____。（2010全国试题）

A．7MB　　B．9MB　　C．10MB　　D．15MB

62．某计算机采用二级页表的分页存储管理方式，按字节编址，页大小为2^{10}字节，页表项大小为2字节，逻辑地址结构为：

页目录号	页号	页内偏移量

大小为2^{16}页，则表示整个逻辑地址空间的页目录表中包含表项的个数至少是____。（2010全国试题）

A．64　　B．128　　C．256　　D．512

63．分区分配内存管理方式的主要保护措施是____。（2009全国试题）

A．界地址保护　　B．程序代码保护　　C．数据保护　　D．栈保护

64．一个分段存储管理系统中，地址长度32位，其中段号占8位，则最大段长是____。（2009全国试题）

A．2的8次方字节　　B．2的16次方字节　　C．2的21次方字节　　D．2的32次方字节

65．在缺页处理过程中，操作系统执行的操作可能是____。（2011全国试题）

Ⅰ．修改页表　　Ⅱ．磁盘I/O　　Ⅲ．分配页框

A．仅Ⅰ、Ⅱ　　B．仅Ⅱ　　C．仅Ⅲ　　D．Ⅰ、Ⅱ和Ⅲ

66．当系统发生抖动(thrashing)时，可以采取的有效措施是____。（2011全国试题）

Ⅰ．撤销部分进程

Ⅱ．增加磁盘交换区的容量

Ⅲ．提高用户进程的优先级

A．仅Ⅰ　　B．仅Ⅱ　　C．仅Ⅲ　　D．仅Ⅰ、Ⅱ

67．在虚拟内存管理中，地址变换机构将逻辑地址变换为物理地址，形成该逻辑地址的阶段是_____。（2011全国试题）

A．编辑　　B．编译　　C．连接　　D．装载

68．下列关于虚拟存储器的叙述中，正确的是_____。（2012全国试题）

A．虚拟存储器只能基于连续分配技术　　B．虚拟存储器只能基于非连续分配技术

C．虚拟存储器容量只受外存容量的限制　　D．虚拟存储器容量只受内存容量的限制

第四章存储器管理选择题参考答案：

1．D	2．D	3．C	4．D	5．B	6．B	7．D	8．D	9．C	10．D
11．A	12．C	13．C	14．D	15．A	16．B	17．C	18．D	19．C	20．D
21．D	22．C	23．C	24．C	25．D	26．D	27．D	28．B	29．A	30．A
31．A	32．D	33．D	34．A	35．B	36．A	37．D	38．D	39．A	40．A
41．B	42．B	43．A	44．B	45．B	46．B	47．A	48．B	49．B	50．B
51．C	52．D	53．B	54．D	55．C	56．B	57．D	58．C	59．A	60．C
61．B	62．B	63．A	64．C	65．D	66．A	67．C	68．C		

2．应用题

1．请求分页系统中，设某进程共有 9 个页，分配给该进程的主存块数为 5(即工作集为 5)，进程运行时，实际访问页面的次序是 0，1，2，3，4，5，0，2，1，8，5，2，7，6，0，1，2。试求：

(1) FIFO页面调度算法，列出其页面淘汰次序和缺页中断次数，以及最后留驻主存的页号顺序。

(2) LRU页面调度算法，列出其页面淘汰次序和缺页中断次数，以及最后留驻主存的页号顺序。

(3) CLOCK页面调度算法，列出其页面淘汰次序和缺页中断次数，以及最后留驻主存的页号顺序。

(4) OPT页面调度算法，列出其页面淘汰次序和缺页中断次数，以及最后留驻主存的页号顺序。

解：(1) 采用FIFO页面调度算法

访问序列	0	1	2	3	4	5	0	2	1	8	5	2	7	6	0	1	2
内存块1	0	0	0	0	0	5	5	5	5	5	5	5	7	7	7	7	7
内存块2		1	1	1	1	1	0	0	0	0	0	0	0	6	6	6	6
内存块3			2	2	2	2	2	2	1	1	1	1	1	1	0	0	0
内存块4				3	3	3	3	3	3	8	8	8	8	8	8	1	1
内存块5					4	4	4	4	4	4	4	2	2	2	2	2	2
淘汰的页	√	√	√	√	√	0	1		2	3		4	5	0	1	8	

因此，页面淘汰顺序为 0、1、2、3、4、5、0、1、8，缺页中断次数为 14 次。最后留驻主存的页号顺序为 7、6、0、1、2。

(2) 采用LRU页面调度算法(因要列出最后的页号顺序，故没采用页号队列演算)

访问序列	0	1	2	3	4	5	0	2	1	8	5	2	7	6	0	1	2
内存块1	0	0	0	0	0	5	5	5	5	5	5	5	5	5	5	1	1
内存块2		1	1	1	1	1	0	0	0	0	0	0	7	7	7	7	7
内存块3			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

内存块4				3	3	3	3	3	1	1	1	1	1	6	6	6	6
内存块5					4	4	4	4	4	8	8	8	8	8	0	0	0
淘汰的页	√	√	√	√	√	0	1		3	4			0	1	8	5	

因此，页面淘汰顺序为0、1、3、4、0、1、8、5，缺页中断次数为13次。最后留驻主存的页号顺序为1 、7、2、6、0。

(3) CLOCK页面调度算法(用蓝色表示指针位置，*号表示访问标志为1)

访问序列	0	1	2	3	4	5	0	2	1	8	5	2	7	6	0	1	2
内存块1	0*	0*	0*	0*	0*	5*	5*	5*	5*	5*	5*	5*	7*	7*	7*	7*	7*
内存块2		1*	1*	1*	1*	1	0*	0*	0*	0*	0*	0*	0	6*	6*	6*	6*
内存块3			2*	2*	2*	2	2	2*	2	2	2	2*	2	2	0*	0*	0*
内存块4				3*	3*	3	3	3	1*	1*	1*	1*	1	1	1	1*	1*
内存块5					4*	4	4	4	4	8*	8*	8*	8	8	8	8	2*
淘汰的页	√	√	√	√	√	0	1		3	4			5	0	2		8

因此，页面淘汰顺序为0、1、3、4、5、0、2、8，缺页中断次数为14次。最后留驻主存的页号顺序为7、6、0、1、2。

(4) OPT页面调度算法

访问序列	0	1	2	3	4	5	0	2	1	8	5	2	7	6	0	1	2
内存块1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
内存块2		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
内存块3			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
内存块4				3	3	3	3	3	3	8	8	8	7	7	7	7	7
内存块5					4	5	5	5	5	5	5	5	5	6	6	6	6
淘汰的页	√	√	√	√	√	4				3			8	5			

因此，页面淘汰顺序为4、3、8、5(或4、3、5、8)，缺页中断次数为9次。最后留驻主存的页号顺序为0、1、2、7、6(或0、1、2、6、7)。(还可有其它组合，从略)

2．页式存储管理中，主存空间按页分配，可用一张“位示图”构成主存分配表。假设主存容量为2M字节，页面长度为512字节，若用字长为32位的字作主存分配的“位示图”需要多少个字？如页号从1开始，字号和字内位号（从高位到低位）均从0开始，试问：第2999页对应于何字何位；99字19位又对应于第几页？

解：(1) 内存总块数=2MB/512B=4096

位示图需要字数=4096/32=128

(2) 字号=(2999-1)/32=93

位号=(2999-1)%32=22

即第2999内存页对应于位示图中93字的22位。

(3) 99*32+19+1=3188

即位示图99字19位对应于内存的3188页

3．某操作系统采用可变分区分配存储管理方法，用户区为512K且始址为0，用空闲分区表管理空闲分区。若分配时采用分配空闲低地址部分的方案，其初始时用户区的512K空间空闲，对下述申请序列：申请300K，申请100K，释放300K，申请150K，申请30K，申请40K，申请60K，释放30K；回答下列问题：

（1）采用首次适应算法，空闲分区中有哪些空闲块（给出始址，大小）？

（2）采用最佳适应算法，空闲分区中有哪些空闲块（给出始址，大小）？

答：(1)

序号	始址	大小
1	150K	30KB
2	280K	20KB
3	400K	112KB

(2)

序号	始址	大小
1	210K	90KB
2	400K	30KB
3	470K	42KB

4．考虑一个 460 字的程序的下述内存访问序列：

10 19 154 170 54 334 185 245 247 456 458 378

（1）假定页面大小为 100 字，试给出页访问串；

（2）假定内存中有 200 个字可供程序使用且采用 FIFO 算法，那么有关该访问串的缺页中断次数是多少?

（3）若使用 LRU 算法，则有关该访问串的缺页中断次数是多少?

答：(1) 页访问串为 0, 0, 1, 1, 0, 3, 1, 2, 2, 4, 4, 3

(2) 分得 2 个内存块。FIFO 的页面置换图以及 LRU 的页面置换图略。

5．某系统对主存采用页式管理，供用户使用的主存区域共 640K 字节，被分成 160 块，块号为 0，1，…，159。现有一作业的地址空间共占 4 页，其页号为 0, 1, 2, 3，被分配到主存的第 2，4，1，5 块中。请回答：

(1) 作业每一页的长度为多少字节？

(2) 写出该作业被装入主存时，其对应的页表。

(3) 把该作业的每一页在主存中的起始地址(用16进制表示)填在下表中：

页号	起始地址
0	
1	
2	
3	

解：(1) 作业每一页的长度为4K字节

(2) 该作业被装入主存时，其对应的页表为

逻辑页号	主存块号
0	2
1	4
2	1
3	5

(3) 该作业的每一页在主存中的起始地址(用16进制表示)如下表所示。

页号	起始地址
0	2000H
1	4000H
2	1000H
3	5000H

6．可变分区存储管理中，作业的撤离必定会修改内存的“空闲区表”，试画出因作业撤离修改“空闲区表”的四种情况。

答：① 回收区与插入点的前一个空闲分区 F1 相邻接(即回收区的低端有邻接区 F1，如图①)：将回收区与 F1 合并，不必增加新表项，只需修改 F1 的大小为两者之和。

② 回收区与高地址 F2 分区邻接(如图②)：此时将回收分区与该分区合并，回收区的首地址为新分区的首地址，大小为两者之和。

图 ①

图 ②

③ 回收区与前后分区 F1 和 F2 都邻接(如图③)：将此 3 个分区合并，F1（前邻接区）的首地址为新分区的首址，大小为三者之和，取消 F2 表项。

④ 回收区与任何空闲区都不邻接(如图④)：在插入点建立一个新表项，填写回收区的首地址和大小。插入到空闲区表的适当位置（后移插入点后的各个表项）

图 ③

图 ④

7．在一个采用页式虚拟存储管理的系统中，有一用户作业，它依次要访问的字地址序列是：115，228，120，88，446，102，321，432，260，167，若该作业的第 0 页已经装入主存，现分配给该作业的主存共 300 字，页的大小为 100 字，请回答下列问题：（此图类似地 4 题）

（1）按FIFO调度算法将产生的缺页中断次数、依次淘汰的页号和缺页中断率各为多少？

（2）按LRU调度算法将产生的缺页中断次数、依次淘汰的页号和缺页中断率各为多少？

答：由题目的已知条件，可得页面走向为：

1, 2, 1, 0, 4, 1, 3, 4, 2, 1

(1) FIFO的页面置换图如下：

页面走向	1	2	1	0	4	1	3	4	2	1
	0	0	0	0	4	4	4	4	4	4
页帧	1	1	1	1	1	1	3	3	3	3
		2	2	2	2	2	2	2	2	1
是否缺页	√	√			√		√			√
被淘汰页号					0		1			2

按FIFO调度算法将产生5次缺页中断，依次淘汰的页号为0,1,2，缺页中断率为5/10=50%。

(2) LRU算法的页面置换图如下：

页面走向	1	2	1	0	4	1	3	4	2	1
	1	2	1	0	4	1	3	4	2	1
页面队列	0	1	2	1	0	4	1	3	4	2
		0	0	2	1	0	4	1	3	4
是否缺页	√	√			√		√		√	√
被淘汰页号					2		0		1	3

按LRU调度算法将产生6次缺页中断，依次淘汰的页号为2,0,1,3，缺页中断率为6/10=60%。

8．分页式存储空间的分配由于块的大小是固定的，可以用一张位示图(Bit map)来构成主存分配表。现设主存有 8192 块，可用字长为 32 位的 256 个字作为位示图。若块号，字号，位号(从高位到低位)分别从 1、0、0 开始，试问 5999 块对应的字号和位号?99 字的 19 位对应哪一块?

9．设某作业占有 7 个页面，如果在主存中只允许装入 4 个工作页面(即工作集为 4)，作业运行时，实际访问页面的顺序是 1，2，3，6，4，7，3，2，1，4，7，5，6，5，2，1。假设开始的 4 个页面已装入主存。

(1) 若试用FIFO页面调度算法，列出各自的页面淘汰顺序和缺页中断次数，以及最后留驻主存4页的顺序。

(2) 若用LRU页面调度算法，列出各自的页面淘汰顺序和缺页中断次数，以及最后留驻主存4页的顺序。

解：(1) 采用FIFO页面调度算法

访问序列	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
内存块1				1	4	4	4	4	4	4	4	5	5	5	5	5
内存块2				2	2	7	7	7	7	7	7	7	6	6	6	6
内存块3				3	3	3	3	2	2	2	2	2	2	2	2	2
内存块4				6	6	6	6	6	1	1	1	1	1	1	1	1
淘汰的页					1	2		3	6			4	7			

因此，页面淘汰顺序为 1、2、3、6、4、7，因开始的 4 个页面已装入主存，缺页中断次数为 6 次。最后留驻主存 4 页的顺序为 5、6、2、1。

(2) 采用LRU页面调度算法

访问序列	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
页				6	4	7	3	2	1	4	7	5	6	5	2	1
面				3	6	4	7	3	2	1	4	7	5	6	5	2
队				2	3	6	4	7	3	2	1	4	7	7	6	5
列				1	2	3	6	4	7	3	2	1	4	4	7	6
淘汰的页					1	2		6	4	7	3	2	1		4	7

因此，页面淘汰顺序为 1、2、6、4、7、3、2、1、4、7，因开始的 4 个页面已装入主存，缺页中断次数为 10 次。最后留驻主存 4 页的顺序为 1、2、5、6。

10．分页式存储空间的分配由于块的大小是固定的，可以用一张位示图(Bit map)来构成主存分配表。现设主存有 8192 块，则可用字长为 32 位的 256 个字作为位示图。若块号、字号、位号(从高位到低位)都是从 0 开始，试问 4999 块对应的字号和位号；129 字的 29 位对应哪一块?

11．有一个虚存系统，某进程占用 3 个内存块，开始时内存为空，执行如下访问页号序列后：

0，1，2，3，1，4，1，2，5，1，2，3，4，5

（1） 采用先进先出（FIFO）淘汰算法，缺页次数是多少？

（2） 采用最近最少使用（LRU）淘汰算法，缺页次数是多少？

（3） 若用最优（OPT）算法呢？

解：（1）FIFO算法：

访问序列	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
内存块	5	5	5	2	2	2	2	4	4	4	0	0	0	0	0	0	0	5	5	5
		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
缺页否	+	+	+	+		+	+	+	+	+	+			+	+			+	+	+

因此，采用FIFO淘汰算法，缺页次数为15次。

（2）LRU算法(通过页面队列进行演算)：

访问序列	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
页面队列	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1

	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0
		5	0	1	2	2	3	0	4	2	2	0	3	3	1	2	0	1	5
缺页否	+	+	+	+		+		+	+	+	+			+		+		+	

因此，采用LRU淘汰算法，缺页次数为12次。

（3）OPT算法：

访问序列	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	5
内存块		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
缺页否	+	+	+	+		+		+			+			+				+		

因此，采用OPT淘汰算法，缺页次数为9次。

12．在采用页式存储管理的系统中，某作业的逻辑地址空间为 4 页（每页 2048 字节），且已知该作业的页表如下表。试借助地址转换图（即要求画出页式存储管理系统地址转换示意图）求出逻辑地址 4688 所对应的物理地址。

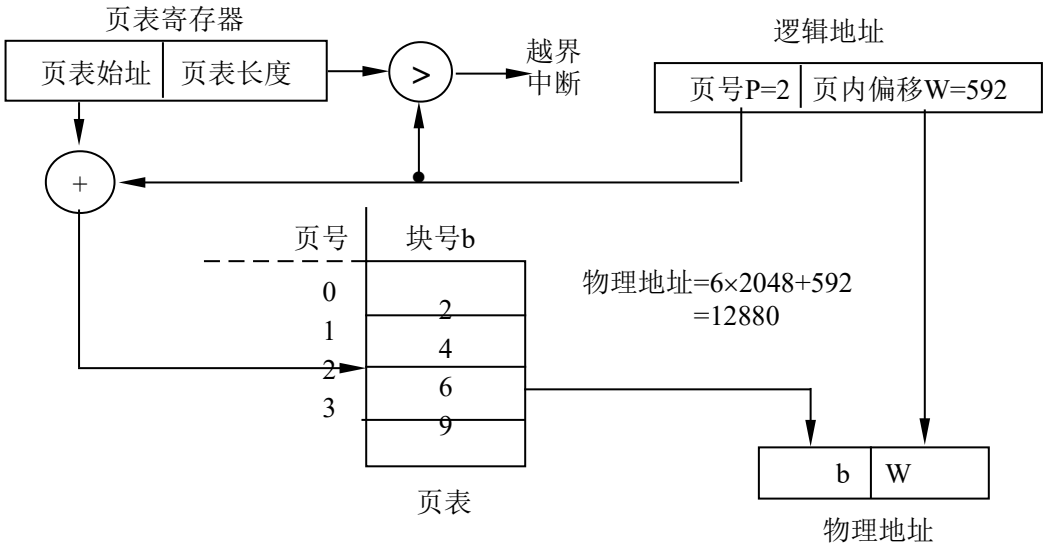
页 表

页 号	内存块号
0	2
1	4
2	6
3	9

解：逻辑地址4688所在的页号和页内偏移分别为：

页号P=4688/2048=2

页内偏移W=4688%2048=592



从上述地址转换图可知，进行地址转换的步骤如下：

（1） 由虚地址计算出页号和页内偏移量；

（2） 根据页号和进程的页表首址，查页表，找到对应的页表项，取出帧号(内存块号)；

（3） 帧号*页面大小+页内偏移形成物理地址。即6×2048+592=12880

13．有一矩阵int a[100][100];）该矩阵按先行后列次序存储。在一个页式虚拟存储系统中，采用LRU页面淘汰算法，设给该进程分配3个内存块(也称页框)，每页可以存放200个整数。其中第1个页框存放程序，

且假定程序已经在内存。

程序A：
```
for (i=0;i<100;i++)
    for (j=0;j<100;j++)
        a[i][j]=0;
```

程序B：
```
for (j=0;j<100;j++)
    for (i=0;i<100;i++)
        a[i][j]=0;
```

分别就程序A和程序B的执行过程计算缺页次数。

解：(1) 先考虑程序A。执行循环语句时，访问程序所在的页(不妨假设为0页)，因程序已在内存，不缺页。

执行第一次循环时，i=0, j=0, 当执行到语句“a[i][j]=0;”时，要访问a[0][0]所在的页(不妨假设为1页)，因0页不在内存，故产生缺页中断，将a[0][0]所在的页装入第2个页框中。内循环控制变量j增1，变为i=0, j=1，访问a[0][1]所在的页，即1号页，因1号页已在内存，不缺页；类似地，j=2，3，…，99，给a[0][2]，…，a[0][99]赋值时也不缺页；外循环控制变量i增1，变为i=2，j=0，1，…，99，对应的a[1][0]，…，a[1][99]仍在1号页中，故对它们赋值时不会发生缺页。

当i=2，j=0时，给a[2][0]赋值，因其所在的2号页不在内存，故产生缺页中断，将a[2][0]所在的页装入第3个页框中。类似地，当j=2，3，…，99，给a[2][1]，…，a[2][99]赋值，以及当i=3，j=0，1，…，99，对应的a[1][0]，…，a[1][99]仍在1号页中，故对它们赋值时不会发生缺页。

当i=4，j=0时，给a[4][0]赋值，因其所在的3号页不在内存，故产生缺页中断，因此时分配的内存块已用完，需置换1个页。因系统采用LRU页面淘汰算法，而程序所在的0号页在不断地访问，因此第1页是最近以来最久未使用的，将1号淘汰，将3号页装入第2个页框中。此后，当i=4，j=2，3，…，99，以及当i=5，j=0，2，…，99，给相应的数组元素赋值时，也不产生缺页中断。

类似地，当i=4、6、…、98时，各发生1次缺页中断。

综上所述，程序A执行过程中，共发生50次缺页。

(2) 再考虑程序B。当j=0, i=0，给a[0][0]赋值时缺页1次；当j=0, i=1给a[1][0]赋值时，因其与a[0][0]在同一页，故不缺页。显然，当j=0，i=2，4，…，98时，各发生1次缺页中断，所以，在j=0的情况下共发生50次缺页。类似地，在j=1，2，…，,99时，各发生50次缺页。

故程序B执行时，共发生50*100=5000次缺页。

14．某操作系统采用可变分区分配存储管理方法，用户区为 512K 且始值为 0，用空闲分区表管理空闲分区。若分配时采用分配空闲低地址部分的方案，其初始时用户区的 512K 空间空闲，对下述申请序列：申请 300K，申请 100K，释放 300K，申请 150K，申请 30K，申请 40K，申请 60K，释放 30K；回答下列问题：

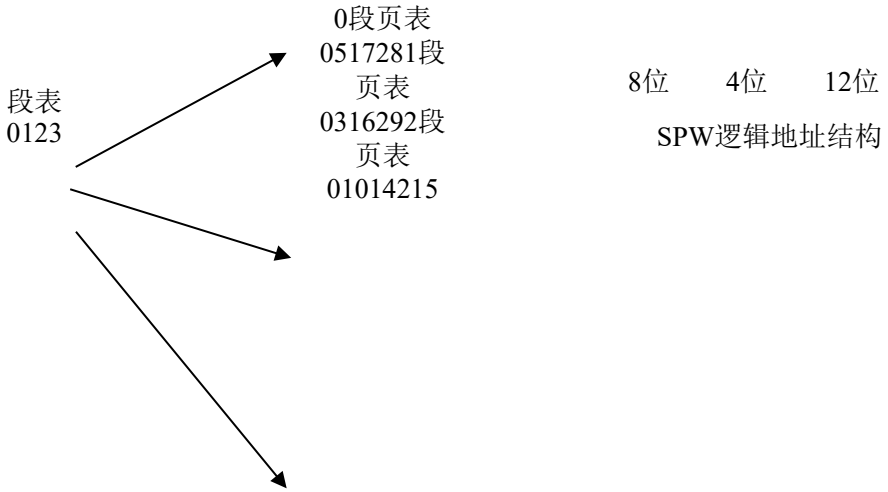
（1）采用首次适应算法，空闲分区中有哪些空闲块（给出始址，大小）？

（2）采用最佳适应算法，空闲分区中有哪些空闲块（给出始址，大小）？

15．页式存储管理中，主存空间按页分配，可用一张“位示图”构成主存分配表。假设主存容量为 2M 字节，页面长度为 512 字节，若用字长为 32 位的字作主存分配的“位示图”需要多少个字？如页号从 1 开始，字号和字内位号（从高位到低位）均从 0 开始，试问：第 2999 页对应于何字何位；99 字 19 位又对应于第几页？

16．分页式存储空间的分配由于块的大小是固定的，可以用一张位示图(Bit map)来构成主存分配表。现设主存有 8192 块，可用字长为 32 位的 256 个字作为位示图。若块号，字号，位号(从高位到低位)分别从 1、0、0 开始，试问 5999 块对应的字号和位号?99 字的 19 位对应哪一块?

17．某系统采用段页式存储管理，其逻辑地址结构和某作业的段表、页表结构如下图所示。请计算该作业中逻辑地址 135468 对应的物理地址（用十进制表示）。



解：W=135468 mod 4096=300　（注：2^{12}=4096）

135468 div 4096=33 (此处 div 表示正数除法)

P=33 mod 16=1　（注：2^4=16）

S=33 div 16=2

即段号为 2，页号为 1，页内偏移为 300。查 2 段页表，1 号页对应的内存块号为 4，故

对应的物理地址=4×4096+300=16684

18．对于页面大小为512字节的计算机，编制了下列汇编语言程序，程序的内存起始地址为1020，其堆栈指针（2字节）指向内存地址8192（向低地址扩展），每条指令占据4个字节。汇编语言程序的功能如下：

1020：（1）装入6144单元的字（2字节字）到0号寄存器；

1024：（2）0号寄存器内容压入堆栈（假定堆栈指针先减1后再入栈）；

1028：（3）调用5120处的子程序，（压入返回地址到堆栈）；

1032：（4）弹出堆栈到0号寄存器；

5120：（5）堆栈指针保存到6000单元；

5124：（6）将立即数7180存入堆栈指针；

5128：（7）堆栈指针所指单元内容减1；

5132：（8）堆栈指针与立即数160相减，结果仍在堆栈指针中；

5136：（9）堆栈指针所指单元内容立即数100比较；

5140：（10）若相等转到5152；（此处假设比较结果是“不相等”）

5144：（11）6000单元内容存回堆栈指针；

5148：（12）子程序返回。

完成下列问题：

（1）给出程序执行的页面访问序列。（本小题6分）

1，12，2，15，2，15，10，11，10，10，14，10，10，13，10，10，11，10，15，2，15

（2）若程序运行时占用5个内存块，采用LRU算法，求页面置换次数。（本小题4分）

页面置换4次

19．一个32位地址的计算机系统使用二级页表，虚地址分为10位顶级页表，10位二级页表，其余是页内偏移。试问：(1) 页面长度是多少？(2) 虚拟地址空间有多少个页面？

20．某计算机有cache、内存、辅存来实现虚拟存储器。如果数据在cache，访问它需要10ns；如果在内存单不在cache，需要60ns将其装入缓存，然后才能访问；如果不在内存而在辅存，需要5ms将其装入内存，再用60ns将其装入cache，然后才能访问。假设cache命中率是0.9，内存命中率为0.8(钱注：指cache未命中时，即余下的10%中内存的命中率为80%)，则数据平均访问时间是多少(ns)？

解：5ms=5,000,000ns

0.9×10+(1-0.9)×0.8× (60+10)+(1-0.9)×(1-0.8)×(5,000,000+60+10)

=9 + 5.6+1,000,01.4

=100,016 (ns)

即数据平均访问时间是100,016 ns。

21．如果一条指令执行时间是1μs，发生一次缺页需要的处理时间为Xμs，若缺页率为平均每Y条指令发生一次，则指令平均执行时间W是多少？

解：W=(1*(Y-1)+X)/Y=(Y-1+X)/Y=1+(X-1)/Y (μs)

22．假定系统为某进程分配了3个物理块，并考虑以下的页面引用串：

7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1。采用Clock置换算法，计算其页面置换次数。

23．设某作业占有5个页面，作业运行时，实际访问页面的顺序是3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4。假设采用FIFO页面调度算法。

(1) 如果分配的主存块数为3，求缺页中断次数（需给出页面置换图）。

(2) 如果分配的主存块数为4，求缺页中断次数（需给出页面置换图）。

Belady现象——对FIFO页面置换算法，有时分配的内存块越多，缺页率越高。

24．设某作业的页面引用串为：1,2,3,4,1,2,5,1,2,3,4,5，指出驻留集大小分别为3、4时，采用FIFO和LRU页面置换算法的缺页中断次数。结果说明了什么？

解：当驻留集为3时：

采用FIFO算法时，缺页次数为9次（此处页面置换图略）

采用LRU算法时，缺页次数为10次（此处页面置换图略）

当驻留集为4时：

采用FIFO算法时，缺页次数为10次（此处页面置换图略）

采用LRU算法时，缺页次数为8次（此处页面置换图略）

结果表明，FIFO 有 Belady 异常现象，即缺页中断次数不随驻留集增大而减少；而 LRU 算法的缺页中断次数随驻留集增大而减少。另外，当驻留集为 3 时，LRU 算法的缺页次数比 FIFO 算法多，所以 LRU 算法不总优于 FIFO 算法，还要看当前页面访问串的特点以及分配的页框数。

25．有一个虚存系统，某进程占用3个内存块，开始时内存为空，执行如下访问页号序列后：

5，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，5，0，1

（1） 采用先进先出（FIFO）页面置换算法，面页置换次数是多少？(需画置换图)

（2） 采用最近最久未使用（LRU）页面置换算法，面页置换次数是多少？(需画置换图)

（3） 采用时钟（Clock）页面置换算法，面页置换次数是多少(需画置换图)？缺页率是多少？

解：(1)采用先进先出（FIFO）页面置换算法，其置换图如下：

5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
5	5	5	2		2	2	4	4	4	0			0	0			5	5	5

	0	0	0		3	3	3	2	2	2			1	1			1	0	0
		1	1		1	0	0	0	3	3			3	2			2	2	1
			√		√	√	√	√	√	√			√	√			√	√	√

由上述演算可知，页面置换次数为12次。

(2) 最近最久未使用（LRU）页面置换算法，其置换图如下：

5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0
		5	0	1	2	2	3	0	4	2	2	0	3	3	1	2	0	1	5
			√		√		√	√	√	√			√		√		√		

由上述演算可知，页面置换次数为9次。

(3) 采用时钟（Clock）页面置换算法，其置换图如下(其中*为指针位置，+号为访问标志)：

5	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	5	0	1
*5+	*5+	*5+	2+	2+	*2+	*2+	4+	4+	4+	*4	3+	3+	3	*3	0+	0+	0	0+	*0
	0+	0+	*0	*0+	0	0+	*0	2+	2+	2	*2	*2+	1+	1+	*1+	*1+	5+	5+	5
		1+	1	1	3+	3+	3	*3	*3+	0+	0+	0+	*0	2+	2+	2+	*2	*2+	1+
			√		√		√	√		√	√		√	√	√		√		√

由上述演算可知，页面置换次数为11次。缺页率为14/20=70%

26．在页式虚存管理系统中，假定驻留集为m个页面（初始所有页帧均为空），在长为p的引用串中具有n个不同的页号(n>m)，对于FIFO、LRU两种页面置换算法，试给出页故障数的上限和下限，说明理由，并举例说明。

答：对于FIFO，页故障数的上限是p，下限是n。因为FIFO淘汰掉先进来的页，而不管其页面以后是否还会用到。在极端情况下，可能刚淘汰的页又接着要使用。故页故障数上限为p；而不同的页至少有一次页故障，故下限是n。

对于 LRU，页故障数的上限是 p，下限是 n。因为同样可能刚淘汰的页又接着要使用。故页故障数上限为 p；而不同的页至少有一次页故障，故下限是 n。

例如，对于页面引用串：0，1，2，3，0，1，2，分配的内存块数为 3

n=4，p=7，m=3，由页面置换图易知，采用 FIFO 和 LRU 页面置换算法，其页故障数皆为 7。

又如，对于页面引用串：0，1，2，3，1，2，3，分配的内存块数为 3

n=4，p=7，m=3，由页面置换图易知，采用 FIFO 和 LRU 页面置换算法，其页故障数皆为 4。

27．某采用段式存储管理的系统为装入主存的一个作业建立了如下段表：

段号	段 长	主存起始地址
0	660	210
1	140	3300
2	100	90
3	580	1237
4	960	1959

（1）计算该作业访问[0，432]，[1，10]，[2，500]，[3，400]（方括号中第一个元素为段号，第二个元素为段内地址）的绝对地址。

（2）总结段式存储管理的地址转换过程。

解：（1）略（其中逻辑地址[2，500]将产生地址越界中断）

（2）段式存储管理的地址转换过程如下：（此题也可画图加简单文字说明来解答）

将逻辑地址中的段号与段表寄存器中该作业的段表长度比较，若超出，则产生地址越界中断；若不超出，则进行如下工作：

由段表寄存器中的段表地址找到该作业的段表，由段号找到该段在段表中的表目；如果逻辑地址中的段内地址不超过该表目中所示长度，则把该表目中的起始地址与段内地址相加，所得的值就是要访问的主存绝对地址；否则，如果逻辑地址中的段内地址超过该表目中所示长度，则产生地址越界中断，暂停作业的运行。

28．在请求分页系统中，某用户程序的逻辑地址空间为16页，每页1KB，分配的内存空间为8KB。假定某时刻该用户的页表如下表所示。

页号	块号
0	3
1	7
2	4
3	1
4	12
5	9
6	61
7	20

试问：

（1）逻辑地址 184BH 对应的物理地址是多少？（用十六进制表示）

（2）逻辑地址 5000(十进制)对应的物理地址是多少？（用十进制表示）

（3）当用户进程欲访问 24A0H 单元时，会出现什么现象？

解：（1）逻辑地址184BH=1 1000 0100 1011B，低10位00 0100 1011B是页内偏移量，高位110B是页号，即页号为6，查页表得内存块号为61，即111101B，与页内偏移地址00 0100 1011B构成物理地址为1111 0100 0100 1011B，即0F44BH。

（2）5000 DIV 1024=4，5000 MOD 1024=904，即逻辑地址5000的页号为4，页内地址为904。查页表知其所在的页框号为12，故对应的物理地址为：12*1024+904=13192。

（3）逻辑地址24A0H=10 0100 1010 0000B，其所在的页号为1001B=9，由题目所给条件可知该逻辑地址所在的页不在内存，故当用户进程欲访问24A0H单元时，会产生缺页中断。

29．某计算机系统的某个进程的页表有4个域：装入时间、上次引用时间、R（读）与M（修改）位，见下表。请问：采用Clock算法、FIFO、LRU和第二次机会页面置换算法，将分别置换哪一页？

钱注：第二次机会页面置换算法实现过程：首先检查FIFO队首页面，如果它的引用位为“0”，则淘汰它；如果它的引用位为“1”，则将其引用位清成0，并把这个页移到队尾，把它看做一个新调入的页，再给它一次机会。该算法与Clock算法本质上没有区别，仅仅是实现方法不同，Clock算法是第二次机会页面置换算法的改进。第二次机会页面置换算法实现时因可能产生频繁的出队入队，实现代价较大。

页号	装入时间	上次引用时间	R	M
0	126	279	0	0
1	230	260	1	0

2	120	272	1	1
3	160	280	1	1

答：采用Clock算法淘汰0#页，FIFO算法淘汰2#页，LRU算法淘汰1#页，第二次机会页面置换算法淘汰0#页。

30．(2010全国试题)设某计算机的逻辑地址空间和物理地址空间均为64KB，按字节编址。若某进程最多需要6页(Page)数据存储空间，页的大小为1KB，操作系统采用固定分配局部置换策略为此进程分配4个页框(Page Frame)。在时刻260前的该进程访问情况如下表所示(访问位即使用位)。

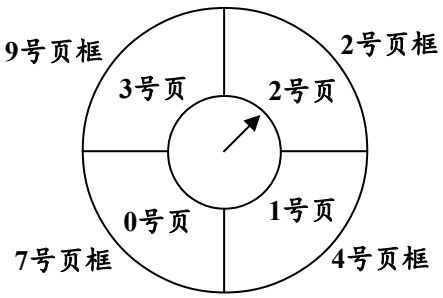
页号	页框号	装入时间	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	160	1

当进程执行到时刻260时，要访问逻辑地址为17CAH的数据。请回答下列问题：

（1）该逻辑地址的对应的页号是多少？

（2）若采用先进先出(FIFO)置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。

（3）若采用时钟(CLOCK)置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程(设搜索下一页的指针沿顺时针方向移动，且当前指向2号页框，示意图如下)。



答案要点：

(1) 17CAH=0001 0111 1100 1010B，表示页号的位是左边6位，即00101B，所以页号为5。

(2) 根据FIFO算法，需要替换装入时间最早的页，故需要置换装入时间最早的0号页，即将5页装入7号页框中，所以物理地址为0001 1111 1100 1010B，换算成十六进制，为1FCAH。

(3) 根据CLOCK算法，如果当前指针所指页框的使用位为0，则替换该页；否则将其使用位清零，并将指针指向下一个页框，继续查找。根据题设和示意图，将从2号页框开始，前4次查找页框顺序为2→4→7→9，并将对应页框的使用位清零。在第5次查找中，指针指向2号页框，因2号页框的使用位为0，故淘汰2号页框对应的2号页，把5号页装入2号页框中，并将对应的使用位置为1，所以对应的物理地址为0000 1011 1100 1010B，换算成十六进制，为0BCAH。

46．(2009全国试题)请求分页管理系统中，假设某进程的页表内容如下表所示。

页表内容

页号	页框(Page frame)号	有效位（存在位）
0	101H	1
1	—	0
2	254H	1

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① TLB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 2362H、1565H、25A5H，请问：

(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1565H的物理地址是多少？请说明理由。

参考答案：

(1) 因页的大小为 4KB，即 2^{12}，故十六进制地址的低 3 位是页内偏移，高位是页号。

2362H：页号 P=2，访问快表 10ns，因初始为空，访问页表 100ns 得到页框号，与页内偏移合成物理地址后访问内存 100ns，共花时间 210ns。

1565H：P=1，访问快表10ns，落空，访问页表100ns缺页，进行缺页中断处理108ns，合成物理地址后访问内存100ns，共计318ns。

25A5H：P=2，访问快表10ns命中，合成物理地址后访问内存100ns，共计110ns。

(2) 因采用LRU算法，故访问1565H时淘汰的是0号页，空出101H号页框存放逻辑地址1565H所在的1号页。由页框号101H和页内偏移565H合成得到虚地址1565H对应的物理地址为101565H。

47. 在某一采用固定分配局部置换策略的请求分页系统中，有一进程逻辑地址空间有10个页，分得了4个页框，每页的装入时间、最后访问时间、访问位R如下表所示(时间用时钟点数表示)。

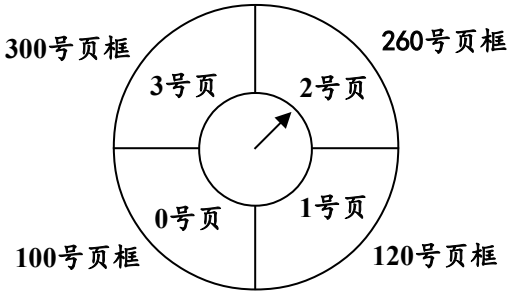
页号	页框号	装入时间	最后访问时间	访问位 R
0	100	126	260	0
1	120	230	200	1
2	260	120	230	1
3	300	160	280	1

假设页的大小为 4KB(4096B)，当进程执行到时刻 300 时，要访问逻辑地址 6AB8H 的数据，请回答下列问题：

（1）若采用先进先出(FIFO)置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。

（2）若采用最近最久未使用(LRU)页面置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。

（3）若采用时钟(CLOCK)置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程(设搜索下一页的指针沿顺时针方向移动，且当前指向 2 号页，示意图如下)。（2010 自编题）



解：

（1）因页的大小为 4KB，故逻辑地址 6AB8H 的页内偏移为 0AB8H，页号为 6。.该逻辑地址所在的页不在内存，产生缺页中断，又因已无空闲页框，故需页面置换。采用先进先出(FIFO)置换算法，因 2 号页最先装入内存，将被置换，其所在的页框号位 260，即 104H，将 104H 与页内偏移 0AB8H 合成，得到物理地址 104AB8H。

（2）采用最近最久未使用(LRU)页面置换算法，因 1 号页是最久没有访问的，将被置换，其所在的页框号位 120，即 78H，将其与页内偏移 0AB8H 合成，得到物理地址 78AB8H。

（3）采用时钟(CLOCK)置换算法，因 0 号页的访问位是 0，将被置换，其所在的页框号位 100，即 64H，将其与页内偏移 0AB8H 合成，得到物理地址 64AB8H。

48. (武汉理工大学2003年试题)假定某采用分页存储管理的系统中，主存容量为4MB，被分成1024块，块号为0，1，2，…，1023。某作业的地址空间占4页，其页号为0，1，2，3，被分配到主存的第28，26，12，57块中。回答：

(1) 主存地址应该用多少位来表示？

(2) 作业每一页的长度为多少？逻辑地址中的页内地址(页内偏移)应占多少位？

(3) 把作业中每一页分配到主存块中的起始地址填入下表：

页号	起始地址	页号	起始地址
0		2	
1		3	

49．一个分页存储器的页表存放在内存。

（1）若内存的存取周期为60ns，则CPU从内存取一条指令（或一个操作数）需要多少时间？

（2）若使用快表(存取周期为10ns)且快表的命中率为75%，则内存的平均存取周期为多少？

答：（1）因页表放在内存，所以取一条指令（或一个操作数）需要2次访问内存，所以需要60ns×2=120ns的时间

（2）假设访问快表的命中率为75%时，此时的平均存取周期为

(10+60)×0.75+60×2×(1-0.75)=82.5(ns)

50．假设当前在处理器上执行的进程的页表如下所示。所有数字为十进制数，每一项都是从0开始计数的，并且所有的地址都是存储器字节地址。页的大小为1024个字节。

I. 正确地描述CPU产生的虚拟地址通常是如何转化成一个物理主存地址的。

II.下列虚地址对应于哪个物理地址(缺页时暂不处理)?

(i) 1052 (ii) 2221 (iii) 5499

虚页号	有效位	访问位	修改位	页帧号
0	1	1	0	4
1	1	1	1	7
2	0	0	0	—
3	1	0	0	2
4	0	0	0	—
5	1	0	1	0

I 答：对于一级页表，进行地址转换的步骤如下：

（1） 由虚地址计算出页号和页内偏移量；

（2） 根据页号和进程的页表首址，查页表，找到对应的页表项，取出帧号；

（3） (帧号*页面大小)+页内偏移形成物理地址。

II 答：p=int(A/L) d=A mod L (A为虚地址，L为页面大小，p为页号，d为页内偏移)

(i) p=int(1052/1024)=1 d=1052 mod 1024=28

根据页号1查页表得到帧号为7，则1052对应的物理地址=7*1024+28=7196

(ii) p=int(2221/1024)=2 d=173

查页表知2号页不在内存，将产生缺页中断。

(iii) p=int(5499/1024)=5 d= 379

根据页号5查页表得到帧号为0，则5499对应的物理地址=0*1024+379=379

51．分页系统存取一次内存的时间是8ns，查询一次快表的时间为2ns，缺页中断的时间是20ns。现一作业的2、3页面已经在内存，并且2页常驻内存。现对作业的2, 4, 2, 3页面进行连续存取。假设页表的查询与快表的查询同时进行，求：在上述4次存取中，每次存取可能需要多少时间？

答：(1) 存取2号页时，因为它已经在内存，可能需要的时间有两种情况：

①查询快表命中(2ns)，得到内存块号形成物理地址后再访问内存(8ns)，2+8=10(ns)

②快表未命中，查页表8ns，再访问内存中的目标单元8ns，8+8=16(ns)

(2) 存取4页面时，它不在内存，发生缺页中断，存取可能的时间是：8ns+20ns+8ns+8ns=44ns

(3) 存取2页面时，因为是第二次存取，可以命中快表，存取的可能时间是：2ns+8ns=10ns

(4) 存取3页面时，因为调入4页面时，有可能将3页面淘汰，也可能没淘汰，存取的可能时间有三种：

①快表命中，8ns+2ns=10ns

②页表命中，8ns+8ns=16ns

③发生缺页，8ns+20ns+8ns+8ns=44ns

52．（7分）某请求分页系统的页面置换策略如下：(2012全国试题)

从0时刻开始扫描，每隔5个时间单位扫描一轮驻留集（扫描时间忽略不计）且在本轮没有被访问过的页框将被系统回收，并放入到空闲页框链尾，其中内容暂时不清空。当发生缺页时，如果该页曾被使用过且还在空闲页链表中，则将其重新放回该进程的驻留集中；否则从空闲页链表头部取出一个页框。

忽略其他进程的影响和系统开销，初始时进程驻留集为空，目前系统空闲页的页框号为32、15、21、41。进程P依次访问的<虚拟页号，访问时刻>为<1?,1> 、<1,2> 、<0,4>、<0,6>、<1,11>、<0,13>、<2,14>。请回答下列问题：

（1） 当虚拟页号为<0,4>时，对应的页框号是什么？

（2） 当虚拟页号为<1,11>时，对应的页框号是什么？说明理由。

（3） 当虚拟页号为<2,14>时，对应的页框号是什么？说明理由。

（4） 这种方法是否适用时间局部性好的程序？说明理由。

答：(1) 当虚拟页号为<0,4>时，对应的页框号是15；

（2）当虚拟页号为<1,11>时，对应的页框号是32。因为虽然在时刻10扫描时，页框32被回收到链表尾部，但当访问虚拟页号为<1,11>发生缺页时，又将其重新放回进程的驻留集中。

（3）当虚拟页号为<2,14>时，对应的页框号是21。因为在时刻14时，进程的驻留集是<0页(页框15)，1页(页框32)>，访问2号虚拟页发生缺页时，从空闲页框链表头取出的是21号页框。

（4）这种方法适用时间局部性好的程序，因为这种方法对于时间局部性好的程序的页要么不易淘汰，要么即使淘汰了也会很快重新从空闲链表取回，这可以减少启动磁盘次数从而效率高。

第五章 设备管理

1．选择题

1．下述关于设备绝对号和相对号的说法中，正确的是____。
A．操作系统为每台设备确定一个绝对号和一个相对号
B．用户进程申请设备时应该指定绝对号和相对号
C．申请设备时指定绝对号可提高设备的利用率
D．申请设备时指定设备类、相对号使设备分配的灵活性强

2．虚拟设备技术是指用____的技术。
A．共享设备代替独占设备　　B．独占设备代替共享设备
C．共享设备模拟独占设备　　D．独占设备模拟共享设备

3．SPOOL系统克服了_____利用率低的缺点。
A．共享设备　　B．独占设备　　C．主存储器　　D．虚拟设备

4．下列算法中可用于磁盘移臂调度的是____。
A．最短计算时间优先　B．电梯算法　　C．时间片轮转　　D．响应比高者优先

5．用户编写程序时使用的设备与实际使用的设备无关，这种特性称为____。
A．设备一致性　　B．设备独立性　　C．设备虚拟性　　D．设备共享性

6．指定扇区旋转到磁头位置所需的时间称为____时间。
A．寻找　　B．延迟　　C．传送　　D．旋转

7．磁盘是共享设备，每一时刻_____进程与它交换信息。
A．可有任意多个　　B．限定n个　　C．至少有一个　　D．最多有一个

8．硬件采用了中断和通道技术，使得_____。
A．CPU与外设能紧密结合　　B．CPU与外设能并行工作
C．CPU速度提高　　D．外设速度提高

9．通道在输入输出操作完成或出错时，就形成____，等候CPU来处理。
A．硬件故障中断　　B．程序中断　　C．外部中断　　D．I／O中断

10．磁盘是可共享的设备，每一时刻____进程与它交换信息。
A．允许有两个　　B．可以有任意多个　　C．最多有1个　　D．至少有1个

11．对磁盘进行移臂调度时，既考虑了减少寻找时间，又不频繁改变移动臂的移动方向的调度算法是__。
A．先来先服务　　B．最短寻找时间优先
C．电梯调度　　D．优先级高者优先

12．为了提高设备分配的灵活性，用户申请设备时应指定____号。
A．设备类相对　　B．设备类绝对　　C．相对　　D．绝对

13．对磁盘进行移臂调度的目的是为了缩短____时间。
A．寻道　　B．延迟　　C．传送　　D．启动

14．操作系统采用 SPOOLing 技术提高了_____利用率。
A．独占设备　　B．文件　　C．主存储器　　D．共享设备

15．从磁盘读取数据的下列时间中，_____对系统效率的影响最大。

A．处理时间　　B．传输时间　　C．延迟时间　　D．寻道时间

16．通道是一种_____。

A．存储器　　B．控制器　　C．I/O处理器　　D．I/O设备

17．在采用SPOOLing技术的系统中，用户作业的打印结果首先被送到_____。

A．硬盘固定区域　　B．终端　　C．内存固定区　　D．打印机

18．对磁盘进行移臂调度的目的是为了缩短_____。

A．查找定位时间　　B．搜索延迟时间　　C．数据传递时间　　D．启动时间

19．通道在输入输出操作完成或出错时，就形成_____，等候CPU来处理。

A．硬盘故障中断　　B．程序中断　　C．外部中断　　D．I/O中断

20．下列算法可用于磁盘移臂调度的是_____。

A．LRU算法　　B．电梯调度算法

C．时间片轮转法　　D．响应比高者优先算法

21．以下_____不属于磁盘的驱动调度。

A．最高响应比优先调度算法　　B．先来先服务调度算法

C．电梯调度算法　　D．单向扫描调度算法

22．操作系统中的SPOOLing技术，实质上是将_____“转化”为共享设备的技术。

A．临界设备　　B．虚拟设备　　C．脱机设备　　D．块设备

23．关于SPOOLing的叙述中，_____是错误的。

A．SPOOLing系统中不需要独占设备

B．SPOOLing系统加快了作业的执行速度

C．SPOOLing系统使独占设备变成了共享设备

D．SPOOLing利用了处理器与通道的并行工作能力

24．为了使多个进程能有效地同时处理输入和输出，最好使用___结构的缓冲技术。

A．单缓冲区　　B．双缓冲区　　C．多缓冲区环　　D．缓冲池

25．通过硬件和软件的功能扩充，把原来的独占设备“改造”成能为若干用户共享的设备，这种设备称为____ 设备。

A．用户　　B．系统　　C．虚拟　　D．临界

26．以下叙述中，正确的是____ 。

A．在现代计算机系统中，只有I/O设备才是有效的中断源

B．在中断处理过程中必须屏蔽中断

C．同一用户所使用的I/O设备也可能并行工作

D．SPOOLing是脱机I/O系统

27．在操作系统中，____指的是一种硬件机制。

A．缓冲池　　B．通道技术　　C．SPOOLing技术　　D．内存覆盖技术

28．在操作系统中，用户在使用 I/O 设备时，通常采用____ 。

A．物理设备名　　B．虚拟设备名　　C．逻辑设备名　　D．绝对设备号

29．SPOOLing技术提高了____的利用率。

A．独占设备　　B．共享设备　　C．文件　　D．主存储器

30．下列有关设备管理的叙述中，____是错误的。

A．所有外围设备的启动工作由系统同一来做

B．来自通道的I/O中断事件由设备管理负责处理

C．编制好的通道程序是放在主存储器中的

D．由用户给出的设备号是设备的绝对号

31．某操作系统中，采用中断驱动I/O控制方式，设中断时，CPU用1ms来处理I/O中断请求，其他CPU时间全部用来计算。若系统时钟中断频率为100Hz，则CPU的利用率为____%。

A．60　　B．70　　C．80　　D．90

32．基本的I/O设备处理程序一般处于____状态。

A．就绪　　B．执行　　C．阻塞　　D．挂起

33．下述各项中，____不是SPOOLing技术的特点。

A．提高了I/O速度　　B．将独占设备模拟成共享设备

C．采用高速缓存(cache)　　D．实现了虚拟设备功能

34．下述有关设备管理的叙述中，错误的是____。

A．通道是处理I/O的软件

B．所有外围设备的启动工作由系统统一来做

C．由用户给出的设备号是设备的相对号（逻辑设备名）

D．编制好的通道程序是存放在主存储器中的

35．I/O软件一般分为4个层次：用户层、与设备无关软件层、设备驱动程序、中断处理程序。以下工作中，不是由设备驱动程序完成的是_____。

A．向设备寄存器写命令

B．检查用户是否有权使用设备

C．将二进制整数转换成ASCII码以便打印

D．解释用户的I/O请求，并将该请求转化为具体的I/O操作

36．I/O系统硬件结构分为4级：①设备控制器；②I/O设备；③计算机；④I/O通道。按级别由高到低的顺序是_____。

A．②-④-①-③　　B．③-①-④-②　　C．②-①-④-③　　D．③-④-①-②

37．本地用户通过键盘登录系统时，首先获得键盘输入信息的程序是_____。（2010全国试题）

A．命令解释程序　　B．中断处理程序

C．系统调用服务程序　　D．用户登录程序

38．单处理机系统中，能并行的是_____。（2009全国试题）

I．进程与进程 II．处理机与设备 III．处理机与通道 IV．设备与设备

A．I、II和III　B．I、II和IV　C．I、III和IV　D．II、III和IV

39．假设磁头当前位于 105 道，正在向磁道号增加的方向移动。现有一个磁道访问请求序列为 35, 45, 12, 68, 110, 180, 170, 195，采用 SCAN 调度（电梯调度）算法得到的磁道访问序列是_____。（2009 全国试题）

A．110,170,180,195,68,45,35,12

B．110,68,45,35,12,170,180,195

C．110,170,180,195,12,35,45,68

D．12,35,45,68,110,170,180,195

40．程序员利用系统调用打开I/O设备时，通常使用的设备标识符是_____。（2009全国试题）

A．逻辑设备名　　B．物理设备名　　C．主设备号　　D．从设备号

41．用户程序发出磁盘I/O请求后，系统的正确处理流程是____。（2011全国试题）

A．用户程序→系统调用处理程序→中断处理程序→设备驱动程序

B．用户程序→系统调用处理程序→设备驱动程序→中断处理程序

C．用户程序→设备驱动程序→系统调用处理程序→中断处理程序

D．用户程序→设备驱动程序→中断处理程序→系统调用处理程序

42．某文件占10个磁盘块，现要把该文件磁盘块逐个读入主存缓冲区，并送用户区进行分析。假设一个缓冲区与一个磁盘块大小形同，把一个磁盘块读入缓冲区的时间为100μs，将缓冲区的数据传送到用户区的时间是50μs，CPU对一块数据进行分析的时间为50μs。在单缓冲区和双缓冲区结构下，读入并分析万该文件的时间分别是_____。（2011全国试题）

A．1500μs、1000μs　　B．1550μs、1100μs

C．1550μs、1550μs　　D．2000μs、2000μs

43．操作系统的I/O子系统通常由四个层次组成，每一层明确定义了与邻层的接口，其合理的层次组织排列次序是_____。（2012全国试题）

A．用户级I/O软件、设备无关软件、设备驱动程序、中断处理程序

B．用户级I/O软件、设备无关软件、中断处理程序、设备驱动程序

C．用户级I/O软件、设备驱动程序、设备无关软件、中断处理程序

D．用户级I/O软件、中断处理程序、设备无关软件、设备驱动程序

44．下列选项中，不能改善磁盘I/O性能的是_____。（2012全国试题）

A．重排I/O请求次序　　B．在一个磁盘设置多个分区

C．预读和滞后写　　D．优化文件物理的分布

第五章设备管理选择题参考答案：

1．D	2．C	3．B	4．B	5．B	6．B	7．D	8．B	9．D	10．C
11．C	12．A	13．A	14．A	15．D	16．C	17．A	18．A	19．D	20．B
21．A	22．A	23．B	24．D	25．C	26．D	27．B	28．C	29．A	30．D
31．D	32．C	33．C	34．A	35．C	36．D	37．B	38．D	39．A	40．A
41．B	42．B	43．A	44．B						

2．应用题

1．假设有一磁盘含有 64000 块，块号记为 1～64000，现用 2000 个 32 位(Bit)的字作该盘的位示图，试问第 59999 块对应于位示图中第几字的第几位(字、位均从 0 开始)；而第 1599 字的第 17 位对应于磁盘的第几块?

解：由块号 b，求字号 i 和位号 j 的公式为：

i=(b-1) div 32　　(div 表示整数除法，32 是字长)

j=(b-1) mod 32　　(mod 表示整数相除取余数)

(59999-1) div 32=1874　　(59999-1) mod 32=30

故 59999 块对应于位示图中第 1874 字的第 30 位。

由位示图的字号 i 和位号 j，求对应的磁盘块号 b 的公式为：

b=i×32+j+1

1599×32+17+1=51186

即第 1599 字的第 17 位对应于磁盘的第 51186 块。第 1599 字的第 17 位对应于磁盘的第几块

2．若干个等待访问磁盘者依次要访问的柱面为 20，44，40，4，80，12，76，假设每移动一个柱面需要 3 毫秒时间，移动臂当前位于 40 号柱面，请按下列算法分别计算为完成上述各次访问总共花费的寻找时间。

（1）先来先服务算法；

（2）最短寻找时间优先算法。

解：（1）采用先来先服务算法

服务顺序为20，44，40，4，80，12，76

总寻道长度=20+24+4+36+76+68+64=292

即总寻道时间=292×3=876(ms)

（2）最短寻找时间优先算法

调度顺序为：40→40→44→20→12→4→76→80

总寻道长度=0+4+24+8+8+72+4=120

总寻道时间=120×3=360(ms)

3．假定一个磁盘共有 100 个柱面，每个柱面上有 4 个磁道，每个盘面分成 16 个扇区。如果内存的字长为 64 位。磁盘地址中指出的柱面号、磁道号、扇区号和块号只需要 64 位二进制位即可表示。如果每个磁盘块的长度是 512 字节。记录磁盘中空闲块有两种方式，即位示图法和空闲块链接法。若采用空闲块链接法中的成组链接方案，在该方案涉及的每一块中，记录空闲块数需用 4 个字节，记录每个磁盘地址需用 4 个字节。请问：

（1）需要用多少内存字来存储关于磁盘空间的位示图。

（2）如果把上述的每一种方法为记录磁盘空闲位置所占用的内存和磁盘空间加起来，就算作这种方案占用存储空间的总数。请用精确的数字说明，在什么情况下，成组链接方案占用的存储空间总数小于位示图法占用的存储空间总数。

答：（1）磁盘空间的总块数=100×4×16=6400

内存字数=6400÷64=100

需用 100 个内存字来存储磁盘空间的位示图。

（2）空闲块数和磁盘地址都用4个字节记录，考虑磁盘空间盘块总数变化的情况。

成组链接方案需要一个专用块，512字节

专用块内容调入内存需占用的字节数为100×4+4=404

即成组链接方案占用存储空间总数为512+404=916字节。

下面计算占用存储空间总数为916字节时，采用位示图方案时磁盘的最大块数：

用1个块存位示图，占512字节磁盘空间；

位示图调入内存后所占用的空间应不超过(916-512)=404个字节，即位示图应不超过404÷8 = 50.5（字），即位示图为50个内存字，50×64 = 3200（块）。

故当空闲块数和磁盘地址都用4个字节且磁盘块数大于3200时，成组链接方案占用的存储空间总数小于位示图法占用的存储空间总数。

4．某移动臂磁盘的柱面由外向里从 0 开始顺序编号，假定当前磁头停在 100 号柱面而且移动方向是向外的，现有一个请求队列在等待访问磁盘，访问的柱面号分别为 190、10、160、80、90、125、30、20、140 和 25。请写出分别采用最短寻找时间优先和电梯调度算法处理上述请求的次序。

5．假设有一磁盘有 6400 块，每块长度为 1024 字节，块号记作 1—6400，现用 400 个 16 位（Bit）的字作该磁盘的位示图，试问第 2999 块对应于位示图中的第几字的第几位（字、位均从 0 开始计）；而第 299 字的第 7 位（同上，从 0 开始）又对应第几块？

6．某系统采用位示图法实现磁盘空间管理，现有一磁盘有 10000 个物理块，位示图的每个字有 32 位，试问：

（1）需要有多少个字？

（2）计算第i个字第j位对应的物理块号（设字号和位号都是从0开始编号）。

（3）求物理块号N对应的字号和位号。

7．假定在某移动臂磁盘上，刚刚处理了访问 143 号柱面的请求，目前正在为访问 125 号柱面的请求服务，同时有若干请求者在等待服务，它们依次访问的柱面号为

86，147，91，177，94，150，102，175，130

请回答下列问题：

（1） 分别写出用先来先服务算法、最短寻找时间优先算法、电梯算法的实际服务次序。

（2） 计算上述算法下移动臂需移动的距离。

8．若递交给磁盘驱动程序的磁盘柱面请求按到达时间顺序分别是 10、22、20、2、40、6 和 38，设磁头初始处于 20 柱面，磁头从一柱面移到另一相邻柱面的时间是 6ms，则对于 FCFS、最近柱面优先、电梯算法（初始磁头向高柱面移动），平均定位时间各为多少？

9．假定一个磁盘组共有 100 个柱面，每个柱面上有 4 个磁道，每个盘面分成 16 个扇区。扇区的容量与磁盘块的容量相等。用位示图法记录磁盘中的各个块是否已经被占用。设内存的字长为 64 位。这里涉及的所有编号，例如柱面号、磁道号、扇区号和块号，以及位示图中的内存字的次序和二进制位等都是从 0 开始编号。请问：

（1）位示图中的第 50 个字的第 20 个二进制位对应的是什么磁盘地址？

（2）第 99 号柱面的第 3 号磁道的第 15 号扇区在位示图中对应第几个字中的第几位？

解：由柱面号、磁道号(磁头号)、扇区号求对应的磁盘逻辑块号的公式为：

块号=柱面号×每柱面扇区数+磁道号×每磁道扇区数+扇区号

=柱面号×每柱面磁道数×每磁道扇区数+磁道号×每磁道扇区数+扇区号

=柱面号×4×16+磁道号×16+扇区号

由磁盘块号求对应的柱面号、磁道号(磁头号)、扇区号的公式分别为：

柱面号=块号 div (每柱面磁道数×每磁道扇区数)= 块号 div 64

磁道号=[块号 mod (每柱面磁道数×每磁道扇区数)]div 每磁道扇区数

=(块号 mod 64) div 16

扇区号=块号 mod 每磁道扇区数=块号 mod 16

(1) 块号=字号×字长+位号=50×64+20=3220

对应的磁盘地址为：

柱面号=块号 div 64=50

磁道号=(块号 mod 64) div 16=(3220 mod 64) div 16=20 div 6=1

扇区号=块号 mod 16=3220 mod 16=4

即位示图中的第 50 个字的第 20 个二进制位对应 50 号柱面 1 号磁道的 4 扇区。

(2) 块号=柱面号×4×16+磁道号×16+扇区号=99×4×16+3×16+15=6399

字号=块号 div 字长=6399 div 64=99

位号=块号 mod 字长=6399 mod 64=63

即第 99 号柱面的第 3 号磁道的第 15 号扇区在位示图中对应第 99 个字中的第 63 位。

10．一台转速为 3600（转分）的磁盘，其存储密度为 16.7（K/道）。已知磁盘由启动到运转平稳的时间为 3ms，磁头臂的移动速度为 0.3（ms/道），请回答：

（1）设磁头的当前位置在第 20 号磁道上，移动方向为磁道号增加的方向。若系统收到 4 条记录访问请求，请求序列如下表所示。

记录号	磁道号
1	18
2	25
3	32
4	7

请写出电梯调度算法的访问序列。

（2）若上述 4 条记录的长度皆为 16.7KB，求系统按电梯调度算法访问磁盘，上述 4 条记录的最长时间为多少？(计算时间时保留 2 位小数)

解：（1）采用电梯调度算法的访问序列为：（20）→25→32→18→7。

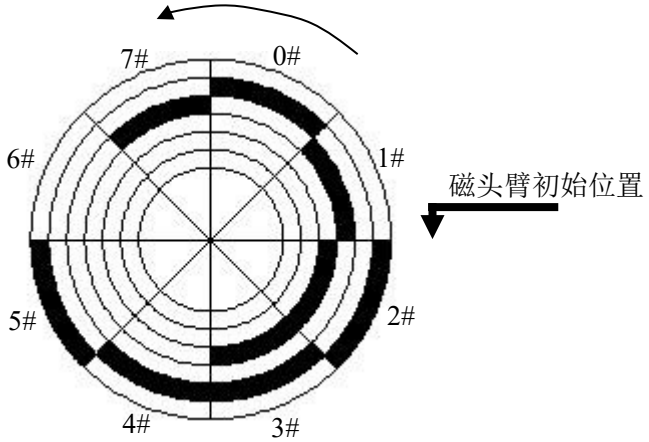
（2）访问2号记录：

移臂时间T2=0.3*5=1.5ms

最大旋转一周时间R2=（60*1000）/3600=50/3ms=16.67 ms
数据传输时间A2=旋转一周时间=16.67 ms
3项时间合计为34.84ms。
访问3号记录：
移臂时间T3=0.3*7=2.1ms
最大旋转一周时间R3=（60*1000）/3600=50/3ms=16.67 ms
数据传输时间A3=旋转一周时间=16.67 ms
3项时间合计为35.44ms。
访问1号记录
移臂时间T1=0.3*14=4.2ms
最大旋转一周时间R1=（60*1000）/3600=50/3ms=16.67 ms
数据传输时间A1=旋转一周时间=16.67 ms
3项时间合计为37.54ms。
访问4号记录
移臂时间T4=0.3*11=3.3ms
最大旋转一周时间R4=（60*1000）/3600=50/3ms=16.67 ms
数据传输时间A4=旋转一周时间=16.67 ms
3项时间合计为36.64ms。
所以，总的访问时间为：3+34.84+35.44+37.54+36.64=147.46ms

11．某系统的磁盘有 65535 个磁道，每个磁道有 255 个扇区。已知该磁盘的转速为 7200 转/分钟，磁头每移动 1 个磁道需 0.5ms，假设磁头当前处于第 20 磁道上，有一用户进程请求访问第 220 磁道的第 3 扇区并立即被系统响应，磁头到达第 220 磁道时恰好处于第 1 扇区的开始位置，试计算该用户进程至少需要等待多长时间才能开始读入所需数据？

12．一个假想的磁盘有 6 个磁道，0#磁道在最外面，其它依次向内排列。每个磁道有 8 个扇区，扇区的排列为：0#扇区，1#扇区，…，7#扇区，如下图所示。
设有一个文件 File 存储在磁盘的 9 个扇区中，每个扇区存 1 个记录。存储情况如图中黑色部分所示。



假定磁头臂移动一个磁道所需时间和磁盘旋转一个扇区所需的时间以及磁头臂从初始位置移动到 0#磁道所需的时间皆为 0.3ms。磁盘的旋转方向及磁头臂的初始位置如图中所示。设磁头臂移动算法采用电梯调度算法，请说明利用该算法读取上述文件时，各个扇区的访问顺序及磁头臂总共移动次数。

解：各个扇区的访问顺序及磁头臂移动情况为：
磁头臂从初始位置移动到0#磁道（第1次移动）
0#磁道，访问5#扇区，再访问2#扇区

磁头臂移动到1#磁道（第2次移动）
1#磁道，访问4#扇区，再访问0#扇区，最后访问3#扇区
磁头臂移动到2#磁道（第3次移动）
2#磁道，访问7#扇区，再访问1#扇区
磁头臂移动到3#磁道（第4次移动）
3#磁道，访问3#扇区，再访问2#扇区
因此，各个扇区的访问顺序为0道5扇区、0道2扇区、1道4扇区、1道0扇区、1道3扇区、2道7扇区、2道1扇区、3道3扇区、3道2扇区。磁头臂总共移动4次。

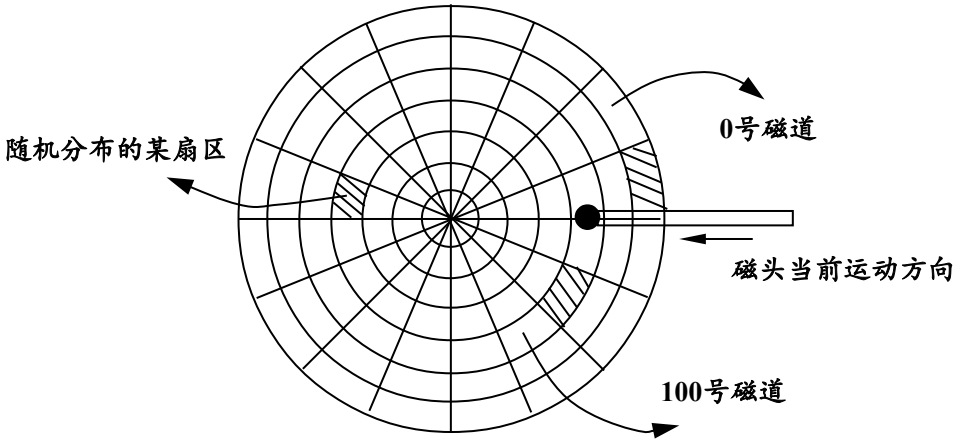
13．在某系统中，数据从磁盘读入缓冲区，然后从缓冲区传入用户区，再在用户区中处理。假设在该磁盘系统中，文件在磁道上非连续存放，磁头从一个磁道移动到相邻磁道需要时间 t1，逻辑上相邻数据块的平均距离为 d 磁道，每块的旋转延迟时间及传输到缓冲区的传输时间分别为 t2 和 t3。问读取 N 个数据块的磁盘访问时间一共是多少？另外，假设将缓冲区数据传送到用户区所花的时间为 t4，且 t4 远远小于读取一个数据块的磁盘访问时间，CPU 对一块数据进行处理的时间为 t5。问分别在单缓冲和双缓冲情况下，平均处理一块数据的总时间为多少？

解：（1）为了读取一数据块，平均需要花费的磁头移动时间为 d×t1，需要等待的旋转延迟时间为 t2，需要传输的时间为 t3。因此读取一块的时间为 d×t1+t2+t3，由此可得，访问 N 块的总时间 t0 为：
t0= N×(d×t1+t2+t3)
（2）在单缓冲情况下，平均处理一块数据的时间为：Max(t0,t5)+t4
（3）在双缓冲情况下，平均处理一块数据的时间为：Max(t0,t5)

14．假定一个磁盘组共有 65536 个柱面，每个柱面上有 16 个磁道（即有 16 个读写磁头），每个盘面分成 256 个扇区。问：
（1）整个磁盘空间共有多少个存储块（扇区）？
（2）如果用字长为32位的单元来构造位示图，共需多少个字？
（3）位示图中18字，16位对应的块号是多少？（字号和位号皆从0开始编号，块号从1开始编号）
（4）由（2）的计算结果分析，你认为大容量磁盘空间的管理，是否适宜采用位示图？

解：(1) 65536×16×256=268435456
即整个磁盘空间共有268435456个存储块（扇区）
(2) 268435456÷32=8388608
即用字长为32位的单元来构造位示图，共需8388608个字。(即32MB)
(3) 块号=字号×字长+位号+1=18×32+16+1=593
(4) 由（2）可知，构造位示图需要8388608个字，即32MB，占用磁盘空间相当大，系统启动后，位示图调入内存后占用内存空间相当大，查找如此巨大的位示图，效率页不高。由此可知，大容量磁盘空间的管理不宜采用位示图。

15．(2010 全国试题)假设计算机系统采用 CSCAN(循环扫描)磁盘调度策略，使用 2KB 的内存空间记录 16384 个磁盘块的空闲状态。
（1）请说明在上述条件下如何进行磁盘块空闲状态的管理。
（2）设某单面磁盘旋转速度为每分钟 6000 转，每个磁道有 100 个扇区，相邻磁道间的平均移动时间为 1ms。若在某时刻，磁头位于 100 号磁道处，并沿着磁道号增大的方向移动(如下图所示)，磁道号请求队列为 50,90,30,120，对请求队列中的每一个磁道需读取 1 个随机分布的扇区，则读完这 4 个扇区总共需要多少时间？给出计算过程。



（3）如果将磁盘替换为随机访问的 Flash 半导体存储器(如 U 盘、SSD 等)，是否有比 CSCAN 更高效的磁盘调度策略？若有，给出磁盘调度策略的名称并说明理由；若无，说明理由。

答案要点：

(1) 用位图表示磁盘的空闲块状态。每一位表示一个磁盘块的空闲状态，共需16384/32= 512个字=512×4个字节=2KB，正好可放在系统提供的内存中。

(2) 采用CSCAN调度算法，访问磁道的顺序为120,30,50,90，则移动磁道长度为20+90+20+40=170，总的移动时间为170×1ms=170ms。

由于转速为6000r/m，则平均旋转延迟时间为60/(6000×2)×1000ms=5ms，总的旋转时间为5ms×4=20ms。

由于转速为6000r/m，则读取一个磁道上的一个扇区的平均读取时间为10ms/100=0.1ms，总的读取扇区的时间=0.1ms×4=0.4ms。

读取上述磁道上的所有4个扇区所花费的总时间=170ms+20ms+0.4ms=190.4ms。

(3) 采用FCFS(先来先服务)调度策略更高效。因为Flash半导体存储器的物理结构不需要考虑寻道时间和旋转延迟，可直接按I/O请求的先后顺序服务。

16．假设有一磁盘含有 64000 块，块号为 1～64000，现用 2000 个 32 位(Bit)的字作该磁盘的位示图，该位视图的字号和位号均从 0 开始编号，试问：

(1) 第32001块对应于位示图中的字号和位号各是多少?

(2) 位视图中，字号为1600、位号为19的位，对应于磁盘的第几块?

(3) 若字号、位号也从1开始编号，写出由块号b求字号i和位号j的计算表达式。

解：(1) i = (32001-1) /32 = 1000 j = (32001-1) mod 32 = 0故第32001块对应于位示图中的字号和位号分别是1000和0

(2) b = 1600*32 + 19 +1 = 51220

位视图中，字号为1600、位号为19的位，对应于磁盘的第51220块。

(3) 若块号、字号、位号均从1开始编号，由块号b求字号i和位号j的计算表达式如下：

i = (b – 1) DIV 32 +1

j = (b – 1) MOD 32 + 1

【注】对于(3)，不少学生做法如下：

由公式　b-1 = (i - 1)×32 + j - 1

即　b = (i – 1)×32 + j

故　i = b DIV 32 + 1

j = b MOD 32

上述推算是错误的，原因是式子b = (i – 1)×32 + j中不能保证j一定小于32。

17．设磁盘组共有 n 个柱面，编号顺序为 0、1、2、. . . 、n-1；共有 m 个磁头，编号顺序为 0、1、2、. . . 、m-

1；每个磁道内的 k 个信息块从 1 开始编号，依次为 1、2、. . . 、k。先用 x 表示逻辑磁盘块号，用 a、b、c 分别表示任一逻辑块的柱面号、磁头号、磁道内块号，则 x 与 a、b、c 可通过如下公式进行转换：

x = k×m×a+k×b+c

a = (x-1) DIV (k×m)

b = ((x-1) MOD (k×m)) DIV k

c = ((x-1) MOD (k×m)) MOD k +1 或 c= (x-1) MOD k +1

若某磁盘组为n=200，m=20，k=10，问：

(1) 柱面号为185，磁头号为12，道内块号为5的磁盘块的逻辑块号为多少？

(2) 逻辑盘块号为1200，它所对应的柱面号、磁头号及磁道内块号为多少？

(3) 若每磁道内的信息块从0开始编号，依次为0、1、2、...、k-1，其余均同题设，试写出x与a、b、c之间的转换公式。

第六章 文件管理

1．选择题

1．逻辑文件的组织结构是由___确定的。
A．操作系统　　B．存储容量　　C．用户　　D．文件长度
2．采用树形目录结构后，不同用户对同一个文件定义的文件名_____。
A．应该相同　　B．不能相同　　C．可以不同　　D．应该不同
3．UNIX文件系统对盘空间的管理采用___。
A．FAT表法　　B．位示图法
C．空闲块链接法　　D．空闲块成组链接法
4．关于多级目录结构的论述，错误的说法是___。
A．便于文件分类
B．查找速度快
C．同一子目录下可以建立同名文件
D．可以实现文件的连接
5．文件系统采用多级目录结构可以___。
A．节省存储空间　　B．解决命名冲突
C．缩短文件传送时间　　D．减少系统开销
6．在有关文件管理的下述叙述中，___是正确的。
A．“在二级目录结构中，不同用户不能用相同的文件名”
B．“逻辑记录的大小与存储介质分块的大小必须一致”
C．“文件系统主要是实现按名存取”
D．“在一级目录结构中，不同用户可以用相同的文件名”
7．为了防止用户共享文件时造成破坏，可以采用_____。
A．对文件设置口令　　B．把文件译成密码
C．对文件加锁　　D．对文件的访问权限进程控制
8．文件系统中文件被按照名字存取是为了___。
A．方便操作系统对信息的管理　　B．方便用户的使用
C．确定文件的存取权限　　D．加强对文件内容的保密
9．系统在接到用户关于文件的___操作命令后，就在文件目录中寻找空目录项进行登记。
A．建立　　B．打开　　C．读　　D．写
10．文件系统与___密切相关，它们共同为用户使用文件提供方便。
A．处理器管理　　B．存储管理
C．设备管理　　D．作业管理
11．如果允许不同用户的文件可以具有相同的文件名，通常采用___来保证按名存取的安全。
A．重名翻译机构　　B．建立索引表
C．建立指针　　D．多级目录结构
12．对记录式文件，操作系统为用户存取文件信息的最小单位是___。
A．字符　　B．数据项　　C．记录　　D．文件

13．对一个文件的访问，常由___共同限制。

A．用户访问权限和文件属性　　B．用户访问权限和用户优先级

C．用户优先级和文件属性　　D．文件属性和口令

14．UNIX系统中，文件存贮器的管理采用的是：___。

A．位图法　　B．空闲块表法　　C．成组连接法　　D．单块连接法

15．逻辑文件存放在到存储介质上时，采用的组织形式是与___有关的。

A．逻辑文件结构　　B．存储介质特性

C．主存储器管理方式　　D．分配外设方式

16．采用直接存取（随机存取）方法来读写磁盘上的物理记录时，效率最低的是_____。

A．连续结构文件　　B．索引结构文件

C．隐式链接结构文件　　D．显式链接结构文件

17．为解决文件重名问题，操作系统的文件系统必须采用_____目录。

A．分段　　B．二级或多级

C．分块存取　　D．标识名

18．文件系统中，索引文件结构中的索引表是用来_____。

A．指示逻辑记录逻辑地址的

B．存放部分数据信息的

C．存放查找关键字项内容的

D．指示逻辑记录和物理块之间对应关系的

19．为了保证文件未经文件主授权，任何其他用户均不得使用该文件，操作系统提供的解决方法为_____。

A．文件复制　　B．文件共享

C．文件保密　　D．文件保护(即文件访问控制)

20．Windows 98 的文件目录（文件夹）采用_____结构。

A．单级目录　　B．二级目录

C．三级目录　　D．树型目录

21．按文件的物理组织结构可将文件分成_____等。

A．数据文件，命令文件，文本文件　　B．命令文件，库文件，索引文件

C．连续文件，链式文件，索引文件　　D．输入文件，输出文件，随机文件

22．在UNIX中文件的物理结构是_____分配方式。

A．顺序　　B．链接　　C．索引　　D．索引顺序

23．从用户观点看，文件系统的主要目的是_____。

A．实现对文件的按名存取　　B．实现虚拟存储

C．提高外存的读写速度　　D．用于存储系统文件

24．UNIX文件系统对磁盘空间的管理采用_____。

A．FAT表法　　B．位示图法　　C．空闲块链接法　　D．空闲块成组链接法

25．逻辑文件必须存放在连续存储空间中的存储结构有_____结构。

A．链接　　B．顺序　　C．索引　　D．流式

26．以下_____不是磁盘存储空间的常用管理方法。

A．位示图　　B．记录的成组操作　　C．空闲块表　　D．空闲块链

27．UNIX系统磁盘存储空间的管理采用_____的管理方法。

A．位示图　　B．记录的成组操作　　C．空闲块表　　D．空闲块成组链接

28．某操作系统的文件系统中，采用3个字节表示磁盘块号，每个磁盘块大小为512字节。该系统中每个（逻辑）磁盘允许的最大容量是____字节。

A．2G　　B．4G　　C．8G　　D．16G

29．若采用位示图（100行，32列）表示磁盘块的使用状态。当分配一个盘块号133号时，其在位示图中的行、列数为_____。（注：行号0～99，列为0～31，首盘块号为0）

A．4和5　　B．5和3　　C．4和3　　D．5和4

30．下列文件中属于逻辑结构的文件是_____。

A．连续文件　　B．系统文件　　C．目录文件　　D．流式文件

31．不包含在文件控制块（又称文件目录项）中的信息是 _____ 。

A．存储介质　　B．文件名

C．存取控制信息　　D．文件的物理结构

32．为了对文件系统中的文件进行安全管理，任何一个用户在进入系统时都必须进行注册，这一级安全管理是____级安全管理。

A．系统　　B．目录　　C．用户　　D．文件

33．位示图可用于____ 。

A．文件目录的查找　　B．磁盘空间的管理

C．主存空间的共享　　D．实现文件的保护和保密

34．在文件的物理结构中，____结构不利于文件长度的动态增长。

A．顺序　　B．链接　　C．索引　　D．Hash

35．UNIX 系统中，____文件用于把一个进程的输出连接到另一个进程的输入。

A．普通　　B．特殊　　C．目录　　D．管道

36．UNIX 系统的多用户环境下，对每个文件设置了____三种权限，从而加强了文件的保密性和安全性。

A．文件的系统、隐含和私有　　B．文件的所有者、同组用户及其他人

C．读、写及执行　　D．读、写、执行及复制

37．下列关于 UNIX 的叙述中，____是不正确的。

A．UNIX是一个多道的分时系统

B．PIPE机制是UNIX的贡献之一

C．提供可动态装卸的文件卷是UNIX的特色之一

D．路径名是UNIX独有的实现文件共享的机制

38．位示图方法可用于____。

A．进程的调度　　B．盘空间的管理

C．文件的共享　　D．进程间通讯

39．下列选项中，____不是删除文件所需要完成的工作。

A．释放文件所占用的存储空间

B．对文件原占用的存储单元全部清零

C．删除该文件的目录项，即文件控制块(FCB)

D．若文件为共享文件，还要对共享设置进行处理

40．在有随机(直接)存取需求和允许文件动态增长的情况下，宜选择____文件形式。

A．顺序　　B．链接　　C．索引　　D．记录式

41．下列对于索引文件的描述中，错误的是____。

A．索引文件和主文件配合使用

B．使用索引文件是为了加快对主文件的检索速度

C．索引文件和顺序文件没有什么联系

D．可以说利用索引文件，是空间换取时间

42．操作系统中对目录管理的主要要求，不包括____。

A．对文件实现按名存取　　B．节省文件存储空间

C．提高对目录的检索速度　　D．允许文件重名

43．以下关于文件组织结构的说法中，错误的是____。

A．文件组织从用户和文件系统的不同角度出发分为逻辑文件和物理文件
B．逻辑文件是用户概念中的文件，分为流式文件和记录式文件
C．磁带文件的物理组织方式一般可以采用顺序结构或链接结构
D．磁盘文件的物理组织方式一般可以采用顺序结构、链接结构或索引结构等

44．某系统中，一个FCB占用32B，盘块大小为1KB，文件目录中共有3200个FCB，查找该目录中的一个文件，平均启动磁盘次数为____。
A．50　　B．64　　C．100　　D．200

45．下列各项描述中，不是树型目录优点的是____。
A．解决了文件重名问题　　B．提高了文件检索速度
C．根目录到指定文件有多条路径　　D．便于进行存储权限控制

46．在UNIX系统V中，如果一个盘块的大小为1KB，每个盘号占4个字节，那么，一个进程要访问某文件中偏移量为23456789字节处的数据时，需要经过____ 。
A．直接寻址（相当于一级索引）　　B．一次间址（相当于二级索引）
C．二次间址（相当于三级索引）　　D．三次间址（相当于四级索引）

47．文件系统采用多级目录结构可以_____。
A．节省存储空间　　B．解决命名冲突
C．缩短文件传送时间　D．减少系统开销

48．在有关文件管理的下述叙述中，_____是正确的。
A．“在一级目录结构中，不同用户可以用相同的文件名”
B．“在二级目录结构中，不同用户不能用相同的文件名”
C．“逻辑记录的大小与存储介质分块的大小必须一致”
D．“从用户的观点看，文件系统主要功能是实现按名存取”

49．在下述存储管理方案中，_____管理方式要求作业的逻辑地址与占有主存的存储区域都是连续的。
A．段页式　　B．页式　　C．段式　　D．可变分区

50．设文件 F1 当前引用计数值为 1，先建立 F1 的符号链接(软链接)文件 F2，再建立 F1 的硬软链接文件 F3，然后删除 F1。此时，F2 和 F3 的引用计数值分别是_____。（2009 全国试题）
A．0．1　　B．1．1　　C．1．2　　D．2．1

51．下面是关于文件的一些操作。若需要读一个文件，那么描述次序正确的是____。
① 将文件的目录信息读入内存
② 向设备管理程序发出I/O请求，完成数据读入操作
③ 指出文件在外存上的存储位置，并进行文件逻辑块号到屋里块号的转换
④ 按存取控制说明检查访问的合法性
⑤ 按文件名从用户打开文件表找到该文件的文件目录项
A．⑤③②④①　　B．①⑤④③②　　C．④①⑤③②　　D．⑤①④③②

52．在UNIX System V中，如果一个盘块的大小为1KB，每个盘块号占4B，那么，一个进程要读取某文件中偏移量为12345678B处的数据时，需要启动____次磁盘（执行读文件操作时，假设该文件的索引结点已经在内存中）。
A．1　　B．2　　C．3　　D．4

53．在UNIX System V中，如果一个盘块的大小为1KB，每个盘块号占4B，那么，该系统中允许的文件最大长度约为____。
A．1GB　　B．16GB　　C．256GB　　D．4TB

54．如果利用200行，30列的位示图来标志盘块的使用情况，在进行盘块分配时，当第一次找到的空闲盘块（即该位为0）处于11行，18列，则相应的盘块号为____。假设行号、列号、盘块号皆从0开始编号。
A．330　　B．348　　C．318　　D．300

55．设文件索引节点中有7个地址项，其中4个地址项是直接地址索引，2个地址项是一级间接地址索引，1个地址项是二级间接地址索引，每个地址项大小为4字节。若磁盘索引块和磁盘数据块大小均为256字节，则可表示的单个文件最大长度是_____。（2010全国试题）

A．33KB　　B．519KB　　C．1057KB　　D．16513KB

56．设置当前工作目录的主要目的是_____。（2010全国试题）

A．节省外存空间　　B．节省内存空间

C．加快文件的检索速度　　D．加快文件的读/写速度

57．下列文件物理结构中，适合随机访问且易于文件扩展的是_____。（2009 全国试题）

A．连续结构　　B．索引结构

C．链式结构且磁盘块定长　　D．链式结构且磁盘块变长

58．文件系统中，文件访问控制信息存储的合理位置是_____。（2009 全国试题）

A．文件控制块　　B．文件分配表

C．用户口令表　　D．系统注册表

59．若一个用户进程通过read系统调用读某个文件，则下列关于read系统调用过程的叙述中，正确的是_。（2012全国试题）（原题不完整，题中蓝字是后加的）

Ⅰ．若该文件的数据不在内存，则该进程进入睡眠等待状态

Ⅱ．请求read系统调用会导致CPU从用户态切换到核心态

Ⅲ．read系统调用的参数应包含文件的名称

A．仅Ⅰ、Ⅱ　　B．仅Ⅰ、Ⅲ　　C．仅Ⅱ、Ⅲ　　D．Ⅰ、Ⅱ和Ⅲ

第六章文件管理选择题参考答案：

1．C	2．C	3．D	4．C	5．B	6．C	7．D	8．B	9．A	10．C
11．D	12．C	13．A	14．C	15．B	16．C	17．B	18．D	19．D	20．D
21．C	22．C	23．A	24．D	25．B	26．B	27．D	28．C	29．A	30．D
31．A	32．A	33．B	34．A	35．D	36．C	37．D	38．B	39．B	40．C
41．C	42．B	43．C	44．A	45．C	46．C	47．B	48．D	49．D	50．B
51．C	52．C	53．B	54．B	55．C	56．C	57．B	58．A	59．C	

2．应用题

【注】题号上带有“*”的题目不作要求。

1*．存放在磁盘上的文件以链接结构组织，假定磁盘的分块大小为每块 512 字节，而文件的逻辑记录的大小为每个记录 250 字节。现有一个文件共有 10 个逻辑记录，请回答：

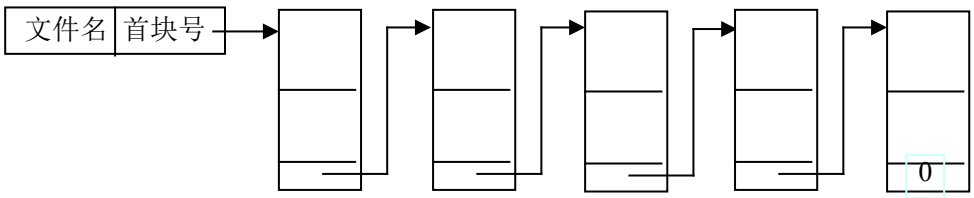
(1)采用成组方式存放该文件时，几个逻辑记录为一组较合适？

(2)画出成组时的链接结构示意图。

(3)当主存缓冲区大小为512个字节时，要读出第7个逻辑记录应启动磁盘多少次？

答：(1) 采用成组方式存放该文件时，2 个逻辑记录为一组较合适。

(2)



(3) 因每个盘块中存 2 个记录，第 7 个逻辑记录存储在第 4 个盘块中，故要读出第 7 个逻辑记录应启动磁盘 4 次。

2．假定磁盘转速为 20 毫秒/周，每个盘面被分成四个扇区，今有 4 个逻辑记录被存放在同一磁道上，每个逻辑记录占一个扇区（如图 6-1）。现有四个请求访问者，他们的请求次序和要求如下表：

请求次序	要求
1	读记录4
2	读记录3
3	读记录2
4	读记录1

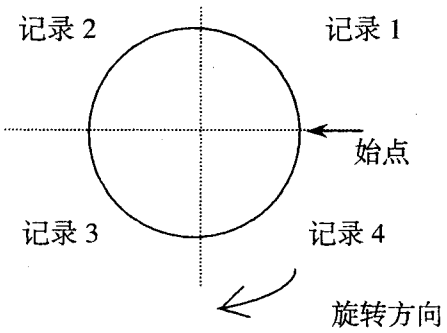


图6-1

设有足够内存缓冲。

回答下列问题：

（1）如果磁头当前在始点位置，按请求次序依次读记录，那么读出这四个记录总共需要多少毫秒？

（2）若对四个请求访问者重新排序，按读记录1，记录2，记录3，记录4次序执行，磁头的当前位置仍在始点，那么读出这四个记录总共需要多少毫秒？

（3）如果当前磁头在记录3开始的位置，那么读出这四个记录最省时间的次序是什么？总共需要多少毫秒？

解：(1) 按请求次序，即依次读记录4、3、2、1，正好是磁盘旋转1周，即总共需要20ms。

(2) 读记录1，需要20ms；

读记录2，需要15ms；

读记录3，需要15ms；

读记录4，需要15ms；

故总共需要65ms。

(3) 读出这四个记录最省时间的次序是3，4，1，2；总共需要20ms。

3*．某用户文件共 10 个逻辑记录，每个逻辑记录的长度为 480 个字符，现把该文件存放到磁带上，若磁带的记录密度为 800 字符/英寸，块与块之间的间隙为 0.6 英寸，回答下列问题：

(1)不采用记录成组操作时磁空间的利用率为_____。

(2)采用记录成组操作且块因子为5时，磁带空间的利用率为_____。

(3)当按上述方式把文件存放到磁带上后，用户要求每次读一个逻辑记录存放到他的工作区。当对该记录处理后，又要求把下一个逻辑记录读入他的工作区，直至 10 个逻辑记录处理结束。系统应如何为用户服务?

答：（1）480/800=0.6(英寸)

0.6/(0.6+0.6)=0.5=50%

即利用率为 50%

（2）480*5=2400 (字符)

2400/800=3(英寸)

3/(3+0.6)=5/6=83%

即利用率为 83%

（3）步骤如下：

① 设置长度为 2400 字符的主存缓冲区；

② 找到该文件的存放位置，启动磁带机读出第一块内容传送到主存缓冲区；

③ 进行记录分解，按用户要求依次把主存缓冲区中的 5 个记录传送到用户工作区；

④ 启动磁导计读入第二块内容到主存缓冲区，分解记录，依次将第 6 至 10 个逻辑记录按用户要求依次送到用户工作区。

4*．假定有一个简单的文件系统，某文件以顺序结构存放在磁盘上。该文件有 10 个等长的逻辑记录组成，每个逻辑记录的长度为 512 个字节。文件存放在磁盘上的起始块号为 28，每个物理块长度为 1K 字节。回答下面问题：

（1）采用记录成组方式存放该文件时，块因子为_____最合适。

（2）存放该文件至少要占用_____个磁盘块。

（3）该文件的第6个逻辑记录所在的磁盘块号为_____。

（4）若要把第 6 个逻辑记录读入到用户区的 1500 单元开始的区域，写出完成该要求的主要过程。

答：(1) 1024/512=2，块因子为 2 时最合适。

(2) 10/2=5，该文件至少需要占 5 个盘块。

(3) 主要过程如下：

① 申请一个长度为 1KB 的系统缓冲区，设其开始地址为 d;

② 该文件第 6 个逻辑记录存储在第 3 个盘块中，即在 30 号盘块中。将盘块号 30 转换成磁盘地址(柱面号，磁头号，扇区号)，启动磁盘，将 30 号盘块的内容读入 d 开始的缓冲区；

③ 把 d+512 开始的 512 个字节传送到用户区 1500 开始区域。

5*．如果因为系统崩溃，存放空闲磁盘块信息的空闲表（链）或位示图完全丢失，会发生什么情况？有什么办法从这个灾难中恢复吗，还是与该磁盘彻底再见？分别就 UNIX 和 FAT 文件系统讨论你的答案。

答：UNIX 采用成组链接管理磁盘空闲块，成组链接的“头指针”(空闲块号栈)存储在超级盘块中，可以在磁盘的另一指定位置存储空闲块号栈的副本，以便在超级块中的数据丢失时，能从这个灾难中恢复。FAT 文件系统也是采用在磁盘中存储 2 个完全相同的 FAT 表的方法，以避免一旦主 FAT 表损坏时，能从灾难中恢复。

6．假设用户甲要用到文件 A、B、C、E，用户乙要用到文件 A、D、E、F。已知：用户甲的文件 A 与用户乙的文件 A 实际上不是同一文件；用户甲与用户乙又分别用文件名 C 和 F 共享同一文件；甲、乙两用户的文件 E 是同一个文件。请回答下列问题：

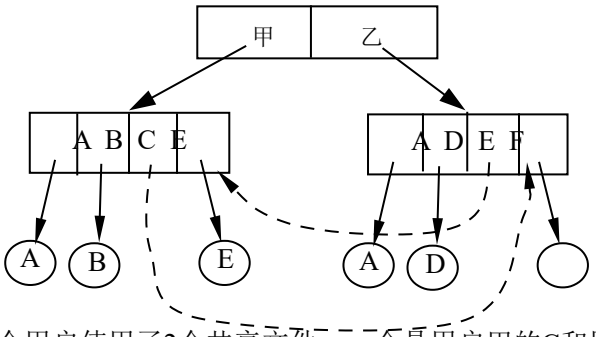
（1）系统应采用怎样的目录结构才能使两用户在使用文件时不致于造成混乱？

（2）画出这个目录结构。

（3）两个用户使用了几个共享文件？写出它们的文件名。

答：(1) 系统应采用二级或多级目录结构才能使两用户在使用文件时不致于造成混乱。

(2)



(3) 两个用户使用了2个共享文件，一个是用户甲的C和用户乙的F，另一个是用户甲的E与用户乙的E。

7．文件系统的性能可以利用在内存设置磁盘缓冲区改进，若磁盘缓冲区的命中率为 h，访问一次缓冲区需 1ms，而访问一次磁盘需 40ms，则平均访问一磁盘块所需时间是____ms。

答：$1 \times h + 40 \times (1 - h) = 40 - 39h$

8．(南开 1999 年试题)某文件系统的目录结构如图 6-2 所示，已知每个目录项占 256B，磁盘的一块为 512B。设当前目录为根目录。

（1）查询文件Wang的路径是什么？

（2）系统需要读取几个文件后才能查到Wang？

（3）计算系统找到Wang，至少读了几个盘块。

（4）给出一种加速文件查找速度的方案。

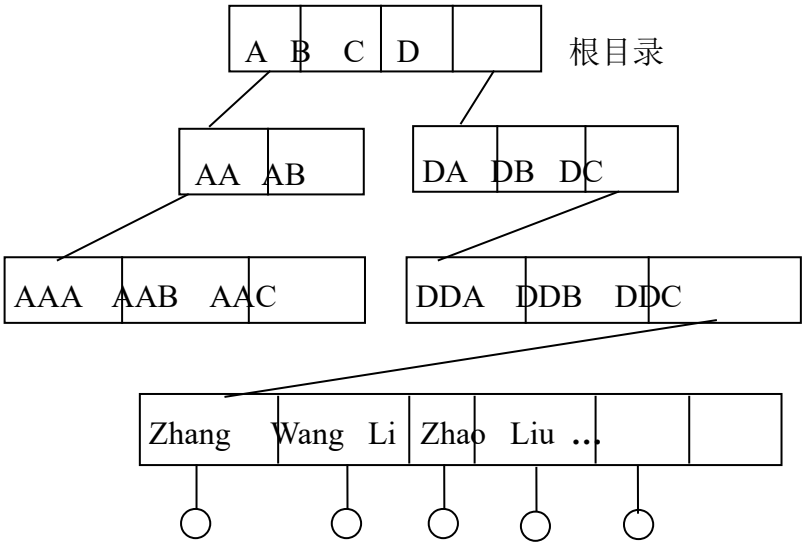


图6-2

答：(1) 查询文件 Wang 的路径是 /D/DC/DDC/Wang

(2) 系统需要读取 D、DC、DDC 等 3 个目录文件才能查到 Wang

(3) 因 1 个盘块中可存储 2 个目录项，读取根目录的第 2 个盘块时才能找到文件 D 的目录项；读取文件 D 的第 2 个盘块时才能找到 DC 的 FCB；读取文件 DC 的第 2 个盘块时才找到文件 DDC 的 FCB；读取 DDC 的第 1 个盘块就能找到 Wang。因此，系统找到 Wang，至少读了 7 个盘块。

(4) 可以采用类似 UNIX 的方法，缩短目录项，例如，目录项中仅包含文件名(12 个字节)和索引节点号(4 个字节)，目录项长度为 16 字节，这样每个盘块可存放 32 个目录项。这样只需读 4 个盘块就可找到 Wang。

9．某系统采用位示图法实现磁盘空间管理，有一磁盘有 10000 个物理块(设字号和位号都是从 0 开始编号，块号从 2 开始)，位示图的每个字有 32 位，试问：

（1）需要有多少个字？

（2）计算第i个字第j位对应的物理块号。

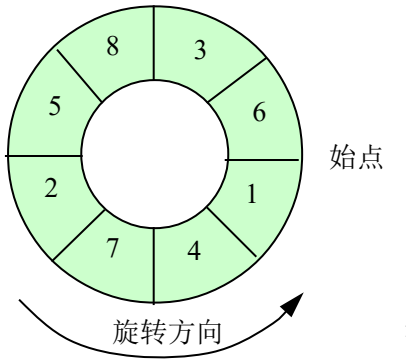
（3）求物理块号N对应的字号和位号。

10．假定磁盘的每个盘面分为 8 个扇区，其旋转速度为 20ms/周。若有 8 个逻辑记录要存放在同一磁道上供处理程序使用，处理程序每次从磁盘读出一个记录后要花费 5ms 进行处理，现在用户要求顺序处理这 8 个记录，请回答：

①　画图说明怎样安排这8条记录，使得它们能保证最高效率。

②　按照最优分布时，读出8个逻辑记录，磁盘需要旋转几周？

解：①因磁盘旋转速度为 20ms/周，每个磁道分 8 个扇区，故读写 1 个扇区的时间(数据传输时间)为 20÷8=2.5ms。程序每读出一个记录要花费 5ms 进行处理，5ms 内读写磁头正好走过 2 个扇区(2 个记录)。为了保证对这 8 条记录处理的最高效率，可 1 号记录存储于起始点开始的扇区中，相隔 2 个扇区后存放 2 号记录，2 号记录后再相隔 2 个扇区存放 3 号记录，……，即可将这 8 条记录安排成图 6-3 所示的形式：



注：图中数字代表记录号

图 6-3

②按照最优分布时，磁盘需要旋转3周。

11．某文件系统以硬盘作为存储器，盘块大小为 512 B，有文件 A，包含 590 个逻辑记录，每个记录占 255B，每个盘块存放 2 个记录。文件 A 在文件目录中的位置如图 6-4 所示。

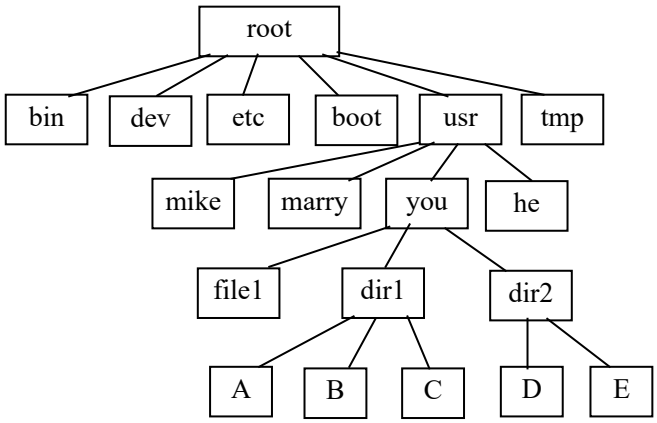


图6-4

此树形目录结构由根目录结点，作为目录文件的中间结点和作为信息文件的叶子结点组成，每个目

录项占 127B，每个物理块存放 4 个目录项。根目录的内容常驻内存。问：

(1) 若文件采用隐式链接结构，设每块的链接字占2B。如果要将文件A读入内存，至少要存取几次硬盘？为什么？

(2) 若文件采用连续文件结构，如果要将文件A的逻辑记录号为480的记录读入内存，至少要存取几次硬盘？为什么？

解：(1) 因根目录内容常驻内存，故不需读盘即可知道目录文件 usr 的地址，第一次由 usr 的地址读一个盘块，因一个盘块存放 4 个目录项，故第一次读盘即可以得到 you 的地址，第二次读盘可以得到 dir1 的地址，第三次读盘可以得到文件 A 的地址。由于每个盘块可存放 2 个记录，而文件 A 包含 590 个记录，故要把文件 A 读入内存，所需读盘次数为 590/2=295 次。所以，为把文件 A 读入内存，需读盘次数为 295+3=298 次。

(2) 当文件为连续结构时，第一次读盘得到you的地址，第二次读盘可以得到dir1的地址，第三次读盘可以得到文件A的地址。得到文件A的地址后，通过计算，只需1次读盘就可读出第480号记录。即一共需要读盘4次，就能将文件A的逻辑记录号为480的记录读入内存。

12．(北京大学 1990 年试题)有如图 6-5 所示的文件目录结构，图中的框表示目录，圈表示文件。请回答下列问题：

(1) 可否进行下列操作，为什么？

① 在目录D中建立一个文件，取名为A。

② 将目录C改名为A。

(2) 若E和G是两个用户各自的目录，问：

① 使用目录E的用户要共享文件M，如何实现？

② 在一段时间内，使用目录G的用户主要使用文件S和T，应如何处置？其目的是什么？

(3) 使用目录E的用户欲对文件I加以保护，不许别人使用，如何实现？

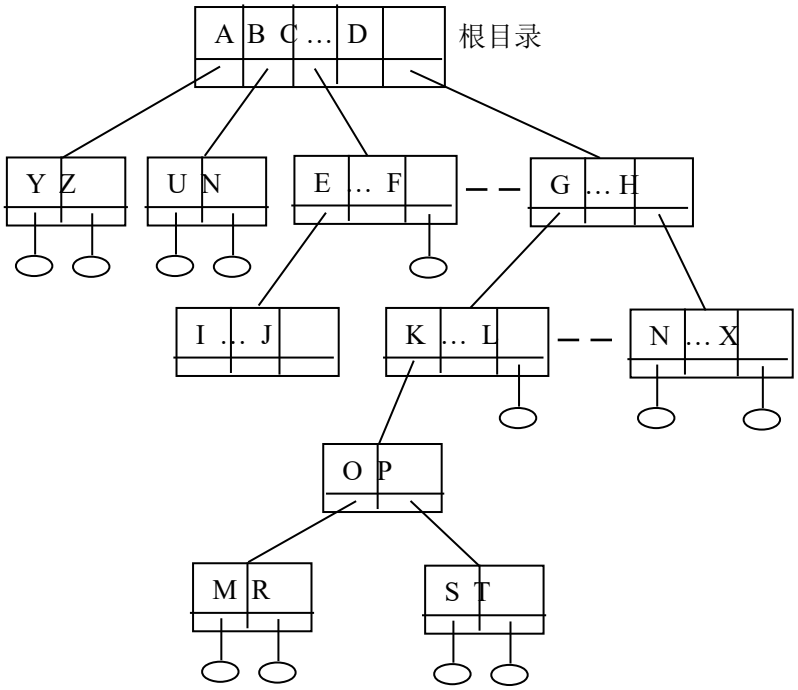


图6-5文件目录结构示意图

答：(1) ① 在目录 D 中可以建立一个文件名为 A 的文件，因为目录 D 中不存在名字为 A 的目录项或文件。

② 目录C不可以改名为A，因为目录C所在的目录（根目录）中已经有名字为A的目录项。

(2) ① 使用目录 E 的用户要共享文件 M，需要用户 E 有文件 M 的访问权限。在访问权限许可的情况

下，用户 E 可通过相应的路径来访问文件 M，即用户 E 通过自己的主目录 E 找到其父目录 C，再访问目录 C 的父目录根目录，然后依次通过目录 D、目录 G、目录 K 和目录 O 访问到文件 M。若用户 E 当前目录为 E，则访问路径为：../../D/G/K/O/M，其中符号“..”表示一个目录的父目录，符号“/”用于分割路径中的各目录名。

② 使用G需要依次访问目录K和目录P，才能访问到文件S及文件T。为了提高访问速度，可以在目录G下建立两个链接文件，分别链接到文件S和文件T上，这样，用户G就可以直接访问这两个文件了。

(3) 使用目录E的用户可以通过修改文件I的存取控制权限来对文件I加以保护，不许别人使用。即在文件I的存取权限中，仅留下用户E的访问权限，而不让其他用户访问。

13．(北京大学 1994 年试题)有一个文件系统如图 6-6 所示。图中的框表示目录，圆圈表示普通文件。根目录常驻内存，目录文件组织成链接文件，不设文件控制块，普通文件组织成索引文件。目录表目指示下一级文件名及其磁盘地址（各占 2 个字节，共 4 个字节）。若下一级文件是目录文件，指示其第一个磁盘块地址。若下级文件是普通文件，指示其文件控制块的磁盘地址。每个目录文件磁盘块最后 4 个字节供拉链使用。下级文件在上级目录文件中的次序在图中为从左至右。每个磁盘块有 512 字节，与普通文件的一页等长。

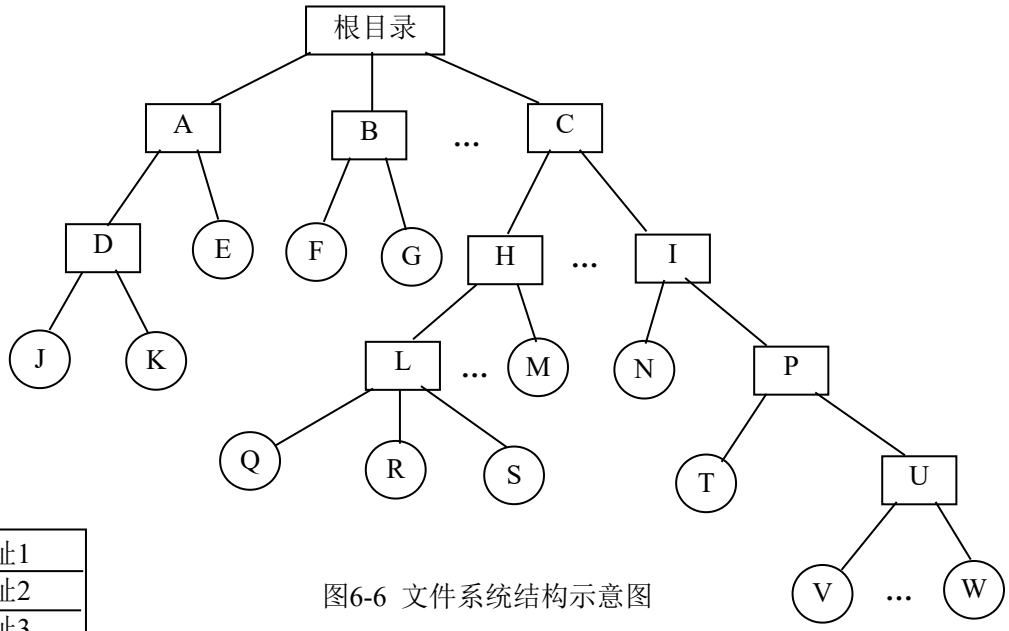


图6-6 文件系统结构示意图

磁盘地址1
磁盘地址2
磁盘地址3
磁盘地址4
磁盘地址5
磁盘地址6
磁盘地址7
磁盘地址8
磁盘地址9
磁盘地址10
磁盘地址11
磁盘地址12
磁盘地址13

图6-7 普通文件的FCB组织

普通文件的文件控制块组织如图6-7所示。其中，每个磁盘地址占2个字节，前10个地址直接指示该文件前10页的地址，第11个地址指示一级索引表地址，一级索引表中每个磁盘地址指示一个文件页地址；第12个地址指示二级索引表地址，二级索引表中每个地址指示一个一级索引表地址；第13个地址指示三级索引表地址，三级索引表中每个地址指示一个二级索引表地址。问：

（1）一个普通文件最多可有多少个文件页？

（2）若要读文件J中某一页，最多启动磁盘多少次？

（3）若要读文件W中的某一页，最少启动磁盘多少次？

（4）就（3）而言，为最大限度减少启动磁盘次数，可采用什么方法？此时，磁盘最多启动多少次？

解：

（1） 由题目中所给条件可知，磁盘块大小512字节，每个磁盘地址占2字节，因此一个一级索引可容纳256个磁盘地址。同样地，一个二级索引表可容纳256个一级索引表地址，一个三级索引表可容纳

256个二级索引表地址。这样，一个普通文件最多可有的页数为：

10+256+256×256+256×256×256=16843018

（2） 从图6-6中可以看出，目录文件A和目录文件D中目录项都只有2个，因此这两个目录文件都不需要拉链。若要读取文件J中某一页，首先从内存的根目录中找到目录文件A的磁盘地址，将其读入内存（第1次读磁盘）。然后再从目录A中找到D的磁盘地址，并将其读入内存（第2次读磁盘）。从D目录中找到文件J的文件控制块地址，将其读入内存（第3次读磁盘）。在最坏的情况下，要访问页的地址需通过三级索引才能找到这时要三次读取磁盘才能将三级索引读入内存（第4、5、6次读磁盘）。最后读入文件J的相应页（第7次读磁盘）。由此可知，若要读文件J中某一页，最多启动磁盘7次。

（3） 从图6-6中可以看出，目录文件C和目录文件U中，目录项数目较多，若目录项数超过127（512/4−1=127，其中有4字节用作链接，故减1），则目录文件的读入可能需要多次读盘（因目录文件组织成链接文件）。在最好情况下，所照的目录项都在目录的第一个磁盘块中。若要读文件W的某一页，首先从内存的根目录中找到目录文件C的地址，将其读入内存（第1次读磁盘）。在最好情况下，能从目录C的第一个盘块中找到目录文件I 的地址，将其读入内存（第2次读磁盘）。从目录I中找到目录文件P的地址，将其读入内存（第3次读磁盘）。从目录P中找到目录文件U的地址，将其读入内存（第4次读磁盘）。在最好情况下，能从目录U的第一个盘块中找到目录文件W的文件控制块的地址，将其读入内存（第5次读磁盘）。在最好情况下，要访问的页在前10页中，这时可以直接读到该页的地址。最后读入文件W的相应页（第6次读磁盘）。由此可知，若要读文件W中的某一页，最少启动磁盘6次。

（4） 由于通过文件控制块访问文件所需的读盘次数无法改变，要减少访问磁盘次数，只有通过访问目录文件的次数来达到。为最大限度地减少启动磁盘次数，可将文件W直接链接在根目录的最左端（其目录项在根目录的前127个目录项内）。这样，若要读文件W的某页时，首先从内存的根目录中找到文件W的文件控制块地址，将文件W的文件控制块读入内存（第1次读磁盘）。在最好情况下，要访问的页在前10页中，这时可以直接读到该页的地址。在最坏的情况下，要访问页的地址需三级索引才能找到，这时要三次访问磁盘（第2、3、4次读磁盘）。最后读入文件W的相应页（最好情况下，第2次读盘；最坏情况下，第5次读磁盘）。由此可知，若将文件W直接链接在根目录的最左端，要读文件W中的某一页，最少启动磁盘2次，最多启动磁盘5次。

14．若磁盘的每个磁道分成 9 个块，现有一文件共有 A、B、…、H、I 9 个记录，每个记录的大小与盘块大小相等，设磁盘转速为 27ms/转，每读出一块后需要 2ms 的处理时间。若忽略其他辅助时间，试问：

（1）如果顺序存放这些记录并顺序读取，处理该文件需要多少时间？

（2）如果顺序读取该文件，记录如何存放处理时间最短？

解：由条件可知，磁盘转速为每转27ms，每磁道存放9个记录，因此读出1个记录的时间为：

27/9=3ms

(1) 读出并处理记录A需要5ms，此时磁头已移到记录B的中间，因此为了读记录B，必须再转将近一圈（从记录B的中间转到记录B的开头，即读出记录A后，要经27ms才能再读记录B）。后续7个记录的读取及处理时间与此相同，但最后一个记录读取与处理时间只需5ms。于是，处理9个记录的总时间为(假定开始时磁头在记录A的开头)：

8×27+9×3+2=245ms

(2) 由于读出并处理一个记录需要5ms，当读出并处理记录A时，不妨假设记录A存放在第一个磁盘块中，磁头已移到第2个盘块中间，为了能顺利读到记录B，应该将它存放在第3个盘块中，即应将记录按如下表所示顺序存放：

盘块号	1	3	5	7	9	2	4	6	8
记录号	A	B	C	D	E	F	G	H	I

这样，处理一个记录并将磁头移到下一个记录的时间是：

读出时间+处理时间+等待时间=3+2+1=6 ms

所以，处理9个记录的总时间为：6×8+5=53 ms

15．有一个交叉存放信息的磁盘，信息在其上的存放方法如图 6-8 所示。每磁道有 8 个扇区，每扇区 512 字节，旋转速度为 3000 转/分。假定磁头已在要读取信息的磁道上，0 扇区旋转到磁头下需要 1/2 转，且设备对应的控制器不能同时进行输入/输出，在数据从控制器传送到内存这段时间内，从磁头下通过的扇区数为 2，问依次读出一个磁道上所有扇区需要多少时间？读出整个磁道的数据传输速度为多少？

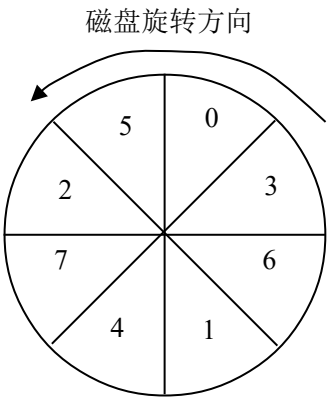


图6-8 交叉存放方式

解：由图 6-8 可知，信息块之间的间隔为 2 个扇区，由已知条件，旋转速度为：3000 转/分=50 转/秒，即 20ms/转

读一个扇区需要的时间：20/8=2.5 ms

读一个扇区并将扇区数据传送到内存需要时间：2.5×3=7.5 ms

读出一个磁道上所有扇区数据需要时间：20/2+8×7.5=70 ms=0.07 s

每磁道数据量：8×512=4KB

数据传输速度为：4KB/0.07=57.14KB/秒

所以，依次读出一个磁道上所有扇区需要0.07秒，其数据传输速度为57.14KB/秒。

16．某软盘有 40 个磁道，磁头从一个磁道移到另一个磁道需要 6 ms。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 磁道，每块的旋转延迟时间及传输时间分别为 100ms 和 25ms，问读取一个 100 块的文件需要多少时间？如果系统对磁盘进行了整理，让同一个文件的磁盘块尽可能靠拢，从而使相邻数据块的平均距离降为 2 个磁道，这时读取 100 块文件需要多少时间？

解：磁盘整理前，读取一块数据的时间为：13×6+100+25=203 ms

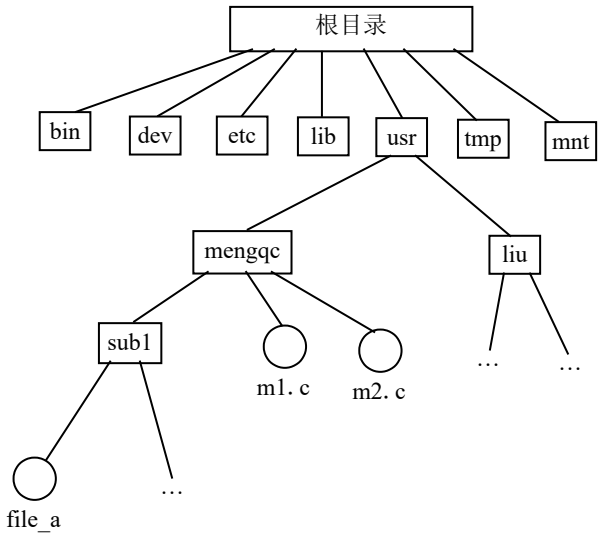
因此，读取一个100块的文件需要时间：203×100=20300 ms

磁盘整理后，读取一块数据的时间为：2×6+100+25=137 ms

因此，读取一个100块的文件需要时间：137×100=13700 ms

17．设UNIX文件系统的目录结构如图6-9所示：

图6-9



（1）设当前工作目录是/usr/mengqc，那么，访问文件 file_a 的绝对路径和相对路径各是什么？

（2）如果用 ls –l 命令列出当前工作目录的内容，其中有如下所示的一项：

rw-r—r--　　2　　mengqc　　……　　m2.c

那么，文件 m2.c 对文件主、同组用户、其他用户分别规定了什么权限？

（3）现在想把工作改到 liu，应使用什么命令（写出完整命令行）？

答：（1）（2）略；（3）cd /usr/liu 或 cd ../liu

18．某文件有 101 块，其第 51 块是中间块。假设该文件首块的物理地址和要添加的信息块已在内存，并且文件的尾部有可增长的物理空间而头部没有，试分别采用连续分配和链接分配的系统，完成下列几种对文件添加或删除一块的操作所需的读写操作的次数（不计把删除的块加入到空闲块表的任何 I/O 操作，也不计因文件首块地址或长度变了而修改磁盘目录项的操作）。

（1）在头部添加一块；　　（2）在尾部添加一块；

（3）在中间块后部添加一块；　　（4）删除首块；

（5）删除尾块；　　（6）删除中间块。

答：（1）在连续分配的系统中，在头部添加一块时，因为整个文件要后移一块，那么 101 个块每块都需要读写 1 次，然后再写入新块，所以总共读写 203 次。

在链接分配的系统中，在头部添加一块时，新块需要写 1 次（新块中指针指向原来的首块），所以只要读写 1 次即可。

（2）在连续分配的系统中，在尾部添加一块时，因为只有新块需要写操作，所以总共需要读写 1 次。

在链接分配的系统中，在尾部添加一块时，需要在第 101 块中将块指针指向新块，这样要读 101 次盘读出第 101 块内容，然后写入新块，最后将 101 块内容中块指针值改为新块块号后重新写入 101 块中，所以总共要读写 103 次。

（3）在连续分配的系统中，在中间块后添加一块时，需要将文件的后 50 块后移一块，这项工作需要读写磁盘 100 次，最后将新块写到第 52 块处，所以总共需要读写磁盘 101 次。

在链接分配的系统中，在中间块后添加一块时，需要在第 51 块中将块指针指向新块，这样要读 51 次盘读出第 51 块内容，获得第 52 块地址，然后写入新块（新块中的块指针指向原第 52 块），最后将 51 块内容中块指针值改为新块块号后重新写入 51 块中，所以总共要读写 53 次。

（4）在连续分配的系统中，删除首块时，需要将后 100 块前移一块，所以总共需要读写 200 次。

在链接分配的系统中，删除首块时，原第 2 块称为首块，因此需要读第 1 块知道第 2 块地址所以总共要读写 1 次。

（5）在连续分配的系统中，删除尾块时，只需修改目录项中的文件长度（块数减 1），不需读写操作，所以总共需要读写 0 次。

在链接分配的系统中，删除尾块时，需要将第 100 块中的块指针改为-1（尾指针标志），因此总共需要读写 101 次。

（6）在连续分配的系统中，删除中间块时，需要将后 50 块前移一块，所以总共需要读写 100 次。

在链接分配的系统中，删除中间块时，需要将第 50 块中块指针指向原第 52 块，因此需要读 51 次，然后将原第 50 块的块指针改为第 52 块的块号后再写入第 50 块，所以总共要读写 52 次。

19．在磁盘上有一个文件系统，磁盘每块 512 字。假定每个文件在目录中占一个目录项，该目录项给出了文件名、第一个索引块的地址、文件长度（块数）。在索引块中前面 511 个字指向文件块，即第 i 个索引项（i=0,1,2,…,510）指向文件的第 i 块，索引块中最后一个字指向下一个索引块，最后一个索引块中最后一个字为-1。假定已在内存中，每个文件的逻辑块号从 0 开始编号，逻辑块长与物理块长相同。对这样索引物理结构的文件，该系统应如何将逻辑块号变换成物理块号？

答：首先通过文件名在目录中找到该文件的目录项，根据给出的逻辑块号 j 与该目录项中文件长度比较，若 j>文件长度，则报错返回。否则，计算逻辑块号 j 对应第几个索引块：

索引块号 n=j div 511　　(div 表示整除运算)

索引块内索引项偏移 w=j mod 511　　(mod 表示相除取余数)

从目录项中指定的第一个索引块地址开始，依次读出 n 个索引块(0~n-1 号索引块)，在读出的最后一个索引块(n-1 号索引块)中，获得 n 号索引块的物理块号 b，读 b 号物理块，从其第 w 个索引项即可查得逻辑块号 j 对应的物理块号。

例如，设逻辑块号 j=100，n=100 div 511=0，w=100 mod 511=100。读出由目录项中指定的第一个索引块，其中第 100 个索引项指定的物理块号，即是逻辑块 100 对应的物理块号；

又如，设 j=600，n=600 div 511=1，w=600 mod 511=89。读出由目录项中指定的第一个索引块，由其中的最后一个字，获得下一个索引块号；读出下一个索引块，其中的第 89 个索引项中指定的物理块号，即是逻辑块 600 对应的物理块号。

20．考虑一个存在于磁盘上的文件系统，其中的文件由大小为 512B 的逻辑块组成。假定每一个文件有一个文件目录项，该目录项包含该文件的文件名、文件长度以及第一块（或第一索引块）和最后一块的位置，而且该目录项位于内存。对于索引结构文件，该目录项指明第一索引块，该索引块又一次指向 511 个文件块（每个索引值占 4B），且有一指向下一索引块的指针(指针占 4B)。针对连续、链接、索引结构的每一种，如果当前位于逻辑块 30（即之前最后一次访问的块是逻辑块 30）且希望访问逻辑块 20(假设逻辑块号从 0 开始编号)，那么，必须分别从磁盘上读多少个物理块？

解：(1) 对于磁盘上的连续结构文件，由文件的逻辑块号、文件块大小、磁盘物理块大小以及文件的首块位置，可以计算该逻辑块所在的物理块号（地址）A：

$A=A_0+(N*L)/S=A_0+20*512/2048=A_0+5$

其中 A_0 为文件第 0 块位置，

N 为逻辑块号(N=20)，

L 为逻辑块长度(L=512)，

S 为磁盘块长度(由已知条件得 S=511*4+1*4=2048)。

因此，无论当前读写位置如何，要访问第 20 个逻辑块，只要直接读出文件的第 6 个物理块，即只需读 1 个磁盘块即可(因目录项已在内存)。

(2) 对于磁盘上的链接结构文件，当前读写了逻辑块 30，要访问逻辑块 20，需要从文件开头开始。由前面分析知，磁盘块大小 2048B，故每个盘块可存放 4 个逻辑块。逻辑块 20 在文件的第 6 个物理块中，因此需依次读出第 1、2、3、4、5 等盘块，从第 5 个物理块获得第 6 个物理块的块号，在读出第 6 物理块，其开头的 512B 即是 20 号逻辑块的内容。所以，需读 6 个物理块。

(3) 对于磁盘上的索引结构文件，若要访问逻辑块20（假定此前在访问逻辑块30时已将索引块保存在内存），可在内存中的索引表中查到逻辑块20所在的第6个物理块号，再将该物理块读出来，其开头512B即是20号逻辑块的内容。因此需要读1个物理块。

21．在实现文件系统时，为了加快文件目录的检索速度，可利用“文件控制块分解法”。假设目录文件存放在磁盘上，每个盘块 512B。文件控制块占 64B，其中文件名占 14B。通常将文件控制块分解成两部分，第 1 部分占 16B（包括 14B 文件名和 2B 文件内部号），第 2 部分占 50B（包括文件内部号和文件其它描述信息）。请按下述要求进行分析：

(1) 假设某一文件目录文件共有280个文件控制块，试分别给出采用分解法前和分解法后，查找该文件目录文件的某一个文件控制块的平均访问磁盘次数。

(2) 一般地，若目录文件分解前占用n个盘块，分解后改用m个盘块存放文件名和文件内部号，请给出访问磁盘次数减少的条件。

解：(1) 采用分解法前：每个盘块存储的文件控制块数为：512/64=8

254个文件控制块占用的盘块数为：280/8=35，即35个盘块

所以，平均访问盘块数为(35+1)/2=18，即平均访问磁盘18次。

采用分解法后：这种情况需要将FCB的第一部分信息集中存放在一起，每个FCB的第一部分占16B，故每个盘块可存放512/16=32个文件的FCB第一部分信息，280个文件需280/32=8.75，即需9个盘块存储。所以文件名查找平均需访问5个盘块。再加上读该文件的目录项的其他信息需读1次盘，所以总共需要访问6次磁盘。

(2) 采用分解法前，由于目录文件占n个盘块，故平均访问磁盘次数为(n+1)/2。

采用分解法后，用m个盘块存放文件和文件内部号，故按文件名查找时，平均访问磁盘次数为(m+1)/2。找到匹配的文件控制块后，还需要进行一次磁盘访问才能读出全部的文件控制块信息，所以，其平均访问磁盘次数为(m+1)/2+1。采用分解法访问磁盘次数减少的条件是：(n+1)/2>(m+1)/2+1

即 n>m+2 或 m<n-2

22．在 UNIX 中，如果一个盘块的大小为 512B，每个盘块号占 4 个字节，磁盘的磁头数为 h，每个磁道的扇区数为 s(扇区大小也是 512B)。设磁盘的柱面号、磁头号、扇区号以及逻辑块号都从 0 开始编号。文件 box 是记录长度为 128B 的定长记录式文件，该文件的索引结点已经读入内存，请写出读取文件 box 的第 500 个记录的处理过程。（2010 自编题）

解：第500个记录的字节偏移量为128×500=64000

64000÷512=125，即第500个记录存储在第125个盘块中

因盘块的大小为 512B，每个盘块号占 4 个字节，即每块可存放 128 个地址。要访问字节偏移量为 64000，需 1 次间接索引。在文件 box 的内存索引结点中，读取一次间接索引的块号 x1；

计算逻辑块号 x1 对应的物理地址：

柱面号a=x1 DIV (h×s)　　DIV表示相除取整数

磁头号b=(x1 MOD (h×s)) DIV s　　MOD表示相除取余数

扇区号 c=x1 MOD s

启动磁盘，由物理地址 a、b、c 定位物理扇区，该扇区中存放着文件 box 的第 11～138 个块号。将该扇区内容读入磁盘缓冲区。在缓冲区中读取第 115 个块号 x2(该缓冲中偏移地址 456 处开始的 4 个字节)，该块中存储着第 500 个记录。

按上述同样方法，由 x2 可计算对应的物理地址 a、b、c，然后启动磁盘，将对应的扇区读入缓冲。显然，第 500 个记录是该扇区中的第 4 个记录。因此，缓冲区中偏移地址 384 处开始的 128 个字节就是第 500 个记录的内容。

23．UNIX 系统中，对磁盘空间使用“空闲块成组联接法”进行管理。图 6-10 为文件卷资源表的当前状态。设某文件删除后回收 3 个物理块(37#、218#、219#)，请图示出回收后的有关表格情况，并简要说明回收过程。(北邮 1999)

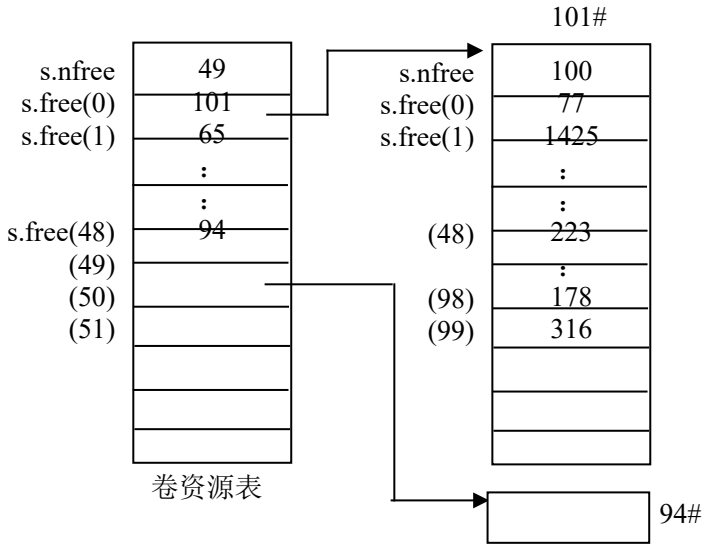


图6-10

解：(1) 回收37#物理盘块的过程：

1　判断 s.nfree<100 成立，执行下面的操作；

2　将回收的盘块号 37 写入 s.free(s.nfree)，即 s.free(49)=37；

3　s.nfree 增 1，即 s.nfree=s.nfree+1。

(2) 回收218#物理盘块的过程：

1　判断 s.nfree<100 成立，执行下面的操作；

2　将回收的盘块号 218 写入 s.free(s.nfree)，即 s.free(50)=218；

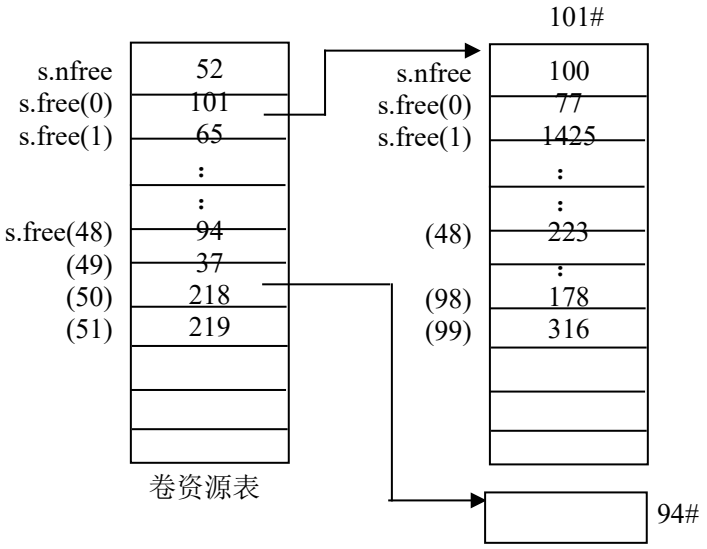
3　s.nfree 增 1，即 s.nfree=s.nfree+1。

(3) 回收219#物理盘块的过程：

1　判断 s.nfree<100 成立，执行下面的操作；

2　将回收的盘块号 219 写入 s.free(s.nfree)，即 s.free(51)=219；

3　s.nfree 增 1，即 s.nfree=s.nfree+1。最终回收后的有关表格情况如下图所示。.



24．UNIX 系统中，假定盘块大小为 1KB，每个盘块号占 4 个字节，文件索引结点中的磁盘地址明细表如图 6-11 所示，如何将下列文件的字节偏移量转换为物理地址(盘块号和块内偏移)?

（1）9000；（2）14000；（3）350000；（4）680000

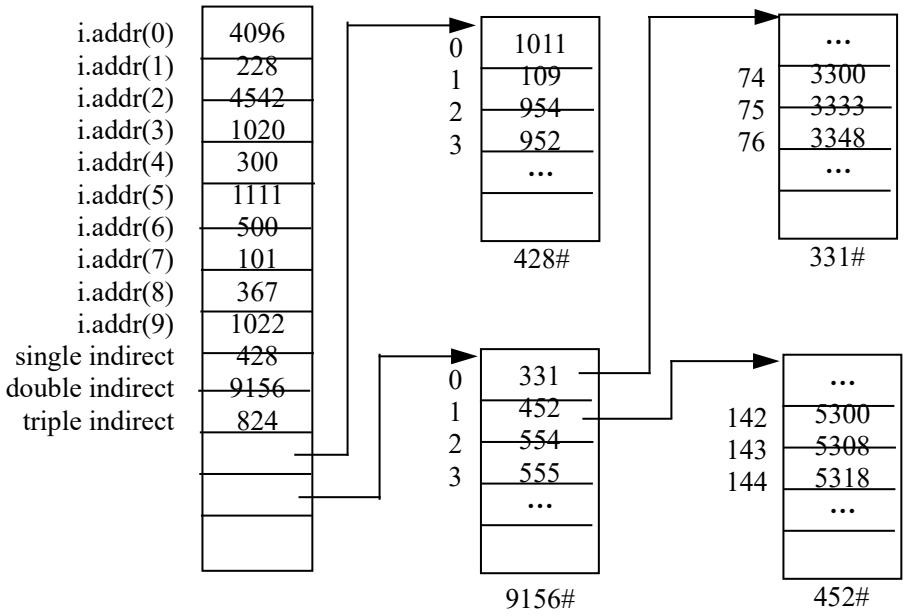


图6-11

解：(1) 字节偏移量为9000

逻辑块号为：9000/1024=8

块内偏移量：9000%1024=808

因逻辑块号小于 10，因此该块为直接块。由图 6-12 可知，其物理块号为 367，该块的第 808 字节即为文件的第 9000 字节。

(2) 字节偏移量为14000

逻辑块号为：14000/1024=13

块内偏移量：14000%1024=688

因逻辑块号 10<13<266，因此该块为一次间接块。

由图 6-12 可知，一次间接索引块号为 428，由于直接索引有 10 个块号，13-10=3。读出 428#块，查得其中第 3 个索引项值为 952。即是逻辑块号 13 对应的物理块号为 952，物理块 952 中的第 688 字节即为文件的第 14000 字节。

(3) 字节偏移量为350000

逻辑块号为：350000/1024=341

块内偏移量：350000%1024=816

因逻辑块号 266<341<65802，因此该块为二次间接块。

由图 6-12 可知，二次间接索引块号为 9156。由于一个一次间接块中可容纳 256 个块号，直接索引有 10 个块号，341-(10+256)=75；75/256=0。读出 9156#盘块，查得其中的第 0 个索引项值为 331。.读 331#盘块，查得其中第 75 个索引项的值为 3333。.因此，字节偏移量 350000 对应的盘块号为 3333，3333#物理块中的第 816 字节即为文件的第 350000 字节。

(4) 字节偏移量为680000

逻辑块号为：680000/1024=664

块内偏移量：680000%1024=64

因逻辑块号 266<341<65802，因此该块为二次间接块。

由图 6-12 可知，二次间接块号位 9156。由于一个一次间接块中可容纳 256 个块号，直接索引有 10 个块号，664-(10+256)=398；398/256=1；398%256=142。读出 9156#盘块，查得其中的第 1 个索引项值为 452。.读 452#盘块，查得其中第 142 个索引项的值为 5300。.因此，字节偏移量 680000 对应的盘块号为 5300，5300#物理块中的第 64 字节即为文件的第 680000 字节。

25．有一磁盘组共有 10 个盘面，每个盘面上有 100 个磁道，每个磁道有 16 个扇区。假定分配以扇区为单位，若使用位示图管理磁盘空间，问：位示图需要占多少空间？若空白文件(空闲盘区表)的每个表目占用 5 个字节，问什么时候空白文件占用的空间大于位示图占用的空间？

解：磁盘组扇区总数为

16×100×10=16000

因此位示图描述扇区状态需要 16000 位，即位示图的大小为 2000 字节。

又，由题目条件可知，空白文件每个表目占 5 个字节，2000 个字节可存放的表目数为：

2000÷5=400

所以，当磁盘中的空白区数目大于 400 时，空白文件所占空间大于位示图占用的空间。

26．(大连理工 2000 试题)一个文件系统目录结构如图 6-12 所示。文件采用的物理结构是串联结构，文件 F1 由 500 个逻辑记录组成，每个盘块可存放 20 个记录。现在欲读取 F1 中第 406#记录，文件系统的根目录现已存在于内存，则最少需读多少次磁盘块，才能取出 F1 的第 406#记录。

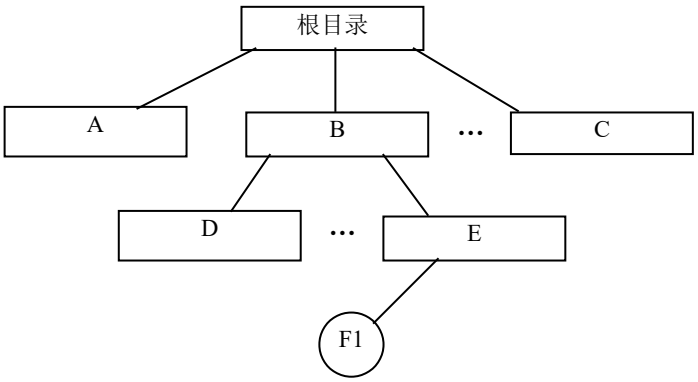


图 6-12

解：在内存的根目录中找到子目录B，获得子目录B的首块号，启动磁盘读入子目录B的首块号对应的盘块，最好的情况是在该盘块读入的目录项中找到子目录E，即至少读一次磁盘块找到E，得到E的首块号。

读入E的首块，最好的情况是在该盘块读入的目录项中找到F1及其首块号，即至少再读1次磁盘块找到F1的首块号。

由于每个盘块可存放20个记录，406÷20=20.3，所以406#记录在文件F1的第21个磁盘块中。因文件采用的物理结构是串联结构(链接结构)，读出前一个盘块才能知道下一个盘块号，故还需读20次磁盘块，才能获得第21个盘块号；读出该盘块，其中的第6个记录，即是F1的第406#记录。即需再读21次磁盘，才能获得第406#记录。

所以欲读取F1中第406#记录，总共需要读23次磁盘块。

27．(北大 2000 试题)为了解决文件系统的不一致性问题，常采用一个实用程序检查文件系统。在进行了块的一致性检查后，得到如下表所示的结果。请解释该文件系统中出现的每一种错误，并给出处理办法。

块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
空闲块	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	2
分配块	0	0	1	1	0	1	0	0	0	2	0	0	0	0	0	0

解：外存空间的任何一个存储块要么存放了信息，要么空闲。因此用两个数据结构分别保存空闲块和分配快，它们的内容应当是互补的，即和为 1。如果不是这样，说明出现了错误。从表中看，系统出现了下述错误：

① 2号块错。该块既是已分配块又是未分配块。应进一步检查2号块是否被某个文件占用，若占用则空闲块标志改为0，否则分配块标志该、改为0。

② 9号块错。可能是分配时登记错，将分配块数减为1。

③ 11号块错。可能是归还时登记错，重新将该块登记到空闲块链表中。

④ 15号块错。可能是归还时出现登记错，将该块的数量减为1。

28．(青岛大学 2001 试题)一台转速为 3600(转/分)的磁盘。其存储密度为 16.7(KB/道)。已知磁盘从启动到运转平稳的时间为 3ms，磁头臂的移动速度为 0.3(ms/道)，请回答：

(1) 设磁头的当前位置在 20 号磁道，移动方向为向磁道号增加的方向。若系统收到 4 条记录访问请求，请求序列如下表所示。

记录号	磁道号
1	18
2	25
3	32
4	7

请写出电梯算法的访问序列。

(2) 若上述 4 条记录的长度皆为 16.7KB，求系统按电梯算法访问磁盘上的上述 4 条记录的最长时间为多少？

解：(1) 采用电梯算法的访问序列为：(20)→25→32→18→7。

(2) 计算磁盘访问时间的公式为：

访问时间=启动时间+移臂时间+旋转等待时间+传输(读写)时间

2号记录的移臂时间T2、最大旋转延迟时间R2和读写时间A2为：

T2=0.3*5=1.5ms

R2=(60*1000)/3600=17.6ms

A2=旋转一周的时间=16.7ms

3项时间合计为34.9ms。

3号记录的移臂时间T3、最大旋转延迟时间R3和读写时间A3为：

T3=0.3*7=2.1ms

R3=(60*1000)/3600=17.6ms

A3=旋转一周的时间=16.7ms。3项时间合计为35.5ms。

1号记录的移臂时间T1、最大旋转延迟时间R1和读写时间A1为：

T1=0.3*14=4.2ms

R1=(60*1000)/3600=17.6ms

A1=旋转一周的时间=16.7ms。3项时间合计为37.6ms。

4号记录的移臂时间T4、最大旋转延迟时间R4和读写时间A4为：

T4=0.3*11=3.3ms

R4=(60*1000)/3600=17.6ms

A4=旋转一周的时间=16.7ms。3项时间合计为36.7ms。

所以，总的访问时间为：3+34.9+35.5+37.6+36.7=147.7(ms)

29．（7分）某文件系统为一级根目录结构，文件的数据一次性写入磁盘，已写入的文件不可修改，但可多次创建新文件。请回答如下问题。（2011全国试题）

（1）在连续、链式、索引三种文件的数据块组织方式中，哪种更合适？要求说明理由。为定位文件数据块，需在FCB中设置哪些相关描述字段？

（2）为快速找到文件，对于FCB，是集中存储好，还是与对应的文件数据块连续存储好？要求说明理由。

解：（1）连续数据块组织方式更合适。因为系统规定文件不可修改，即文件长度不可能动态增长或缩小，在这种情况下，连续组织方式的缺点不再存在，而它的优点是顺序访问容易、读写文件速度快，而且磁盘空间利用率高。为定位文件数据块，需在FCB中设置首块号和文件长度值。

（2）为快速找到文件，对于FCB集中存储好，因为这样目录查找速度比FCB与对应的文件数据块连续存储时快得多。

30．(2012自编题)UNIX系统中某记录式文件长度为3200KB，其磁盘存储结构如图6-13所示。

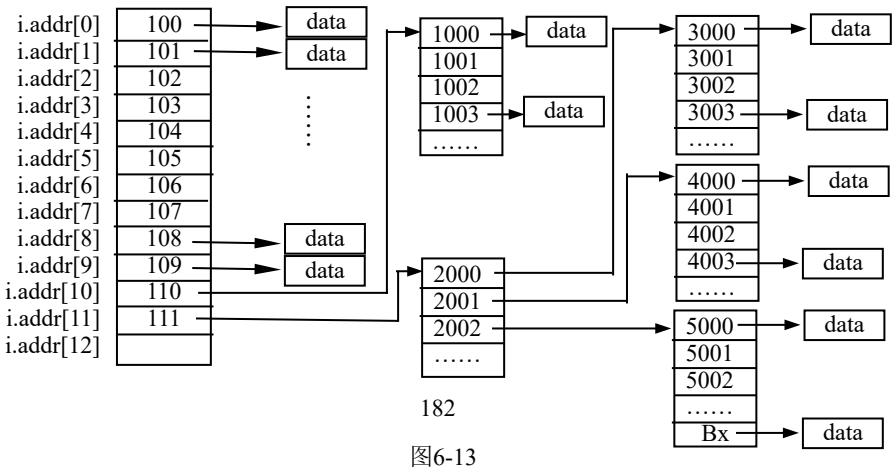


图6-13

假设该文件的逻辑记录长度为256B，磁盘块的大小为2KB，磁盘的逻辑块号占4B，又设每个索引块中的盘块号是连续的。请回答下列问题：

(1) 该文件占用的磁盘块数(不含索引结点所在的盘块)是多少？请给出计算过程。

(2) 图6-13中，Bx是存储该文件的最后一个盘块号，Bx的数值为多少？请说明计算过程。

(3) 假设某用户进程要读取该文件的第5000条记录(文件记录号从1开始编号)，写出系统为完成此工作的操作过程。

答：(1) 文件数据占用的盘块数为3200÷2=1600

文件长度为3200KB，需要二次间接索引。其中一次间接索引需1个索引块，存放文件数据的盘块数为2048÷4=512。需二次间接索引的文件数据盘块数为1600-(512+10)=1078，需3个二级索引块。故二次间接索引结构中共需4个索引块。即该文件占用的盘块数共计1605个。

(2) 由上面分析可知，二次间接索引的文件数据盘块数为1078，每个索引块最多可存储512个块号，1078个盘块号需3个索引块存储，其中第3个索引块存储1078-1024=54个盘块号，故图中的Bx应该是5053

(3) 每个盘块可存放的记录数为2048÷256=8

5000÷8=625，即该文件的第5000条记录存放在第625个盘块中

625-10-512=103。由图6-13可知，该记录存放在3102号盘块中。系统读取第5000条记录的操作过程如下：

① 分配1磁盘缓冲区，其容量为2KB；

② 由磁盘的物理参数(磁头数，每磁道扇区数等)计算出逻辑块号3102对应的柱面号，磁头号以及扇区号，启动磁盘，读取该盘块的数据到磁盘缓冲区；

③ 设磁盘缓冲区的首地址为B，因第5000条记录是该盘块中的第8条记录，故从B+1792地址开始，读取256B数据到用户进程的工作区，即完成了读取该文件第5000条记录的工作。

31．（8分）某虚拟文件系统空间的最大容量是4TB（1TB=2^{40}），以磁盘块为基本分配单元，磁盘块大小为1KB，文件控制块（FCB）包含一个512B的索引表区。请回答下列问题：(2012全国试题)

（1）假设索引表区仅采用直接索引结构，索引表区存放文件占用的磁盘块号，……索引项中块号最少占多少字节？可支持的单个文件的最大长度是多少字节？

（2）假设索引表区采用如下结构：第0~7字节采用<起始块号，块数>格式表示文件创建时预分配的连续存储空间。其中起始块号占6B，块数占2B，剩余504字节采用直接索引结构，一个索引项占6B，则可支持的单个文件的最大长度是多少字节？为了使单个文件的长度达到最大，请指出起始块号和块数分别所占字节数的合理值并说明理由。

答：(2012-4-18自编答案)

(1) 盘块数为$2^{40}÷2^{10}=2^{30}$，故索引项中块号最少占4字节。512÷4=128，即512B的索引表区最多存放128个盘块号，因此可支持的单个文件的最大长度是1KB×128=128KB。

(2) 预分配的连续存储空间最大为1KB×(2^{16}-1)=65535KB

直接索引部分的最大空间为：1KB×(504÷6)=84KB

故可支持的单个文件的最大长度是65535+84=65619KB

若起始块号所占字节数尽可能少，即采用4个字节，则块数最大可达4个字节，即块数可达2^{32}，这样预分配的连续存储空间最大可达1KB×2^{32}=4TB

即单个文件长度理论上可达4TB，即达到了虚拟文件系统空间的最大容量。

32．设某个文件由100个物理盘块存储，对于连续文件、链接文件和索引文件，分别计算执行下列操作时的启动磁盘I/O次数(假如头指针和索引表均在内存中)：

(1)　把一块加在文件的开头；

(2)　从文件的开头删去一块。

解：(1) 把一块加在文件的开头

① 对于连续文件：

一般情况下该文件存储处的前一个物理盘块不会正好是空闲的，因此需要寻找磁盘中有101个空闲盘块的连续空间，找到后将新增块写入该连续空闲空间的第一个盘块中（启动1次磁盘I/O）；然后读出文件原第一个块，写入该连续空间的第二个盘块中(启动2次磁盘I/O)，……，文件的原100个物理块搬家，共需启动200次磁盘I/O；将修改后的PCB存盘，需启动1次磁盘I/O。故把一块加在连续文件的开头，一般情况下，总共需要启动202次磁盘I/O。

② 对于链接文件：

将新增块内容及文件头指针写入一空闲盘块中(启动1次磁盘)，将新的头指针写入该文件的PCB中(启动1次磁盘)，故总共需要启动2次磁盘I/O。

③ 对于索引文件：

将新增内容写入一空闲盘块中；将修改后的内存索引表(假设是一级索引)写入索引盘块中(1次I/O，若该文件是多级索引，则其索引表由多个盘块存储，就需要启动多次磁盘I/O).。故对于一级索引文件，总共需要启动2次磁盘I/O。

(2) 从文件的开头删去一块：

① 对于连续文件：

读出该文件的PCB(启动1次磁盘I/O)，修改其首块号为文件原第2块的块号，文件长度(盘块数)减1，将修改后的PCB存盘(启动1次磁盘)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动2次磁盘I/O。

② 对于链接文件：

对隐式链接，为读出第1块以获得第2块的块号(启动1次磁盘)，将内存中的头指针修改为第2块的块号，读出该文件的FCB(启动1次磁盘)，修改PCB中的头指针，将修改后的PCB存盘(启动1次磁盘)。因此，在不考虑回收盘块的启动磁盘操作的情况下，总共需要启动2次磁盘I/O。

对显式链接，将修改后的内存FAT表写入磁盘(最多涉及2个盘块中的FAT，故最多启动2次磁盘)。总共最多需要启动2次至少启动1次磁盘I/O。

③ 对于索引文件：

将修改后的内存索引表(假设是一级索引)写入索引盘块中(1次I/O，若该文件是多级索引或混合索引，则其索引表由多个盘块存储，就需要启动多次磁盘I/O).。故对于一级索引文件，总共需要启动1次磁盘I/O。

33．设某个文件由100个物理盘块存储，对于连续文件、链接文件和索引文件，分别计算执行下列操作时的启动磁盘I/O次数(假如头指针和索引表均在内存中)：

(1)　把一块加在文件的中间(新加块成为第 51 块)；

(2)　从文件的中间删去一块(删去第 51 块)。

解：(1) 把一块加在文件的中间(第51块)：

① 对于连续文件：

最好的情况是文件末尾盘块的下一盘块是空闲的，此时将文件的后50个块搬家，需启动100次磁盘；写入新增块需启动1次磁盘；将修改后的PCB存盘，需启动1次磁盘I/O。故最好情况下总共需要启动102次磁盘I/O。一般情况下，整个文件需要搬家，这时总共需要启动202次磁盘I/O。

② 对于链接文件：

对隐式链接，为读出第51块的块号，需依次读出前50个盘块(启动50次)；将新增内容及第51块的块号写入一空闲盘块中(启动1次磁盘)；将读出的第50块的链接指针改为新增盘块号后写入原第50块中(启动1次磁盘)。总共需要启动52次磁盘。

对显式链接，将新增内容写入一空闲盘块中(启动1次磁盘)；将修改后的内存FAT表写入磁盘(最多涉及2个盘块中的FAT，故最多启动2次磁盘)；总共最多需要启动3次磁盘I/O。

③ 对于索引文件：

将新增内容写入一空闲盘块中；将修改后的内存索引表(假设是一级索引)写入索引盘块中(启动1次磁盘I/O，若该文件是多级索引或混合索引，则其索引表由多个盘块存储，就可能需要启动多次磁

盘I/O).。故对于一级索引文件，总共需要启动2次磁盘I/O。

(2) 从文件的中间删去一块(删去第51块)：

① 对于连续文件：

将第52块搬家到原第51块处(需启动2次磁盘I/O)；同样，原第53、54、……、100块搬家，各需启动2次磁盘，共启动96次磁盘；读出文件的FCB，修改其长度后存盘(2次磁盘I/O)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动100次磁盘I/O。

② 对于链接文件：

对隐式链接：为读出第52块的块号，需依次读出前51个盘块(启动51次)；假设第50块的内容读入缓冲区B1中，将读出的第51块的指针域(第52块的块号)写入B1的指针域中，将修改后的B1内容写入原第50块中(启动1次磁盘I/O)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动52次磁盘I/O。

对显式链接：将修改后的内存FAT表写入磁盘(最多启动2次磁盘)；总共最多需要启动2次磁盘I/O。

③ 对于索引文件：

将修改后的内存索引表(假设是一级索引或UNIX的混合索引)写入索引盘块中(启动1次磁盘I/O，若该文件是多级索引，则其索引表由多个盘块存储，就可能需要启动多次磁盘I/O).。故对于一级索引或UNIX的混合索引文件，总共需要启动1次磁盘I/O。

34．设某个文件由100个物理盘块存储，对于连续文件、链接文件和索引文件，分别计算执行下列操作时的启动磁盘I/O次数(假如头指针和索引表均在内存中)：

(1) 把一块加在文件的末尾；

(2) 从文件的末尾删去一块。

解：(1) 把一块加在文件的末尾：

① 对于连续文件：

最好的情况是文件末尾盘块的下一盘块是空闲的，将新增内容写入该空闲盘块中(启动1次磁盘)；将修改后的PCB存盘，需启动1次磁盘I/O。故最好情况下总共需要启动2次磁盘I/O。一般情况下，文件需要搬家，这时总共需要启动202次磁盘I/O。

② 对于链接文件：

对隐式链接，为读出第100块以便修改其链接指针，需启动100次磁盘；将新增内容及链表尾指针写入一空闲盘块中(启动1次磁盘)；将读出的第100块的链接指针改为新增盘块号后写入原第100块中(启动1次磁盘)。总共需要启动102次磁盘。

对显式链接，将新增内容写入一空闲盘块中(启动1次磁盘)；将修改后的FAT表写入磁盘(最多启动2次磁盘)；总共最多需要启动3次磁盘I/O。

③ 对于索引文件：

将新增内容写入一空闲盘块中；将修改后的内存索引表(只涉及该文件的最后一个索引盘块)写入索引盘块中(启动1次磁盘I/O).。故对于索引文件，总共需要启动2次磁盘I/O。

(2) 从文件的末尾删去一块：

① 对于连续文件：

读出该文件的PCB(启动1次磁盘I/O)，文件长度(盘块数)减1，将修改后的PCB存盘(启动1次磁盘)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动2次磁盘I/O。

② 对于链接文件：

对隐式链接，为读出第100块的块号，需依次读出前99个盘块(启动99次磁盘I/O)；将读出的第99块的指针改为尾指针后仍存入原第99块中(启动1次磁盘I/O)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动100次磁盘I/O。

对显式链接，将修改后的内存FAT表写入磁盘(最多涉及2个盘块中的FAT，故最多启动2次磁盘)。总共最多需要启动2次至少启动1次磁盘I/O。

③ 对于索引文件：

将修改后的内存索引表存盘(只涉及最后一个索引盘块，故只需启动1次磁盘I/O)。因此，在不考虑回收盘块的磁盘操作的情况下，总共需要启动1次磁盘I/O。

第七章 操作系统接口

（【注】2009考研大纲中不包含这部分内容）

1．选择题

1．____中断是自愿中断。
A．程序　　B．外部　　C．I/O　　D．访管

2．操作系统作业管理的主要功能是_____。
A．作业的调度和控制　　B．作业的输入
C．作业的建立　　D．作业的编制

3．在下述关于操作使用接口的不同论述中，正确的是_____。
A．操作使用接口是用户程序与操作系统之间的接口，因此它不是命令接口
B．操作使用接口是键盘和屏幕
C．操作使用接口是用户程序
D．操作使用接口是一组键盘命令及命令解释程序

4．交互作业的操作使用接口不包含_____。
A．菜单技术　　B．窗口技术　　C．操作控制命令　　D．作业控制语言

5．关于Unix系统不正确的叙述是：_____。
A．Unix由内核和外壳两部分组成
B．系统程序在核心态运行
C．终端用户通过shell使用系统
D．仅当一条shell命令执行完成以后，才能打入下一条shell命令

6．作业的交互控制方式也可称为_____。
A．联机控制方式　　B．批处理控制方式
C．脱机控制方式　　D．多进程控制方式

7．用户程序请求操作系统服务是通过 _____ 实现的。
A．子程序调用指令　　B．访管指令
C．条件转移指令　　D．以上三种都可以

8．用户通过终端使用计算机系统控制作业执行的方式称为___方式。
A．自动　　B．联机　　C．脱机　　D．假脱机

9．UNIX 操作系统的 SHELL 是负责___的模块。
A．解释并执行来自终端的命令　　B．解释并执行来自终端的内部命令
C．解释并执行来自终端的外部命令　　D．进行功能调用

10．系统调用是通过_____进入操作系统。
A．系统中断　　B．外部中断　　C．访管中断　　D．I/O中断

11．当用户程序执行访管指令时，中断装置将使中央处理器_____工作。
A．维持在目态　　B．从目态转换到管态
C．维持在管态　　D．从管态转换到目态

12．下列选项中，操作系统提供给应用程序的接口是_____。（2010全国试题）
A．系统调用　　B．中断　　C．库函数　　D．原语

13．计算机系统中设置的访管指令，____执行。

A．只能在目态　　B．只能在管态

C．既可在目态又可在管态　　D．在目态和管态下都不能

14．下列选项中，不可能在用户态发生的事件是_____。（2012全国试题）

A．系统调用　　B．外部中断　　C．进程切换　　D．缺页

第七章操作系统接口选择题参考答案：

1．D　2．A　3．D　4．D　5．D　6．A　7．B　8．B　9．A　10．C

11．B　12．A　13．A　14．C