

操作系统

▼ 操作系统

▼ 第1章 绪论

- 操作系统的基本概念
- 多道程序设计技术和分时技术
- 操作系统的基本类型

▼ 第2章 操作系统的组织架构

- 操作系统虚拟机
- 处理机的状态
- 中断机制

▼ 第4章 并发处理

- 进程概念
- 进程的相互制约关系
- 同步机构
- 生产者-消费者问题
- 进程通信(IPC)

▼ 第5章 资源分配与调度

- 资源分配策略
- 死锁

▼ 第6章 处理机调度

▼ 作业调度

- 作业调度算法
- 调度算法性能的衡量

▼ 第7章 主存管理

- 7.2 主存管理的功能

▼ 分区存储管理

- 放置策略
- 7.4 淘汰策略
- 页式存储管理
- 段式存储管理
- 段页式存储管理

▼ 第8章 输入/输出管理

▼ 输入输出管理概念

- 共享分配
- 虚拟分配

- 输入/输出控制方式

▼ 第9章 文件系统

- 文件的物理结构
- UNIX文件系统的主要结构
- 位示图
- FAT表
- 磁盘调度策略
- 打开文件管理机构

第1章 绪论

操作系统的基本概念

- 计算机的操作系统是一种**系统软件**

操作系统负责计算机系统的软硬件资源的分配和管理; 控制和协调并发活动; 提供用户接口, 使用户获得良好的工作环境

- 操作系统的主要功能
 - 处理机分配
 - 存储管理
 - 设备管理
 - 软件资源(文件)管理
- 操作系统的特性
 - 并发: 指处理多个同时性活动的的能力
 - 共享: 指多个计算任务对资源的同时共享
 - 不确定性: 能处理随机发生的多个事件(这些事件随时都有发生的可能)

多道程序设计技术和分时技术

- 多道程序设计技术

多道程序设计技术是在计算机主存中同时存放几道相互独立的程序, 使它们在管理程序的控制之下, 相互穿插地运行

当某道程序因某种原因不能继续运行下去时(如等待外部数据传输数据时), 管理程序便将另一道程序投入运行

多道程序设计技术可以实现同时被接受进入计算机的若干道程序相互穿插地运行

- 多道程序运行的特征:
 - 多道
 - 宏观上并行

- 微观上串行
- 分时技术
 - 是把处理机时间划分成很短的时间片轮流地分配给各个联机作业使用,如果某个作业在分配的时间片用完之前还未完成计算,该作业就暂时中断,等待下一轮继续计算
- 实时系统
 - 实时系统的响应时间是根据被控对象的要求决定的,一般要求秒级、毫秒级、微秒级甚至更快的响应时间

操作系统的基本类型

- 批量操作系统
 - 特点: 批量的作业
 - 优点: 系统资源的利用率比较高、系统的吞吐率高
 - 缺点: 用户不能与机器直接对话、对用户的响应时间(提交作业到获得系统的处理信息这一段时间)较长
- 分时操作系统
 - 特点: 并行性、独占性(时间片轮转)、交互性
- 实时操作系统
 - 实时操作系统对外部输入的信息,能够在规定的时间内(截至期限deadline)处理完毕并作出反应

第2章 操作系统的组织架构

操作系统虚拟机

在裸机上配置了操作系统程序后就构成了操作系统虚拟机, 操作系统的核心在裸机上运行,而用户程序则在扩充后的机器上运行.

扩充后的虚拟机不仅可以使⽤原来裸机提供的各种基本硬件指令,而且还可以使⽤操作系统中所增加的许多其他"指令"

处理机的状态

- 核态
 - 有的系统将管理程序执行时的机器状态进一步分为核态和管态, 这时, 管态的权限有所变化, 管态只允许使⽤一些在用户态下所不能使⽤的资源, 但不能使⽤修改机器的状态指令。而核态(kernel mode) 就具有上述管态所具有的所有权
 - 核态>管态>目态
- 管态

操作系统的管理程序执行时机器所处的状态,此状态下允许处理机使用全部资源和指令

- 目态

用户程序执行时机器所处的状态,此状态禁止使用特权指令

- 如何区分管态和目态

设置一个系统状态位(模式位),就可以区分当前执行的是系统程序还是用户程序

- 管态目态相互转换

用户程序执行时,若要请求操作系统服务,则要通过一种受控方式进入操作系统,将用户态转为管态,由操作系统得到控制块,在核态下执行其相应的服务例程,服务完毕后,返回到用户态

中断机制

- 中断的概念

中断是指某个事件(电源掉电,定点加法溢出或I/O传输结束等)发生时,系统中止现行程序的运行,引出处理该事件程序进行处理,处理完毕后返回断点,继续执行

- 用户程序发出磁盘I/O请求后,系统的正确处理流程是

用户程序 -> 系统调用处理程序 -> 设备驱动程序 -> 中断处理程序

第4章 并发处理

进程概念

- 进程是指一个具有一定独立功能的程序关于某个数据集合的一次运行活动

每一个进程都有生命期,即从创建到消亡的时间周期。

当操作系统为一个程序构造一个进程控制块并分配地址空间之后,就创建了一个进程。

用户可以任意的取消用户的作业,随着作业运行的正常或不正常结束,进程也被撤销了。

- 进程与程序的联系与区别

(1) 程序是指令的有序集合,其本身没有任何运动的含义,它是一个静态概念;而进程是程序在处
理机上的一次执行过程,它是一动态概念

(2) 进程是一个能独立运行的单位,能与其它进程并行地活动

(3) 进程是竞争计算机系统有限资源的基本单位,也是进行处理机调度的基本单位

- 进程类型

- 系统进程: 管态下工作/分配资源

- 用户进程: 目态下工作/分配资源

- 就绪状态(ready)

当进程获得了除CPU之外所有的资源,它已经准备就绪,一旦得到了CPU控制权,就可以立即运行,该进程所处的状态为就绪状态

- 运行状态(run)

当进程调度/分派模块分派后,得到中央处理机控制权,它的程序正在运行,该进程所处的状态为运行状态

- 等待状态(wait)

若一进程正在等待某一事件发生(如等待输入/输出操作的完成)而暂停执行,这时即便给它CPU控制权它也无法执行,则称该进程处于等待状态,又称为阻塞状态

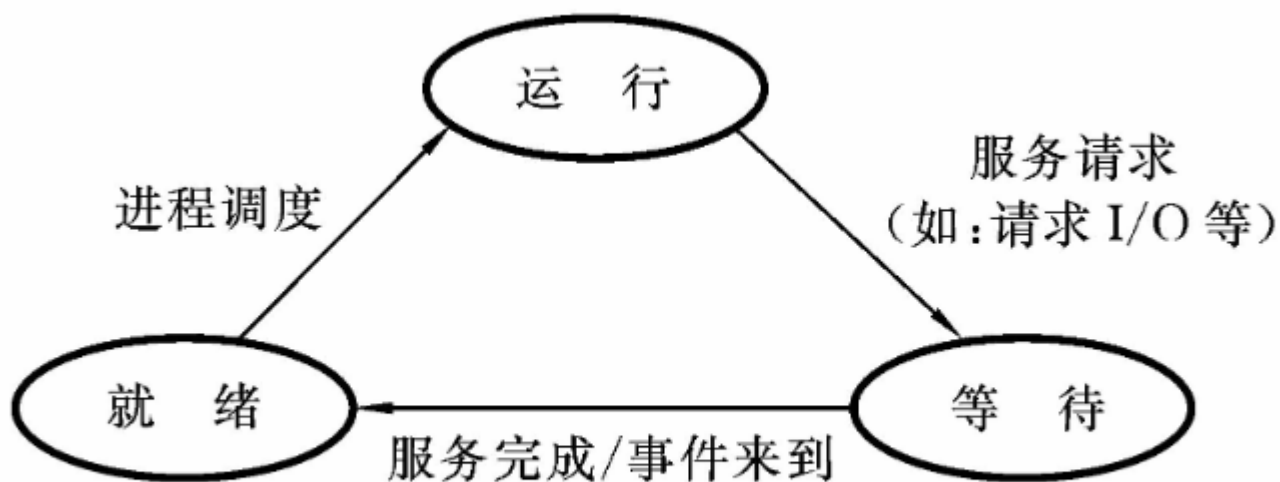


图4.8 具有进程基本状态的变迁图

tip: 时间片到: 运行->就绪

- 线程的概念

- (1) 线程是进程中的一条执行路径
- (2) 它有自己私用的堆栈和处理机执行环境(尤其是处理器寄存器)
- (3) 它共享分配给父进程的主存
- (4) 它是单个进程所创建的许多个同时存在的线程中的一个

- 进程与线程的联系和区别

(不支持多线程时)进程是任务调度的单位,**进程是系统资源的分配单位**;
当系统支持多线程处理时,**线程是任务调度的单位**,但不是系统资源的分配单位;

- 同一进程的多线程,同时多核上都能运行,这种性质称作线程的**并行性**
- 程序访问的局部性原理使得**高速缓存**成为可能

进程的相互制约关系

- 临界资源: 一次仅允许一个进程使用的资源
- 临界区

是进程中对公共变量(或存储区)进行审查与修改的**程序段**,称为相对于该公共变量的临界区

- 进程互斥

在操作系统中,当某一进程正在访问某一存储区域时,就不允许其它进程来读出或者修改该存储区的内容; 否则就会出现无法估计的错误,通常将进程之间的这种相互制约关系称为互斥

- 进程同步

两个进程并发执行,一个进程等待另一个进程发来的消息,或者建立某个条件后才向前推进,这种制约被称为进程的同步

- 多个进程实体能存在于同一内存中, 在**一段时间内**都得到运行, 这种性质称作进程的**并发性**

同步机构

- 在大多数同步机构中,必须用一个标志来代表某种资源的状态,比如标志为0表示未被使用,则表示已被使用,这一标志经常被称为锁或信号灯

- 信号灯

信号灯是一个确定的二元组(s, q), s是一个具有非负初值的整型变量, q是一个初始状态为空的队列. 其可能的取值范围是负整数/零/正整数(信号灯数值仅能由PV操作改变)

- 整型信号量不能避免忙等

P、V操作是操作系统中进程低级通信原语

- P操作(进程对资源的申请)

- S值减1
- 若相减结果大于或等于0,则进程继续执行
- 若相减结果小于0,则进程被封锁,并将它插入到该信号灯的等待队列中,然后转入进程调度程序

- V操作(进程对资源的释放)

- S值加1
- 若相加结果大于零,则进程继续执行
- 若相加结果小于或等于零,则从该信号灯的等待队列中移出一个进程,解除它的等待状态,然后返回本进程继续执行

- $S > 0$: 有S个资源, 请求资源的进程可以继续执行
- $S = 0$: 无资源, 请求进程必须等待
- $S < 0$: 有|S|个进程正在等待

生产者-消费者问题

1. 有一个有界的多缓冲区
2. 多个生产者, 生产产品后, 放入缓冲区
3. 多个消费者, 取出产品后, 进行消费

- 两个同步信号灯, 一个互斥信号灯

```
int full = 0; // 满缓冲区的数量
int empty = n; // 空缓冲区的数量
int mutex = 1; // 互斥信号灯(锁)
```

- 生产者

```
空缓存区 - 1
p(互斥信号灯)
放入缓冲区
v(互斥信号灯)
满缓冲区 + 1
```

- 消费者

```
满缓冲区 - 1
p(互斥信号灯)
取产品
v(互斥信号灯)
空缓冲区 + 1
消费产品
```

进程通信(IPC)

- 进程通信的概念

进程通信(inter process communication, IPC)是一个进程与另一个进程间共享消息的一种方式。消息(message)是发送进程形成的一个信息块, 通过信息的语法表示传送内容到接收进程。IPC是利用消息, 明确地从一个进程的地址空间拷贝到另一个进程的地址空间, 而不使用共享存储器的一种通信机制。

第5章 资源分配与调度

资源分配策略

- 目的: 吞吐率尽可能地高, 响应时间尽可能地短

- 先请求先服务(FIFO)
 - 优点: 次序不变,系统开销小
 - 缺点: 对短作业不利(响应时间长)
- 优先调度
 - 优点: 比较灵活,可以优先照顾需要尽快处理的作业或进程,以及它们的各种请求
 - 缺点: 复杂,系统开销大

死锁

- 死锁的概念
 - 在两个或多个并发进程中,如果每个进程持有某种资源而又等待着别的进程释放它或它们现在保持着的资源,在未改变这种状态之前都不能向前推进,称这一组进程发生了死锁
- 死锁产生的根本原因是: 系统能够提供的资源个数比要求该资源的进程数要少
- 产生死锁的必要条件
 - 互斥条件
 - 涉及的资源是非共享的,即一次只有一个进程使用
 - 不剥夺条件(非抢占)
 - 资源在未使用完毕之前,不能被其它进程强行夺走
 - 占有并等待(部分分配)
 - 在等待一新资源的同时,进程继续占有已分配到的资源
 - 环路条件(循环等待)
 - 存在一种循环链,链中每一个进程已获得的资源同时被链中下一个进程所请求
- 解决死锁问题的策略
 - 采用资源静态分配方法(预先分配所有共享的资源)**预防**死锁
 - 采用资源动态分配、有控分配方法(有序资源分配算法、银行家算法)来**避免**死锁
 - 当死锁发生时检测出死锁,并设法修复
 - 忽略死锁,认为死锁不会发生,为大多数操作系统所采用(如UNIX)
- 在设备分配中,可能会发生死锁的分配方式是**多请求方式**

第6章 处理机调度

作业调度

作业调度算法

- 先来先服务调度算法
 - 优点: 实现容易
 - 缺点: 效率较低,对短作业不利
- 短作业优先调度算法
 - 优点: 实现容易且效率较高
 - 缺点: 只照顾短作业的利益,没有考虑长作业的利益
- 响应比高者优先调度算法
 - 响应比 = 响应时间 + 执行时间
 - 响应时间 = 等待时间 + 执行时间
 - 优点: 既照顾了短作业,又不使长作业等待时间过长
 - 缺点: 调度性能一般
- 优先调度算法
 - 优先数 = 等待时间² - 要求执行时间 - 16 x 输出量(I/O事件)
- 可剥夺调度方式
 - "重要而紧迫"的进程一到,便暂停正在执行的进程,立即把处理机分配给它,这种方式称为可剥夺调度方式

调度算法性能的衡量

- 周转时间
 - 周转时间是指从作业进入系统开始,到作业完成为止所经历的时间,包括作业在外存后备队列上等待作业调度的时间、进程在就绪队列上等待进程调度的时间、进程在cpu上执行的时间和进程等待I/O操作完成的时间。
- 平均周转时间
 - 平均周转时间的计算方法是: 将所有作业的周转时间相加,然后除以作业的数量。
- 带权平均周转时间
 - 带权周转时间: 周转时间/执行时间
 - 计算所有作业的带权周转时间求平均值

第7章 主存管理

7.2 主存管理的功能

- 虚拟存储器

- 最大容量: $2^{\text{CPU寻址范围}}$
- 实际容量: $\min(\text{内存}+\text{外存}, 2^{\text{CPU寻址范围}})$

虚拟存储器 (virtual memory) 将用户的逻辑主存与物理主存分开, 这是现代计算机对虚存的实质性的描述。更为一般的描述是: 计算机系统在处理应用程序时, 只装入部分程序代码和数据就启动其运行, 由操作系统和硬件相配合完成主存和外围联机存储器之间的信息的动态调度, 这样的计算机系统好像为用户提供了一个其存储容量比实际主存大得多的存储器, 这个存储器称为虚拟存储

- 地址映射

物理地址是计算机主存单元的真实地址, 又称为绝对地址或实地址

逻辑地址是用户的程序地址(指令地址或操作数地址), 对于逻辑地址来说, 在主存中并没有一个固定的, 真实的物理单元与之对应, 它是一个虚地址或称为相对地址

逻辑地址与被共享的物理地址之间有一定的映射关系

- 程序运行时, 必须将逻辑地址正确地转换为物理地址, 此即为地址映射(地址重定位)
- 地址重定位的结果是得到**可执行程序**

做题:

先将逻辑地址分解成页号P和页内地址W两部分(虚地址 \bmod 页面大小, 商为页号P, 余数为页内地址W)

根据页号P查页面变换表, 可得页号P对应的物理块号B

物理地址 = 物理块号B \times 页面大小 + 页内地址W

- 编程或编译时确定地址映射关系
- 静态地址映射

在装载程序时确定映射关系(装入主存), 也称静态重定位

- 动态地址映射

在程序执行过程中实现的

动态地址映射是指在程序执行期间, 随着每条指令和数据的访问自动地, 连续地进行地址映射.

这种重定位的实现需要硬件提供手段, 且一般是靠硬件地址变换机构实现的, 比如重定位寄存器.

- 在存储器管理中, 页面是信息的**物理**单位, 分段是信息的**逻辑**单位

页面大小由**系统/硬件**确定, 分段大小由**用户程序**确定

- 存储管理的功能

- 映射逻辑地址到物理主存地址
- 在多用户之间分配物理主存
- 对各用户区的信息提供保护措施
- 扩充逻辑主存区

- 覆盖技术

覆盖技术的实现是把程序划分为若干个功能上相对独立的程序段, 按照其自身的逻辑结构使那些不会同时运行的程序段共享同一块内存区域。程序段先保存在磁盘上, 当有关程序的前一部分执行结束后, 把后续程序段调入内存, 覆盖前面的程序段。

- 交换技术

在分时系统中, 用户的进程比内存能容纳的数量更多, 系统将那些不再运行的进程或某一部分调出内存, 暂时放在外存上的一个后备存储区, 通常称为交换区, 当需要运行这些进程时, 再将它们装入内存

- 在存储管理中, 采用覆盖与交换技术的目的是**节省主存空间**

分区存储管理

- 分区存储管理

分区存储管理是满足多道程序设计的最简单的一种存储管理方法。它允许多个用户作业共享主存空间, 这些作业在主存内是以划分分区而共存的。早期的分区存储管理技术只有固定式分区方法, 后来才发展形成动态分区方法

- 地址空间是一维线性的
- 可变分区存储管理, 即动态分区分配, 会产生外碎片。
- 固定分区存储分配, 会产生内碎片。
- 单道程序设计不提供地址保护

放置策略

- 首次适应算法(First Fit)

空闲区按其位置(地址由低到高)的顺序链在一起, 即每个后继空闲区的起始地址总比前者要大。当要分配一个分区时, 总是从低地址空闲区开始查寻, 直到找到第一个足以满足该作业要求的空闲区为止, 并把该空闲区分为两个区, 一个为已分配区, 其大小等于所要求的大小; 另一个仍为空闲区, 并留在原来的链接位置上

- 首次适应算法要求对空闲区表项**地址从小到大**排列
- 最佳适应算法(Best Fit)

空闲区队列按空闲区大小递增(由小到大)的顺序链在一起, 空闲区队列指针free总是指向最小的一个空闲区

在进行分配时总是从最小的一个空闲区开始查寻,因此找到的第一个能满足要求的空闲区便是最佳的一个

- 最佳适应算法要求对空闲区表项**尺寸从小到大**排列
- 最坏适应算法(Worst Fit)

最坏适应算法就是把一个作业程序放入主存中最不适合它的空闲区,即最大的空闲区内在大空闲区中放入作业后,剩下的空闲区常常也很大,于是也能装下一个较大的新作业

7.4 淘汰策略

- 置换算法

当要索取一页面并送入主存时,必须将该作业已在主存中的某一页面淘汰掉. 用来选择淘汰哪一页的规则就叫做置换算法

- 颠簸(抖动)

刚被淘汰出去的页,不久之后又要访问它,因而又要把它调入,而调入后不久又再次淘汰,如此反复,使得整个系统的页面置换非常频繁,以致大部分的机器时间花费在来回进行页面的调度上,只有一小部分时间用于程序的实际执行,从而影响整个系统的效率.

导致系统效率急剧下降的主存和辅存之间的频繁页面置换现象称为颠簸(抖动)

- 在任何时刻,若访问的页已在主存,则称此次访问成功; 若访问的页不在主存,则称此次访问失败,并产生缺页中断
- 缺页中断

当CPU访问的页面不在物理内存时, 便会产生一个缺页中断, 请求操作系统将所缺页调入到物理内存

- 支持虚拟存储器的硬件是**缺页中断**

- 最佳算法(OPT)

最佳算法(最佳淘汰策略)是当要调入一新页而必须淘汰一旧页时,所淘汰的那一页应是以后不再使用的,或是在最长时间段之后才会用到的页

- 先进先出算法(FIFO)

总是选择在主存中居留时间最长(即进入最早)的一页淘汰

- 最久未使用淘汰算法(LRU)

选择最长时间未被使用的那一页淘汰

页式存储管理

- 页式存储管理的思想

程序等分为页, 内存等分为块; 块长等于页长, 块为内存分配的最小单位, 将程序的一页放入到内存的某块, 用页表登记页与块的关系, 以实现地址映射

- 地址空间是一维线性的
- 有效访问时间

从进程出发指定逻辑地址的访问请求, 经过地址变换, 到在内存中找到对应的实际物理地址单元并取出数据, 所需要花费的总时间, 称之为内存的有效访问时间。

在基本分页存储管理中, 有效访问时间分为第一次访问内存时间(即查找页表对应的页表项所耗费的时间 t)与第二次访问内存时间(即页表项中的物理块号与页内地址拼接成实际物理地址所耗费的时间 t)之和, 计算公式为

$$EAT = 2t + \lambda - t * a$$

λ 表示查找快表所需要的时间, a 表示命中率, t 表示访问一次内存所需的时间

段式存储管理

- 段式系统

作业由若干个逻辑分段组成, 如可由代码分段、数据分段、栈段组成。

- 地址空间是二维的
- 程序地址的一般形式由 (s, w) 组成, 这里 s 是段号, w 是段内位移
- 在分段管理中, 以段为单位分配, 每段是一个连续存储区

段页式存储管理

- 段页式存储管理的思想:

段页式存储管理的用户地址空间是二维的,按段划分的.在段中再划分成若干大小相等的页,这样地址结构就由段号,段内页号和页内位移三部分组成

用户使用的仍是段号和段内相对地址,由地址变换机构自动地将段内相对地址的高几位解释为段内页号,将剩余的低位解释为业内位移

用户地址空间的最小单位不是段而是页,而主存按页的大小划分,按页装入, 这样,一个段可以装入到若干个不连续的主存块内,段的大小不再受主存可用区的限制了

- 地址空间是二维的
- 在段页式存储系统中, 一个作业对应一个段表, 多个页表

第8章 输入/输出管理

输入输出管理概念

- 设备独立性

设备独立性是指用户在编制程序时所使用的设备与实际使用的设备无关,也就是在用户程序中仅使用逻辑设备名. 逻辑设备名是用户自己指定的设备名(或设备号),它是暂时的,可更改的.物理设备名是系统提供设备的标准名称(永久的,不可更改的)

- 设备独立性的类型

- 一个程序应该独立于分配给它的某种类型的具体设备
- 一个程序应该尽可能与它使用的I/O设备类型无关

共享分配

- 共享设备

外部设备中如磁盘等直接存取设备都能进行快速的直接存取,它们往往不是让一个作业独占而是被多个作业、多进程共同使用,这类设备是共享设备

对共享设备采用共享分配方式,即进行动态分配:当进程提出资源申请时,由设备管理模块进行分配,进程使用完毕后,立即归还

共享设备必须是可寻址的和随机访问的设备

虚拟分配

- Spool(假脱机系统)

为了克服独占设备的缺点,操作系统提供外部设备联机同时操作的功能,又称假脱机系统

Spool系统利用通道和中断技术,在主机控制之下,由通道完成输入/输出工作,该系统包括预输入程序、缓输出程序、井管理程序和预输入表、缓输出表等数据结构。它在联机方式下实现了输入收存和输出发送的功能,使外部设备和主机能并行操作,所以称为假脱机系统。

输入/输出控制方式

- 通道方式

I/O通道是用来控制**外设与主存**之间进行成批数据传输的部件

它一旦被启动,就能独立于CPU运行,这样就可使CPU和通道并行操作,而且CPU和各种外部设备也能并行操作

- (判断题)使用异步方式, I/O写调用可能需要阻塞直到数据写到设备上 **✗**

第9章 文件系统

- 文件系统在创建一个文件时,为它建立一个**文件控制块(FCB)**

文件的物理结构

- 连续文件

连续文件结构是由一组分配在磁盘连续区域的物理块组成的

- 优点: 连续存取时速度较快
- 缺点: 长度一经固定便不易改变,因此不利于文件的增生和扩充

- 串联文件

串联文件采用的是一种非连续的存储结构,文件的逻辑记录可以存放到不连续的物理块中,能较好地利用辅存空间

- 优点: 串联文件结构比较易于修改
- 缺点: 只适用于顺序存取方式,不适用于直接存取方式

- 索引文件

将逻辑文件顺序地划分为长度与物理存储块长度相同的逻辑块,然后为每个文件分别建立逻辑块号与物理块号的对照表

- 优点: 可以直接读写任意记录,而且便于文件的增删
- 缺点: 每次存取都涉及索引表的查找,所以采用的查找策略对文件系统的效率有很大影响

- 索引表的组织

- 一级间接索引

若磁盘块的大小为512B,用于登记磁盘块号的表项占用2B,这样一个磁盘块可以登记256个表项

- 二级间接索引

文件目录项中有一组表项,其内容登记的是第二级索引表块的块号. 第二级索引表块中的索引表项登记的是第一级索引表的块号, 第一级索引表项中登记的是文件逻辑记录所在的磁盘块号

- 二级索引结构文件大小的计算:

$$n \times 256^2 \times 512B$$

n为文件目录项中用于索引的表项数目

逻辑记录大小为512B

用于登记磁盘块号的表项占用2B

- 逻辑文件存放到存储介质上时, 采用的组织形式是与**存储介质**有关的

UNIX文件系统的主要结构

- 在UNIX中, 通常把设备作为**特殊**文件来处理

书上原话: 将设备和文件一样看待, 这是UNIX系统采用的方法。使用文件系统的系统调用进行设备的读、写操作等。设备作为特殊文件也有相应的文件目录表项 (在UNIX系统中称为索引节点), 根据文件类型 (设备是特殊文件) 可以查找该文件的索引节点, 从而进入该类设备的驱动程序

- 目录项

UNIX系统采用的办法是将目录项中除了名字以外的信息全部移到另一个数据块上, 而系统中所有这类数据块都放入磁盘中约定的位置, 在物理上连续存放, 并顺序编号, 这种数据块就是索引节点(index node), 简称为i节点

位示图

2. (本题 5 分) 某操作系统的数据文件共有 1233k 字节, 磁盘存放为 4k 字节为 1 块, 若用字长为 32 位的位示图管理盘空间, 请回答 (要求给出计算过程):

(1) 位示图需要多少个字? (3 分) $\lceil 1233 / 4 \rceil = 309$ $\lceil 309 / 32 \rceil = 10$

(2) 第 i 字第 j 位对应的块号是多少? (假设 i、j 和块号均从 0 开始) (2 分)

块号: $(32 * i) + j$

FAT表

2. (本题 7 分) 采用显示链接的方式进行磁盘空间管理, 假定磁盘块的大小的 1KB, 对于 540MB 的磁盘, 其文件分配表 FAT 需要占用多少存储空间? (说明: 假定盘块号所占存储空间以字节为单位。例如, 当盘块号占 12 个二进制位时, 它实际应占用 16 位)

FAT表
 $1B = 8bit$
向上取整
答: 磁盘 540MB, 磁盘块大小为 1KB, 则磁盘共有 $540MB / KB = 540K$ 个
因为 $512 < 540 < 1024$, 所以盘块号需要 2^{20} , 即 20 位二进制表示, 占 2.5B。
向上取整, 实际占 3B
所以 FAT 表占用存储空间为 $540K \times 3B = 1620KB$ 。

磁盘调度策略

- 电梯调度算法
 - SCAN 算法
电梯在一个方向上, 按照楼层从低到高或者从高到低的顺序进行服务, 直到到达最顶层或者最底层后, 改变方向继续服务。
 - 最短寻道时间优先算法(SSTF)
每次选择距离当前磁头所在磁道最近的磁道进行访问, 以最小化寻道时间。
- 对磁盘进行移臂调度的目的是为了缩短**寻道时间**
- 等待当前磁道上的某指定扇区旋转到磁头下所需的时间称为**延迟时间**

打开文件管理机构

- 文件描述符表
一个进程的用户描述符表的前三项一般作为固定使用。
 - 0#打开文件称为标准输入文件
 - 1#打开文件称为标准输出文件
 - 2#打开文件称为标准错误文件