

密训·资料

操作系统概论（全国）

1904

MI XUN ZI LIAO

编前语

试卷说明：本科目考试题型为单选、填空、简答和综合题。各题型分值分布情况如下表所示：

题型	分值
单选题	1 分×20 道=20 分
填空题	2 分×10 道=20 分
简答题	4 分×5 道=20 分
综合题	10 分×4 道=40 分

资料说明：本资料提供的都是考试重点，但是在重点中，按照知识点的重要程度，对知识点进行了标星。如下表：

星级	说明
1-2 星	易考单选、填空，2 星相对 1 星更重要。
3-4 星	易考简答，4 星相对 3 星更重要。
5 星	易考综合题

星标数目越多，表示知识点越重要，越可能考到。但要注意的是，资料提供的是知识重点，并不是押题。文中加粗的是需要重点了解的地方。这份资料是为了指导学员考前最后的复习，帮助学员理清知识点间的逻辑，加速记忆。学员在拿到资料后，可根据星标数、资料里的逻辑整理和标识，了解哪些知识点是重点，然后重点复习。

目录

第一章 操作系统简介.....4

 第一节 什么是操作系统..... 4

 第二节 操作系统的发展☆☆..... 4

 第三节 操作系统的特征☆☆..... 5

 第四节 操作系统的功能☆☆..... 5

 第五节 操作系统的体系结构☆..... 5

 第六节 指令的执行..... 6

第二章 进程管理.....6

 第一节 进程的描述..... 6

 第二节 进程的控制☆..... 7

 第三节 操作系统内核..... 8

 第四节 进程同步☆☆..... 9

 第五节 进程通信..... 10

 第六节 线程..... 10

第三章 进程调度与死锁.....11

 第一节 进程调度的功能与时机..... 11

 第二节 进程调度算法..... 11

 第三节 实时系统中的调度..... 12

 第五节 多处理器调度..... 13

 第六节 死锁..... 14

第四章 内存管理.....15

 第一节 存储器的层次结构..... 15

 第二节 程序的链接和装入☆☆..... 16

 第三节 连续分配存储管理方式..... 16

 第四节 基本分页存储管理方式..... 16

 第五节 基于分页的虚拟存储系统..... 18

 第六节 分段存储管理..... 19

 第七节 Linux 的伙伴系统..... 20

第五章 文件系统.....21

 第一节 文件..... 21

 第二节 目录☆..... 21

第三节 文件系统的实现.....	21
第六章 I/O 设备管理.....	23
第一节 I/O 系统的组成☆☆.....	23
第二节 I/O 控制方式☆☆.....	23
第三节 缓冲管理☆☆.....	24
第四节 设备分配☆☆.....	24
第五节 I/O 软件原理.....	25
第六节 磁盘管理.....	25



第一章 操作系统简介

第一节 什么是操作系统

1.1.0 操作系统的定义☆☆

操作系统 (OS) 是一种复杂的系统软件, 是不同程序代码、数据结构、数据初始化文件的集合, 可执行。

1.1.1、1.1.2 操作系统的功能☆☆

用户与硬件之间的接口	硬件接口和软件接口
资源的管理者	处理器管理、存储管理、文件管理和设备管理。
为应用程序的运行提供环境	

第二节 操作系统的发展☆☆

系统		特点	优缺点
无操作系统		因等待人工操作暂停——运行, 这样一种不能连续自动工作的状态。	
单道批处理系统		(1) 自动性 (2) 顺序性 (3) 单道性	优点: 减少了等待人工操作的时间。 缺点: CPU 资源不能得到充分利用。
多道程序系统	多道批处理系统	(1) 多道性 (2) 无序性 (3) 调度性 (4) 复杂性	优点: 能够提高 CPU、内存和 I/O 设备的利用率和系统吞吐量。 缺点: 系统平均周转时间长, 缺乏交互能力。
	分时操作系统	(1) 多路性 (2) 独立性 (3) 及时性 (4) 交互性	优点: 向用户提供了人机交互的方便性, 使多个用户可以通过不同的终端共享主机。
实时操作系统		(1) 多路性 (2) 独立性 (3) 及时性 (4) 交互性 (5) 可靠性	用于实时控制和实时信息处理领域 优点: 比分时系统要求有更高的可靠性。

1.2.7 七、操作系统产品现状☆

主机操作系统	运行在大型主机上的操作系统, 主要提供三类服务: 批处理、事务处理和分时处理。
服务器操作系统	运行在网络服务器上的操作系统, 可以通过网络同时为众多用户服务, 允许用户共享硬件和软件资源。服务器可提供打印服务、文件服务和 Web 服务。
微机操作系统	也称个人机操作系统, 现代微机操作系统都支持多道程序处理, 就是通常所说的支持多任务。微机操作系统为单个用户提供良好的应用环境和应用软件开发环境。
嵌入式操作系统	特征是小巧、实时性、可装卸、代码固化、弱交互性、强稳定性、接口统一、低能耗。 应用领域有掌上电脑、智能手机、自动售货机、自动取款机等。

第三节 操作系统的特征☆☆

并发	两个或多个事件在同一时间间隔内发生，多道程序系统可以实现并发执行。
共享	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。资源共享有两种方式，即互斥共享和同时共享。
虚拟	指通过某种技术把一个物理实体变成若干逻辑上的对应物。
异步性	进程以不可预知的速度向前推进。主要指时间不确定。

第四节 操作系统的功能☆☆

内存管理 (目的：提高内存的利用率)	内存分配	主要任务是为每道程序分配内存空间。可采用两种分配方式，即静态分配方式和动态分配方式。
	内存保护	任务：(1)使操作系统内核的空间不会被用户随意访问，以保证系统的安全和稳定；(2)确保每道用户程序都在自己的内存空间中运行，互不干扰。 实现方式：可以采用界限寄存器存放允许程序访问的地址区间的上限和下限值。
	地址映射	CPU 执行程序过程中访问内存时，需把程序的逻辑地址转变为物理地址，这个转换的过程称为地址映射。
	内存扩充	任务是借助虚拟存储技术，从逻辑上扩充内存容量，使系统能够向用户提供比物理内存大的存储容量。 以实现功能：请求调入功能和置换功能。
进程管理	包括进程的描述与组织、进程控制、进程同步、进程通信及进程调度。	
设备管理	主要完成用户的 I/O 请求，为用户分配 I/O 设备。应具有以下功能：(1)缓冲管理(2)设备分配(3)设备处理(4)设备独立性和虚拟设备。	
文件管理	(1)文件存储空间的管理：为每个文件分配必要的外存空间，提高外存利用率，并能有助于提高访问文件的速度。 (2)目录管理：为每个文件建立目录项并对众多目录项进行有效组织。 (3)文件的读、写管理和存取控制。	
提供用户接口	为了方便用户使用操作系统，操作系统向用户提供了用户与操作系统之间的接口。操作系统向 最终用户 提供 命令行和图形用户接口 ，向 程序员 提供应用程序与操作系统之间的接口即 系统调用 。 命令接口又可分为联机用户接口和脱机用户接口。	

第五节 操作系统的体系结构☆

简单的监控程序模型	功能简陋，任意时刻系统中只能运行一个任务
单体结构模型	所有的软件和数据结构都放置在一个逻辑模块中，对外层的用户程序提供一个完整的内核界面——系统调用。整个系统由若干具有一定独立功能的子程序组成，允许任一子程序调用其他子程序。

层次结构模型	基本思想：将操作系统分解为多个小的、容易理解的层，系统功能被隔离在不同层中。
微内核结构	微内核技术是操作系统发展的一个里程碑，它产生了一种完全不同的操作系统体系结构，提供了操作系统发展的新途径。其好处是系统结构清晰，具有较高的灵活性、可靠性和可维护性；一个严重的缺点就是效率不高。
动态可扩展结构模型	基本思想：在运行过程中，能够动态地实现系统行为扩展的结构，也可称之为弹性结构。

第六节 指令的执行

1.6.1 一、指令周期☆

一个单一指令需要的处理称为指令周期。一个指令周期可以划分成两个步骤：取指周期和执行周期。

1.6.2 二、取指令和执行指令☆☆

(1) 取指令：在每个指令周期开始时，处理器从存储器中取一条指令。在典型的固定长度指令的处理器中，**程序计数器 (PC)** 保存有下一次要取的指令的地址。

(2) 执行指令：取到的指令被放置在处理器的**指令寄存器 (IR)** 中。指令中包含确定处理器将要采取动作的位，处理器解释指令并执行要求的动作。

【习题演练】

1、以下不属于分时系统基本特征的是 ()

A:多路性 B:独立性 C:原子性 D:交互性

答案：C

解析：分时系统的特点是：多路性、独立性、及时性和交互性。

2、操作系统的异步性是指 ()

A:程序的运行结果不确定

B: 程序的运行次序不确定

C:程序多次运行的时间不确定

D:程序的运行结果，运行次序以及多次运行的时间都不确定

答案：C

解析 操作系统的异步性是指进程以不可预知的速度向前推进。内存中的每个程序何时执行、何时暂停、以怎样的速度向前推进，以及每道程序总共需要多少时间才能完成等，都是不可预知的。主要强调的是时间的不确定性。所以选 C。

第二章 进程管理

第一节 进程的描述

2.1.1 一、程序的并发执行☆

程序的顺序执行	先进入内存的程序先执行，在一个程序执行完毕之前，不能执行其他程序。特点：(1) 顺序性；(2) 封闭性；(3) 可再现性。
程序的并发执行	指在同一时间间隔内运行多个程序。特点：(1) 间断性；(2) 失去封闭性；(3) 不可再现性。

2.1.2 二、进程的概念

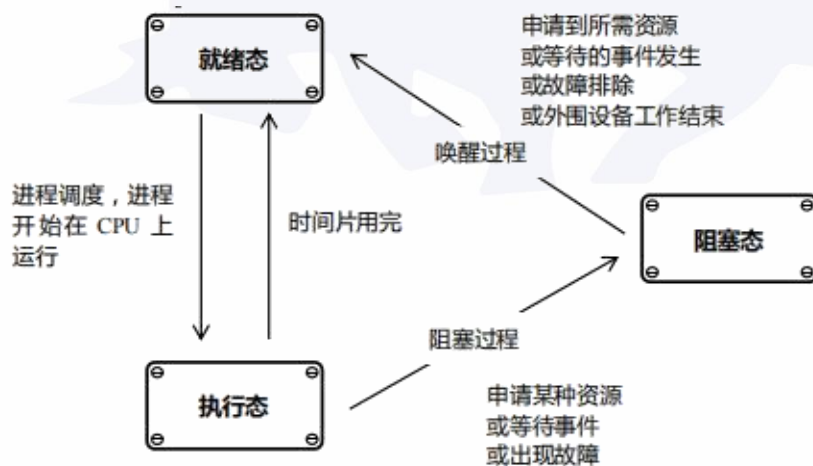
进程定义	定义 1：进程是允许并发执行程序在某个数据集合上的运行过程。 定义 2：进程是由正文段、用户数据段及进程控制块共同组成的执行环境。
进程特征☆☆	(1) 并发性 (2) 动态性 (3) 独立性 (4) 异步性 (5) 结构特征。
进程和程序的比较	区别：(1) 进程是动态的，程序是静态的；(2) 进程是暂时的，程序是永久的。 联系：(1) 进程总是对应至少一个特定的程序。(2) 一个程序可以对应多个进程。

2.1.3 三、进程控制块☆

应用程序对应的进程由**程序**、**用户数据**和操作系统管理进程所需要的**进程控制块**构成。

定义	进程实体存在的标志是操作系统管理进程所使用的数据结构。是操作系统感知进程存在的唯一标志。
构成	<p>进程标识符信息：进程标识符用于唯一标识一个进程。</p> <p>处理机状态信息：进程控制块中保留的处理机状态信息通常包括通用寄存器、指令计数器、程序状态字 PSW 和用户栈指针。</p> <p>进程调度信息：进程状态信息、进程优先级和进程调度所需的其他信息。</p> <p>进程控制信息：程序和数据的地址、进程同步和通信机制、资源清单，以及链接指针。</p>

2.1.4 四、进程的状态☆☆



2.1.5 五、进程的组织

链接方式	具有相同状态的进程的进程控制块用其中的链接字链接成一个队列。
索引方式	根据所有进程的状态，建立几张索引表，索引表的每一个表项指向一个 PCB 的物理块。
进程队列	把具有相同状态的进程放在同一个队列中，具有不同状态的进程就形成了不同的进程队列。

第二节 进程的控制☆

	条件	过程
进程的创建	(1) 用户登录；(2) 作业调度；(3) 提供服务；(4) 应用请求。	(1) 申请空白 PCB；(2) 为新进程分配资源；(3) 初始化进程控制块；(4) 将新进程插入就绪队列。
进程的阻塞	(1) 请求系统服务；(2) 启动某种操作；(3) 新数据尚未到达；(4) 无新工作可做。	(1) 将进程的状态改为阻塞态；(2) 将进程插入相应的阻塞队列；(3) 转进程调度程序，从就绪进程中选择进程为其分配 CPU。
进程的唤醒		(1) 将进程从阻塞队列中移出；(2) 将进程状态由阻塞态改为就绪态；(3) 将进程插入就绪队列。
进程的终止	(1) 当进程正常执行完毕，调用终止进程的系统调用，请求操作系统删除该进程；(2) 一个进程调用适当的系统调用，终止另外一个进程。	(1) 从进程 PCB 中读进程状态；(2) 若进程正在执行，则终止进程的执行；(3) 若进程有子孙进程，在大多数情况下需要终止子孙进程。(4) 释放资源。(5) 将终止进程的 PCB 移出。

第三节 操作系统内核

定义	是计算机硬件的第一次扩充，内核执行操作系统与硬件关系密切，执行频率高的模块，常驻内存。
功能	(1) 支撑功能：中断处理、时钟管理和原语操作。 (2) 资源管理功能：进程管理、存储器管理和设备管理。

2.3.1 一、中断

定义	是改变处理器执行指令顺序的一种事件。
过程☆	(1) 系统关闭中断，保护断点 (2) 转中断处理程序 (3) 执行中断处理子例程 (4) 恢复现场，开中断。

2.3.2 二、时钟管理

重要性	时钟是计算机系统的脉搏。
计算机系统时钟	实时时钟 RTC→BIOS→OS 时钟 →应用程序
操作系统的时钟机制	OS 时钟管理硬件（可编程间隔定时器 PIT）：主要由 3 部分构成：晶振、计数器和保持寄存器。 时钟软件——时钟驱动程序，也称为时钟中断处理程序。

2.3.3 三、系统调用

系统调用	是一群预先定义好的模块，它们提供一条管道让应用程序或一般用户能由此得到核心程序的服务。系统调用是系统程序与用户程序之间的接口
用户态执行	用户空间是指用户进程所处的地址空间，一个用户进程不能访问其他进程的用户空间，只有系统程序才能访问其他用户空间。当 cpu 执行用户空间的代码时，称该进程在用户态执行；

系统态执行	系统空间是指含有一切系统核心代码的地址空间,当 CPU 执行系统核心代码时,称进程处于系统态执行。
系统调用与一般函数的区别☆☆☆☆	系统调用与一般函数的区别:(1)系统调用运行在系统态(核心态),而一般函数运行在用户态;(2)系统调用与一般函数调用的执行过程不同;(3)系统调用要进行“中断处理”,比一般函数调用多了一些系统开销。

第四节 进程同步☆☆

2.4.1、2.4.2

概念	进程同步有两个任务:(1)对具有资源共享关系的进程,保证诸进程以互斥的方式访问临界资源。(2)对具有相互合作关系的进程,保证相互合作的诸进程协调执行。 临界资源是必须以互斥方式访问的共享资源。 临界区是进程中访问临界资源的那段代码。
准则	(1)空闲让进(2)忙则等待(3)有限等待(4)让权等待

2.4.3 三、信号量机制☆☆☆☆

整型信号量机制	wait(s)	{while s≤0 do no-op; s=s-1;}
	signal(s)	{s=s+1; }
	实现:将临界资源 CS 放入 wait(s) 和 signal(s)之间	
记录型信号量机制☆☆☆☆	wait(s)	var s: semaphore; begin s.value = s.value - 1; if s.value < 0 then block(s.L); end
	signal(s)	var s: semaphore; begin s.value = s.value + 1; if s.value ≤ 0 then wakeup(s.L); end.
	实现:将临界资源 CS 放入 wait(s) 和 signal(s)之间	
AND 型信号量机制	Swait	if s1 ≥ and ... sn ≥ 1 then for i = 1 to n do si = si - 1 end for
	Ssignal	For i = 1 to n do Si = Si + 1;
	基本思想是将进程在整个运行过程中所需要的所有资源一次性地全部分配给进程,待该进程使用完后在一起释放。 调用 Swait 操作申请资源,调用 Ssignal 操作释放资源。	

2.4.4 四、经典的进程同步问题☆☆☆☆☆（用记录型信号量机制）

生产者—消费者问题	生产者进程生产消息，并将消息提供给消费者进程消费。在生产者进程和消费者进程之间设置了一个具有 n 个缓冲区的缓冲池，生产者进程可以将它所生产的消息放入缓冲池的一个缓冲区中，消费者进程可以从一个缓冲区中取得一个消息消费。
读者—写者问题	D 是多个进程共享的数据区，允许多个进程同时读 D 区，仅允许一个进程写 D 区，且有进程写 D 区时，不能有任何其他进程读或写 D 区。

2.4.5 五、管程☆

定义	是描述共享资源的数据结构和在数据结构上的共享资源管理程序的集合。其中包括：变量的定义、变量的初始化代码，以及管理共享资源的过程
应用	需建立的管程名为 PC ，其中包括了两个过程： $enter(item)$ 过程，生产者进程调用该过程向缓冲池中投放消息；另一个是 $remove(item)$ 过程，消费者进程调用该过程从公共缓冲池中取消息。

第五节 进程通信

进程间通信的基本方式	描述
共享存储器系统	相互通信的进程共享某些数据结构或共享存储区。分为基于共享数据结构的通信方式；基于共享存储区的通信方式。
消息传递系统	进程间通过操作系统提供的一组通信程序传递格式化的消息。分为直接通信方式和间接通信方式。
管道通信	发送进程通过管道（连接读写进程的一个特殊文件）把数据以字符流的形式发送给接收进程。
消息缓冲队列	通过消息缓冲区（数据结构）、发送原语和接收原语通信。消息缓冲队列机制广泛用于本地进程之间的通信。

第六节 线程

定义	在支持线程的操作系统中，线程是进程的一个实体，线程是被系统独立调度和分派的基本单位；进程是资源分配的基本单位。
分类	(1) 用户级线程 (2) 内核级线程
线程与进程的关系☆☆☆☆	<p>(1) 资源和调度。线程是程序执行的基本单位，进程是拥有资源的基本单位。</p> <p>(2) 地址空间资源。不同进程的地址空间是相互独立的，而同一进程中的各线程共享同一地址空间。</p> <p>(3) 通信关系。进程之间的通信必须使用操作系统提供的进程间通信机制，而同一进程中的各线程间可以通过直接读或写全局变量来进行通信。</p> <p>(4) 并发性。多进程之间可以并发执行，多线程之间也可以并发执行，而且同一进程中的多个线程之间也可并发执行。</p> <p>(5) 系统开销。线程开销很小。且同一进程中的线程的上下文切换要更快。</p>

【习题演练】

1、下列不属于进程特性的是 ()

A:多样性 B:动态性 C:并发性 D:异步性

答案：A

解析：进程的特征：并发性、动态性、独立性、异步性和结构特征。

2、进程从执行状态进入就绪状态的原因可能是 ()

A:被选中占有处理机 B:等待某一事件

C:等待的事件已发生 D:时间片用完

答案：D

解析：如果进程在 CPU 上运行的时间片递减为 0，系统将该进程的状态由执行态变为就绪态。

第三章 进程调度与死锁

第一节 进程调度的功能与时机

功能	按照某种策略和算法从就绪态进程（在 Linux 中是可执行进程）中为当前空闲的 CPU 选择在其上运行的新进程。
时机	当一个进程运行结束（包括正常结束和异常结束）、进程阻塞、中断返回、在支持抢占式调度的系统中有比当前运行进程优先级更高的进程到来、当前运行进程的时间片用完等。

第二节 进程调度算法

准 则 ☆☆	(1) 周转时间短	<p>周转时间：指从作业被提交给系统开始，到作业完成为止的这段时间间隔。它包括 4 部分时间：作业在外存后备队列上等待调度的时间，进程在就绪队列上等待进程调度的时间，进程在 CPU 上执行的时间（服务时间 T_s），以及进程等待 I/O 操作完成的时间。</p> <p>平均周转时间：如果系统中有 n 个作业，系统的平均周转时间 T 等于 n 个作业的周转时间之和除以 n，其公式为：$T = \frac{1}{n} \left[\sum_{i=1}^n T_i \right]$</p> <p>带权周转时间：$W$ 是作业的周转时间 T 与系统为它提供的服务时间 T_s 之比，即 $W=T/T_s$。n 个作业的平均带权周转时间表达式为：$W = \frac{1}{n} \left[\sum_{i=1}^n \frac{T_i}{T_s} \right]$</p>
	(2) 响应时间快 (3) 截止时间的保证 (4) 系统吞吐量高 (5) 处理机利用率好	
算 法 ☆☆ ☆☆ ☆	先来先服务调度算法 (FCFS)	<p>在进程调度中，FCFS 就是从就绪队列的队首选择最先到达就绪队列的进程，为该进程分配 CPU。</p> <p>优缺点：FCFS 适合长进程，不利于短进程。有利于 CPU 繁忙型进程，不利于 I/O 繁忙型进程。</p>

短进程优先调度算法 (SPF)	从就绪队列中选择估计运行时间最短的进程，将处理机分配给它，使它立即执行并一直执行完成，或发生某事件而被阻塞放弃处理机时，再重新调度。 优点：能有效降低进程的平均等待时间，提高系统的吞吐量。 缺点：对长进程不利；不能保证紧迫进程的及时处理；不一定能真正做到短进程优先。	
优先权调度算法	系统将 CPU 分配给就绪队列中优先权值最高的进程。	非抢占式优先权调度算法： 有高优先权进程到来，系统也不能剥夺当前进程的 CPU 使用权，高优先权进程只能先进入就绪队列。
		抢占式优先权调度算法： 如果新到达进程的优先权高于当前正在运行进程的优先权，那么系统会抢占 CPU，把它分配给新到达的高优先权进程，而正在执行的低优先权进程暂停执行。
	问题：低优先权进程无穷等待问题，可通过老化技术解决。	
时间片轮转调度算法 (RR)	在现代分时系统中广泛使用。 系统将所有的就绪进程按先来先服务的原则，排成一个队列，每次调度时把 CPU 分配给队首进程，并令其执行一个时间片。当时间片用完时，调度程序终止当前进程的执行，并将它送到就绪队列的队尾。 时间片大小的确定： 系统响应时间为 T ，进程数目为 N ，时间片为 q ，有 $T = Nq$ (1) 时间片的大小与响应时间成正比；(2) 时间片的大小就与系统允许的最大进程数成反比。(3) 系统的处理能力。	
多级队列调度	将就绪队列分成多个独立队列，根据进程的某些属性，如需要占用的内存大小等，进程会被永久地分配到一个队列。	
多级反馈队列调度	在采用多级反馈队列调度的系统中建立多个优先权不同的就绪队列，为每个队列赋予大小不同的时间片。	

第三节 实时系统中的调度

3.3.1 一、实现实时调度的基本条件☆☆☆☆

提供必要的调度信息	就绪时间、开始截止时间和完成截止时间、处理时间、资源要求、优先级。
系统处理能力强☆☆☆ ☆	<p>(1) 假定系统中有 m 个周期性的硬实时进程，它们的处理时间可表示为 C_i，周期时间表示为 P_i，则在单处理机情况下，必须满足如下限制条件：</p> $\sum_{i=1}^m \frac{C_i}{P_i} \leq 1 (1 \leq i \leq m)$ <p>(2) 采用多处理机系统可以提高实时系统的处理能力。若系统的处理机个数</p>

	为 n , 处理能力的限制条件如下 : $\sum_{i=1}^m \frac{C_i}{P_i} \leq n (1 \leq i \leq m)$
采用抢占式调度机制	抢占式调度算法根据抢占 CPU 的时机不同 , 可以分为基于时钟中断的抢占和立即抢占。
具有快速切换机制☆☆	应具有的能力 (1) 对外部中断的快速响应能力 : 要求系统具有快速的硬件中断机构 , 还应使禁止中断的时间间隔尽可能短。 (2) 快速的进程切换能力 : 应使系统中的每个运行功能单位适当地小 , 以减少进程切换的时间开销。

3.3.2 二、常用的几种实时调度算法☆☆

最早截止时间优先算法	根据进程的开始截止时间确定进程的优先级。
最低松弛度优先算法	松弛度用来表示一个实时进程的紧迫程度。松弛度越小 , 进程的优先级越高 , 越优先获得处理机。 如果一个进程的完成截止时间为 T , 当前时间为 T_c , 处理完该任务还需要的时间为 T_s , 则松弛度 L 的计算式表示为 : $L = T - T_c - T_s$

第五节 多处理器调度

3.5.1、3.5.2 一、多处理器系统 (MPS) 的类型、进程分配方式

	类型	进程分配方式
根据处理器的耦合程度	紧密耦合 的多处理器系统 : 共享主存储器系统和 I/O 设备。	
	松弛耦合 的多处理器系统 : 每台计算机都有自己的存储器和 I/O 设备 , 每一台计算机都能独立工作。	
根据处理器结构是否相同	对称 多处理器系统 : 属于同构的多处理器系统 , 各处理单元在功能和结构上都是相同的。	静态分配
	非对称 多处理器系统 : 有多种类型的处理单元 , 它们的功能和结构各不相同。其中只有一个主处理器 , 有多个从处理器。	动态分配 主—从式

3.5.3 三、进程 (线程) 调度方式

	描述	优缺点
自调度	最常用的调度方式之一 , 也是最简单的一种调度方式。 在系统中设置有一个公共的就绪队列 , 任何一个空闲的处理器都可以自行从该就绪队列中选取一个进程或者一个线程运行。	优点是易移植和有利于提高 CPU 的利用率。 缺点是瓶颈问题、低效性和线程切换频繁。

成组调度	由系统将一组相互合作的进程或线程同时分配到一组处理器上运行,进程或线程与处理器——对应。	优点是减少线程切换和减少调度开销。时间分配有两种方式:面向所有的应用程序平均分配处理器时间和面向所有的线程平均分配处理器时间。
专用处理器分配	在一个应用程序执行期间,专门为该应用程序分配一组处理器,每个线程一个,这组处理器供该应用程序专用,直至应用程序完成。	优点是加速了应用程序的运行速度和避免了进程切换。缺点:会造成处理器资源的严重浪费。

第六节 死锁

3.6.1 一、产生死锁的原因和必要条件☆

定义	由于多个进程竞争共享资源而引起的进程不能向前推进的僵死状态称为死锁
原因	竞争共享资源且分配资源的顺序不当。
产生死锁的必要条件	(1) 互斥条件 (2) 请求和保持条件 (3) 不剥夺条件 (4) 环路等待条件

3.6.2 二、处理死锁的基本方法、3.6.3 三、银行家算法、3.6.4 四、死锁的检测和解除

		具体方法
死锁的预防 ☆☆	摒弃请求和保持条件	①所有进程执行前要一次性地申请在整个运行过程中所需要的全部资源。 ②对某些进程在申请其他资源前要求该进程必须释放已经分配给它的所有其他资源。
	摒弃不剥夺条件	一个已保持了某些资源的进程,当它再提出新的资源要求而不能立即得到满足时,必须释放它已经保持的所有资源。
	摒弃环路等待条件	进程必须按规定的顺序申请资源。
死锁的避免 ☆☆☆☆☆	当系统能找到一个进程执行序列,使系统只要按此序列为每个进程分配资源,就可以保证进程的资源分配和执行顺利完成,不会发生死锁时,称系统处于 安全状态 。若系统不存在这样的安全序列,则称系统处于 不安全状态 。 系统处于安全状态,死锁就不会发生。系统进入不安全状态,便可能进入死锁状态。 设系统有一类数量为 M 的独占性资源,系统中 N 个进程竞争该类资源,每个进程对资源的最大需求为 W。当 M、N、W 满足 $N(W-1) + 1 \leq M$ 时,系统处于安全状态。	银行家算法 过程:①进行资源试分配; ②对试分配后系统的状态做安全性检测。

死锁的检测 ☆☆	死锁定理为：S 为死锁状态的充分条件是当且仅当 S 状态的资源分配图是不可能完全简化的。
死锁的解除 ☆☆	进程终止：终止所有进程或一次只终止一个处于死锁的进程，直到死锁解除。 资源抢占：逐步从进程中抢占资源给其他进程使用，直到死锁环被打破为止。

【习题演练】

1、以下对短进程优先调度算法的说法中，正确的是（ ）

- A:相比 FCFS 而言，对短进程不利
B:相比 FCFS 而言，长进程可能会长时间得不到调度
C:相比 FCFS 而言，紧迫进程更能得到及时处理
D:相比 FCFS 而言，降低了系统的吞吐量

答案：B

解析：与 FCFS 算法相比，短进程优先的算法能有效降低进程的平均等待时间，提高系统的吞吐量。算法的缺陷是对长进程不利，如果系统中不断有短进程到来，长进程可能长时间得不到调度。该算法不能保证紧迫进程的及时处理，因为该算法不考虑进程的紧迫程度。

2、设系统有一类数量为 M 的独占性资源，系统中 5 个进程竞争该类资源，每个进程对该类资源的最大需求为 3。为确保系统不会发生死锁，M 至少应该等于_____。

答案：11

解析：设系统有一类数量为 M 的独占性资源，系统中 N 个进程竞争该类资源，每个进程对资源的最大需求为 W。当 M、N、W 满足 $N(W-1) + 1 \leq M$ 时，系统处于安全状态，不会发生死锁。或者这样想，每个进程分别分配 2 个资源，即 $5 \times 2 = 10$ 个，再给其中一个进程分配 1 个资源，该进程即可执行，执行完毕后释放 3 个资源，其他进程可顺利执行，故 M 至少应该等于 11。

第四章 内存管理

第一节 存储器的层次结构

层次结构	<p>更小, 更快, (每字节) 更贵的存储设备</p> <p>更大, 更慢, (每字节) 更便宜的存储设备</p> <p>L0: 寄存器</p> <p>L1: 芯片内的 L1 高速缓存 (SRAM)</p> <p>L2: 芯片外的 L2 高速缓存 (SRAM)</p> <p>L3: 主存储器 (DRAM)</p> <p>L4: 本地二级存储 (本地磁盘)</p> <p>L5: 远程二级存储 (分布式文件系统, Web 服务器)</p> <p>CPU 寄存器保存取自高速缓存存储器的字</p> <p>L1 高速缓存保存取自 L2 高速缓存的高速缓存行</p> <p>L2 高速缓存保存取自存储器的缓存行</p> <p>主存储器保存取自本地磁盘的磁盘块</p> <p>本地磁盘保存取自远程网络服务器上的磁盘的文件</p>
局部性原理	<p>(1) 时间局部性：如果程序中的某条指令一旦执行，则不久后该指令可能再次执行。</p> <p>(2) 空间局部性：一旦程序访问了某个单元，在不久之后，其附近的存储单元也将被访问。</p>

第二节 程序的链接和装入☆☆

程序的链接	静态链接	静态链接是在程序运行前，用链接程序将目标模块链接成一个完整的装入模块。静态链接程序的任务一是对逻辑地址进行修改，二是变换外部调用符号。优点：运行速度较快；缺点：内存开销大，程序开发不灵活。
	动态链接	可将某些目标模块的链接推迟到这些模块中的函数被调用执行时才进行。优点：节省内存和外存空间，方便了程序开发。
程序的装入	绝对装入方式	按照装入模块的物理地址将程序和数据装入内存。
	可重定位装入方式(静态重定位)	在程序装入时对目标程序中的指令和数据地址的修改过程称为重定位。
	动态运行时装入(动态重定位)	一个进程在被换出之前所在的内存位置与后来被从外存重新调入内存时所在的内存位置不同，在这种情况下，地址映射必须延迟到进程执行时再进行，把这种装入方式称为动态运行装入。

第三节 连续分配存储管理方式

单一连续分配	内存中只有一个用户区，任意时刻内存中只能装入一道程序，这种分配方式仅适用于单用户、单任务的系统。	
固定分区分配	将内存用户区划分成若干个固定大小的区域，每个区域中驻留一道程序。固定分区分配的用户分区大小固定，分区数目固定。 结构体包括 4 个字段：分区编号、分区大小、分区起始地址和分区状态。	
动态分区分配☆☆	原理	系统动态地对内存进行划分，根据进程需要的空间大小分配内存。内存中分区的大小和数量是变化的。动态分区方式比固定分区方式显著地提高了内存利用率。
☆☆	数据结构	(1) 空闲分区表：每个表项中包括 分区编号、分区大小和分区起始地址。 (2) 空闲分区链：每个结点包括分区大小、分区起始地址、指向前一个空闲分区结点的指针，以及指向后一个空闲分区结点的指针。
	算法	(1) 首次适应算法 FF (2) 循环首次适应算法 NF (3) 最佳适应算法 BF
	流程	(1) 内存分配流程：检索空闲分区链；如果 $m.size - u.size \leq size$ ，则直接把该空闲分区分配给进程 将分配给进程的分区起始地址返回给内存分配程序的调用者；修改空闲分区链表。 (2) 内存回收流程：释放一块连续的内存区域；如果被释放区域与其他空闲区相邻，则合并空闲区；修改空闲分区链。 合并空闲区 ：①仅回收区的前面有相邻的空闲分区②仅回收区的后面有相邻的空闲分区③回收区的前、后都有相邻的空闲分区，把回收区与空闲分区 R1、R2 合并成一个空闲分区。

第四节 基本分页存储管理方式

4.4.1 一、分页存储管理的基本原理

概念☆☆	页	将一个进程的逻辑地址空间分成若干个大小相等的片
	页框	将物理内存空间分成与页大小相同的若干个存储块，或称页帧
	分页存储	在为进程分配内存时，以页框为单位将进程中的若干页分别装入多个可以不相邻接的页框中。
	页内碎片	进程的最后一页一般装不满一个页框，而形成了不可利用的碎片，称为“页内碎片”，是一种内部碎片。
	页表	页表是系统为进程建立的数据结构，页表的作用是实现从页号到页框号的映射。
地址结构☆☆	逻辑地址结构包含两部分：页号 P (12~31 位) 和页内偏移量 W (0~11 位)。若 A 为逻辑地址，L 为页大小，P 为页号，W 为页内偏移量，则有以下计算关系： $P = \text{INT} (A/L)$ ； $W = \text{MOD} (A/L)$	
分页地址变换☆☆☆☆	基本任务是实现逻辑地址到物理地址的变换。 (1) 进程执行，PCB 中页表起始地址和页表长度送 CPU 的页表寄存器。 (2) CPU 访问逻辑单元 A。 (3) 由分页地址变换硬件自动将 A 分为页号和页内偏移两部分。 (4) 由硬件检索页表，得到 A 所在的页对应的页框号。页号对应的页表项起始地址=页表起始地址+页表项长度*页号(页表项中存有页框号)。从该地址指示的内存单元中读取页框号。 (5) 页框号和页内偏移地址送物理地址寄存器，计算物理地址。物理地址=页框大小*页框号+页内偏移量。	

4.4.2 二、快表

定义	也称转换后援缓冲 (TLB)，是为了提高 CPU 访存速度而采用的专用缓存，用来存放最近被访问过的页表项。
地址变换	(1) CPU 产生分页的逻辑地址页号和页内偏移后，将该逻辑地址的页号提交给 TLB。 (2) 查找 TLB，如果找到页号，则把该页所在的页框号用于形成物理地址。否则 (TLB 失效) 查找内存页表，从内存页表中找到相应的页表项，读取页所在的页框号，以形成物理地址。 (3) 如果所查找的页表项不在 TLB 中，在访问完内存页表后，要把找到的页表项中的页号和页框号写到 TLB 中。如果 TLB 中的条目已满，系统会根据某种策略 (如最近最少使用替换) 选择一个 TLB 中的条目，用刚访问的页表项信息替换选中的这个 TLB 条目。
性能☆☆	在 TLB 中找到某一个页号对应的页表项的百分比称为 TLB 命中率。 有 TLB：有效访存时间=一次访问 TLB 的时间加上一次访问内存的时间。 没有 TLB：访存时间=一次访问 TLB 的时间加上两次访问内存 (一次访问内存页表，一次访问内存读写数据或指令) 的时间。

4.4.3 三、两级和多级页表

两 级 页 表 ☆☆	<p>将页表再进行分页，使每个页表分页的大小与内存页框的大小相同，并为它 们编号。页目录表的表项中存放了每一个页表在物理内存中所在的页框号。</p> <div><div>页目录号</div><div>页号</div><div>页内偏移地址</div><div>p1</div><div>p2</div><div>d</div></div> <p>(1) 逻辑地址结构：</p> <p>(2) 地址映射的过程：页表所在的页框号在内存中的地址=页 H 录起始地址+PI * 页表项长度。进程页所在的页框号在内存中的地址=页表的起始地址+p2 × 页表项长度。页表的起始地址=页表所在的页框块号*页框大小。A 的物理地址=进程页所在的页框号*页框大小+页内地址 d。</p>
----------------------	--

第五节 基于分页的虚拟存储系统

4.5.0 虚拟存储系统☆☆☆☆

定义	指具有请求调人功能和置换功能，能从逻辑上对内存容量进行扩充的一种存储器系统。在虚拟存储器系统中，进程无需全部装入，只要装入一部分即可运行。
虚拟存储技术的好处	1) 提高内存利用率。2) 提高多道程序度。3) 把逻辑地址空间和物理地址空间分开，使程序员不用关心物理内存的容量对编程的限制。
特征	(1) 离散性、(2) 多次性、(3) 对换性、(4) 虚拟性。

4.5.1 一、请求分页中的硬件支持☆☆

页表	<p>是支持请求分页系统最重要的数据结构。</p> <div>页表结构：<div>页框号</div><div>状态位 P</div><div>访问字段 A</div><div>修改位 M</div><div>保护位</div></div>
缺 页 异 常机构	<p>(1) 分页硬件通过页表完成逻辑地址到物理地址的映射时，通过检查页表中的状态位 P 判断当前被访问的页是否在内存中。如果不在，则产生缺页异常信号。</p> <p>(2) 执行操作系统的缺页异常处理过程。先在内存中为请求调人的页找一个空闲页框。然后，调度磁盘操作，把需要的页装入找到的空闲页框中。</p> <p>(3) 修改页表，更新已经调入页的存在位、在内存中的页框号、访问位和保护位等字段的值。</p> <p>(4) 重新开始执行因缺页而被中断的指令。</p>
地 址 变 换	<p>(1) 由分页地址变换机构从逻辑地址中分离出页号和页内偏移地址。</p> <p>(2) 以页号为索引查找快表，若快表中有该页的页表项，则读出页框号，计算物理地址。</p> <p>(3) 若快表中无该页信息，转到内存表中查找。若页表中的状态位 P 显示该页已调入内存，则从相应的页表项读出页所在的页框号，并计算物理地址，然后把该页表项写入快表。</p> <p>(4) 若该页尚未调入内存，则产生缺页异常，请求操作系统从外存中把该页调入内存，然后修改页表，重新执行被中断的指令。</p>

4.5.2 二、页分配策略☆☆

页分配策略	固定分配	组合成以下 3 种页分配和置换策略： (1) 定分配局部置换 (2) 可变分配全局置换 (3) 可变分配局部置换
	可变分配	
页置换策略	局部置换	
	全局置换	

4.5.4 四、页置换算法☆☆☆☆☆

最佳置换算法和先进先出置换算法 (FIFO)	最佳置换算法：选择以后永远不会被访问的页或者在未来最长时间内不再被访问的页作为换出页。(理论) 先进先出页置换算法 (FIFO) :最简单的页置换算法。实现这种算法的一种方式是为每个页记录该页调入内存的时间,当选择换出页时,选择进入内存时间最早的页。
最近最久未使用 LRU 置换算法	择最近最久未使用的页换出(最近最久未使用的页在最近的将来被访问的可能性也比较小)。(常用) 实现:为每个在内存中的页配置一个移位寄存器;利用一个特殊的栈来保存当前使用的各个页的页号;为每个页表项增加一个时间字段,并为 CPU 增加一个逻辑时钟或计数器。
其他置换算法	(1)附加引用算法(2)简单 Clock 置换算法(3)改进型 Clock 算法(4)最少使用置换算法(5)页缓冲算法

4.5.5 五、请求分页系统的性能分析

缺页率对有效访问时间的影响	假设 P 为缺页率, ma 为存储器访问时间:有效访问时间=(1-P)*ma + P*缺页异常时间。有效访问时间正比于缺页率。
工作集	为了能有效降低缺页率,从而提高访存的时间效率。工作集就是在某段时间间隔Δ里,进程实际要访问的页的集合。
抖动	定义:多道程序度太高,使运行进程的大部分时间都用于进行页的换入、换出,而几乎不能完成任何有效工作的状态称为抖动。 原因:系统中的进程数量太多,每个进程能分配到的页框太少,以至于进程运行过程中频繁请求调页。 预防:采取局部置换策略;在 CPU 调度程序中引入工作集算法;挂起若干进程。

第六节 分段存储管理

4.6.1 一、分段机制的引入☆☆

在使用分段存储管理的系统中,程序员使用二维的逻辑地址,一个数用来表示段,另一个数用来表示段内偏移。

优点:方便编程、分段共享、分段保护、动态链接,以及存储空间动态增长。

4.6.2 二、分段系统的基本原理☆☆

原理	进程的地址空间被划分成若干个段。段的大小不一样,每个段的逻辑地址从 0 开始,采用一段连续的地址空间。
逻辑地址结构	二维的,由段号和段内地址组成。

构	<div><div>段号</div><div>段内偏移</div></div> <div>3116150</div> <div>图 4-52 32 位系统中分段地址形式</div>
段表	段表是由操作系统维护的用于支持分段存储管理地址映射的数据结构。段表由段表项构成。每个段表项包含段号、段基址（段的起始地址）和段长（即段大小）3 个部分。
地址变换	以段号 s 作索引，从段表中找到段号为 s 的段表项。 从找到的段表项中读出 s 段的基地址和段大小。 如果 $d \leq \text{段大小}$ ，则将段基址与段内偏移 d 相加，得到与逻辑单元 s : d 相应的物理单元地址。
分页和分段的主要区别 ☆☆☆☆	（1）页是按物理单位划分的，段是按逻辑单位划分的。 （2）页的大小是固定的，而段的大小不固定。 （3）分页的地址空间是一维的，分段的地址空间是二维的。

4.6.3 三、段页式存储管理

原理	将用户进程的逻辑空间先划分成若干个段，每个段再划分成 若干个页。
地址变换过程	(1) 以段号 s 作索引，找到段 s 的段表项，得到该段页表的起始地址。 (2) 通过分页机制从段内偏移 d 中分离出页号 P 和页内偏移 W。 (3) 以段内页号 P 作索引，从段 s 的页表中搜索页号 P 对应的页表项。 (4) 从页表项中得到页所在的页框号。 (5) 由页框号与页内偏移 W 得到某逻辑地址对应的物理地址。 物理地址=页框号 x 页框大小+页内偏移 W

第七节 Linux 的伙伴系统

分配连续物理内存块	每个块的第一个页框的物理地址是该块大小的整数倍。
物理框的释放与合并	满足以下条件的两个块称为伙伴。 (1) 两个块具有相同的大小，记作 b。 (2) 它们的物理地址是连续的，起始地址是 2b 的整数倍。

【习题演练】

- 1、在程序装入时对目标程序中的指令和数据地址的修改过程称为（ ）
A:加载 B:链接 C:重定位 D:编译
答案：C
解析：在程序装入时对目标程序中的指令和数据地址的修改过程称为重定位。
- 2、基本分页存储管理方式的逻辑地址结构包括两部分，即页内偏移量和（ ）
A:页号 B:页内地址 C:页框号 D:段号
答案：A
解析：基本分页的逻辑地址结构包含两部分：页号 P 和页内偏移量 W。

第五章 文件系统

第一节 文件

5.1.0 零、文件☆☆

文件系统管理是操作系统的重要功能之一,它为用户提供了在计算机系统中对数据信息进行长期、大量存储和访问的功能。

文件系统的用户接口,包括文件的命名、类型、属性和对文件的操作。

5.1、5.2、5.3、5.4、5.5、5.6 文件☆☆

文件命名	多数操作系统都支持文件名用圆点隔开分为两部分
文件结构	(1) 无结构字节序列:也称为流式文件,操作系统所见到的就是字节。 (2) 固定长度记录序列:具有固定长度的记录。 (3) 树形结构:文件由一棵记录树构成,记录长度不定,在记录的固定位置包含一个关键字域,记录树 按关键字域排序。
文件类型	(1) 正规文件,一般分为 ASCII 文件和二进制文件。ASCII 文件由多行正文组成,在某些系统中每行用回车符结束,某些则用换行符结束,而有些系统还同时采用回车符和换行符,各行的长度不必相同。二进制文件:二进制文件具有一定的内部结构,如可执行的.exe 文件。 (2) 目录文件 (3) 字符设备文件 (4) 块设备文件
文件存取	常用的文件存取方式有两种:顺序存取和随机存取。
文件属性	为方便管理,除了文件名和文件数据外,文件系统还会保存其他与文件相关的信息,如文件的创建日期、文件大小和修改时间等,这些附加信息称为文件属性。
文件操作	CREATE:创建文件。OPEN:打开文件。CLOSE:关闭文件。READ:从文件中读取数据。WRITE:往文件中写数据。APPEND:该操作是 WRITE 调用的限制形式,它只能在文件末尾添加数据。SEEK:对于随机存取文件,要指定从何处开始取数据。GETATTRIBUTES:获取文件属性。SETATTRIBUTES:修改属性。RENAME:修改已有文件的名件名。

第二节 目录☆

层次目录系统	目录文件的结构:属性放在目录项中和放在 i 结点中。 目录类型:(1) 单层目录 (2) 两级目录 (3) 树形目录
路径名	绝对路径名:由从根目录到文件的路径组成。 相对路径名: "." 指当前目录, ".." 指当前目录的父目录。
操作	CREATE:根据给定的目录文件名,创建目录。DELETE:删除目录。 OPENDIR:目录内容可以被读取。CLOSEDIR:关闭目录。REaddir:以标准格式返回打开目录的下一级目录项。RENAME:更换目录名。

第三节 文件系统的实现

5.3.1 一、实现文件☆☆☆☆

连续分配	把每个文件作为一连串连续数据块存储在磁盘上。
------	------------------------

	<p>优点：实现简单、读操作性能好。</p> <p>缺点：随着时间的推移，磁盘会变得零碎。当删除文件时，文件所占的簇被释放，这些空闲的连续簇形成“空洞”。</p>
使用磁盘链接表的分配	<p>为每个文件构造簇的链接表，每个簇开始的几个字节用于存放下一个簇的簇号，簇的其他部分存放数据，每个文件可以存放在不连续的簇中。在目录项中只需存放第一个数据块的磁盘地址，文件的其他块可以根据这个地址来查找。</p> <p>优点：可以充分利用每个簇，不会因为磁盘碎片（除了最后一块中的内部碎片）而浪费存储空间，管理也比较简单。缺点：随机存取相当缓慢。</p>
使用内存的链接表分配	<p>将文件所在的磁盘的簇号存放在内存的表(文件分配表)中。访问文件时，只需从内存文件分配表中顺着某种链接关系查找簇的簇号。不管文件有多大，在目录项中只需记录文件的第一块数据所在簇的簇号，根据它查找到文件的所有块。</p> <p>缺点：必须把整个表都存放在内存中。不适合大容量的磁盘。</p> <p>MS-DOS 就使用这种方法进行磁盘分配。</p>
i-结点	<p>为每个文件赋予一个被称为 i 结点数据结构，其中列出了文件属性和文件块的磁盘地址。</p>

5.3.2 二、实现目录☆☆☆

CP/M 中的目录	<p>字节 8 3 1 2 1 16</p> <p>用户码 磁盘块号</p> <p>CP/M 是以簇而不是以字节为单位来记录文件长度的。</p>
MS-DOS 中的目录	<p>8 3 1 10 2 2 2 4</p> <p>MS-DOS 用文件分配表即 FAT 作为索引表来存放文件数据所在簇的簇号。</p>
UNIX 中的目录	<p>字节 2 14</p> <p>当打开某个文件时，文件系统必须要获得文件名并且定位文件所在的第一个簇。</p>

5.3.3 三、磁盘空间管理☆☆☆

簇大小	一般簇大小是 2 的整数次幂个连续的扇区。				
记录空闲块	<table><tr><td>空闲簇链接表</td><td>用一些空闲簇存放空闲簇的簇号。对于 1 KB 大小的簇，可以存放 256 个 32 位 的簇号(有一个存放指向下一个块的指针)</td></tr><tr><td>位图</td><td>位图方法所需空间少，因为每个簇只用一个二进制位标识，而在链接表方法中，每一个簇号都要用 32 位。</td></tr></table>	空闲簇链接表	用一些空闲簇存放空闲簇的簇号。对于 1 KB 大小的簇，可以存放 256 个 32 位 的簇号(有一个存放指向下一个块的指针)	位图	位图方法所需空间少，因为每个簇只用一个二进制位标识，而在链接表方法中，每一个簇号都要用 32 位。
空闲簇链接表	用一些空闲簇存放空闲簇的簇号。对于 1 KB 大小的簇，可以存放 256 个 32 位 的簇号(有一个存放指向下一个块的指针)				
位图	位图方法所需空间少，因为每个簇只用一个二进制位标识，而在链接表方法中，每一个簇号都要用 32 位。				

【习题演练】

- 1、为方便管理，文件系统会保存一些与文件相关的信息，如文件的创建日期、文件大小和修改时间等细节，这些信息称为（ ）
- A:文件名 B:文件属性 C:文件结构 D:文件数据

答案：B

解析：为方便管理，除了文件名和文件数据外，文件系统还会保存其他与文件相关的信息，如文件的创建日期、文件大小和修改时间等，这些附加信息称为文件属性。

2、以磁盘文件系统为例，文件存储的几种常用方式中，连续分配的缺点是（ ）

A:读操作性能不好

B:随着时间的推移会形成很多“空洞”

C:可以充分利用每个簇

D:打开文件时需要频繁读取硬盘

答案：B

解析：连续分配优点：实现简单、读操作性能好。

连续分配缺点：随着时间的推移，磁盘会变得零碎。当删除文件时，文件所占的簇被释放，这些空闲的连续簇形成“空洞”。

第六章 I/O 设备管理

第一节 I/O 系统的组成☆☆

I/O 系统的结构	微机 I/O 系统	总线型 I/O 系统：CPU 与内存之间可以直接进行信息交换，但是不能与设备直接进行信息交换，必须经过设备控制器。
	主机 I/O 系统	四级结构，包括主机、通道、控制器和设备。一个通道可以控制多个设备控制器，一个设备控制器也可以控制多个设备。
I/O 设备的分类	按传输速率分类	低速设备、中速设备、高速设备。
	按信息交换的单位分类	块设备和字符设备
	按设备的共享属性分类	独占设备、共享设备和虚拟设备
设备控制器	功能	(1)接收和识别命令(2)数据交换(3)设备状态的了解和报告(4)地址识别)(5)数据缓冲(6)差错控制
	逻辑构成	(1)设备控制器与处理机的接口数据线、控制线和地址线。 (2)设备控制器与设备的接口：设备与设备控制器接口中的3类信号为数据、状态和控制信号。 (3)I/O 逻辑：I/O 逻辑主要由指令译码器和地址译码器两部分功能部件构成，将 CPU 的命令和地址分别译码，控制指定设备进行 I/O 操作。
I/O 通道	是大型主机系统中专门用于 I/O 的专用计算机。	

第二节 I/O 控制方式☆☆

轮询	使 CPU 经常处于由于输入/输出而造成的循环测试状态，造成 CPU 的极大浪费，影响整个系统的吞吐量。
中断	能使 CPU 与 I/O 设备在某些时间段上并行工作，提高 CPU 的利用率和系统的吞吐量。

DMA 控制方式	<p>DMA 控制器的逻辑组成包括 3 部分：主机与 DMA 的接口、DMA 与设备的接口，以及 I/O 控制逻辑。</p> <p>DMA 控制器中 4 类寄存器：命令/状态寄存器 CR、内存地址寄存器 MAR、数据寄存器 DR 和数据计数器 DC。</p>
-----------------	---

第三节 缓冲管理☆☆

缓冲的引入	原因：（1）处理数据流的生产者与消费者之间的速度差异。（2）协调传输数据大小不一致的设备。	
单缓冲	<p>最简单的缓冲技术，操作系统在主存储器的系统区中只设立一个缓冲区。</p> <p>（1）对于面向块的设备：输入数据被传送进入系统缓冲区。当传送完成时，进程把该块移到用户空间，并立即请求另一块系统缓冲区。</p> <p>（2）对于面向流的 I/O：在每次传送一行的方式下，或者每次传送一个字节的方式下。如键盘、打印机和传感器等。</p>	
双缓冲	当一个进程往这一个缓冲区中传送数据（或从这个缓冲区中读取数据）时，操作系统正在清空（或填充）另一个缓冲区，这种技术称为双缓冲，或缓冲交换。	
循环缓冲	组成	<p>（1）多个缓冲区：空缓冲区 R、已装满数据的缓冲区 G、现行工作缓冲区 C。</p> <p>（2）多个指针：Nextg、Nexti、Current。</p>
	使用	申请：Getbuf 过程；释放：Releasebuf 过程。
	进程同步	<p>（1）Nexti 指针追上 Nextg 指针，全部缓冲区已满，需要阻塞生产者进程。</p> <p>（2）Nextg 指针追上 Nexti 指针，全部缓冲已空，需要阻塞消费者进程。</p>
公共缓冲池	组成	<p>(1) 3 种类型的缓冲区：空缓冲区、装满输入数据的缓冲区和装满输出数据的缓冲区。</p> <p>(2) 3 种队列：空缓冲队列 emq、输入队列 inq、输出队列 outq。</p> <p>(3) 4 种工作缓冲区：收容输入数据的缓冲区、提取输入数据的缓冲区、收容输出数据的缓冲区、提取输出数据的缓冲区。</p>
	使用	申请：Getbuf 过程；释放：Puthuf 过程。
	工作方式	收容输入、提取输入、收容输出和提取输出。

第四节 设备分配☆☆

设备分配中的数据结构	设备控制表 DCT、控制器控制表 COCT、通道控制表 CHCT、系统设备表 SDT	
设备分配	设备的固有属性	独占设备、共享设备、可虚拟设备
	设备分配算法	先来先服务、基于优先权的分配算法
	设备分配方式	安全分配方式、不安全分配方式

设备独立性	含义	应用程序独立于具体使用的物理设备
	意义	(1) 应用程序与物理设备无关，系统增减或变更外围设备时不需要修改应用程序。 (2) 易于处理输入/输出设备的故障。例如，某台打印机发生故障时，可用另一台不同型号、不同品牌的打印机替换，而不用修改应用程序。 (3) 提高了系统的可靠性，增加了设备分配的灵活性。
独占设备的分配程序	步骤	(1) 分配设备 (2) 分配控制器 (3) 分配通道
SPOOLing 技术	含义	在联机情况下实现的同时外围操作称为 SPOOLing
	组成	(1) 输入井和输出井 (2) 输入缓冲区和输出缓冲区 (3) 输入 SP _i 和输出进程 SP _o (4) 请求 I/O 队列
	特点	(1) 提高了 I/O 速度；(2) 将独占设备改造为共享设备；(3) 实现了虚拟设备功能。

第五节 I/O 软件原理

6.5.0 零、I/O 软件原理☆☆

设备管理软件的层次结构：1) 用户层软件：进行 I/O 调用，格式化 I/O。2) 与设备无关的软件层：命名、保护、阻塞、缓冲、分配。3) 设备驱动程序：包括中断处理程序、设备驱动程序。4) 中断处理程序（底层）：执行 I/O 操作。

6.5.1、6.5.2、6.5.3、6.5.4☆☆

设备管理软件的功能	(1) 实现 I/O 设备的独立性 (2) 错误处理 (3) 异步传输 (4) 缓冲管理 (5) 设备的分配和释放 (6) 实现 I/O 控制方式
中断处理程序	当 I/O 结束时唤醒设备服务子程序
设备驱动程序	建立设备寄存器、检测状态
与硬件无关的 I/O 软件的功能	(1) 设备命名 (2) 设备保护 (3) 提供独立于设备的块大小 (4) 为块设备和字符设备提供必要的缓冲技术 (5) 块设备的存储分配 (6) 分配和释放独立设备 (7) 错误处理。

第六节 磁盘管理

6.6.1 一、磁盘结构☆☆

数据的组织和格式	磁盘密度：每英寸中所存储的位数，显然是内层磁道的密度较外层磁道的密度高。 磁盘上存储的物理记录数目是由扇区数、磁道数及磁盘面数所决定的。 每个扇区包括：标识符字段、数据字段。
类型	(1) 固定头磁盘 (2) 移动头磁盘
访问时间	(1) 寻道时间：指把磁头移动到指定磁道上所经历的时间； (2) 旋转延迟时间：指将指定扇区移动到磁头下面所经历的时间； (3) 传输时间：指把数据从磁盘读出或向磁盘写入数据时所经历的时间。

	访问时间=寻道时间+旋转延迟时间+传输时间。其中，花费时间最长的是寻道时间和旋转延迟时间。
--	---

6.6.2 二、磁盘调度☆☆☆☆

先来先服务 (FCFS)	最简单的磁盘调度算法。根据进程请求访问磁盘的先后顺序进行调度。 优点：公平、简单
最短寻道时间优先 (SSTF)	其要求访问的磁道与当前磁头所在的磁道距离最近，以使每次的寻道时间最短。
扫描 (SCAN) 算法	解决“饥饿”现象。 不仅考虑到要访问的磁道与当前磁道的距离，更优先考虑磁头当前的移动方向。又称电梯调度算法。
循环扫描 (CSCAN) 算法	规定磁头是单向移动
NStepSCAN 算法	解决“磁臂粘着”现象。 将磁盘请求队列分成若干个长度为 N 的子队列，磁盘调度将按 FCFS 算法依次处理这些子队列。每处理一个队列时又是按 SCAN 算法，对一个队列处理完后，再处理其他队列。
FSCAN 调度算法	NStepSCAN 算法的简化。只将磁盘请求队列分成两个子队列。

6.6.3 三、提高磁盘 I/O 速度的方法

(1) 提前读 (2) 延迟写 (3) 优化物理块的分布 (4) 虚拟盘 (5) 磁盘高速缓存。

【习题演练】

1、在磁盘调度算法中，规定磁头是单向移动的算法是 ()

- A:先来先服务 B:最短寻道时间优先
C:扫描 (SCAN) 算法 D:循环扫描 (CSCAN) 算法

答案：D

解析：目前常用的磁盘调度算法有：

- (1) 先来先服务：根据进程请求访问磁盘的先后顺序进行调度。
(2) 最短寻道时间优先：要求访问的磁道与当前磁头所在的磁道距离最近，以使每次的寻道时间最短。
(3) 扫描 (SCAN) 算法：为防止进程出现“饥饿”现象。不仅考虑要访问的磁道与当前磁道的距离，更优先考虑磁头当前的移动方向的算法。
(4) 循环扫描 (CSCAN) 算法：规定磁头是单向移动。
(5) NStepSCAN 和 FSCAN 调度算法：解决“磁臂粘着”现象。FSCAN 是 NStepSCAN 的简化，即 FSCAN 只将磁盘请求队列分成两个子队列。

2、对磁盘的访问时间中，将指定扇区移动到磁头下面所经历的时间是指 ()

- A:寻道时间 B:启动磁臂的时间
C:传输时间 D:旋转延迟时间

答案：D

解析：对磁盘的访问时间分成以下 3 部分：

（1）寻道时间：指把磁头移动到指定磁道上所经历的时间；（2）旋转延迟时间：指将指定扇区移动到磁头下面所经历的时间；（3）传输时间：指把数据从磁盘读出或向磁盘写入数据时所经历的时间。

