第 1 章　操作系统概论

1　早期操作系统设计的主要目标是什么?

- 方便性：方便用户使用计算机。
- 有效性：使计算机系统能高效可靠地运转，提高系统资源的利用率。
- 还要便于操作系统的设计、实现和维护。

2　操作系统是资源管理程序，它管理系统中的什么资源?

进程----进程表　存储器----存储表　I/O 设备----I/O 设备表　文件----文件表

3　为什么要引入多道程序系统?它有什么特点?

1)　**引入多道程序设计技术的根本目的**是提高 CPU 的利用率，充分发挥系统设备的并行性。这包括程序之间、CPU 与设备之间、设备与设备之间的并行操作。

2)　**特点**：多道、宏观上并行（不同的作业分别在 CPU 和外设上执行）、微观上串行（在单 CPU 上交叉运行）。

4　叙述操作系统的基本功能。

(1)处理机管理：进程管理。处理机如何调度的问题：FCFS、优先级、时间片轮转?

(2)存储器管理：主存管理。存储分配、存储保护、主存扩充。

(3)设备管理：涉及对系统中各种输入、输出设备的管理和控制。分配设备，控制设备传输数据

(4)文件管理将程序、数据、操作系统软件等组织成文件，存放在磁盘或磁带上，方便用户访问。

5　批处理系统、分时系统和实时系统各有什么特点?各适合应用于哪些方面?

批处理系统:优点: 系统吞吐量大，资源利用率高

缺点: 不能直接控制作业运行，作业的周转时间太长。

适合计算量大、自动化程度高的成熟作业。

分时系统:同时性：若干用户同时使用一台计算机。

独立性：每个用户占有一台终端，彼此独立操作，互不干扰。

交互性：用户可通过终端与系统进行人机对话。

及时性：用户的请求能在较短时间内得到响应。

适用于短小作业

实时系统:实时性。计算机对随机发生的外部事件能够及时地响应和处理。

可靠性。要具有容错能力，可采用双工机制：一台主机；一台后备机。

确定性。是指系统按照固定的、预先确定的时间执行指定的操作。其可确定性取决于系统响应中断的速度和处理能力。

适用于实时过程控制，实时信息处理

6　操作系统的特性?

(1)**并发性**：并发是指系统中存在着若干个逻辑上相互独立的程序，它们都已被启动执行，都还没有执行完，并竞争各种资源。

(2)**共享性**：是指系统中的资源可供内存中多个并发执行的进程共同使用。如打印机、磁带机、磁盘等。支持系统并发性的物质基础是资源共享

(3)**虚拟性**：把共享资源的一个物理实体变为若干个逻辑上的对应物。如，CPU 的分时共享；虚拟存储器技术。

(4)**异步性（随机性）**：有限的共享资源使并发进程之间产生相互制约关系。各个进程何时执行、何时暂停、以怎样的速度向前推进、什么时候完成等都是不可预知的。

7　　衡量 OS 的性能指标有哪些？什么是吞吐量、响应时间和周转时间？

- **资源利用率**：指在给定时间内，系统中某一资源（如 CPU、存储器、外部设备等）实际使用时间所占比率。
- **吞吐量(Throughput)**：指单位时间内系统所处理的信息量。它通常是用每小时或每天所处理的作业个数来度量。
- **周转时间**：指从作业进入系统到作业退出系统所用的时间。而平均周转时间是指系统运行的几个作业周转时间的平均值。

8　什么是嵌入式系统？

以实际应用为中心、以计算机技术为基础、软硬件可裁剪的适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。软件要求固化存储。

9　　什么是对称多处理？它有什么好处？

对称多处理(SMP)：操作系统和用户程序可安排在任何一个处理机上运行，各处理机共享主存和各种 I/O 设备。

优势：1）增加了系统吞吐率。多个作业可以分配在任何一个处理机上执行，大大增加了系统的吞吐率。2）.增加了系统的可靠性。一个处理机的失效，只是性能的降低，不会影响整个系统

10 为了实现系统保护，CPU 通常有哪两种工作状态？各种状态下分别执行什么程序？什么时候发生状态转换？状态转换由谁实现的？

核心态（管态）和用户态（目态）。在核心态下， 允许执行处理机的全部指令集，访问所有的寄存器和存储区；在用户态下，只允许执行处理机的非特权指令，访问指定的寄存器和存储区。通过中断和异常，CPU 能从用户程序的运行转入操作系统内核程序的运行。

用户态到核心态的转换由硬件完成；管态到目态的转换由操作系统程序执行后完成。

11　什么是系统调用？什么是特权指令？特权指令执行时，CPU 处于哪种工作状态？

系统调用就是操作系统内提供的一些子程序。编程人员通过使用系统调用命令,向操作系统提出资源请求或获得系统的一些功能服务,以取得操作系统的服务。

特权指令是指关系系统全局的指令.这类指令只允许操作系统使用,不允许用户使用。

工作在核心态.

12　操作系统通常向用户提供哪几种类型的接口?其主要作用是什么？

操作接口：命令语言或窗口界面是用户使用计算机系统的主要接口。

用户利用该接口来组织和控制作业的执行.

编程接口：系统调用是用户与操作系统之间的编程接口。

编程人员在程序中利用该接口,箱操作系统提出资源请求和一些功能服务.

第 2-3 章　进程管理

1　程序顺序执行的特点

封闭性：程序在运行时独占全机资源。因此，这些资源的状态只能由这个运行的程序决定和改变。不受外界因素影响。

可再现性：只要初始条件相同，无论程序是连续运行，还是断断续续地运行，程序的执行结果与其执行速度无关。

优点：　由于顺序程序的封闭性和可再现性，为程序员调试程序带来了很大方便。

缺点：　由于资源的独占性，使得系统资源利用率非常低。

2　何谓进程，进程由哪些部分组成？试述进程的四大特性（动态性、独立性、并发性、结构性）及进程和程序的区别。

1)进程是可以和其他程序并行执行的程序关于某个数据集合的一次执行过程。

2)进程是由程序，数据和进程控制块三部分组成的。

至少一个可执行程序；一个独立的地址空间；一个执行栈区（用于子程序调用，系统调用，进程切换）；一些要使用的文件和 I/O 设备

3） 动态性。进程是程序的一次执行过程，是临时的，有生命期的。

独立性。进程是系统进行资源分配和调度的一个独立单位。

并发性。多个进程可在处理机上交替执行。

结构性。系统为每个进程建立一个进程控制块。

4）①进程是动态的，程序是静态的。程序是有序代码的集合，进程是程序的执行，没有程序就没有进程。通常，进程不可以在计算机之间迁移，而程序可以复制。

②进程是暂时的，程序是永久的。进程包括程序、数据、进程控制块。

③通过多次执行，一个程序可对应多个进程；通过调用关系，一个进程可包括多个程序。进程可创建其他进程，而程序并不能形成新的程序。

3　进程控制块的作用是什么?它主要包括哪几部分内容?

- PCB 是进程存在的唯一标识。含有进程的描述信息和管理控制信息。

1） 进程标识数：用于唯一地标识一个进程，通常是一个整数。

2） 进程的状态、调度、存储器管理信息：是调度进程所必需的信息，包括进程状态、优先级、程序在主存地址、在外存的地址等。

3） 进程使用的资源信息：分配给进程的 I/O 设备、正在打开的文件等。

4） CPU 现场保护区：保存进程运行的现场信息。包括：程序计数器(PC)、程序状态字、

通用寄存器、堆栈指针等。

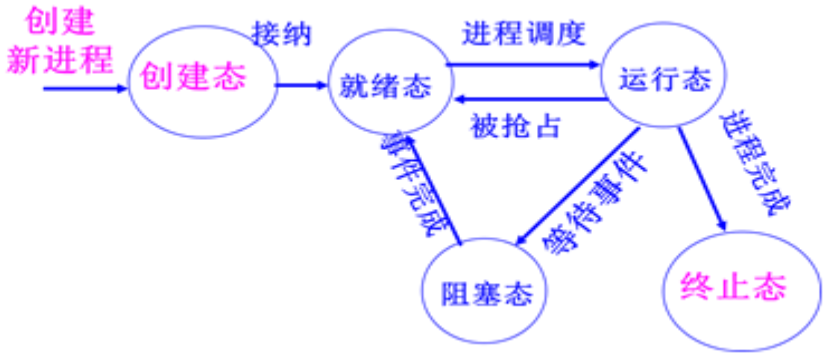
5） 记帐信息：包括使用 CPU 时间量、帐号等。

6） 进程之间的家族关系：类 UNIX 系统，进程之间存在着家族关系，父/子进程。Windows 进程之间不具有父子关系。

7） 进程的链接指针：链接相同状态的进程。

4 进程的基本状态，试举出使进程状态发生变化的事件并描绘它的状态转换图。

（五个状态，三个基本状态）



1)运行态(running)：进程正在 CPU 上运行。单 CPU 系统一次只有一个运行进程；多 CPU 系统可能有多个运行进程。

(2)阻塞态(blocked)：又称等待态。当一个进程因等待某个条件发生而不能运行时所处的状态。等待 I/O 完成，等待一个消息

(3)就绪态(ready)：已获得除 CPU 之外的全部资源，只要再获得 CPU，就可执行。

（4）创建态：刚刚建立，未进就绪队列。

（5）终止态：已正常结束或故障中断，但尚未撤销。暂留在系统中，方便其他进程去收集该进程的有关信息。

5 什么是原语?什么是进程控制？

原语：是由若干条指令组成的完成某种特定功能的一段程序，具有不可分割性。即原语的执行必须是连续的，在执行过程中不允许被中断。

进程控制：是指系统使用一些具有特定功能的程序段来创建、撤消进程，以及完成进程各状态之间的转换。进程控制是由操作系统内核完成的。是属于原语一级的操作，不能被中断。

6 进程调度的功能、方式、时机、算法。作业调度，交换调度。作业的周转时间和作业的带权周转时间?（P32-36，结合课后习题好好看）

进程调度就是为进程分配处理机

功能：(1) 记录系统中各进程的执行状况。

管理系统中各进程的控制块,将进程的状态变化及资源需求情况及时地记录到PCB中。

(2) 选择就绪进程真正占有 CPU

(3) 进行进程上下文的切换。将正在执行进程的上下文保存在该进程的 PCB 中，将刚选中进程的运行现场恢复起来，以便执行。

方式：① 非抢先方式(非剥夺方式)。 某一进程占用 CPU,直到运行完或不能运行为止，其间不被剥夺。用在批处理系统。主要优点：简单、系统开销小。

② 抢先方式(剥夺方式) 允许调度程序基于某种策略（优先级、时间片等）剥夺现行进程的 CPU 给其它进程。用在分时系统、实时系统。

时机：（1）现行进程完成或错误终止；

（2）提出 I/O 请求，等待 I/O 完成时；

（3）在分时系统，按照时间片轮转，分给进程的时间片用完时；

（4）优先级调度，有更高优先级进程就绪；

（5）进程执行了某种操作原语，如阻塞原语和唤醒原语时，都可能引起进程调度。

算法：进程调度所采用的算法是与整个系统的设计目标一致的。

批处理系统：增加系统吞吐量和提高系统的资源利用率。

实时系统：保证对随机发生的外部事件作出实时响应。

作业调度（1-3）：（1）先来先服务(FCFS)：简单，节省机器时间。缺点：容易被大作业垄断，使得平均周转时间延长。

（2）最短作业优先(SJF)：选取运行时间最短的作业运行。对短作业有利，作业的平均周转时间最佳。 若系统不断进入短作业，长作业就没有机会运行，出现饥饿现象。

（3）响应比高者优先(HRN)

Rp =(作业等待时间+作业估计运行时间)/作业估计运行时间=1+作业等待时间/作业估计运行时间

交换调度（4-6）：（4）优先级调度法

将 CPU 分配给就绪队列中优先级最高的进程

（5）轮转法(Round Robin)

（6）多级反馈队列轮转法

7 线程的定义，线程与进程的比较。系统对线程的支持（用户级线程、核心级线程、两级组合）。

线程是进程中一个可执行实体，是被操作系统调用的一个独立单位。

线程与进程的比较：

（1）拥有的资源

进程拥有一个独立的地址空间，若干代码段和数据段，若干打开文件、主存以及至少一个线程。一个进程内的多线程共享该进程的所有资源，线程自己拥有很少资源。

（2）调度

进程调度需进行进程上下文的切换，开销大。

同一进程内的线程切换，仅把线程拥有的一小部分资源变换了即可，效率高。同一进程内的线程切换比进程切换快得多。不同进程的线程切换…

（3）并发性

引入线程后，使得系统的并发执行程度更高。 进程之间、进程内的多线程之间可并发执行。

（4）安全性

同一进程的多线程共享进程的所有资源，一个线程可以改变另一个线程的数据，而多进程实现则不会产生此问题。共享方便。

8 并发执行的进程在系统中通常表现为几种关系?各是在什么情况下发生的?

（1）对资源的共享引起的互斥关系

进程之间本来是相互独立的，但由于共享资源而产生了关系。间接制约关系，互斥关系。

（2）协作完成同一个任务引起的同步关系

一组协作进程要在某些同步点上相互等待发信息后才能继续运行。直接制约关系。同步关系。

（3）进程之间的前序关系

由于进程之间的互斥同步关系，使得进程之间具有了前序关系，这些关系决定了各个进程创建和终止的时间。

9 什么叫临界资源?什么叫临界区?对临界区的使用应符合的四个准则（互斥使用、让权等待、有空让进、有限等待）。

临界资源：就是一次仅允许一个进程使用的资源。

临界区(critical section)：就是并发进程访问临界资源的那段必须互斥执行的程序。

四个准则：不能同时有两个进程在临界区内执行——**互斥使用**

等待进入临界区的进程，应释放处理机后阻塞等待——**让权等待**

在临界区外运行的进程不可阻止其他进程进入临界区——**有空让进**

不应使要进入临界区的进程无限期等待在临界区之外——**有限等待**

10 解决进程之间互斥的办法：

<开、关中断，加锁、开锁（又叫测试与设置，通常由一条机器指令完成），软件方法，信号量与 P、V 操作。>

(1) 关中断 最简单的方法。在进程刚进入临界区后，立即禁止所有中断；在进程要离开之前再打开中断。因为 CPU 只有在发生时钟中断或其它中断时才会进行进程切换。

（2）使用测试和设置硬件指令锁位变量 W ：为每个临界资源设置一个，以指示其当前状态。W=0，表示资源空闲可用；W=1，表示资源已被占用。

（3）信号量与 P、V 操作：两个或多个进程可以通过简单的信号进行合作，一个进程可以被迫在某一位置停止，直到它接收到一个特定的信号。为了发信号，需要使用一个称作信号量的特殊变量。

11 若信号量 S 表示某一类资源，则对 S 执行 P、V 操作的直观含意是什么？当进程对信号量 S

执行 P、V 操作时，S 的值发生变化，当 S>0、S=0、和 S<0 时，其物理意义是什么？

P 操作相当于申请资源，V 操作相当于释放资源

P 操作：S>=0，则执行 P 操作的进程继续执行；S<0，则执行 P 操作的进程变为阻塞状态，并排到与该信号量有关的队列中等待。

V 操作：S<=0，则执行 V 操作的进程从与该进程有关的队列中释放一个进程，使他由阻塞变为就绪状态；S>0，则执行 V 操作的进程继续前进。

12 在用 P/V 操作实现进程通信时，应根据什么原则对信号量赋初值？

13 经典的 IPC 问题。(P47-49)

14 进程高级通信有哪些实现机制？

- 高级通信：是指进程采用系统提供的多种通信方式来实现通信。如消息缓冲、信箱、管道、共享主存区等。
- 发送进程和接收进程的消息通信方式：
  - 非阻塞发送，阻塞接收
  - 非阻塞发送，非阻塞接收
  - 阻塞发送，阻塞接收

15 死锁产生的必要条件及解决死锁的方法

必要条件：1）互斥条件。独占性的资源。

2） 保持和等待条件。进程因请求资源而阻塞时，对已经获得的资源保持不放。

3）不剥夺条件。已分配给进程的资源不能被剥夺，只能由进程自己释放。

4）循环等待条件。存在一个进程循环链，链中每个进程都在等待链中的下一个进程所占用的资源。

产生死锁的根本原因：是对独占资源的共享，并发执行进程的同步关系不当。

解决办法：1）鸵鸟算法。忽略死锁。

2） 死锁的预防。通过破坏产生死锁的四个必要条件中的一个或几个，来防止发生死锁。

3）死锁的避免。是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。

4）死锁的检测和恢复。允许死锁发生，通过设置检测机构，及时检测出死锁的发生，然后采取适当措施清除死锁。

16 理解银行家算法的实质。能够利用银行家算法避免死锁。（P61）

第 4 章 存储器管理

1 存储器管理的功能。名字空间、地址空间、存储空间、逻辑地址、物理地址。

功能：1）存储器分配：解决多道程序或多进程共享主存的问题

（2）地址转换或重定位：研究各种地址变换方法及相应的地址变换机构。

（3）存储器保护：防止故障程序破坏 OS 和其它信息

（4）存储器扩充：采用多级存储技术实现虚拟存储器及所用的各种管理算法。

（5）存储器共享：并发执行的进程如何共享主存中的程序和数据。

符号名字空间：源程序中的各种符号名的集合所限定的空间。

存储空间：物理存储器中全部物理存储单元的集合所限定的空间。

逻辑地址空间：经编译连接后的目标代码所限定的空间。用地址码替换符号地址。

存储空间是由字或字节组成的一个大的阵列，每一个字或字节都有它自己的编号地址。这些编号就叫物理地址。

2　什么是地址重定位?分为哪两种?各是依据什么和什么时候实现的?试比较它们的优缺点。

地址重定位：把程序地址空间的逻辑地址转换为存储空间的物理地址。<静态重定位和动态重定位>

1）**静态**：在进程执行前，由装入程序把用户程序中的指令和数据的逻辑地址全部转换成存储空间的绝对地址。

特点：1）无硬件变换机构；2）为每个程序分配一个连续的存储区；3）在程序执行期间不能移动，主存利用率低；4）难以做到程序和数据的共享；5）用于单道批处理系统。

2）动态：装入程序把程序和数据原样装入到已分配的存储区中。程序运行时，把该存储区的起始地址送入重定位寄存器。需硬件地址转换机构。

优点：A）主存利用充分。可移动用户程序。移动后，只需修改重定位寄存器。B）程序不必占有连续的存储空间。C）便于多用户共享存储器中的同一程序和数据。

3　内存划分为两大部分：用户空间和操作系统空间。

通常存储器划分为两部分：一部分是操作系统占用区，另一部分是用户进程占用区（或用户区）。存储器管理是指对用户区的管理。

4 存储保护的目的是什么?对各种存储管理方案实现存储保护时，硬件和软件各需做什么工作?

防止地址越界：进程运行时产生的所有存储器访问地址都要进行检查，确保只访问为该进程分配的存储区域。

正确地进行存取：对所访问的存储空间的操作方式（读、写、执行）进行检查，以防止由于误操作，使其数据的完整性受到破坏。

5　试述可变式分区管理空闲区的方法及存储区的保护方式。覆盖与交换有什么特点？

根据作业的大小动态地划分分区，使分区的大小正好等于作业大小。各分区的大小是不定的；内存中分区的数目也是不定的。

（1）首次适应(first fit)法：

要求空闲区表或空闲区链中的空闲区按地址从小到大排列。分配内存时，从起始地址最小的空闲区开始扫描，直到找到一个能满足其大小要求的空闲区为止。分一块给请求者，余下部分仍留在其中。

（2）最佳适应(best fit)法：

存储分配程序要扫描所有空闲区，直到找到能满足进程需求且为最小的空闲区为止。

缺点：因为要查找所有的分区，所以比首次适应算法效率低。可能把主存划分得更小，出现很多无用的碎片。改进：从小到大对空闲区排序。

（3）最坏适应(worst fit)法：

要扫描所有的空闲区，直到找到满足进程要求且为最大的空闲区为止。一分为二，一部分分给进程，另一部分仍留在链表中。

目的：使剩下的空闲区可用。

缺点：要扫描所有的空闲区；大空闲区的不断分割，可能满足不了大进程的要求。

改进：从大到小对空闲区排序，以提高查找速度。

覆盖的特点：打破了必须将一个进程的全部信息装入主存后才能运行的限制。在逻辑上扩充了主存。小主存可运行大进程。（交换主要是在进程之间进行，而覆盖则主要在同一个进程内进行。）

交换的特点：打破了一个程序一旦进入主存，便一直运行到结束的限制。

6　页表的作用是什么?简述页式管理的地址变换过程。能利用页表实现逻辑地址转换成物理地址。管理内存的数据结构有哪些?

页表：系统为每个进程建立一张页面映像表，记录逻辑页与主存块的映射关系。

管理内存的数据结构有两种：存储分块表，位示图；

7　什么是页式存储器的内零头?它与页的大小有什么关系?可变式分区管理产生什么样的零头（碎片）?

8　段式存储器管理与页式管理的主要区别是什么?

(1)段是由用户划分的；页是为了方便管理由硬件划分的，对用户是透明的。

(2)页的大小固定；段的大小不固定。

(3)段式用二维地址空间；页式用一维地址空间。

(4)段允许动态扩充，便于存储保护和信息共享。

(5)段可能产生主存碎片；页消除了碎片。

(6)段式管理便于实现动态链接，页式管理只能进行静态链接。

(7)段与页一样，实现地址变换开销大，表格多。

9 什么是虚拟存储器。虚拟存储器的容量能大于主存容量加辅存容量之和吗?

虚拟存储器：是系统为了满足应用对存储器容量的巨大需求而构造的一个非常大的地址空间。其容量由计算机的地址结构确定。系统的指令地址部分能覆盖的地址域大于实际主存的容量。

10 实现请求页式管理，需要对页表进行修改，一般要增加状态位、修改位、访问位。试说明它们的作用。

(1) 有效位（状态位）：用来指示某页是否在主存。为 1 表示该页在主存，完成正常的地址变换；

为 0 表示该页不在主存，由硬件发出一个缺页中断，转操作系统进行缺页处理。

(2) 修改位：指示该页调入主存后是否被修改过。"1"表示修改过，"0"表示未修改过。

(3) 访问位（引用位）：指示该页最近是否被访问过，"1"表示最近访问过，"0"表示最近未访问。

11 产生缺页中断时，系统应做哪些工作?

（1）根据当前执行指令中的逻辑地址查页表的状态位。（2）状态位为 0，缺页中断。（3）操作系统处理缺页中断，寻找一个空闲的内存页。（4）若有空闲页，则把从磁盘读入信息装入该页面。（5）若无空闲页，则按某种算法选择一个已在内存的页面，暂时调出内存。若修改过还要写磁盘。调入需要的页。之后要修改相应的页表和内存分配表。（6）恢复现场，重新执行被中断的指令。

12 会利用 FIFO、LRU、OPT 以及时钟页面置换算法描述页面置换过程，计算产生的缺页率。Belady 异常。（P89，好好看）

13 什么是程序的局部性原理？什么叫系统抖动?工作集模型如何防止系统抖动?

时间局部性：程序中往往含有许多循环，在一段时间内会重复执行该部分。

空间局部性：程序中含有许多分支，在一次执行中，只有满足条件的代码运行，不满足条件的代码不运行。即使顺序执行程序，程序的地址域在短时间内变化不大。在进程运行过程中，用到哪部分程序或数据再由系统自动装入。

14 多级页表的概念，多级页表中页表建立的时机。写时复制技术的概念。

大页表：(地址空间 4GB)/(页 4KB)=1M 个页表项。若每个页表项占 4B，则最大的页表为 4MB。

多级页表结构：页表在内存不必连续存放。

页表的建立不再是在进程装入主存时，而是推迟到要访问页时，才为包含该页的页表分配空间和建立页表页。

第 5 章　文件系统

1 什么是文件和文件系统？文件系统的主要功能。UNIX 系统如何对文件进行分类？它有什么好处？

文件是存储在外部存储器上的具有符号名的相关信息的集合。

文件系统：OS 中管理文件的软件机构。包括管理文件所需的数据结构、相应的管理软件和被管理的文件。

功能：1.管理文件存储器。记录空间使用情况，分配空间，调整或回收空间。2.实现按名存取。利用目录结构快速定位文件。3.应具有灵活多样的文件结构和存取方法，便于用户存储和加工处理信息。4.提供一套使用方便、简单的操作命令。5.保证文件信息的安全性。6.便于文件的共享。

UNIX 系统为了方便用户操作文件，按文件的组织和处理方式划分为三类：**普通文件，目录文**

件，特别文件

便于系统对不同类型的文件进行不同的管理，以实现文件的保护和共享等。

2　文件目录的作用是什么?文件目录项通常包含哪些内容？文件控制块。

主要作用是使用户实现按名存取文件。主要记录文件的名字机器存放物理地址的一张映射表，表中包括了许多文件控制块（FCB）

3　文件的逻辑结构有几种形式？文件的存取方法？

1）无结构的字节流式文件。由无结构的先后到达的相关字节组成，其文件长度就是所包含的字节个数。

有结构的记录式文件。分为定长记录式文件和变长记录式文件。

2）文件的存取方法

（A）顺序存取：按照文件信息的逻辑顺序依次存取。是在前一次存取的基础上进行的。

（B）直接存取（随机存取）基于文件的磁盘模型，磁盘允许对任意文件块进行随机读和写

4　文件的物理结构有哪几种?对于不同的结构，文件系统是如何进行管理的?

（1）连续文件（顺序文件）：文件内容连续存放。优点：简单。支持顺序存取和随机存取。存取速度快。只要访问一次文件的管理信息，就可方便地存取到任一记录。缺点：不灵活，容易产生碎片。

（2）链接文件：不要求文件内容连续存放。把文件所占用的物理块用链接指针链接起来。优点：可以解决外存的碎片问题，提高了外存空间的利用率；允许文件动态增长。缺点：只能按文件的指针链顺序存取，查找效率较低。

（3）索引文件：为每个文件建立一张索引表。用索引表记录文件内容的存放地址，即记录文件的逻辑块号和对应的物理块号之间的关系。优点：文件可动态修改；随机、顺序存取。缺点：索引表的使用增加了存储空间的开销；降低了文件的存取速度。

（4）索引顺序文件： 它是索引文件和顺序文件的组合形式。他将顺序文件中的所有记录分成若干组，并为其建立索引表。

5　DOS 文件卷的结构，DOS 系统的文件物理结构是什么？

MS-DOS 就是使用文件分配表（FAT）来分配和管理磁盘空间的。DOS 系统的文件采用链接结构。（索引顺序文件）

6　　了解记录的组块和分解。

对于记录式文件，记录大小是由文件的性质决定的，而存储介质上的块的划分与存储介质的特性有关，块的大小通常不是固定不变的。因此，实际一个逻辑记录与物理块的大小是不相等的。当用户文件的逻辑记录远小于物理块大小时，一个逻辑记录存放在一个物理块总，将会造成极大的浪费。为此，可把多个记录存放在一个物理块中，也允许一个逻辑记录跨块存放。这样可以提高存储器的利用率。

7　文件存储空间的管理方法有几种?它们各是如何实现文件存储空间的分配和回收的?

（1）空白文件目录 ：（是一种最简单的方法）系统为所有这些空白文件建立一张表。每个空白文件占用一个表目。适合于文件的静态分配（连续文件的分配）

（2）空闲块链表：A）空闲块链 把所有空闲块连接成一个链表。优点：简单。适合文件动态分配。缺点：工作效率低；分配和回收多个盘块时要次访问磁盘才能完成。B）**空闲块成组链表** 利用盘空闲块来管理盘上的空闲块，每个磁盘块记录尽可能多的空闲块而成一组。各组之间也用链指针链接在一起。便于空闲块的分配与回收。适合连续文件、链接文件和索引文件的存储分配。

（3）位映像表(bit map)或位示图：是适合文件静态分配和动态分配的最简单方法

位映像表（位向量）：每一个二进制位对应一个物理盘块。为 1 时表示块已分配，为 0 时空闲。

8 建立多级目录有哪些好处?文件的重名和共享问题是如何得到解决的?

（层次结构清晰，便于管理保护，有利于文件分类，解决文件重命名，提高检索速度，控制存取权限）

9.文件系统中，常用的文件操作命令有哪些？它们的具体功能是什么?打开和关闭文件命令的目的是什么?

1) **创建(Create)文件** 主要功能：在指定设备上为指定路径名的文件建立一个目录项，并设置文件的有关属性。
2) **删除(Delete)文件** 主要功能：根据文件的路径名找到指定的目录项，回收其占用的各个物理块，再将该目录项置为空。
3) **打开(Open)文件** 根据文件路径名找到目录项，将文件的目录项复制到主存一个专门区域，返回文件在该区域的索引。建立进程与文件的联系。目的：避免多次重复地检索文件目录。
4) **关闭(Close)文件** 释放文件在主存专门区域中的目录项，切断用户与文件的联系。若该目录项被修改过，则复制到磁盘。若文件作过某些修改，应将其写回辅存。
5) **读(Read)文件** 命令中必须指出要读的数据个数，以及存放数据的主存地址。根据文件所在设备、文件类型的不同，系统设置不同的读命令。
6) **写(Write)文件** 命令中必须指出要写的数据个数，以及存放数据的主存地址，将主存中的数据写到指定的文件中。
7) **追加(Append)文件** 限制了写文件的形式，将数据追加到文件尾。
8) **随机存取(Seek)文件** 重新定位文件的读/写位置指针。
9) **得到文件属性(Get Attributes)** 进程在执行时常常需要了解文件的属性。
10) **设置文件属性(Set Attributes)** 修改文件的一些属性，以适应用户的要求。
11) **重命名(Rename)文件** 重新命名一个已经存在的文件。

10 存取控制表 ACL 的概念。（P117）

为存取控制矩阵中的每一列建立一张存取控制表(ACL)，用一有序对(域, 权集)表示

11　理解内存映射文件（memory mapped file）的过程。（P120）

将文件映射到进程地址空间的一个区域，返回虚拟地址，仅当需要对文件存取时，才传输实际的数据

第 6 章　设备管理

1　I/O 设备通常大致可分为哪两大类？各自传输的信息单位有什么特点？（字符设备和块设备）

字符设备：人机交互设备。是以字符为单位发送和接收数据的，通信速度比较慢。键盘和显示器、鼠标、扫描仪、打印机、绘图仪等。

版本块设备：外部存储器。以块为单位传输数据.常见块尺寸：512B～32KB,如磁盘、磁带、光盘等。

网络通信设备：主要用于与远程设备的通信。传输速度比字符设备快，比块设备慢。如网卡、调制解调器等。

时钟：按预先规定好的时间间隔产生中断。

2　常用的四种数据传输方式。（P126）

程序查询方式（polling），中断方式，直接存储器访问(DMA)方式，通道控制方式

3　根据设备的使用方式，设备被分为几种类型？何为虚拟设备？它是通过什么技术实现的？

虚拟设备，共享设备，独占设备。

虚拟设备：是指设备本身是独占设备，而经过虚拟技术处理，可以把它改造成共享设备，供多个进程同时使用。常用可共享的高速设备来模拟独占的慢速设备。能有效提高独占型设备的利用率。 Spooling 技术是实现虚拟设备的具体技术。它利用可共享磁盘的一部分空间，模拟独占的输入/输出设备。以空间换时间

4　按照设备管理的层次结构，I/O 软件划分为几层？各层主要实现哪些功能？

1) **中断处理程序**　①进程在启动一个 I/O 操作后阻塞起来，I/O 操作完成，控制器产生一个中断。②CPU 响应中断，执行中断处理程序。③检查设备状态。④中断返回被中断的进程，或转进程调度。

2） 设备驱动程序　①设备初始化。②启动设备传输数据的例程。③中断处理例程。

3）独立于设备的软件　实现所有设备都需要的功能，且向用户提供一个统一的接口。

4）用户层的 I/O 接口　程序运行期间，库函数 read 将其连接在一起形成一个可执行文件

5　何为设备的独立性？

设备独立性是指用户及用户程序不受系统配置的设备类型和具体设备的台号的影响。用户只是使用逻辑设备，具体的映射由操作系统完成。

6　什么是 SPOOLING 技术？以输出为例，说明它的实现原理。（SPOOLING 技术是以空间换时间）

是实现虚拟设备的具体技术。它利用可共享磁盘的一部分空间，模拟独占的输入/输出设备。以空间换时间。

Spooling 实际是一种缓冲技术。进程要打印时，系统并不为它分配打印机，而是把待打印的数据缓冲到一个独立的磁盘文件上，形成待打印文件队列。之后，Spooling 系统一次一个地将打印队列上的文件送打印机打印。这种技术又叫缓输出技术。

7　一个特定磁盘上的信息如何进行编址？

盘面号、磁道号 和扇区号（或柱面号、磁头号和扇区号）。

8　要将磁盘上一个块的信息传输到主存需要系统花费哪些时间？

寻道时间、旋转延迟时间和读/写传输时间

9.常用的磁盘调度算法：（P136）

先来先服务、最短寻道时间优先、扫描法（SCAN, C_SCAN, LOOK, C_LOOK）。

第 7 章　Linux 进程管理

1　进程控制块，其中与进程管理、存储器管理和文件管理有关的一些字段，线程组标识符。

（1）进程控制块，又称为进程描述符（用 task_struct 类型的数据结构表示）：描述进程的数据结构，它包含了与进程相关的所有信息。

（2）进程标识符（PID）：存放在进程描述符 pid 字段中。新创建的进程的 PID 通常是前一个进程的 PID 加 1.默认情况下，最大的 PID 号是 32767.

（3）金成刚从用户态切换到核心态时，其核心栈为空，只要将栈顶指针减去 8K，就能得到 thread info 结构的地址。

2　与进程创建有关的函数：fork( )、vfork( )、clone( )。

（1）创建子进程函数 fork()：创建成功之后，子进程采用写时复制技术读共享父进程的全部地址空间，仅当父或子要写一个页时，才为其复制一个私有的页的副本。

（2）创建轻量级进程函数 clone()：实现对多线程应用程序的支持。共享进程在内核的很多数据结构，如页表、打开文件表等等。

（3）vfork()系统调用：创建的子进程能共享父进程的地址空间，为了防止父进程重写子进程需要的数据，先阻塞父进程的执行，直到子进程退出或执行了一个新的程序为止。

3　理解进程切换的过程。涉及到页目录表、核心栈、硬件上下文。

（1）进程切换只发生在核心态。在发生进程切换之前，用户态进程使用的所有寄存器值都已被保存在进程的核心栈中。

（2）进程硬件上下文存放在进程描述符的 thread_struct thread 中。该结构包含的字段涉及到大部分 CPU 寄存器，但象 eax、ebx 等通用寄存器的值仍被保留在核心栈中。

（3）进程切换：第一步，切换页目录表以安装一个新的地址空间；第二步，切换核心栈和硬件上下文。由 schedule()函数完成进程切换。

4　进程调度方式。进程调度时机。

（1）**进程调度的方式**：Linux2.6 系统采用可抢先式的动态优先级调度方式。其内核是完全可重入的。无论进程处于用户态还是核心态运行，都可能被抢占 CPU，从而使高优先级进程能及时被调度执行，不会被处于内核态运行的低优先级进程延迟。

（2）**进程调度的算法**：Linux 系统的调度算法是基于进程过去行为的启发式算法，以确定进程应该被当作交互式进程还是批处理进程。

（3）**实时进程调度的时机**：1）出现了更高优先级的实时进程。（2）进程执行了阻塞操作而进入睡眠状态。（3）进程停止运行或被杀死。（4）进程调用自愿放弃处理机。（5）在基于时间片轮转的实时进程调度过程中，进程用完了自己的时间片。

5　Linux 有很多内核线程，了解 0 号进程和 1 号进程的作用。

（1）0 号进程就是一个内核线程，0 号进程是所有进程的祖先进程，又叫 idle 进程或叫做 swapper 进程。其进程描述符存放在 init_task 变量中。每个 CPU 都有一个 0 号进程。

（2）1 号进程是由 0 号进程创建的内核线程 init，负责完成内核的初始化工作。在系统关闭之前，init 进程一直存在，它负责创建和监控在操作系统外层执行的所有用户态进程。

第 8 章　Linux 存储器管理

1　进程地址空间的划分？管理进程私有地址空间的数据结构？链接虚拟内存区域的单链表和红黑树。指向映射文件对象的指针字段？指向进程页目录表的指针字段？

（1）Linux 把地址空间分成两部分。进程的私有空间是前 3G，进程的公有空间是后 1G 的内核虚空间。

（2）**vm_area_struct**

struct vm_area_struct {

struct mm_struct * vm_mm; 虚拟内存描述符

unsigned long vm_start;　　/*起始地址*/

unsigned long vm_end;　　/*结束地址*/

struct vm_area_struct　*vm_next;

struct rb_node　vm_rb;　　/*红-黑树*/

struct file * vm_file;　指向文件映射对象或为 NULL

struct raw_prio_tree_node　prio_tree_node;

内存映射文件时，用此结构构造 radix 优先级搜索树

……}

（3）进程通过一个单链表把所拥有的各个虚拟内存区域按照地址递增顺序链接在一起。当进程需要的虚拟内存区域数较少，即只有一二十个时，使用链表来管理虚拟内存区域是很方便的。红黑树是一棵排好序的平衡二叉树。必须满足四条规则：①树中的每个节点或为红或为黑；②

树的根节点必须为黑；③红节点的孩子必须为黑；④从一个节点到后代诸叶子节点的每条路经，都包含相同数量的黑节点，在统计黑节点个数时，空指针也算作黑节点。这四条规则保证：具有 n 个节点的红黑树，其高度至多为 2*log(n+1)。

（4）如上（3）中的加粗字体

（5）struct mm_struct {

struct vm_area_struct *mmap;

struct rb_root mm_rb; /*指向红-黑树的根*/

pgd_t *pgd; /*指向页目录表*/

atomic_t mm_users; /*次使用计数器*/

atomic_t mm_count; /*主使用计数器*/

struct list_head mmlist; 双向链表

unsigned long start_code, end_code; /*可执行代码所占用的地址区间*/

……};

2 Linux 堆的管理：malloc( )，free( )。

malloc(size)：请求 size 个字节的动态内存。

free(addr)：释放内存。

3 管理物理内存页框的数据结构？ 内存管理区 zone 结构，伙伴系统？分区页框分配器分配页框的过程。

（1）struct page {

unsigned long flags; /*页框状态标志*/

atomic_t _count; /*页框的引用计数*/

atomic_t _mapcount; /*对应的页表项数目*/

unsigned long private; /*由伙伴系统使用*/

struct address_space *mapping;页高速缓存

pgoff_t index; 在页高速缓存中以页为单位偏移

struct list_head lru; 链入活动页框链表或非活动..

void *virtual; /*页框所映射的内核虚地址*/

};

（2）Linux 把内存节点划分为 3 个管理区 zone。

ZONE_DMA：包含低于 16MB 的常规内存页框。用于对老式的基于 ISA 设备的 DMA 支持。

ZONE_NORMAL：包含高于 16MB 且低于 896MB 的常规内存页框。

ZONE_HIGHMEM：包含从 896MB 开始的高端物理页框。内核不能直接访问这部分页框。在 64 位体系结构上，该区总是空的。

（3）采用伙伴系统(buddy system)管理连续的空闲内存页框，以解决外碎片问题。

外碎片就是夹杂在已分配页框中间的那些连续的小的空闲页框。

伙伴算法把空闲页框组织成 11 个链表，分别链有大小为 1，2，4，8，16，32，64，128，256，512 和 1024 个连续页框的块。

（4）负责处理对连续物理页框的分配请求。 管理区分配器负责搜索一个能满足动态内存请求的内存管理区。当空闲页框不足时，管理区分配器应当触发页框回收算法。 在每个管理区内的页框，除了一小部分页框被保留为每 CPU 页框高速缓存外（以满足本地 CPU 发出的对单个页框的请求），其它的由伙伴系统来管理。

4 理解 slab 分配器的原理。slab 分配器的作用？

（1）**原理**：slab 分配器把小内存区看作对象，slab 分配器对不再引用的对象只是释放但内容保留，以后再请求新对象时，就可直接使用而不需要重新初始化。

从分区页框分配器获得几组连续空闲页框

slab 分配器为不同类型的对象生成不同的高速缓存，每个高速缓存存储相同类型的对象。

高速缓存由一连串的 slab 构成，每个 slab 包含了若干个同类型的对象。

（2）**作用**：slab 分配器用于为只有几十或几百个字节的小内存区分配内存。如，file 对象。

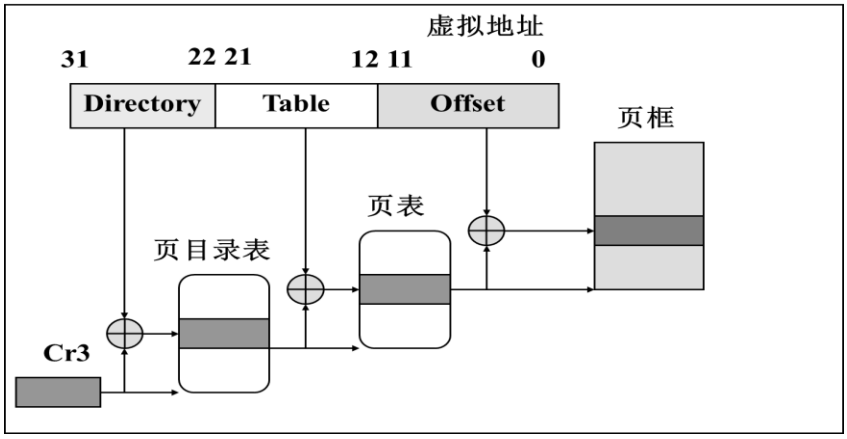
5 进程页表建立的时机？了解页目录表项或页表项所包含的字段。逻辑地址的划分，利用两级页表实现地址转换的过程。

（**1**）页表一直推迟到访问页时才建立，以节约内存。

（2）页表项所包含的字段

页表项中的字段	说明
Present 标志	为 1，表示页（或页表）在内存；为 0，则不在内存。
页框物理地址(20 位)	页框大小为 4096，占去 12 位。20+12=32
Accessed 标志	页框访问标志，为 1 表示访问过
Dirty 标志	每当对一个页框进行写操 作时就设置这个标志
Read/Write 标志	存取权限。Read/Write 或 Read
User/Supervisor 标志	访问页或页表时所需的特权级
PCD 和 PWT 标志	设置 PCD 标志表示禁用硬件高速缓存，设置 PWT 表示写直通
Page Size 标志	页目录项的页大小标志。置 1，页目录项使用 2MB/4MB 的页框
Global 标志	页表项使用。在 Cr4 寄存器的 PGE 标志置位时才起作用

（3）利用两级页表实现地址转换



6　请求调页。所缺的页可能存放的地方。

（1）请求调页机制是把页框的分配一直推迟到进程要访问的页不在 RAM 中时引起一个缺页异常，才将所需的页调入内存。请求调页增加了系统中的空闲页框的平均数。

（2）所缺页的存放处：

1.该页从未被进程访问过，且没有相应的内存映射。

2.该页属于非线性内存映射文件。非线性内存映射的是文件数据的随机页。给定文件的所有非线性映射虚拟内存区域描述符都存放在一个双向链表中。

3. 该页已被进程访问过，但其内容被临时保存到磁盘交换区上。

4.该页在非活动页框链表中。

5.该页正在由其它进程进行 I/O 传输过程中。

7　了解盘交换区空间的管理方法。

盘交换区用来存放从内存暂时换出的数据页，每个盘交换区都由一组 4KB 的页槽组成。盘交换区的第一个页槽用来存放该交换区的有关信息，有相应的描述符。存放在磁盘分区中的交换区只有一个子区，存放在普通文件中的交换区可能有多个子区，原因是磁盘上的文件不要求连续存放。内核尽力把换出的页存放在相邻的页槽中，减少访问交换区时磁盘的寻道时间。

第 9-10 章　Linux 文件系统

1　Ext2 文件卷的布局？各部分的作用是什么？

Ex2 文件卷有若干磁盘块组成（1 个引导块和 n 个块组）。每个块组又由超级块、块组描述符、数据块位图、文件的索引节点位图、索引节点区和文件数据区组成。

（1）引导块。Ex2 文件卷的第一个盘块不被 Ex2 文件系统使用，他用作 Linux 系统的引导块或保留不用。引导块用以读入并启动操作系统。只有根文件系统的引导块才起作用。

（2）块组结构。每组包含存放在相邻磁道的数据块和索引节点。块组的大小相等并顺序安排。

采用这种结构，可以用较少的平均寻道时间访问在磁盘一个单独块组中的文件。

（3）超级块。存放整个文件卷的资源管理信息。

（4）数据块位图。记录文件数据区各个盘块的使用情况。

（5）索引节点位图。记录索引节点区各个索引节点的使用情况。

（6）索引节点区。存放文件的索引节点，一个索引节点存放一个文件的管理和控制信息。

（7）文件数据区。存放普通文件和目录文件。

2　Linux 系统把一般的文件目录项分成哪两部分？这样做的好处是什么？

把通常的文件目录项分成简单目录项和索引节点两部分。简单目录项包含了文件名和文件的索引节点号等，文件的控制管理信息放在文件的索引节点中。

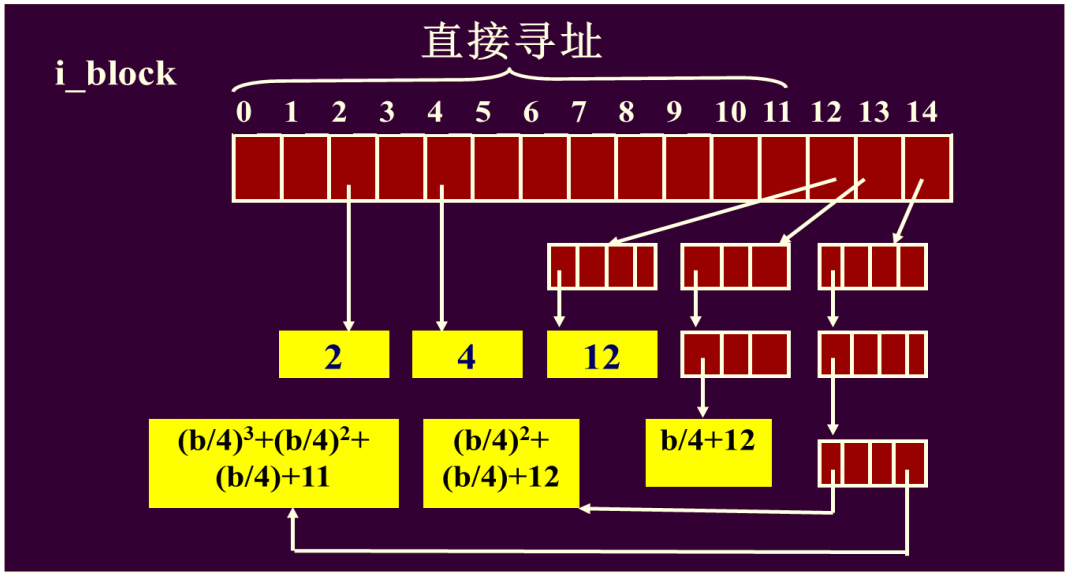
一方面，可以提高文件目录的检索速度和文件系统的使用效率；另一方面，通过多条路径共享信息文件时

系统只保留一个索引节点，减少文件的控制管理信息的冗余，提高了信息的一致性。

3　Linux 文件系统的索引节点中，索引表划分成几级？文件的索引表是如何增长的？要求能够利用索引表实现将文件中的字节地址转换成文件的物理块的操作。

（1）分为四级。直接索引，占前 12 项（0~11）；一次间接索引，占第 12 项；二次间接索引，占第 13 项；三次间接索引，占第 14 项。分别用来存储小型，中型，大型和巨型文件。

（2）



最初的 12 个元素是直接索引项，给出文件最初的 12 个逻辑块号对应的物理块号。

索引 12 是一次间接索引块，是一个存放盘块号的一维数组。对应的文件逻辑块号从 12 到（b/4）+11，b 是盘块大小，每个逻辑块号占 4B。

索引 13 是二次间接索引块，对应的文件逻辑块号从 b/4+12 到 $(b/4)^2+(b/4)+11$。

索引 14 是三次间接索引块，对应的文件逻辑块号从 $(b/4)^2+(b/4)+12$ 到$(b/4)^3+(b/4)^2+(b/4)+11$。

4　硬链接和符号链接的区别？

符号链接与硬连接的区别在于，他不与文件节点的索引建立连接。同时这种链接允许在不同的文件系统之间建立。当为一个文件建立符号链接时，索引节点的硬链接计数不变。

5　Linux 文件系统如何管理空闲存储空间？

（1）磁盘块和索引节点的分配和回收。（2）文件的数据块和其索引节点尽量在同一个块组中。（3）文件和它的目录项尽量在同一个块组中。（4）父目录和子目录尽量在同一个块组中。（5）每个文件的数据块尽量连续存放。

6　VFS 通用文件模型中的四个主要对象？

超级块对象：VFS 超级块对象是由各种具体的文件系统在安装时在内存建立的。

索引节点对象：对应具体文件系统一个文件的索引节点。

目录项对象：它代表一个目录项，存放目录项与对应文件进行链接的信息。

文件对象：他记录了由进程打开的文件与进程之间进行交互所必需的信息。是进程和文件的桥梁。

7　Linux 系统中，进程打开一个磁盘文件要涉及哪些数据结构？它们各有哪些关键字段？他们的作用是什么？参考图 10.2（P208）

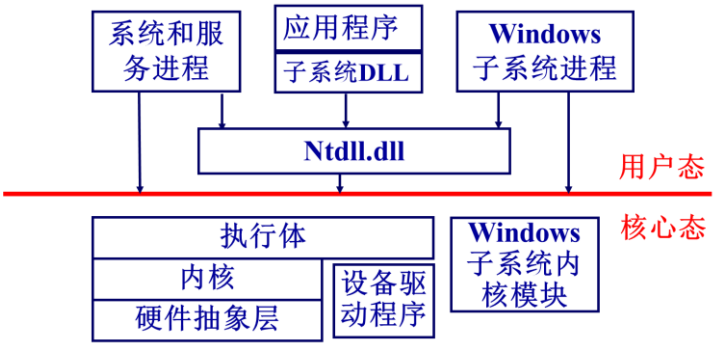
8　一个文件在使用与不用时各占用系统哪些资源？

9　安装表的作用是什么？

保存安装点与被安装的文件系统的信息。

第 14 章　Windows 2000/XP 模型

1.Windows 采用什么样的体系结构？



系统分为两种状态：核心态和用户态。粗线上方代表用户态进程，下方是核心态的操作系统服务。用户态的进程只能运行在受保护的地址空间。因此，四种类型的用户态进程都有各自的私有进程空间，核心态的操作系统服务组件运行在统一的核心地址空间。核心态组件包括：执行体、内核、文件和设备驱动程序、硬件抽象层，以及窗口和图等。

2.硬件抽象层 HAL 的作用是什么？

直接操纵硬件。HAL.dll 是一个可加载的核心态模块。HAL 隐藏各种与硬件有关的细节。使内核、设备驱动程序和执行体免受特殊硬件平台差异的影响。 系统可移植性好。

3.Windows 系统组件的基本机制包括：

陷阱调度、执行体对象管理器、同步（自旋锁、内核调度程序对象）、本地过程调用 LPC 等。

4.理解：延迟过程调用 DPC，异步过程调用 APC

（1）DPC 被用来执行一些相对于当前高优先级的任务来说不那么紧急的任务。

有时内核在进行系统嵌套调用时，检测到应该进行重调度。为了保证调度的正确性，内核用 DPC 来延迟请求调度的产生。

硬件中断服务例程可以把一些相对不紧急的事情放到一个 DPC 对象中处理，从而缩短处理机停留在高 IRQL 的时间。

DPC 队列是一种机制，它能记住有哪些工作尚未处理。

当 IRQL 降低到 DPC/Dispatcher 级别以下时，DPC 中断就产生。调度程序依次执行 DPC 队列中的每个例程，直至 DPC 队列为空。

DPC 队列是系统范围的。

（2）为用户程序和系统代码提供了一种在特定用户线程环境中执行代码的方法。

如果需要从内核空间复制一个缓冲区到某一用户进程地址空间缓冲区，那么复制过程需要在用户进程上下文运行，这样页表才能包含内核缓冲区和用户缓冲区。

每个线程都有自己的 APC 队列。 APC 队列也由内核管理。

5.Windows 中有哪些对象，都有什么作用？

两种类型对象：执行体对象和内核对象。

执行体对象就是执行体的各种组件实现的对象。执行体组件：进程和线程管理器、内存管理器、I/O 管理器、对象管理器等。内核对象是由内核实现的一批初级对象，对用户态代码不可见，仅供执行体使用。内核对象提供基本的能力，如执行体对象之间的同步。一个执行体对象可以包含一个或多个内核对象。

6.在多处理机系统中，提供了哪些同步和互斥机制？

内核引入自旋锁实现多处理机互斥机制。内核以内核对象的形式给执行体提供其他的同步机构—“调度程序对象”，包括：进程对象、线程对象、事件对象、信号量对象、互斥体对象、可等待的定时器对象及文件对象等。每个同步对象都有“有信号”或“无信号”两种状态

7.线程如何实现等待一个同步对象的操作？（P281）

一个线程通过等待一个对象的句柄可以与调度程序对象同步，这是，内核将线程挂起并把他的状态从运行态改变为等待态，且排列到等待对象的线程队列中。之后线程一直处于等待状态，直到所等待的程序调度对象句柄有无信号状态变为有信号状态。这是，内核将线程由等待态改为就绪态。线程通过调用由对象管理器提供的 WaitForSingLeObject 和 WaitForMultipleLeObject 系统服务来与调度程序对象同步。无论什么时候当内核将一个对象设置为有信号状态时，他将检查是否有线程在等待这个对象。如果有，内核会唤醒一个或几个线程，使它们能继续执行。

第 15 章　Windows 进程和线程管理

1. 管理进程和线程的数据结构：执行体进程块 EPROCESS、执行体线程块 ETHREAD、内核进程块 KPROCESS、内核线程块 KTHREAD。

2. 创建进程：CreateProcess( )；

① 打开将在进程中执行的映像文件(.exe)，创建一个区域对象，建立映像与主存之间的映射关系。

② 创建执行体进程对象，包括申请并初始化执行体进程控制块，创建并初始化进程地址空间、内核进程块和进程环境块等。

③ 创建一个主线程。

④ 通知 Win32 子系统，对新进程和线程进行一系列初始化。

⑤ 完成地址空间的初始化，开始执行程序

创建线程：CreateThread( )

① 在进程的地址空间为线程创建用户态堆栈。

② 初始化 CPU 硬件描述表

③ 调用 NtCreateThread（）创建执行体线程对象，填写有关内容，并将线程置为挂起状态，返回线程的标识和对向句柄。

3. 线程的 7 种状态，及其解释。(P 290)

1) 就绪状态(ready)
2) 备用状态(standby)。已选好处理机，正等待描述表切换，以便进入运行状态。
3) 运行状态(Running)
4) 等待状态(waiting)
5) 传输状态(transition)。核心栈被调到外存的就绪态。
6) 终止状态(terminated)
7) 初始化状态(Initialized)。正在创建过程中。

4. 线程调度：基于优先级的抢先式的多处理机调度系统。线程调度程序的数据结构：32 个就绪线程队列、32 位线程就绪队列位图、32 位处理机空闲位图。

负责记录各线程的状态。

① 32 个就绪队列。每个优先级对应一个。

② 32 位掩码的就绪位图。每一位指示一个调度优先级的就绪队列中是否有线程等待运行。

③ 32 位掩码的空闲位图。每一位指示一个处理机是否处于空闲状态。

5. 线程优先级的提升时机。

1) I/O 操作完成后的线程。
2) 信号量或事件等待结束的线程。
3) 前台进程中的线程完成一个等待操作。
4) 由于窗口活动而唤醒图形用户接口线程。

5)　线程处于就绪状态超过一定时间，仍未能进入运行状态(处理器饥饿)。

第 16 章　Windows 存储器管理

1　两种数据结构：虚拟地址描述符 VAD、区域对象，这两种结构各有什么作用？

（1）当一个线程要求分配一块连续虚存时，存储器管理器并不立即为其构造页表，而是为它建立一个 VAD 结构，记录该地址空间的相关信息：被分配的地址域、该域是共享的还是私有的、该域的存取保护以及是否可继承等信息。存储管理器通过维护一组 VAD 结构，记录每个进程地址空间的状态。一个进程的一组 VAD 结构构成一棵自平衡二叉树，以便快速查找。

（2）“区域对象”(section object)被称为“文件映射对象”，是一个可被多个进程共享的存储区。一个区域对象可被一个或多个进程打开。区域对象可被映射到页文件或磁盘上的其他文件。主要作用：1. 利用区域对象可将一个可执行的映像装入主存。然后访问这个文件就象访问主存中的一个大数组，而不是对文件进行读/写操作。

2. 使用区域对象可将一个大于进程地址空间的文件映射到进程地址空间。3. 高速缓存管理器利用区域对象访问一个被缓冲文件中的数据。

2　虚存内存区域：空闲的、保留的、提交的

进程私有的 2G 地址空间的页可能是空闲的，或被保留，或被提交。

被保留：已预留虚存，还没分配物理主存

被提交：已分配物理主存或交换区。

分配主存时，可以先保留地址空间，后提交物理主存；也允许保留和提交同时实现。

3　管理物理内存的数据结构：页框数据库。页框的 8 种状态：活动、转换、备用、更改、更改不写入、空闲、零初始化、坏，页框的状态转换图 16.9。原型页表项的概念。

1）页框数据库的作用

✓　进程页表用于跟踪虚拟页在物理主存的位置，页框数据库用于跟踪物理主存的使用情况。

✓　页框数据库是一个数组，其索引号从 0 到主存的页框总数-1。

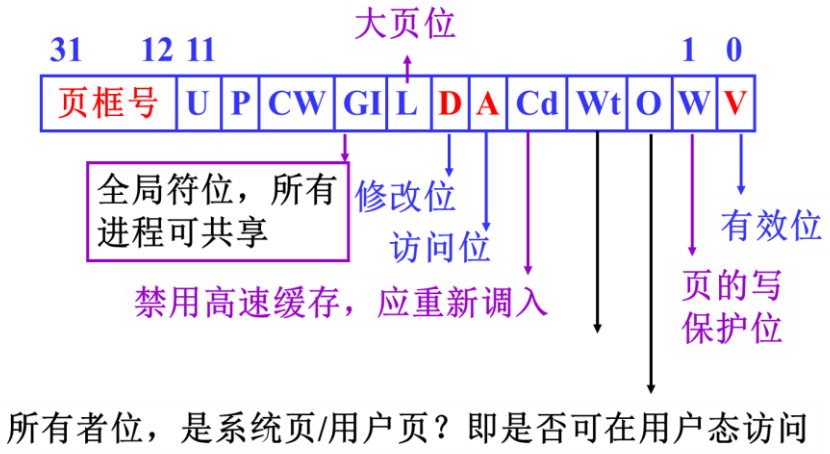
2）主存中的页框可能处于以下八种状态：

1. 活动（有效）：是进程工作集的一部分。
2. 过渡 ：不在进程工作集中，但未被破坏或该页框的 I/O 正在进行中
3. 备用：已不属于工作集，页表项仍然指向该页，但被标记为无效和处于过渡状态。
4. 更改：已不属于工作集，修改未写磁盘，页表项仍指向该页，被标记为无效和处于过渡状态。
5. 更改不写入 ：更改但不将该页框写入磁盘
6. 空闲 ：不属于任何一个用户进程的空闲页框
7. 零初始化 ：由零初始化线程进行了清零的空闲页框
8. 坏页框。

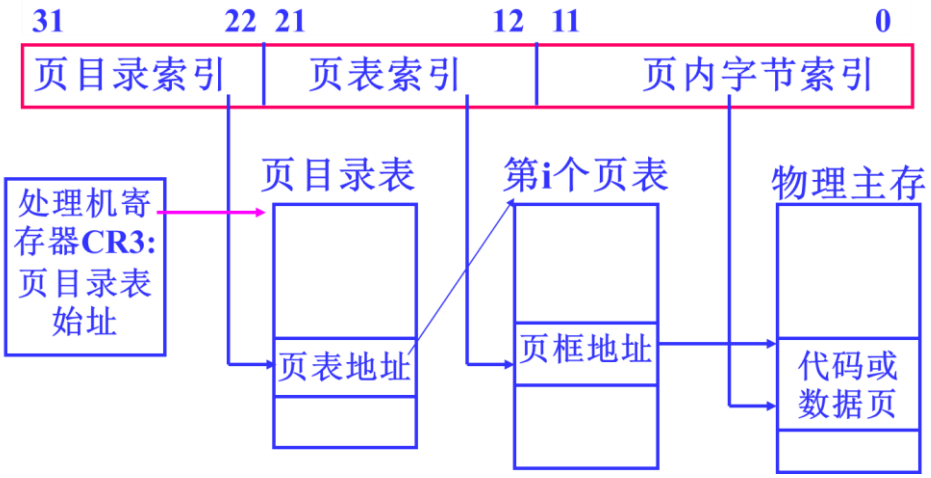
3）当一个页框被两个或多个进程共享时，存储器管理器依靠一个称为“原型页表项”（Prototype PTE）的软件结构来映射这些被共享的页框。当一个区域对象第一次被创建时，这些原型页表项“按段”同时被创建。

4　32 位逻辑地址，二级页表。页目录表项和页表项具有相同的数据结构，该数据结构包含哪些数据项？进程页表建立的时机。进程的地址转换过程。

（1）页表和页目录表的结构相同



（2）虚拟地址变换过程



5　Windows 采用的页替换策略是什么？

① 调页策略：将所缺的页及其前后的一些页装入主存。试图减少调页 I/O 次数。

② 置页策略：将虚拟页放到物理主存。

③ 置换策略：在多处理器系统中，采用了局部先进先出置换策略。而在单处理器系统中，

更接近于最近最久未使用策略(LRU，也称为“时钟页面置换算法”)。实现局部和全局置换的一种组合模式。

第 17 章　Windows 文件系统

1　Windows 所支持的文件系统类型有哪些？

支持两种格式的文件系统：FAT 和 NTFS 文件系统。FAT 表文件系统是支持向下兼容的文件系统，包括 FAT12，FAT16 和 FAT32 文件系统。12、16 和 32 分别为描述磁盘块簇地址使用的位数。 NTFS 使用的是 64 位的磁盘地址，规定使用 64 位的簇编号，但限制用 32 位来表达。

2　虚拟簇号和逻辑簇号的概念。

FAT 和 NTFS 将卷划分成若干簇，并从卷头到卷尾进行编号，称为**逻辑簇号(LCN)**。

NTFS 支持的文件的物理结构是索引式的。通过索引表建立文件的**虚拟簇号(VCN)**与磁盘的逻辑簇号之间的映射。

3　NTFS 卷的结构，主控文件表 MFT 的作用。

NTFS 卷结构：

MFT 是 NTFS 卷的管理控制中心，包含了卷上所有的文件、目录及空闲未用盘簇的管理信息；

- MFT 由若干个记录构成，记录的大小固定为 1KB。每个记录描述一个文件或目录。
- MFT 中的前 16 个记录是为 NTFS 元数据文件保留的。每个元数据文件具有一个以“$”开头的文件名，但该符号是隐藏的。
- 16 个元数据文件之后是一般文件和目录的记录

4　NTFS 文件的物理结构：索引顺序结构。

5　管理文件的目录结构采用 B-树。