

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：便于用户使用；

有效性：使系统高效可靠地运转，提高系统资源利用率；

便于设计、使用和维护；

2 操作系统是资源管理程序，它管理系统中的什么资源？

处理机；（进程管理，处理机调度）

存储器；（主存管理）

文件；（软件组成文件存放在磁盘或磁带上）

设备；（输入输出设备的管理和控制）

3 为什么要引入多道程序系统？它有什么特点？

提高系统利用率和系统的并行性；

主存上同时存放多个作业，同时处于运行状态，共享系统中的各种资源；

4 叙述操作系统的基本功能。

处理机管理，即进程管理和处理机的调度；

存储器管理：即主存的管理；

文件管理：将软件组成文件存储在磁盘或磁带上，并进行管理；

设备管理：管理和控制系统的输入、输出设备；

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？

① 批处理：系统吞吐量大、资源利用率高；单用户与作业无法交互，作业

平均周转时间长；

适用于计算量大、自动化程度高的成熟作业；

② 分时：同时性——若干用户同时使用一台计算机；

独立性——每个用户占据一台终端，独立操作；

交互性——用户可通过终端与系统人机对话；

及时性——用户请求可在较短时间内响应；

适用于短小的、要求用户与系统交互的作业；

③ 实时：实时性——响应用户请求的时间又被控制对象所能承受的延迟决定；

可靠性——系统具有容错能力；

确定性——系统执行指定操作按固定、预先确定的时间进行；

适用于仅处理外部响应触发的任务的系统；（不以作业为单位）

6 操作系统的特性？

并发性：若干逻辑上独立的程序，某时间均处于运行状态，相互竞争系统资源；

共享性：系统中的资源可供多个并发的进程共同使用；

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；

异步性：有限资源共享使并发进程相互制约，各进程的执行、推进、完成时刻均不可预知；

7 对于衡量 OS 的性能指标，什么是资源利用率、吞吐量、响应时间和周转时间？

- 资源利用率：给定时间内，系统某资源利用时间占总时间的比率；（注意存储器一直都被使用，因此它的利用率以空间计算）
吞吐量：单位时间内系统处理的信息量；（如几个作业/小时）
响应时间：提交请求到产生响应所需时间；
周转时间：作业进入系统到退出所用的时间；
- 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，软硬件可裁剪的专用计算机系统；
- 9 什么是对称多处理？它有什么好处？
操作系统和用户程序可安排在任意处理机上运行，各处理机共享主存和各种 I/O 设备；（不分主/从处理机）
提高系统资源利用率和系统并行程度；
- 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？
用户态（目态）和核心态（管态）；
用户态执行用户程序，核心态执行操作系统内核程序；
状态转换：在发生系统调用、中断和异常时产生。
管态→目态：操作系统程序；目态→管态：硬件；
- 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：操作系统提供的一些子程序；
特权指令：具有特殊权限的指令，一般不直接提供给用户使用；
特权指令执行时，系统处于核心态/管态；
- 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？
命令接口；程序接口；图形界面接口；
主要作用：便于用户通过操作系统使用计算机；

第 2-3 章 进程管理

- 1 程序顺序（串行）执行的特点
顺序性：自上而下的顺序执行；
封闭性：独占全机资源；
可再现性：初始条件相同，则程序执行结果一定；
- 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。
进程：又称任务，是程序的一次动态执行过程，即程序在一个任务集上顺序执行发生的活动；
进程=程序+数据+PCB；
动态性：程序的动态执行过程，有生命周期；
独立性：进程是系统分配资源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：系统为每个进程均建立一个 PCB；
- 3 进程控制块 PCB 的作用是什么？它主要包括哪几部分内容？
进程存在的唯一标识，可静态描述进程；
包括以下描述内容：可执行程序、地址空间、执行栈区、打开的文件、申请的 I/O 设备等；

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。

基本状态：运行、阻塞、就绪；

就绪→运行：系统调度；

运行→就绪：被抢占；

运行→阻塞：等待事件；

阻塞→就绪：事件完成；

5 什么是原语？什么是进程控制？

原语：若干条指令组成的、完成一定功能的一个进程；（重要性质：不可中断）

进程控制：系统使用一些具有特定功能的程序段创建和撤销进程，以及完成进程各状态间的转换；（内核实现，属原语级操作，不可中断）

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？

作业调度：高级调度；多道批处理系统中，多个用户作业成批提交到外存，形成后备作业序列，其中的作业被作业调度选进内存，就进入运行态；

交换调度：中级调度；将暂不具备运行条件的进程（阻塞/就绪）换出到外存交换区，将外存交换区中具备运行条件的进程换入主存；

进程调度：低级调度；

进程调度功能：记录系统中各进程执行状况，管理 PCB；

选择就绪进程占有 CPU；

进行进程上下文的切换（保存原现场，恢复新现场）；

方式：非抢先（非剥夺）和抢先（剥夺），区别在于是否允许剥夺现行进程的 CPU 给另一进程；

时机：现行进程完成或错误终止；

提出 I/O 请求，等待 I/O 完成；

分时系统时间片用完；

优先级调度有更高优先级进程就绪；

进程执行了某种原语操作；

算法：FCFS、SJF、HRN、优先级、时间片轮转；（注意 HRN 是 1+等待/运行）

周转时间：作业进入到退出的时间；

带权周转时间：周转时间/估计运行时间；

周转时间：作业从进入系统到退出的时间；

带权周转时间：周转时间乘以权值（权值即 1/运行时间）

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）。

进程逻辑上表示 OS 要做的一个作业，线程是组成该作业的许多可能的子任务；

进程有独立地址空间，线程则共享所属进程的资源，自己拥有很少的资源；

进程调度需要恢复现场，开销大；线程则只需变换一小部分的资源，效率高且更快；

引入线程之后系统并发程度更高；进程间，进程内的多线程之间均可以并发；

多线程共享所有资源时，一个线程可以改变另一线程的数据，安全性不如多进程（多进程无此问题）；

用户级线程：由用户程序通过调用用户态运行的线程库完成线程管理，线程调度算法需用户自己设计；内核以进程为单位进行调度，一个线程阻塞，其依附的进程也阻塞；

核心级线程：由内核完成线程管理，应用程序通过系统调用进行线程的创建和撤销；一个线程阻塞，不影响其他线程（以及整个进程）的执行；

- 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？

互斥：由于共享资源产生了关系，间接制约；

同步：某些同步点上，一些进程要相互等待发信息，之后才可继续进行，直接制约；

- 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：一次仅允许一个进程使用的系统临界资源；

临界区：并发进程访问临界资源时必须互斥执行的程序；（所以一直说是“临界区代码”）

互斥使用：互斥访问临界区；

让权等待：等待使用临界区的进程应当阻塞等待，避免处理机空转；

有空让进：处理机有空闲就允许进入，临界区外的任何进程都不能阻止其他进程进入临界区；

有限等待：不能使某进程无限期等待进入临界区；

- 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。

开关中断：允许中断的关闭，从而让临界区内的进程执行完；

test&set：Windows 内核实现进程互斥的一种机制，即使用硬件指令“测试与设置”；其为每个临界资源设置一个锁位变量，表示临界资源是否空闲，在进程中设置循环判断，以让进程获知临界资源何时空闲；（但会造成 CPU 忙等）

软件方法：允许进入标记 **turn**、申请即将进入标记数组 **flag**（**flag** 数组方法的死锁，类比两个胖子挤一扇门，谁都要进但又等对方先进）；二者结合的方法（**turn** 表示允许谁进的裁决，**flag** 表示谁要进）；

P、V 操作：原语一级，不允许中断。

P：申请一个资源，等待队列可能+1；也可表示发信息；

V：释放一个资源，等待队列可能-1；也可表示接收信息；

- 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？**

直观上，就是申请一个资源 S 和释放一个资源 S；

$S>0$ 表示有剩余资源，针对申请完成分配；

$S=0$ 表示正好没有剩余资源，但本次分配也完成；

$S<0$ 表示无剩余资源，当前申请的进程等待，S 的绝对值大小就是等待队列的长度；

- 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？

互斥信号量：允许几个进程同时访问，就设初值为几；

同步信号量：利用生产者消费者的观点，分析资源的数目，并设置对应数值为信号量初值；

13 经典的 IPC（进程间通信）问题。

详见生产者和消费者、读者和写者、理发师、哲学家进餐、阅览室问题；

14 进程高级通信有哪些实现机制？

消息缓冲：发送进程申请一个缓冲池中的缓冲区，将消息放入其中，再将缓冲区链接到接收进程的消息队列中；

信箱：发送进程将消息发送到信箱，接收进程从中接收消息；

管道：用于连接一个读进程和一个写进程的共享文件，先进先出，通过操作系统管理的核心缓冲区实现；

共享存储区：直接在主存划出一大块共享存储区，连接到各自地址空间，通过对其读写来共享数据；

15 死锁产生的必要条件及解决死锁的方法

四大必要条件：

互斥：独占性资源；

保持和等待：等待时均持有各自的资源，不释放；

非剥夺：不允许剥夺等待进程的资源；

循环等待：进程资源图中必存在一条循环等待链；

解决：①鸵鸟算法：忽略死锁，不解决；

②预防死锁：

使用虚拟设备相关技术如 spooling 破坏互斥条件，但不能实质性破坏设备的互斥性质，且不是每台设备都可以使用，并且磁盘空间不足仍会死锁；

使用静态分配破坏保持和等待条件，预先为进程分配好全部资源，若不能全部分配则不分配；但静态分配会造成资源浪费，多数情况无法预知各自需要资源的数量，盲目分配更会导致资源浪费；

使用强制剥夺等待进程资源破坏非剥夺条件，但恢复被剥夺资源的进程的运行现场代价非常大；

使用全局编号，并按编号升序为进程分配资源来破坏循环等待条件，但无法找到合适的全局编号；

③避免死锁：允许资源的动态申请，每次申请前使用银行家算法判断此次申请会否使系统进入不安全状态；

④检测与恢复：使用进程资源图结合 DFS 算法判断是否有循环等待链；之后使用故障终止进程或资源剥夺进行死锁恢复；

16 理解银行家算法的实质。能够利用银行家算法避免死锁。

银行家算法：分三个矩阵：分配矩阵、剩余请求矩阵、最大需求矩阵，再有一个剩余资源向量记录剩余的资源；

每次的申请先和剩余请求比较，看其是否合理；之后再看剩余资源能否满足申请，能则从中拿出这些申请，添加到分配矩阵对应行里，并相应减小剩余请求矩阵对应行；

若剩余请求矩阵这一行全为 0，则说明可以完成，之后从分配矩阵中拿出这一行所有资源加回剩余资源向量，之后分配可不考虑这一行；

针对当前的剩余资源向量，逐行判断剩余请求矩阵哪一行可以被满足，之后循环上面的判断过程直至无法继续满足或者全都能满足；

无法继续满足——系统会死锁，当前申请不安全，不可完成；

可全部满足——当前申请安全，可完成分配；

此算法既可以检查当前状态是否安全，又可以判断一次分配是否合理；

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：存储器分配（解决主存的共享）；

地址转换与重定位（完成地址变换）；

存储器保护（防止故障程序破坏 OS 和其他信息）；

存储器扩充（多级存储实现虚拟存储器及其管理算法）；

存储器共享（并发进程如何共享主存中的程序和数据）；

地址空间：源程序各种符号名限定的空间；

存储空间：物理存储器内全部物理存储单元构成的集合限定的空间；字或字节组成的大阵列；

地址空间虚拟，存储空间物理；

逻辑地址：编译链接后目标代码限定空间中的地址码；

物理地址：存储空间中字或字节对应的编号地址；

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：程序地址空间中的逻辑地址变换为物理地址；

分为静态重定位和动态重定位；前者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为存储空间物理地址；后者在装入时将逻辑地址原样装入，之后借助硬件地址变换机构，在程序运行中进行地址变换；

静态：优点在于无需硬件变换机构；

缺点在于程序执行期间存储区不能移动，主存利用率低，难以进行程序和数据共享；

动态：优点在于主存利用充分，用户程序可移动，程序不必占有连续存储空间，便于共享；

缺点在于需要硬件变换机构；

3 内存划分为两大部分：用户空间和操作系统空间。存储器管理是针对用户空间进行管理的。

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（防止访问超出为进程分配的存储区域），正确地进行存取（检查可读可写可执行，防止误操作破坏数据完整性）；

固定式分区：上下界寄存器限制访问上下界；

可变式分区：基址和限长寄存器限制访问逻辑地址的上下界；

页式管理：使用越界中断的判断机构判断是否越界，软件上为页表增加该页的操作方式位以表示可读可写可执行；

段式管理：硬件上，控制寄存器和段表内部均进行越界中断判断；软件上，为段表增加操作方式位；

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？

可变式分区使用动态重定位，在运行进程时 CPU 给出的是逻辑地址；

其使用基址和限长寄存器完成存储保护，分区首地址装入基址寄存器，程序大小装入限长寄存器；每个逻辑地址小于限长寄存器的内容；

覆盖：允许同一主存区可被不同程序段重复使用，实质上就是让不同时执行的程序段共用同一主存区；需要系统提供覆盖管理控制程序，用户提供覆盖结构；主要用于系统内部的程序管理；

交换：系统根据需把主存中暂不运行的进程的部分或全部信息移到外存，而把外存中的进程移进主存并投入运行；

覆盖与交换的区别：覆盖多用于早期 OS，针对同一进程内部；交换多用于现代 OS，针对不同进程之间；

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？

页表的作用是将进程虚页转换成主存物理块；

页式管理的地址变换过程为：首先取页号与控制寄存器中的内容进行越界判断，若不越界，则与基址相加得到页表对应页；之后从对应页取出页号对应的物理内存块号，取出逻辑地址作为页内地址，进行地址计算；

地址计算的方法为：物理块号 \times 物理块大小+页内地址；

页式存储器管理使用的数据结构主要有页表和页框。页框是将主存划分为的大小相等的若干块，可将页放入页框中，以实现页的离散放置和主存的离散分配；页表是系统为每个进程建立的页面映像表，记录逻辑页与主存块的映射关系；

7 什么是页式存储器的内零头（碎片）？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）？

内零头存在于页式管理的最后一页，若这一页未用完，则未用完的部分就是内零头即碎片；一般内零头的大小和页的大小成正比；

而可变式分区管理产生的碎片主要来自于空闲区的分配，分配算法要求从空闲块中取出和作业大小相当的一部分，剩下的部分若大小较小而无法使用，就成为了碎片；可变式分区的碎片是因为其要求每道程序必须占用主存中的连续空间而产生的；

8 段式存储器管理与页式管理的主要区别是什么？

① 段式存储器管理的地址码是二维的（段号+段内地址，用户地址空间被人为分了很多段），页式存储器管理的地址码是一维的（用户地址空间还是连续的）；

② 段式存储器管理的段长可以不相同，但页式存储器管理的页的大小是相同的；

③ 段是由用户划分的，而页是为管理方便由硬件划分的，对用户不可见；

④ 段允许动态扩充，而页不可以；

⑤ 段可能存在主存碎片，而页消除了碎片；

⑥ 段式管理便于动态链接，而页式只能进行静态链接；

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满足用户对存储器容量的需求而构造的一个非常大的地址空间；

- 虚存的容量不可能大于主辅存容量之和，最大值就是二者之和；
- 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。**
状态位：指示某页是否在主存，为 1 表示在主存，为 0 表示不在主存，硬件产生缺页中断，转中断处理程序进行缺页处理；
修改位：指示某页调入主存后是否被修改过，1 表示是，0 表示否；
访问位：指示某页最近是否被访问过，1 表示是，0 表示否（用于页面置换算法）；
- 11 **产生缺页中断时，系统应做哪些工作？**
①检查状态位是否为 0，若是则产生缺页中断；
②转操作系统处理缺页中断，寻找一个空闲的页框；
③若找到，则从磁盘读入页，并装入；
若未找到，则按某种页面调度算法将主存中某页换出主存，若修改过页面，还要写磁盘，再将外存中所需的页调入主存的这一页框；
④调入的页的页表和内存分配表要做相应的修改；
⑤中断结束，恢复现场，继续执行下一条指令；
- 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。**
FIFO：先进先出，新进入的页直接放在最上方，挤掉最下方的页；可能出现抖动和 Belady 异常（即为进程分配的主存块增加时，缺页中断次数可能反而增加）；
LRU：最近最少使用，新进入的页放在最上方，挤掉下面的页，但被访问的页会从任何位置直接升到最上方；有效避免了 Belady 异常；
OPT：最优，新进入的页挤掉的是下一次访问距现在最远的页；（若在访问队列的末尾附近，可能也看上一次访问距现在最远）
时钟置换：一种第二次机会算法，从时钟状的循环链当前指针位置出发，直至找到一个引用位为 0 的页，将其引用位改为 1，并置换页；之后指针移到下一位；
- 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？**
局部性原理：分为三方面；时间局部性指程序存在循环，一段时间内循环执行某部分；空间局部性指程序含有许多分支，但一次执行中只执行满足条件的分支；此外，进程运行过程中，用到某部分程序或数据时，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。**
大页表可以不在主存中连续存放，而是将其分散到许多页中存放，并为这些页建立相应的页表，即多级页表；
多级页表中，页表的建立不再在进程装入主存时，而推迟到要访问要访问页时才为包含该页的页表分配和建立页表项；
写时复制技术指若有进程写某个页，则系统把此页复制到主存另一个页框中，并更新该进程的页表，并置该页为可读/写；
- 15 **页的共享问题。需要一个专门数据结构来记录进程间共享页。**
页的共享可避免出现多个同一页的副本，节约主存；
通常只读页可共享，而可读/写的数据页不可共享；

共享页除了要专门的数据结构记录以外，还需要在页表中添加引用计数项，仅当此项为 0 时才允许调出或释放盘空间；

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：存储在外存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、相应管理软件和被管理文件；

UNIX 的文件分类为：普通文件——通常的文件；目录文件——文件目录构成的一类用来维护文件系统结构的文件；特别文件——输入设备和输出设备（字符型特别文件）、I/O 型文件（字符块型特别文件）、管道文件；（UNIX 完全以文件方式进行处理）

可以使用文件的观点统一处理文件、目录和设备，可靠性高；

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

文件目录的作用主要是使用户按名存取文件；

文件目录项通常包含：文件名、文件结构信息、文件存取控制信息、文件管理信息；操作系统使用文件控制块管理文件，它将文件控制块保存在目录文件中，每个文件占用一个目录项；

3 文件的逻辑结构有几种形式？文件的存取方法？

文件的逻辑结构包括：无结构的字节流文件——无结构、先后到达的字节流组成，文件长度即包含的字节数；有结构的记录式文件——分为定长式和变长式两种；

文件的存取方法有：顺序存取——按文件信息逻辑顺序依次存取，在前一次存取基础上进行；直接存取——基于文件的磁盘模型，允许对任意文件块随机读写（记录式：根据记录编号实现，字节流：利用系统调用移动读写指针实现）；

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？

文件的物理结构：

① 连续文件——线性表，文件内容连续存放；

② 链接文件——不要求文件内容连续存放，文件所占用物理块用链接指针连接；

③ 索引文件——为每个文件建立一张索引表，以记录文件内容存放地址；（记录文件逻辑块号与对应物理块号的关系）

④ 索引顺序文件——索引结构的变形结构；索引表记录每个文件逻辑块开始的 VCN 与开始的 LCN，VCN 为虚拟簇号（逻辑的地址），LCN 为逻辑簇号（虽然叫逻辑，但因为物理上相邻，故也可看做物理的地址）；

管理方式：连续文件——空白文件目录；链接文件、索引文件、索引顺序文件——空闲块链表；索引顺序文件——还可使用位映像表（即位示图）；

5 DOS 文件卷的结构，DOS 系统的文件物理结构是什么？

DOS 文件卷为 FAT 文件系统结构，使用文件分配表 FAT 分配和管理磁盘空间；

MS-DOS 系统的文件物理结构为链接文件结构；

其使用无结构的字节流形式组织和处理文件；

6 了解记录的组块和分解。

一个物理块可存放若干个逻辑记录，一个逻辑记录又可存放在若干个物理块中；称一个块中存放的逻辑记录个数为块因子；（记录组块）

信息交换以块为单位进行，用户将要写的记录先存入主存缓冲区，其满时再写磁盘，即必须使用主存缓冲区；

用户使用记录时，先将包含该记录的物理块读入内存缓冲区，之后再进行记录分解，得到其中的逻辑记录；（记录分解）

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？

文件存储空间的管理方法：空白文件目录、空闲块链表、位映像表；空闲块链表又分为空闲块链（单链）和空闲块成组链表；

空白文件目录：为所有空白文件建立一张表，每个空白文件占用一个表目；

空闲块链：把所有空闲块链接成一张链表；

空闲块成组链表：利用盘空闲块管理盘上的空闲块，一个磁盘块记录尽可能多的空闲块而成为一组，各组之间使用链指针链接；

位映像表：使用位示图进行管理；涉及到磁盘相对块号与位示图字节号和位号之间的转换；

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？

多级目录的优点：①层次结构清晰，便于管理维护；②有利于文件分类；③可解决重名问题；④提高文件检索速度；⑤可控制文件存取权限；⑥便于共享；

重名的解决：重名文件的完全路径名不同，从而得以区分；

共享的解决：可使用符号链接文件（一种很小的文件）存储欲共享文件的绝对路径名；

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？

打开 Open，关闭 Close，读 Read，写 Write，追加 Append，随机存取 Seek，得到文件属性 Get Attributes，设置文件属性 Set Attributes，重命名 Rename；

打开文件目的：根据文件名找到文件目录项，进而找到 FCB，将 FCB 复制到内存并记录到系统打开文件表；

关闭文件目的：释放文件在主存专门区域的目录项，切断与用户文件的联系；

10 存取控制表 ACL 的概念。

直接使用存取控制矩阵记录各个域对象的存取控制权限开销太大，因此使用 ACL 记录。

ACL 为存取控制矩阵中每列建立，用有序对(域,权集)表示，ACL 在文件的 FCB 中；例如 File1(D1,rwx)表示 File1 在域 D1 中，可读可写可执行；

11 理解内存映射文件（memory mapped file）的过程。

内存映射文件，即针对某个具体的文件创建一个它的文件映射对象，又称区域对象；创建区域对象即将文件映射到进程地址空间中一个区域，返回虚拟地址，仅当需要对文件存取时才传输实际数据；

其存取机制类似于请求页式存储器管理，当映射的页不存在时，也会产生缺页中断，之后转操作系统进行缺页处理；

第 6 章 设备管理

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

I/O 设备分类：字符设备，块设备，网络通信设备，时钟；

字符设备：以字符为单位发送和接收数据，通信速度较慢；

块设备：以块为单位传输数据，传输的数据成块，通信速度较快；

2 常用的四种数据传输方式。

程序查询方式：向设备控制器发写命令，读设备控制器的状态，若完成，则从内存读数据并向设备控制器写入，否则循环判断设备控制器是否完成；此方式 CPU 和设备串行工作，循环等待的过程会造成 CPU 忙等；

中断方式：向设备控制器发写命令后，CPU 转去执行其他程序，同时设别驱动程序启动设备工作，直至传输完成发出中断；CPU 响应中断，中断当前服务的例程，之后检查传输是否出错，无错则完成，有错则重试或故障终止；此方法传送完一部分就要让 CPU 中断一次，效率仍不够高；

直接存储器访问 DMA：让 DMA 控制器接管地址总线控制权，直接控制控制器内部缓冲区与主存进行数据交换；其仅在开始和结束时需要 CPU 干预，中间 CPU 可做其他事，减少了中断次数，提高了 CPU 运行效率；

通道控制方式：使用专用的 I/O 处理机——通道处理多个设备的 I/O 请求，有自己的指令系统，若干条通道命令连接成通道程序，其所需的 CPU 干预更少，且可做到一个通道控制多台设备；

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

设备：独占性设备——对应临界资源，共享时只允许一个进程在某时间段内使用；

共享设备——共享时允许多个进程的交叉访问；

虚拟设备——设备本身是独占设备，但通过虚拟技术处理后可改造成共享设备；

虚拟设备使用了虚拟设备技术，使用可共享的高速设备模拟独占的慢速设备，以提高独占设备利用率；

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？

I/O 软件分层为：中断处理程序——处理中断，检查设备状态，若正常完成，则唤醒等待进程，并处理后面可能的待处理 I/O 请求，若传送出错，则重新传输或向上层报告设备错误信息；

设备驱动程序——每种设备驱动程序处理一种类型的设备；

独立于设备的软件——实现所有设备都需要的功能，如设备命名、保护、提供与设备无关的块尺寸、缓冲技术解决主存与设备速度不匹配的问题、设备分配与调度、出错处理等；

用户层的 I/O 接口——用于发出一些系统调用如读文件；

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 技术是一种以空间换时间的技术，利用可共享磁盘的一部分空间模拟独占 I/O 设备；

例如输出时，系统把待输出数据缓冲到一个独立磁盘空文件上，形成待输出文件队列，之后 spooling 系统一次一个地将待输出队列上的文件输出；

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？

花费的时间分为寻道时间、旋转延迟时间、读/写传输时间；（可以这样理解：先寻磁道，之后旋转，最后进行读写操作，三步都有时间开销）

9 常用的磁盘调度算法：**先来先服务、最短寻道时间优先、扫描法**（SCAN, C_SCAN, LOOK, C_LOOK）。

FCFS：按照磁盘请求序列自带的顺序进行寻道；

SSTF：最短寻道时间，贪心算法，每步都选择距当前位置最近的磁道移动；

扫描：SCAN——扫到头反向；

C_SCAN——扫到头回最底下；

LOOK——扫到最后一个记录反向，反向后也在最后一个记录处不继续；

C_LOOK：C_SCAN+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，几个唯一标识符。

Linux 的 PCB 的具体实现就是 `task_struct`；其一些重要字段有：

`thread_info`：进程管理，当前进程的基本信息；

`mm_struct`：存储器管理，指向进程虚拟内存描述符；

`fs_struct`、`files_struct`：文件管理，分别指向文件系统信息和打开文件的信息；

Linux 进程和线程中常用的唯一标识符有：

`PID`：进程的唯一标识符；

`tgid`：线程组标识符；

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

`fork()`：创建子进程，创建成功后子进程采用写时复制技术读父进程全部地址空间，仅当父进程或子进程要写一页时，才为其复制一个私有的页副本；

`vfork()`：阻塞父进程执行地创建子进程（直至子进程退出或执行一个新程序）；

`clone()`：创建轻量级进程；

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切换页目录表，以安装一个新的地址空间；

②切换核心栈和硬件上下文；

4 **进程调度方式。进程调度时机。**

Linux 采用的是可抢占的动态优先级调度方式；

进程调度时机包括：当前进程阻塞等待信息或执行完；当前进程故障中断；有高于当前进程优先级的进程处于就绪态；

5 Linux 有很多内核线程，了解 0 号进程和 1 号进程的作用。

0 号：内核进程，所有进程的祖先进程；

1 号：0 号进程创建的内核线程 `init`，负责内核初始化；系统关闭前 `init` 一直存在，负责创建和监控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？

32 位机有 4GB 进程地址空间，Linux 进程前 3G 为私有，后 1G 为公有；

Linux 使用虚拟内存描述符描述和管理进程私有地址空间；

虚拟内存描述符中，指向映射文件对象的字段为 `*vm_file`，指向单链表的字段为 `*vm_next`，红黑树字段为 `vm_rb`；

指向进程页目录表的字段为 `mm_struct` 下的 `*pgd` 字段；

2 **Linux 堆的管理：**`malloc()`，`free()`。

3 **管理物理内存页框的数据结构？分区页框分配器分配页框的过程。**

管理物理内存页框的数据结构为 `zone` 结构；

分区页框分配器针对大的块请求，使用伙伴系统，从与块请求尺寸最接近的连续页框块链表中寻找页框块，若不能满足则在更大的页框块链表中寻找，直至找到，并一部分留给请求，另一部分按大小插入某尺寸页框块的链表中；

而针对较小的请求，使用 `slab` 分配器，获得几组连续页框，为不同类型对象生成不同高速缓存，每个高速缓存存储同类对象；

4 **理解 `slab` 分配器的原理。`slab` 分配器的作用？**

原理：获得连续页框，为不同类型对象生成不同的高速缓存放在页框里，形成每 `CPU` 页框高速缓存；

作用：为只有几十或几百 `B` 的内存区分配内存；

5 **进程页表建立的时机？** 逻辑地址的划分，利用两级页表实现地址转换的过程。

进程页表一直推迟到访问页时才建立，以节约内存；

逻辑地址分为 3 个域：前 10 位为页目录索引；中间 10 位为页表索引；后 12 位为页内偏移；

转换过程即先由页目录索引得到页目录表对应位置，之后再得到对应页表始址，由页表索引得到页的对应始址；最后利用页内偏移与页的对应始址计算得到物理地址；

6 **请求调页。所缺的页可能存放的地方。**

Linux 请求调页使用的是最少使用页面置换策略 `LFU`；

所缺的页可能存放的地方为：某个从未被进程访问过，且无相应内存映射的页处；

已被进程访问过，但其内容被临时保存在磁盘交换区上的页处；

非活动页框链表中的页处；

正由其他进程进行 `I/O` 传输过程中的页处；

7 **了解盘交换区空间的管理方法。**

每个盘交换区都由一组 `4KB` 的页槽组成；其第一个页槽存储改建换取的有关信息，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（连续的），存放在普通文件中的交换区可能有多个子区（离散的），因为磁盘上的文件不要求连续存放；

内核尽力把换出的页放在相邻页槽中，减少访问交换区时的磁盘寻道时间；

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？**

Ext2 把磁盘块分为组，每组包含存放在相邻磁道的数据块和索引节点；块组大小相等、顺序安排；

Ext2 使用块组描述符描述这些块组本身的结构信息，同时将超级块和所有块组描述符重复存储于每个块组中；**Ext2** 通过位图（盘块位图和索引节点位图）管理每个块组中的磁盘块和索引节点；

各部分的作用为：①超级块存放整个文件卷的资源管理信息；

②索引节点存放文件的管理控制信息；

③盘块位图和索引节点位图用于管理块组中的磁盘块和索引节点；

④索引节点区和文件数据区存放真正的索引节点和文件数据；

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？

Linux 中，一般的文件目录项被分为简单目录项和索引节点两部分；
简单目录项包含文件名和索引节点号等，可提高文件目录检索速度；
索引节点系统只需保留一个即可实现多条路径共享文件，减少信息冗余；

3 Linux 文件系统的索引节点中，索引表划分成几级？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。

索引表划分成四级：直接索引项、一次间接索引项、二次间接索引项、三次间接索引项；

直接索引项：直接给出前 12 个逻辑块号对应的物理块号；

一次间接索引项：给出的是逻辑块号 $(12 - b/4 + 11)$ ；

二次间接索引项：给出的是逻辑块号 $(b/4 + 12 - (b/4)^2 + b/4 + 11)$ ；

三次间接索引项：给出的是逻辑块号 $((b/4)^2 + b/4 + 12 - (b/4)^3 + (b/4)^2 + b/4 + 11)$ ；

4 硬链接和符号链接的区别？

硬链接：为同一文件取新的符号名；

符号链接：使用符号链接文件（一种极小的文件）存储欲共享文件的绝对路径；

5 Linux 文件系统如何管理空闲存储空间？

Linux 使用位映像表管理空闲存储空间；

注意位示图方法的公式！

6 VFS 通用文件模型中的四个主要对象？

超级块对象、索引节点对象、目录项对象、文件对象；

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

涉及的数据结构依次有：task_struct、files_struct、file、dentry、ext2_inode_info；

涉及的关键字段依次有：task_struct 的 files；files_struct 的 fd_array[]；file 的 f_dentry；dentry 的 d_inode；

各关键字段的作用为：task_struct 的 files 记录进程打开的文件信息；

files_struct 的 fd_array[] 记录某打开文件中的各文件对象；

file 的 f_dentry 记录某文件对象对应的目录项对象；

dentry 的 d_inode 指向具体文件的 inode；

8 一个文件在使用与不用时各占用系统哪些资源？

文件在不使用时仅占用外存资源，而文件在使用时会占用主存（内存）资源，因为 Linux 的虚拟文件系统 VFS 会在主存中建立一系列的数据结构便于主存使用文件；

9 安装表的作用是什么？

由于 Linux 的文件系统在使用之前必须注册（获得 VFS 提供的调用文件系统方法的接口）和安装（将文件系统安装到根文件系统某个目录节点上），故需要使用安装表 vsmount 结构保存各文件系统的安装点和对应文件系统的信息；

第 14 章 Windows 2000/XP 模型

1. Windows 采用什么样的体系结构？

体系结构见图；

2. 硬件抽象层 HAL 的作用是什么？

HAL 直接操控硬件，对应的 HAL.dll 是一个可加载的核心态模块；

其隐藏各种有关硬件的细节，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，其系统可移植性好；
3. **Windows 系统组件的基本机制。**
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**
DPC 用于执行一些相对当前高优先级任务不那么紧急的任务；相对不紧急的事情放入 DPC 对象，形成 DPC 队列，之后当中断请求级别 IRQL 降低至 DPC/Dispatcher 以下，就产生 DPC 中断，调度程序依次执行 DPC 队列中的例程直至其为空；
APC 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，各线程均有其 APC 队列；
理解“特定环境”：如要从内核空间复制一个缓冲区到用户进程地址空间，则这个复制过程必须在用户进程上下文进行，才能使页表包含内核缓冲区和用户缓冲区；
 5. **Windows 中有哪些对象，都有什么作用？**
Windows 中有两种类型对象：执行体对象和内核对象。
执行体组件：进程和线程管理器、内存管理器、I/O 管理器、对象管理等。
内核对象是由内核实现的一个初级对象集，对用户态代码不可见，仅供执行体使用。
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？**
内核同步：多处理器系统中，内核引入自旋锁实现多处理机互斥机制，通过硬件指令“测试与设置”实现；（详见第三章）
执行体同步：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？**
对象管理器提供了两个系统服务程序，使线程与调度程序对象同步；
`WaitForSignalObjects();` //等待单个调度程序对象（即同步对象）；
`WaitForMultipleObjects();` //等待多个调度程序对象（即同步对象）；

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()`；创建线程：`CreateThread()`
3. **线程的 7 种状态，及其解释。**
初始化状态：正在创建过程中；
就绪状态：已具备所有运行的条件，只是未选好处理机；
传输状态：核心栈被调到外存的就绪状态；
备用状态：已选好处理机，等待描述表切换以运行；
运行状态：线程处于运行中；

等待状态：线程等待某信号，停止运行中；
终止状态：线程执行完成或错误终止，停止执行；

4. 线程调度。

Windows 的线程调度使用基于优先级的抢先式多处理机调度系统。

使用的数据结构有：32 个就绪线程队列、32 位线程就绪队列位图（对应 32 个就绪线程队列）、32 位处理机空闲位图（对应 32 台处理机）。

5. 线程优先级的提升时机。

- ① I/O 完成后；
 - ② 信号量或事件等待结束的线程；
 - ③ 前台进程完成一个等待操作；
 - ④ 窗口活动导致 GUI 线程被唤醒；
 - ⑤ 线程处于就绪状态超过一定时间，但仍未进入运行状态，导致处理器饥饿；
- （注意进程只有在普通优先级 7/9 才能提升优先级，但线程提升优先级的限制就比这小很多，在如上的情况下提升。）

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？

VAD：线程要求分配一块连续的虚存时，系统并不立即为其建立页表，而是先为其建立一个 VAD 结构，记录被分配的地址域、域的共享/私有、域的存储保护和是否可继承等信息；

区域对象：即文件映射对象，是一个可以被多进程共享的存储区；一个区域对象可被多个进程打开；使用区域对象，访问文件就像访问主存上的一个数组一样，无需文件 I/O 操作；

区域对象的作用为：① 可将一个可执行文件装入主存；

② 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 可被缓存管理器用于访问被缓存文件中的数据；

2 虚拟内存区域

空闲的：未使用的虚存区域；

保留的：已预留虚存，未分配物理主存；

提交的：已分配物理主存或交换区；

3 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。

数据项包括：页框号、全局符位、大页位、修改位、访问位、禁用高速缓存标志、写直通位、所有者位、页的写保护位、有效位；

进程的页表在访问页时才建立（同 Linux），称为懒惰方式；

虚地址变换的过程就是标准的二级页表虚地址变换的过程：先用页目录索引找页表地址，再用页表索引结合页表地址找页框地址，最后用页内地址结合页框地址计算物理地址；最终再访问物理主存中的代码或数据页；

4 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。

5 原型页表项，区域对象的页表。虚拟页式中，采用原型页表实现多进程共享页。

原型页表是区域对象的页表，用于记录被共享的页框

进程访问区域对象中的页（即共享一个页）时，需要使用原型页表项并将其信息填

写到进程的页表中；

6 Windows 采用的页替换策略是什么？

Windows 使用一种最接近 LRU 的策略替换页，实质就是时钟页面置换；

进程访问的页放在一个时钟状的循环链中，每一页的页表要附设最近访问位；使用一个指针完成操作，替换一页时，用指针顺时针找到第一个最近访问位为 0 的页（最近未访问），将其替换为新页，并置新页的最近访问位为 1；之后指针向后移动一位；

第 17 章 Windows 文件系统

1 Windows 所支持的文件系统类型有哪些？

Windows 支持文件系统 FAT12、FAT16 和 FAT32 以及 NTFS；

2 NTFS 文件的物理结构：索引顺序结构。虚拟簇号和逻辑簇号的概念。

Windows 的 NTFS 文件卷使用的是索引顺序的文件结构；其涉及虚拟簇号和逻辑簇号；

虚拟簇号 VCN：在文件卷上的编号，是一个虚地址；

逻辑簇号 LCN：块上的编号，虽然是“逻辑编号”，但也可以当做物理地址处理；

3 NTFS 卷的结构，主控文件表 MFT 的作用。

NTFS 文件卷的结构依次包含：分区引导扇区、主控文件表区、文件数据区；

分区引导扇区记录卷的布局、文件系统结构和引导代码等信息；

主控文件表区记录卷上所有文件、目录和空闲盘簇的管理信息，是 NTFS 文件卷的管理控制中心；

4 管理文件的目录结构采用 B-树。