

## 第一章习题参考答案

1. 什么是计算机的操作系统？

答：操作系统是一组计算机程序的集合，主要用以控制和管理计算机的硬件和软件资源，合理地组织计算机的工作流程，向应用程序和用户方便、快捷、友好的使用接口。

2. 叙述操作系统的基本功能，结合 windows XP 操作系统举例说明它提供哪几种用户使用计算机的接口？

答：操作系统管理计算机系统资源的基本功能可以分成以下几个方面：进程管理、存储管理、设备管理、文件管理。随着计算机网络的日益普及和安全问题的不断受到重视，现在的操作系统也都具备了基本的网络通信与服务功能和相应的安全与保护机制。

Windows XP 操作系统提供的用户接口有：API、命令界面、图形界面、批处理控制命令

3. 操作系统与编译、语言、数据库、中间件(J2EE、.Net、CORBA)等其它软件的关系如何？

答：在裸机之上的第一层是操作系统内核，它扩充了裸机的功能，实现了操作系统核心安全可信部分的功能；在此基础上可以建立不同特征的操作系统。在操作系统的上面是语言编译系统、数据库管理系统、中间件等支撑软件，它们使计算机成为具有不同功能的虚拟机。也就是说，由于有了语言层，计算机可以识别、运行相应语言程序；由于有了数据库系统层，计算机可以操作相应的数据表格。由于中间件(CORBA、J2EE、.NET)进一步补充、扩展了内层操作系统分布计算功能、系统集成能力。

4. 操作系统分为哪些类型？每种类型的主要特点是什么？

答：根据操作系统在用户界面的使用环境和功能特征的不同，操作系统一般可分为三种基本类型，即批处理操作系统、分时操作系统和实时操作系统。随着计算机体系结构的发展，又出现了许多种操作系统，它们是嵌入式操作系统、个人操作系统、网络操作系统和分布式操作系统。

批处理操作系统中，用户将作业交给系统操作员，系统操作员将许多用户的作业组成一批作业，之后输入到计算机中，在系统中形成一个自动转接的连续的作业流，然后启动操作系统，系统自动、依次执行每个作业。最后由操作员将作业结果交给用户。

分时操作系统是适应用户人机交互需要而出现的。分时系统使用了多道程序设计的方法。它把主机 CPU 的时间划分成很多个小的时间单位，称为时间片，一般多为几十毫秒。在时钟中断机构的支持下，轮流把时间片分配给不同的用户，每道用户程序每次只运行一个时间片。

“实时”就是立即的意思，是指计算机要能及时响应外部事件的请求，并以足够快的速度完成对事件的处理。实时操作系统初期主要用于过程控制和实时信息处理系统。嵌入式操作系统指运行在嵌入式(计算机)环境中，对整个系统各种部件和资源进行统一协调、处理、指挥和控制的系统软件。它具有通常操作系统的基本功能，它与一般操作系统相比又有很大不同，主要体现在微型化、可定制、实时性、可靠性、易移植、开发工具与使用环境密切相关等特点。

个人操作系统面向现个人计算机开发的操作系统，在计算时用户可以完全自由地控制个人计算机。

网络操作系统除了具有通常单机操作系统资源管理功能外，必须提供高效而可靠的网络进程通信能力和基本的分布计算服务（远程文件服务、网络进程管理等）。

分布操作系统也可以定义为通过通信网络将物理上分布的具有自治功能的数据处

理系统或计算机系统互连起来，实现信息交换和资源共享，协作完成任务。分布式计算机系统是计算机网络系统的高级形式，由多台计算机组成，计算机之间没有主次之分。分布式系统的特点是数据和控制及任务的分布性、整体性、资源共享的透明性、各节点的自制性和协同性。

5. 简述操作系统的基本特性。

答：1. 并发性

并发性是两个或多个事件在同一时间间隔内发生的、同时处于活动状态的特性。在单处理器多道程序环境下，在一段时间内有多道程序同时处于活跃状态，每一时刻仅能执行一道程序，微观上这些程序是在交替执行，是串行的，宏观上这些程序都在运行，是并发的。

2. 共享性

所谓共享，主要是指资源共享。内存中并发执行的多个程序可以共享计算机的硬件和软件资源。

3. 虚拟性

虚拟性是指将一个物理实体映射为一个或多个逻辑对象。物理实体是客观存在的，而逻辑对象是虚构的，是一种感觉上的存在，是主观上的一种标识。

4. 随机性

随机性也叫异步性，指的是每道程序在何时执行，各个程序执行的顺序，以及每道程序所需的时间都是不确定的，也是不可预知的。

6. 操作系统的基本结构有哪些？

答：1. 无结构操作系统、2. 模块化结构、3. 分层结构、4. 客户/服务器结构、5. 虚拟机结构、6. 面向对象的结构

7. 什么叫多道程序设计？其主要特点是什么？

答：多道程序设计的基本思想就是在内存中同时存放几道程序，它们交替执行。多道程序设计充分地利用 CPU，并使计算机的其他资源，如通道和通道控制的输入输出设备更好地并行运行，大大提高了计算机系统的性能。在多道程序系统中，多道程序在微观上是交替地顺序执行，即在某一时刻只有一道程序在执行。在宏观上，各道程序都向前推进了。

## 习题 2

1-7 题，见教材相关章节内容。

8. Job1 与 Job2 同时到达，根据短作业优先算法，这时应该先执行 Job2，Job2 执行完毕时 9.0，系统中有 job1 与 job3 两个作业，job3 较短，所以先执行 job3，job3 执行完毕时 9.7，系统中有 job1 和 job4，job4 较短，先执行 job4，再执行 job1，所以他们四个作业的执行顺序为 Job2、Job3、Job4、Job1；

Job2 的开始时刻是 8.0，完成时刻是 9.0、周转时间 1，带权周转时间 1；

Job3 的开始时刻是 9.0，完成时刻是 9.7、周转时间 1.3，带权周转时间 1.86；

Job4 的开始时刻是 9.7，完成时刻是 10.0，周转时间 0.8，带权周转时间 2.7；

Job1 的开始时刻是 10.0，完成时刻是 12.0，周转时间 4.0，带权周转时间 2；

平均周转时间为 1.78；

平均带权周转时间为 1.89。

9. 盘子是个互斥使用的通用类型缓冲区，可存放两种不同类型数据，父母=两个特定类型数据/产品生产者，子女=两个特定类型数据/产品消费者。

```
enum{apple, orange} plate;
semaphore sCap, sOrg, sApp;    //盘子里可以放水果数量, 盘子里有无桔子/苹果
sCap = 1;    // 盘子容量: 一个水果
sOrg = 0;    // 盘子里没有桔子
sApp = 0;    // 盘子里没有苹果
cobegin
process father {
    while(true){
        削一个苹果;
        P(sCap);
        把苹果放入 plate;
        V(sApp);
    }
}
process mother {
    while(true){
        剥一个桔子;
        P(sCap);
        把桔子放入 plate;
        V(sOrg);
    }
}
coend

process son{
    while(true){
        P(sOrg);
        从 plate 中取桔子;
        V(sCap);
        吃桔子;
    }
}
process daughter {
    while(true){
        P(sApp);
        从 plate 中取苹果;
        V(sCap);
        吃苹果;
    }
}
```

10.

```
semaphore S=1, SO=0, SS=0, SW=0;    //容器是否可用, 容器中是浓缩汁/糖/水
```

```

enum { sugar, water, orange } container;
cobegin
process Provider {
    while(true){
        P(S);
        将原料装入容器内;
        if (cantainer==orange) V(SO);
        else if (cantainer==sugar) V(SS);
        else V(SW);
    }
}
process P1 {
    while(true){
        P(SO);
        从容器中取浓缩汁;
        V(S);
        生产橙汁;
    }
}
coend

process P2 {
    while(true){
        P(SS);
        从容器中取糖;
        V(S);
        生产橙汁;
    }
}

process P3 {
    while(true){
        P(SW);
        从容器中取水;
        V(S);
        生产橙汁;
    }
}

```

11.

答：系统的剩余资源向量  $A=(1,2,3,0)$ ，各进程的剩余请求矩阵

$$R = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 2 & 1 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 6 & 2 & 2 \end{pmatrix}$$

(1) 由于系统存在一个进程完成的安全序列 P4、P1、P2、P3、P5，故系统状态安全；

进程	可用资源	请求资源	占有资源	分配释放量	完成情况
P1		1,2,0,0	0,0,1,2		
P2		0,7,5,0	1,0,0,0		
P3		1,2,1,2	1,1,4,4		
P4	1,2,3,0	0,2,2,0	0,6,3,2		1 true
P5		0,6,2,2	0,0,1,4		

进程	可用资源	请求资源	占有资源	分配释放量	完成情况
P1	1,8,6,2	1,2,0,0	0,0,1,2	1,8,7,4	2 true
P2		0,7,5,0	1,0,0,0		
P3		1,2,1,2	1,1,4,4		
P4	1,2,3,0	0,2,2,0	0,6,3,2	1,8,6,2	1 true
P5		0,6,2,2	0,0,1,4		

进程	可用资源	请求资源	占有资源	分配释放量	完成情况
P1	1,8,6,2	1,2,0,0	0,0,,1,2	1,8,7,4	2 true
P2	1,8,7,4	0,7,5,0	1,0,0,0	2,8,7,4	3 true
P3		1,2,1,2	1,1,4,4		
P4	1,2,3,0	0,2,2,0	0,6,3,2	1,8,6,2	1 true
P5		0,6,2,2	0,0,1,4		

进程	可用资源	请求资源	占有资源	分配释放量	完成情况
P1	1,8,6,2	1,2,0,0	0,0,,1,2	1,8,7,4	2 true
P2	1,8,7,4	0,7,5,0	1,0,0,0	2,8,7,4	3 true
P3	2,8,7,4	1,2,1,2	1,1,4,4	3,9,11,8	4 true
P4	1,2,3,0	0,2,2,0	0,6,3,2	1,8,6,2	1 true
P5		0,6,2,2	0,0,1,4		

进程	可用资源	请求资源	占有资源	分配释放量	完成情况
P1	1,8,6,2	1,2,0,0	0,0,,1,2	1,8,7,4	2 true
P2	1,8,7,4	0,7,5,0	1,0,0,0	2,8,7,4	3 true
P3	2,8,7,4	1,2,1,2	1,1,4,4	3,9,11,8	4 true
P4	1,2,3,0	0,2,2,0	0,6,3,2	1,8,6,2	1 true
P5	3,9,11,8	0,6,2,2	0,0,1,4	3,9,12,12	5 true

(2) 进程 P3 提出对资源 R3 的剩余请求为 1，由于系统剩余资源向量  $A=(1, 2, 2, 0)$ ，所以假定分配给它，此时仍能找到一个与 (1) 相同的安全序列，即可以分配；

(3) 系统初始配置的各类资源分别为 (3, 9, 12, 12)。

补充 (二)

一、分析 P、V 操作执行期间为什么要屏蔽中断？试举例说明。

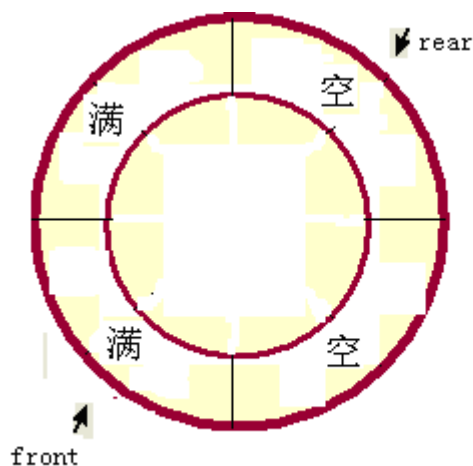
参考解答：

假设现有 A、B 两进程要互斥进入临界区 CS，互斥信号量 S.value 初值为 1。假设 P 操作不采用原语方式，如果此时 A 进程欲进入临界区 CS，那么它首先作 P (S) 操作。在执行完 S.value: =S.value -1 后 S.value 值变为 0，此时由于某种原因,分配给 A 进程的时间片结束，停止 A 进程运行，系统转而调用 B 进程。由于此时 S.value =0，B 进程作 P (S) 操作，S.value=S.value -1，S.Value 的值变为-1，B 进程被封锁，无法进入临界区。下次 A 进程重新轮到执行时，它从断点处开始，A 进程直接作 if S.Value <0 then Block(S.blocklist)由于此时 S 值为-1，因此 A 进程也被封锁，无法进入临界区 CS，此时临界区 CS 是处于空闲的状态，但 A、B 进程都不能进入此临界区，造成错误。由此可以知道 P 操作需要采用原语方式，即在它执行过程中不允许中断。对于 V 操作，请读者自己举例说明也必须是原语。

由上面分析，P、V 操作实现临界区进程互斥，是用简短的 P、V 操作屏蔽中断，即采

用原语的方式执行，从而使临界区可以开放中断，同时实现进程互斥。

二、m 个生产者进程和 n 个消费者进程共享缓冲区 Buffer[i] (i=1, 2, 3...k)。生产者进程循环地生产产品、把产品依次送入缓冲区，消费者循环地依次从中取出产品消费。缓冲区构成一个环形缓冲池。如下图所示，“满”的单元表示该缓冲区单元放有产品，“空”的单元表示该缓冲区单元为空。rear 指向生产者下次存入产品的单元，front 表示消费者下次取出产品的单元。利用 p、v 操作，实现他们的同步（设置信号量、设置其初值、画出同步流程）。说明您设置的信号量可能的最大值和可能的最小值，何时出现可能的最大值和可能的最小值的情况。



答：设缓冲区空信号量 empty, empty.value=k;  
 设缓冲区满信号量 full, full.value=0;  
 设进程互斥信号量 mutex, mutex.value=1;  
 生产进程                      消费进程

Loop :P(empty)	loop: p(full)
P(mutex)	p(mutex)
向缓冲区存产品	从缓冲区取产品
V(mutex)	V(mutex)
V(full)	V(empty)
Goto loop	goto loop

Empty 可能最大值为  $k$ , 最小值为  $-m$ 。

生产进程都没有向缓冲区装入产品，缓冲区全空，Empty 取得可能的最大值  $k$ 。

缓冲区全满,  $m$  个生产进程继续要求向缓冲区装入产品, 一个个都做了  $P(empty)$ ,  $empty$  取得可能的最小值  $-m$ 。

Full 可能最大值为  $k$ , 最小值为  $-n$

生产进程向缓冲区装满产品，缓冲区全满，full 取得可能的最大值。

缓冲区全空,  $n$  个消费进程要求消费, 一个个都做了  $P(full)$ ,  $full$  取得可能的最小值  $-n$ 。

$mutex$  最大值为 1, 如果  $m+n \geq k$ ,  $mutex$  最小值为  $-(k-1)$   
· 如果  $m+n < k$ ,  $mutex$  最小值为  $-(m+n-1)$

无进程进入缓冲区时,  $mutex$  有最大值 1。

## 第3章 习题参考答案

### 1. 内存管理的基本功能是什么？

答案：操作系统的内存管理主要包括以下功能：

1. 内存的分配和回收：操作系统根据用户程序的请求，在内存中按照一定算法把找到一块空闲，将其分配给申请者；并负责把释放的内存空间收回，使之变为空闲区。
2. 提高内存的利用率：通过多道程序共享内存，提高内存资源的利用率。
3. 通过虚拟存储技术“扩充”内存容量：使用户程序在比实际内存容量大的情况下，也能在内存中运行。
4. 内存信息保护：保证各个用户程序或进程在各自规定的存储区域内操作，不破坏操作系统区的信息，并且互不干扰。

### 2. 什么是逻辑地址，什么是物理地址，为什么要实现地址转换？

答案：用户的源程序通常用高级语言编写，源程序通过编译或汇编后得到目标程序。目标程序使用的地址称为逻辑地址，也叫相对地址；程序在物理内存中的实际存储单元称为物理地址，也叫绝对地址。只有把程序和数据的逻辑地址转换为物理地址，程序才能正确运行，该过程称为地址转换或地址重定位。

### 3. 可变分区存储管理中有哪些内存分配方法？比较它们的优缺点。

答案：有最先适应分配算法，循环首次适应分配算法，最优适应分配算法，最坏适应分配算法，快速适应算法。

从搜索空闲区速度及内存利用率来看，最先适应分配算法、循环首次适应分配算法和最优适应算法比最坏适应算法性能好。如果空闲区按从小到大排列，则最先适应分配算法等于最优适应分配算法。反之，如果空闲区按从大到小排列，则最先适用分配算法等于最坏适应分配算法。空闲区按从小到大排列时，最先适应分配算法能尽可能使用低地址区，从而，在高地址空间有较多较大的空闲区来容纳大的作业。下次适应分配算法会使存储器空间得到均衡使用。最优适应分配算法的内存利用率最好，因为，它把刚好或最接近申请要求的空闲区分给作业；但是它可能会导致空闲区分割下来的部分很小。在处理某种作业序列时，最坏适应分配算法可能性能最佳，因为，它选择最大空闲区，使得分配后剩余下来的空闲区不会太小，仍能用于再分配。由于最先适应算法简单、快速，在实际的操作系统中用得较多；其次



是最优适应算法和下次适应算法。

#### 4. 什么是紧凑技术，什么情况下采用？

答案：紧凑技术就是把内存中的作业改变存放区域，使分散的空闲区能够汇聚成一个大的空闲区，从而有利于大作业的装入。

当系统运行一段时间后，内存被多次分配和回收，会产生许多不连续的空闲空间。有可能出现这样的现象：内存中每一块空闲空间都不能满足某一作业的内存请求，而所有空闲空间的总和却能满足该作业，这时可采用紧凑技术。

#### 5. 什么是交换技术？什么情况下采用？（将“移动”更正为“交换”）

答案：为了释放部分内存空间，由操作系统根据需要，将某些暂时不运行的进程或程序段从内存移到外存的交换区中；当内存空间富余时再给被移出的进程或程序段重新分配内存，让其进入内存，这就是交换技术，又称为“对换”或“滚进/滚出(roll-in/roll-out)。

当剩余内存不足，而又有新的进程要进入内存时，可应用交换技术。

#### 6. 叙述页式存储管理系统中的地址转换过程。

答案：当处理器给出某个需要访问的逻辑地址时，地址转换机构自动地从逻辑地址的低地址部分得到页内偏移，从高地址部分得到页号。将页号与页表寄存器中的页表长度进行比较，如果页号大于或等于页表长度，表示该页在页表中没有相应项，本次所访问的地址已经超越进程的地址空间，会产生地址越界中断；否则，从页表寄存器得到页表在内存中的起始地址。将页号和页表项长度的乘积再加上页表的起始地址，得到该页的页表项在页表中的位置，从而可以查到该页在内存中的物理块号。最后，将页内偏移装入物理地址寄存器的低位字段中，将物理块号装入物理地址寄存器的高位字段中，此时物理地址寄存器中的内容就是地址转换机构给出的物理地址。

#### 7. 比较分页与分段存储管理的差异。

答案：段是信息的逻辑单位，由源程序的逻辑结构所决定，用户可见，段长可根据用户需要来规定，段起始地址可以从任何地址开始。在分段方式中，源程序(段号，段内偏移)经连结装配后仍保持二维结构。

页是信息的物理单位，与源程序的逻辑结构无关，用户不可见，页长由系统确定，页面

只能以页大小的整倍数地址开始。在分页方式中，源程序(页号，页内偏移)经连结装配后变成了一维结构。

#### 8. 页式存储管理中，试分析大页面与小页面各自的优缺点。

答案：如果页面较小，页数就要增加，页表也随之扩大，为了控制页表所占的内存空间，应选择较大的页面尺寸。

内存以块为单位，一般情况下进程的最后一个页面总是装不满一个物理块，会产生内部碎片，为了减少内部碎片，应选择小的页面尺寸。

作业存放在辅助存储器上，从磁盘读入一个页面的时间包括等待时间（移臂时间+旋转时间）和传输时间，通常等待时间远大于传输时间。显然，加大页面的尺寸，有利于提高 I/O 的效率。

综合考虑以上几点，现代操作系统中，页面大小大多选择在 512 B 到 4KB 之间。

#### 9. 段页式存储管理中怎样划分逻辑地址空间？

答案：段页式存储管理的基本原理。

1. 程序根据自身的逻辑结构划分成若干段，这是段页式存储管理的段式特征。
2. 内存的物理地址空间划分成大小相等的物理块，这是段页式存储管理的页式特征。
3. 将每一段的线性地址空间划分成与物理块大小相等的页面，于是形成了段页式存储管理。

段号(s)	段内页号 (p)	页内位移(d)
-------	----------	---------

4. 逻辑地址分 3 个部分：段号、段内页号和页内位移，其形式为：

#### 10. 如果一个分页系统能够向用户提供的逻辑地址最大为 16 页，页面大小为 2K，内存总共有 8 个存储块。请问逻辑地址应该为多少位？内存空间为多大？

答案：因为逻辑地址最大页数量为  $16=2^4$ ，所以逻辑地址中，表示页号需要用 4 位；

又因为页面大小为  $2K=2^{11}$ ，表示业地址需要用 11 位，所以整个逻辑地址应该为  $4+11=15$ （位）

因为页面大小为  $2K$ ，所以内存块大小也是  $2K$ ，又因为共有 8 个物理内存块，所以总共内存空间为  $8*2K = 16K$ 。

11. 什么是虚拟存储技术？叙述实现虚拟存储管理的基本思想。

答案：将作业不执行的部分暂时存放在外存，待到进程需要时，再将其从外存调入内存。将外存作为内存的补充，从逻辑上扩充内存，这就是虚拟存储技术。

虚拟存储技术的思想：将外存作为内存的扩充，作业运行不需要将作业的全部信息放入内存，将暂时将不运行的作业信息放在外存，通过内存与外存之间的对换，使系统逐步将作业信息放入内存，最终达到能够运行整个作业，从逻辑上扩充内存的目的。

12. 请求页式虚拟存储管理中的页面置换算法有哪几种？各有何特点？

答案：**先进先出（FIFO）页面置换算法**：先进先出页面置换算法开销低、容易编程实现，适合于线性顺序特性好的程序。但是该算法没有考虑到页面的访问频率，很可能刚被换出的页面马上又要被访问，使得缺页率偏高。

**最佳（OPT）页面置换算法**：最佳页面置换算法具有最低的缺页率。最佳页面置换算法只是一种理想化的页面调度算法，很难实现。但是，该算法可以作为评判其它的置换算法的准则

**最近最久未使用（LRU）页面置换算法**：LRU 算法能够合理地预测程序运行状态，具有很好的置换性能，被公认为是一种性能好且可以实现的页面置换算法，但是 LRU 算法在实现起来比较复杂。

**时钟（clock）置换算法**：兼顾了效率与公平，是一种性能较好的算法。

13. 请指出缺页中断和一般中断的区别。

答案：1) CPU 检测中断的时间不同。对一般的中断信号，CPU 是在一条指令执行完后检测其是否存在，检测时间以一个指令周期为间隔。而对缺页中断信号，CPU 在一条指令执行期间，只要有中断信息就可检测，不需要等待一个指令周期。因此，CPU 检测缺页中断更及时。

2) CPU 可以多次处理。如果在一个指令周期中多次检测到缺页中断，CPU 都会及时处

理。

14. 一个 32 位地址的计算机系统使用二级页表，虚拟地址为：顶级页表占 9 位，二级页表占 11 位。请问：页面长度为多少？虚拟地址空间有多少个页面？  
答案：页内偏移量所在地址位数为  $32-9-11=12$  位，所以页面长度为  $2^{12}=4K$ ，虚拟地址空间为  $2^{32}$ ，所以页面个数为  $2^{32}/2^{12}=2^{20}=1M$  个。

15. 在一个请求分页的虚拟存储器管理中，一个程序的运行页面走向为：

1、2、3、4、2、3、5、6、3、1、4、6、7、5、2、4、1、3、2

如果为程序分配的物理块分别为 3 个、4 个，请分别用 FIFO、OPT 和 LRU 算法求出缺页中断次数和缺页率。

答案：(1)页框为 3：

FIFO:

页面走向	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	2			3	4	5	6	3	1	4	6	7	5	2	4	1
		2	2	3			4	5	6	3	1	4	6	7	5	2	4	1	3
			3	4			5	6	3	1	4	6	7	5	2	4	1	3	2
是否缺页				缺			缺	缺	缺	缺	缺	缺	缺	缺	缺	缺	缺	缺	缺

FIFO 缺页中断次数为 14；缺页率为 14/19。

OPT:

页面走向	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	4			4	4		4			4	4	4			3	
		2	2	2			5	6		6			7	5	2			2	
			3	3			3	3		1			1	1	1			1	
是否缺页				缺			缺	缺		缺			缺	缺	缺			缺	

OPT 缺页中断次数为 8；缺页率为 8/19。

LRU:

页 面 走 向	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	4			5	5		1	1	1	7	7	7	4	4	4	2
		2	2	2			2	6		6	4	4	4	5	5	5	1	1	1
			3	3			3	3		3	3	6	6	6	2	2	2	3	3
是 否 缺 页				缺			缺	缺		缺	缺	缺	缺	缺	缺	缺	缺	缺	缺

LRU 缺页中断次数为 13；缺页率为 13/19。

(2) 页框为 4:

FIFO:

页 面 走 向	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	1			2	3		4			5		6	1		7	
		2	2	2			3	4		5			6		1	7		2	
			3	3			4	5		6			1		7	2		4	
				4			5	6		1			7		2	4		3	
是 否 缺 页							缺	缺		缺			缺		缺	缺		缺	

FIFO 缺页中断次数为 7；缺页率为 7/19。

OPT:

页 面 走 向	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
	1	1	1	1			1	1					1	1	1				
		2	2	2			5	6					7	5	2				
			3	3			3	3					3	3	3				
				4			4	4					4	4	4				
是 否 缺 页							缺	缺					缺	缺	缺				

页																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

OPT 缺页中断次数为 5；缺页率为 5/19。

LRU:

页 面	1	2	3	4	2	3	5	6	3	1	4	6	7	5	2	4	1	3	2
走 向																			
	1	1	1	1			5	5		5	4		4	4	2	2	2	2	
		2	2	2			2	2		1	1		1	5	5	5	5	3	
			3	3			3	3		3	3		7	7	7	7	1	1	
				4			4	6		6	6		6	6	6	4	4	4	
是 否 缺 页							缺	缺		缺	缺		缺	缺	缺	缺	缺	缺	

LRU 缺页中断次数为 10；缺页率为 10/19。

补充（一）:

1、某虚拟存储器的用户编程空间共32个页面，每页为1KB，内存为16KB。假定某时刻一用户页表中已调入内存的页面的页号和物理块号的对照表如下：

页号	物理块号
0	5
1	10
2	4
3	7

则逻辑地址 0A5C(H)所对应的物理地址是什么？

参考解答：125C（H） （要求写出计算步骤）

[分析]页式存储管理的逻辑地址分为两部分：页号和页内地址。

由已知条件“用户编程空间共 32 个页面”，可知页号部分占 5 位；由“每页为 1KB”， $1K=2^{10}$ ，可知内页地址占 10 位。由“内存为 16KB”，可知有 16 块，块号为 4 位。

逻辑地址 0A5C（H）所对应的二进制表示形式是：000 1010 0101 1100，根据上面的分析，下划线部分为页内地址，编码“000 10”为页号，表示该逻辑地址对应的页号为 2。查页表，得到物理块号是 4（十进制），即物理块地址为：01 00，拼接块内地址 10 0101 1100，得 01 0010 0101 1100，即 125C（H）。

2、某电脑采用页式存储管理方式，该系统允许作业最大的逻辑地址有 64 页，每页 1024 个字节，主存则被划分为 256 块。请问，逻辑地址需要多少位来表示？表示主存的地址又需多少位？

解：因为逻辑地址=逻辑页号+页内位移，逻辑地址有 64 页，表示 64 需要二进制 6 位（ $64=2^6$ ），每页 1024 个字节，表示页内位移需要 10 位二进制数（ $1024=2^{10}$ ）。所以逻辑地址需 16 位。同理，可知表示主存的物理地址需要 18 位。

3、下表给出了某系统中的空闲分区表，系统采用可变式分区存储管理策略。现有以下作业序列：96K、20K、200K。若采用首次适应算法和最佳适应算法来处理这些作业序列，试问哪一种算法可以满足该作业序列的请求，为什么？（4分）

分区号	大小	起始地址
1	32K	100K
2	10K	150K
3	5K	200K
4	218K	220K
5	96K	530K

分析：

首次适应算法要求空闲分区按地址递增的次序排列，在进行内存分配时，总是从空闲分区表首开始顺序查找，直到找到第一个能满足其大小要求的空闲分区为止。然后，再按照作业大小，从该分区中划出一块内存空间分配给请求者，余下的空闲分区仍留在空闲分区表中。

最佳适应算法要求空闲分区按大小递增的次序排列，在进行内存分配时，总是从空闲分区表首开始顺序查找，直到找到第一个能满足其大小要求的空闲分区为止。如果该空闲分区大于作业的大小，则与首次适应算法相同，将剩余空闲区仍留在空闲区表中。

解：

若采用首次适应算法，在申请 96K 存储区时，选中的是 4 号分区，进行分配后 4 号分区还剩下 122K；接着申请 20K，选中 1 号分区，分配后剩下 12K；最后申请 200K，现有的五个分区都无法满足要求，该作业等待。显然采用首次适应算法进行内存分配，无法满足该作业序列的需求。这时的空闲分区表如下表所示。

采用首次适应算法分配后的空闲分区表

分区号	大小	起始地址
1	12K	120K
2	10K	150K
3	5K	200K
4	122K	316K
5	96K	530K

若采用最佳适应算法，在申请 96K 存储区时，选中的是 5 号分区，5 号分区大小与申请空间大小一致，应从空闲分区表中删去该表项；接着申请 20K 时，选中 1 号分区，分配后 1 号分区还剩下 12K；最后申请 200K，选中 4 号分区，分配后剩下 18K。显然采用最佳适应算法进行内存分配，可以满足该作业序列的需求。为作业序列分配了内存空间后，空闲分区表如下表所示。

最佳适应算法分配后的空闲分区表

分区号	大小	起始地址
1	12K	120K
2	10K	150K
3	5K	200K
4	18K	420K

4、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），且已知该作业的页面映象表如图。试借用地址变换图求出有效逻辑地址 4865 所对应的物理地址。

页号	块号
0	2
1	4
2	6
3	8

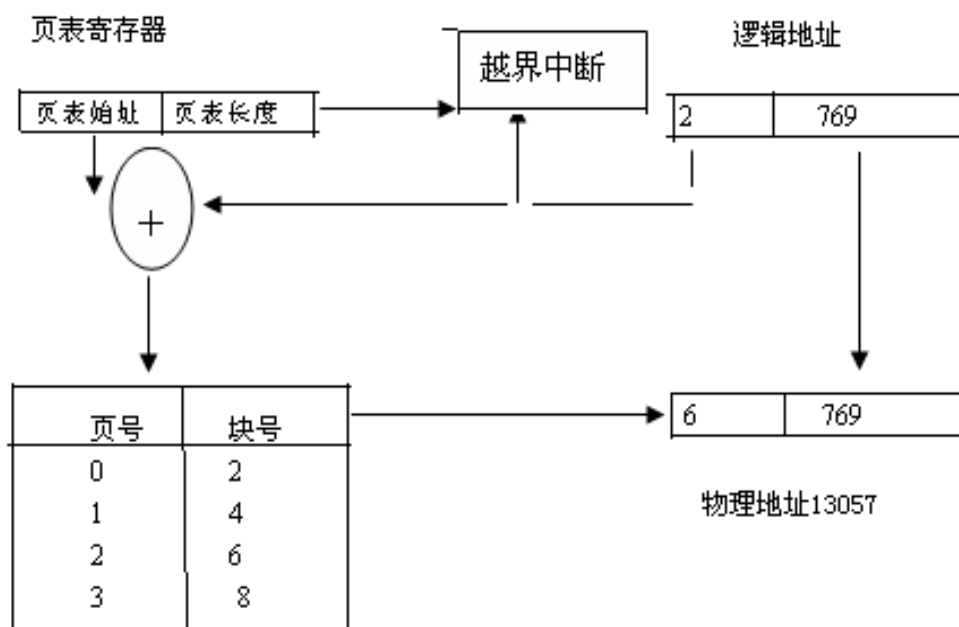
一页大小为 2048 字节，则逻辑地址 4865 的页号及页内位移为：

页号：4865/2048=2

页内位移：4865%2048=769

通过页表得知页面 2 存放在物理块 6 中，将物理块号与逻辑地址中的页内位移拼接，形成物理地址，即：6\*2048+769=13057

其地址变换过程如图：



5、在一个分段存储管理系统中，其段表如图，求逻辑地址对应的物理地址是什么？



段号	内存起始地址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

在分段存储管理系统中，为了实现从逻辑地址到物理地址的转换，系统将逻辑地址中的段号与段表长度进行比较，若段号超过了段表，则表示段号太大，于是产生越界中断信号，若未越界，则根据段表始址和段号计算出该段对应段表项的位置，从中读取该段在内存的起始地址，然后，再检查段内地址是否超过该段的段长，若超过则同样发出越界中断信号。若未越界，则将该段的起始地址与段内位移相加，从而得到了要访问的物理地址。

(1) 由于第 0 段的内存始址为 210，段长为 500，故逻辑地址 (0.430) 是合法地址。逻辑地址 (0.430) 对应的物理地址为：210+430=640

(2) 由于第 1 段的内存始址为 2350，段长为 20，故逻辑地址 (1.10) 是合法地址，逻辑地址 (1.10) 对应的物理地址为 2350+10=2360

(3) 由于第 2 段起始地址为 100，段长为 90，逻辑地址 (2.500) 的段内位移超过了段长，故该地址为非法地址。

(4) 由于第 3 段的内存始址为 1350，段长为 590，故逻辑地址 (3.400) 是合法地址。逻辑地址 (3.400) 对应的物理地址为：1350+400=1750

(5) 由于第 4 段的内存始址为 1938，段长为 95，逻辑地址 (4.112) 的段内位移超过了段长，故该地址为非法地址。

(6) 由于系统中不存在第 5 段，故逻辑地址 (5.32) 是非法地址。

补充 (二)

一、设有 8 页的逻辑空间，每页有 1024 字，它们被映射到 32 块的物理主存区中。那么逻辑地址的有效位是 (13) 位，物理地址至少为 (15) 位。

二、如果一个分页系统能够向用户提供的逻辑地址最大为 16 页，页面大小为 2K，内存总共有 8 个存储块。请问逻辑地址应该为多少位？内存空间为多大？

三、某请求页式管理系统，用户编程空间有 40 个页面，每个页面为 200H 字节，假定某时刻用户页表中虚页号和物理块号对照表如下：

虚页号：	0	2	5	17	20
物理块号：	5	20	8	14	36

求：虚地址 0A3CH、223CH 分别对应的物理地址。(提示：可以都转换成二进制做，也可

以转换成十进制做)

(一) 转换成十进制计算过程:

页的大小 200H, 即 512 字节

(1) 虚地址 0A3CH 转换成十进制为 2620,  $2620/512$  得到页号为 5, 页内地址为 60, 查页面得对应块号为 8. 因此该地址对应的物理地址为:  $8*512+60=4156$  即 103CH。

(2) 虚地址 223CH 转换成十进制为 8764,  $8764/512$  得到页号为 17, 页内地址为 60, 查页面得对应块号为 14. 因此该地址对应的物理地址为:  $14*512+60=7228$ , 即 1C3CH。

(二) 二进制计算过程:

页的大小 200H, 即 512 字节  $=2^9$ , 页内偏移量需要 9 个位。逻辑页 40 个, 需要 6 个位来表示。

(1) 虚地址 0A3CH 转换成二进制为 0000 1010 0011 1100, 低 9 位 (0 0011 1100) 页内偏移量, 剩余高位 (0000101) 为页号, 即 5 号页, 查页面得对应块号为 8, 即 0001000, 所以, 对应物理地址为: 0001 0000 0011 1100, 即 103CH。

(2) 虚地址 223CH 转换成二进制为 0010 0010 0011 1100, 低 9 位 (0 0011 1100) 页内偏移量, 剩余高位 (0010001) 为页号, 即 17 号页, 查页面得对应块号为 14, 即 0001110, 所以, 对应物理地址为: 0001 1100 0011 1100, 即 1C3CH。

## 第四章习题参考答案

1、举例说明四种 I/O 控制方式的特点及其使用的场合。

答：按照 I/O 控制器与 CPU 之间联系方式的不同，可把 I/O 设备的控制方式分为四类：查询方式、中断方式、DMA 方式和通道方式，它们的主要差别在于中央处理器和外围设备并行工作的方式不同，并行工作的程度不同。中央处理器和外围设备并行工作有重要意义，它能大幅度提高计算机效率和系统资源的利用率。

在简单的微型计算机系统中，经常采用程序循环查询 I/O 方式。CPU 要不断的发送 I/O 测试指令用以测试设备控制器的忙/闲 (busy) 标志位，若设备不忙，则主存与外部设备交换一个字符或一个字。若设备忙，则 I/O 测试指令不断对它进行测试，直到设备空闲为止。由于 CPU 的速度远高于 I/O 设备的速度，使得 CPU 绝大部分时间都处于等待 I/O 完成的循环测试之中。显然，这对 CPU 是极大的浪费。但是，它的控制简单，在 CPU 速度慢、要求不高的场合下常被采用。

在查询方式中，CPU 之所以要不断地测试 I/O 设备的状态，就是因为在 CPU 中无中断机构，使 I/O 设备无法向 CPU 报告它已完成了一个字符的输入操作。引入中断机构是为了消除设备驱动程序不断地轮询控制器状态寄存器的开销，当 I/O 操作结束后，由设备控制器“自动地”通知设备驱动程序。在采用中断驱动进行传输数据的过程中，由于无须 CPU 干预，仅当输完数据时，才需 CPU 花费极短的时间去做些中断处理，因而可使 CPU 与 I/O 设备在一定程度上并行工作，从而提高了整个系统的资源利用率及吞吐量。例如，从键盘键入一个字符的时间约为 200 ms，而将字符送入键盘缓冲区的时间小于 0.2 ms。采用程序 I/O 方式，CPU 约有 199.8 ms 的时间因为查询忙而处于等待中。若采用中断驱动方式，CPU 可利用这 199.8 ms 的时间去做其它事情，而仅用 0.2 ms 的时间来处理由控制器发来的中断请求。可见，中断驱动方式可以很大程度上提高 CPU 的利用率。

中断处理方式提高了主机的利用率，但是每次中断都要保存现场信息、恢复现场等，仍占用大量 CPU 时间。在传送数据量大、速度高的情况下，频繁地中断就不适应了。目前个人计算机块设备传输系统中，都普遍采用了 DMA (Direct Memory Access) 方式，这是一种直接内存存取方式，通过 DMA 控制器 DMAC 控制从内存向设备输入输出。当进程要求输入时，CPU 首先为 DMA 设置输入的字节数、输入到内存的地址，开放中断，启动 DMA。然后 CPU 和 DMA 并行工作。DMA 每传送一个数据并不产生中断，只是字数计数器减 1，内存地址寄存器加 1。传送字数计数器减至 0 时，也就是本次 DMA 传送的数据全部传送完毕时，产生中断，请求 CPU 进行结束处理。

DMA 方式的具有三个特点：一是数据传输的基本单位是数据块，即每次传送至少一个数据块；二是所传送的数据是从设备直接送入内存，或者直接读出内存的；三是在传输时 CPU 参与更少，仅在传送一个或多个数据块的开始和结束时，才需 CPU 干预，整块数据的传送是在控制器的控制下完成的。

DMA 仅负责在高速外围设备与内存之间批量数据的传输工作，由于不对数据做再加工，进一步提高了 CPU 与 I/O 设备的并行操作程度。

I/O 通道方式是 DMA 方式的发展，它可进一步减少 CPU 的干预，即把对一个数据块的读(或写)为单位的干预，减少为对一组数据块的读(或写)及有关的管理和控制为单位的干预。可以把通道看成一个比 DMA 功能更强的接口设备，它有专门用于 I/O 的处理单元。在进行 I/O 操作时，接受 CPU 的委托，独立地执行自己的通道程序来实现内存与外设之间的数据传输。当主机委托的 I/O 任务完成后，通道发出中断，请求 CPU 进行结束处理。这使 CPU 从对 I/O 设备的繁忙直接控制中解脱出来，极大地提高了 CPU 与外设并行工作的程度，从而

更有效地提高整个系统的资源利用率。例如，当 CPU 要完成一组相关的读(或写)操作及有关控制时，只需向 I/O 通道发送一条 I/O 指令，以给出其所要执行的通道程序的首址和要访问的 I/O 设备，通道接到该指令后，通过执行通道程序便可完成 CPU 指定的 I/O 任务。

2、何谓设备独立性，它有什么好处？在操作系统中如何体现？

答：所谓设备独立性，也称为设备无关性，是指在用户程序中不要直接使用物理设备名（或设备的物理地址），而只能使用逻辑设备名。逻辑设备是实际物理设备属性的抽象，它不限于某类具体设备。逻辑设备究竟和哪一个具体的物理设备相对应，还要由系统根据当时的设备忙、闲情况来决定或由系统管理员指定。在应用程序中使用逻辑设备名来请求使用某类设备，而系统在实际执行时，使用物理设备名。

为了实现设备独立性，就必须在驱动程序之上设置一层与设备无关的软件，其主要功能如下：

(1) 向用户层软件提供统一接口。无论哪种设备，它们向用户所提供的接口相同。例如对各种设备的读操作，在应用程序中都用 read,而写操作则都用 write。

(2) 设备无关程序负责将设备名映射到相应的设备驱动程序。

(3) 设备保护。操作系统应向各个用户赋予不同的设备访问权限，以实现设备的保护。

(4) 协调不同设备数据块的差异。设备无关软件屏蔽了不同设备使用的数据块大小可能不同的特点，向用户软件提供了统一的逻辑块大小，即缓冲。

(5) 差错控制。由于在 I/O 操作中的绝大多数错误都与设备关，所以主要由设备驱动程序来处理，而与设备无关软件只处理哪些设备驱动程序无法处理的错误。

3、简述 SPOOLing 技术的目的和实现方法。

答：当系统中引入了多道程序技术后，可以利用其中的一道程序，来模拟脱机输入输出时的外围控制机功能，把低速 I/O 设备上的数据传送到高速磁盘上；或者把数据从磁盘传送到低速输出设备上。这样，便可在主机的直接控制下，实现脱机输入输出功能。此时的外围操作与 CPU 对数据的处理同时进行，这种在联机情况下实现的同时外围操作称为 SPOOLing，或称为假脱机系统。

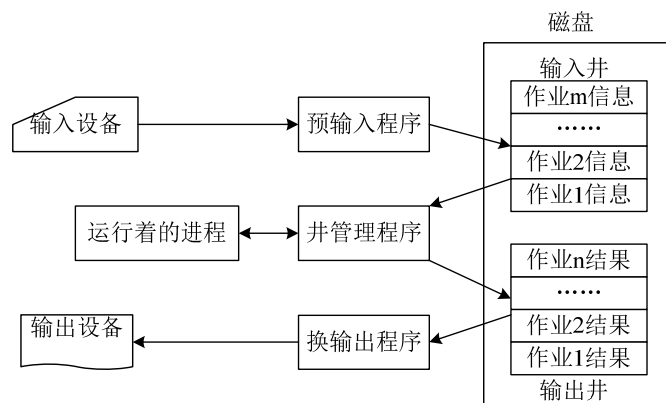


图 4.19 SPOOLing 系统结构

如图4.19所示，是SPOOLing系统的典型构成。其中输入井模拟假脱机输入，用于收容输入的数据；输出井模拟脱机输出，用于收容用户程序的输出数据；预输入程序模拟脱机输入时的外围控制机，将输入设备的输入信息送到输入井，当相关进程需要输入数据时，直接从输入井读入到内存中的用户程序区；缓输出程序模拟脱机输出时的外围控制机，把用户要求输出的信息从用户程序区送到输出井，待输出设备空闲时，将输出井中的信息送到输出设备上；井管理程序负责管理输入井与输出井的协调工作。在进程执行过程中，如果请求启动某台设备的输入或者输出工作，操作系统得到该请求并调出井管理程序，控制从相应输入井读取信息或将信息送至输出井内。当输入井中的信息被进程取出后，相应的井区应归还。同

样的方法，也可以实现信息的输出。通过预输入（出）管理程序从输入（出）并读入（出）信息和通过设备管理从设备上输入（往设备上输出）信息，其结果是一样的。

4、在多道程序环境下，中断处理过程与单道程序环境下的处理有何不同？

答：在多道程序环境下，转入中断处理过程前，根据操作情况保存相应信息，这些信息一开始是保存在设备表中，系统中的每台设备在表中都有对应的表项，如最初调用的返回地址，以及 I/O 操作的一些特定参数等。然后 CPU 就可以分配给其他进程使用了，为此，设备管理调用进程管理的调度程序，原进程的执行就被挂起了。然后进入中断处理程序，在中断处理程序完成后，还要恢复现场信息。而在单道程序环境下，进入中断处理程序前后，不需要保留和恢复现场信息的。

5、设备管理引入缓冲的目的何在？

答：引入缓冲技术主要有以下目的：

第一，它能改善中央处理器与外围设备之间速度不匹配的矛盾，提高 CPU 和 I/O 设备的并行性。例如一个程序，它时而进行长时间的计算而无输出，时而又产生阵发性的输出送到打印机。由于慢速打印机跟不上这个要求，CPU 不得不停下来等待。如果设置了缓冲区，程序输出的数据先送到缓冲区暂存，然后由打印机慢慢地输出。于是，CPU 不必等待，而可继续执行程序，使 CPU 和打印机得以并行工作。

第二，它能减少 I/O 对 CPU 的中断次数和放宽对 CPU 中断响应时间的要求。如果 I/O 操作每传送一个字节就产生一次中断的话，那么设置了 n 个字节的缓冲区后，则可以等到缓冲区满才产生中断。这样，中断次数就减少为  $1/n$ ，而且中断响应时间也可相对放宽。

第三个，缓冲技术还能协调逻辑记录大小与物理记录大小不一致的问题。用户程序输出一个逻辑记录时，不是直接向设备输出，而是申请一个等于磁盘扇区的缓冲区，首先向缓冲区输出，多个逻辑记录拼接成等于或接近磁盘块长，然后向磁盘输出。同样用户程序从磁盘读入一个逻辑记录时，首先把逻辑记录所在的扇区读入一个缓冲区，然后根据需要从缓冲区读出相应的逻辑记录。

6、通道与 DMA 有何区别？

答：DMA 通常采用偷窃总线控制权的方法，由 DMAC 送出内存地址和发出内存读、设备写或者设备读、内存写的控制信号来完成内存与设备之间的直接数据传送，而不用 CPU 干预。当进程要求输入时，CPU 首先为 DMA 设置输入的字节数、输入到内存的地址，开放中断，启动 DMA。然后 CPU 和 DMA 并行工作。DMA 每传送一个数据并不产生中断，只是字数计数器减 1，内存地址寄存器加 1。传送字数计数器减至 0 时，也就是本次 DMA 传送的数据全部传送完毕时，产生中断，请求 CPU 进行结束处理。

I/O 通道方式是 DMA 方式的发展，它可进一步减少 CPU 的干预，即把对一个数据块的读(或写)为单位的干预，减少为对一组数据块的读(或写)及有关的管理和控制为单位的干预。可以把通道看成一个比 DMA 功能更强的接口设备，它有专门用于 I/O 的处理单元。在进行 I/O 操作时，接受 CPU 的委托，独立地执行自己的通道程序来实现内存与外设之间的数据传输。当主机委托的 I/O 任务完成后，通道发出中断，请求 CPU 进行结束处理。这使 CPU 从对 I/O 设备的繁忙直接控制中解脱出来，极大地提高了 CPU 与外设并行工作的程度，从而更有效地提高整个系统的资源利用率。

7、系统在进行设备分配时，应考虑哪几个因素？设备分配程序的功能是什么？设备分配中为什么可能出现死锁？

答：进行设备分配时，应考虑设备的使用性质、运行状态、是否可能死锁等。

设备分配程序的功能：设备分配和回收的任务是按照一定的算法将设备有关资源分配给申请进程，在进程使用完毕后还要负责收回相关设备资源。

由于设备资源有限，而且有些设备是临界资源，如果不进行合理分配，有可能造成一些

进程相互等待对方进程占用的设备资源，从而引发死锁。

8、Windows 7 针对设备管理有哪些新改进？

答：图 4.23 是 Windows 7 的设备系统结构层次图，其中物理层设备是硬件开发商提供的设备；Windows Shell、功能发现模块、手持设备、打印机、扫描仪、传真机等设模块继承了原有 Windows 系统的功能模块；设备元数据系统、设备显示对象、设备与打印机、Windows 功能模块是 Windows 7 新增加的模块；而 Web 应用、Windows 应用、设备显示 XML 则由硬件提供商或者第三方软件开发商实现。

Web应用		Windows应用			Windows功能	
设备显示XML						
设备与打印机						
Windows Shell						
手持设备	打印机	扫描仪	传真机	… …	设备显示对象	
					功能发现模块	
					设备元数据系统	
物理层设备（USB、蓝牙、WiFi等）						

图 4.23 Windows 7 设备系统结构图

习题答案：

## 第五章

1. 什么是文件？文件系统？文件系统的主要功能是什么？

答：在计算机系统中，文件是指存储在外部存储介质上的、由文件名标识的一组相关信息的集合。文件系统是指操作系统中与文件管理有关的那部分软件和被管理的文件，以及实现管理所需要的一些数据结构的总体。一个理想的文件系统应具备以下功能：

- (1) 实现文件的“按名存取”功能
- (2) 实现能够快速定位文件的目录结构，如树型目录；考虑如何组织目录文件，即目录项的设计和文件控制块的存储组织方法，这也直接影响到检索文件的速度。
- (3) 向用户提供一套使用方便、简单的操作命令。
- (4) 管理磁盘、磁带等组成的文件存储器。记录哪些空间被占用，哪些空间空闲，以便用户创建文件时，为其分配空闲空间；修改或删除文件时，调整或收回相应空间。
- (5) 实现逻辑文件到物理文件的转换。
- (6) 保证文件信息的安全可靠。
- (7) 便于文件的共享。

2. 简述计算机系统中文件的分类。

答：(1) 按照文件的逻辑结构的不同，可以把文件分成流式文件和纪录式文件。

(2) 按照用途将文件分为系统文件、库文件和用户文件。

(3) 按照性质可以把文件分为普通文件、目录文件和特殊文件。

(4) 按照保护级别将文件分为只读文件、只写文件，可读可写文件、可执行文件和不保护文件等。例如只读文件只允许授权用户读，但不能写。

(5) 按照文件数据的形式将文件分为源文件，目标文件和可执行文件。

(6) 按照保存期限可以分为临时文件和永久文件。

3. 把一些字符设备也看成是文件，其根据是什么？这样做有什么好处？

答：特殊文件在使用形式上和普通文件相同，通过对特殊文件的操作即可完成相应设备的操作。例如文件/dev/lp 表示打印机设备。特殊文件可以把所有 I/O 设备统一在文件系统下，那么用户就可以用统一的观点看待和处理驻留在不同存储介质上的信息，系统也可以对设备和文件实施统一的管理，从而简化设备管理和文件管理的接口设计

4. 何谓目录文件？它在树形目录结构中起着什么作用？

答：为了加快文件的查找速度，通常把FCB集中起来进行管理，文件控制块的有序集合称为文件目录，即一个文件控制块就是一个文件目录项。文件目录也是以文件的形式保存在外存上的，这就形成了目录文件。

目录文件是树形目录结构中实现查找文件和“按名存取”的重要数据结构，它的组织方法直接影响到检索文件的速度，对整个文件系统的效率、性能和可靠性都有很大的影响。

5. 顺序存取和随机存取(直接存取)的主要区别是什么？它们对于有结构文件和无结构文件的操作有何不同？

答：顺序存取是按照记录的排列顺序或字符的先后顺序逐个存取的方法叫做顺序存取，一般要设置存取文件的读写指针。对于有结构文件读写指针每次定位的是一条记录；对于无结构文件读写指针每次定位的是一个字符。

直接存取是根据需要任意存取文件中的记录或物理块的方法。对于有结构的文件可以直

接读出第*i*个编号的记录，这时会把读写指针直接移到读写处进行操作；对于无结构文件可以按照块号直接请求存取某个物理块，即允许随意读出或写入块。

6. 何谓文件的逻辑结构和文件的物理结构？文件在外存上有哪几种存放方式？它们与文件存取方法有什么关系？

答：文件的逻辑结构是指从用户的观点出发观察到的文件组织形式，用户可以直接处理，独立于文件的物理特性。文件的物理结构是指逻辑文件在物理存储空间中存放方法和组织关系，又称文件的存储结构。

文件在外存上的存放方式对应了该文件的物理结构，主要有连续文件、链接文件、索引文件和直接文件。

连续文件和链接文件适合顺序存取，索引文件和直接文件适合按键存取和直接存取。

7. 为了加快文件检索过程和提高文件系统工作效率，UNIX操作系统采取了哪几项措施？

答：

(1) 采用索引节点的目录文件组织方式：为了减少检索文件访问的物理块数，UNIX文件系统把文件目录项中的文件名和其它描述信息分开，后者单独组成定长的一个数据结构，称为索引节点（*i-node*或*inode*也称*i*节点），这样，每个文件都有唯一的一个索引节点，一个文件系统中的所有文件的索引节点都集中存放在文件存储器上的索引节点区，并对每个索引节点进行编号，该编号也称索引节点号（*i*节点号）。在文件目录项中仅留14个字节的文件名和2个字节的*i*节点号。

(2) 采用树形目录的目录结构

(3) 采用了混合索引的外存分配方式，兼顾了小、中、大以及特大型文件

8. 多重索引结构和多级目录结构是一回事吗？它们各指的是什么？

答：不是一回事。前者指文件的物理结构，即外存的索引分配方式。后者是指目录之间的组织结构。

9. 打开文件和关闭文件的目的是什么？

答：用户使用文件之前先通过“打开”操作，把此文件的文件目录信息（包括索引节点信息）复制到指定的内存区域，当不再使用这个文件时，使用“关闭”操作撤销该文件存放在内存的使用信息，以切断用户进程和该文件目录的联系。这样，文件被打开后，很多信息就已经调入内存，可被用户多次使用，直至文件被关闭或撤销，大大减少访盘次数，提高了文件系统的效率。

10. 什么是文件静态共享和文件动态共享？文件动态共享又有哪两种方式？

答：文件或目录的共享关系不管用户是否正在使用系统都存在的共享方式称为静态共享。所谓文件的动态共享，就是系统中不同的用户进程或同一用户的不同进程并发地访问同一文件。这种共享关系只有当用户进程存在时才可能出现，一旦用户的进程消亡，其共享关系也就自动消失。根据是否共享读写位移指针，动态共享分为两种方式，一种是共享位移指针的动态共享方式，一种是不共享位移指针的动态共享方式。

11. FAT 含义是什么？

答：FAT是File Allocation Table（文件分配表）的缩写。FAT是一个简单的文件系统，它是根



据其物理文件组织形式而命名的，即用文件分配表来记录每个文件中所有盘块之间的链接。

12. 在一个操作系统中，inode节点中分别含有10个直接地址的索引和一、二、三级间接索引。若设每个盘块中可存放128个盘块地址，则一个1MB的文件占用多少间接盘块？一个25MB的文件占用多少间接盘块？（缺少盘块大小的已知条件，设为512字节）

答：盘块默认大小为 512 字节，即0.5KB

$1\text{M}/0.5\text{K}=2^{11}$ 个（即2048个盘块）

$25\text{M}/0.5\text{K}=51200$ 个

利用直接地址能表示的文件最大盘块数为10个

利用一级索引能表示的文件最大盘块数为 $128+10=138$ 个

利用二级索引能表示的文件最大盘块数为 $10+128+128*128=16522$ 个

利用三级索引能表示的文件最大盘块数为 $10+128+128*128+128*128*128=2113674$ 个

所以1MB的文件占用10个直接盘快，128个一次间接盘块， $2048-138=1910$ 个二次间接盘块

25MB的文件占用10个直接盘快，128个一次间接盘块，16384个二次间接盘块， $51200-16522=34678$ 个三次间接盘块。

13. 假定一个盘组共有 100 个柱面，每个柱面上有 8 个磁道，每个盘面分成 4 个扇区，用位示图方法表示，位示图需占多少存储单元？

答：磁盘扇区总数为： $8*4*100=3200$  个，故位示图占用  $3200/8=400$  字节

14. 文件目录在文件系统中的地位是什么？文件目录中应该包含文件的哪些信息？

答：文件目录是文件系统中实现文件查找和文件“按名存取”的重要数据结构。

文件目录中包含了该文件的文件控制块地详细信息，即文件的属性信息，如果是按索引节点方式组织，只要包含该文件的名称和文件的索引节点号。