# 磁盘与文件

## 磁盘调度

磁盘访问延迟 = 队列时间 + 控制器时间 + 寻道时间 + 旋转时间 + 传输时间

磁盘调度的目的是减小延迟，其中前两项可以忽略，寻道时间是主要矛盾。

### 磁盘调度算法

* FCFS：先进先出的调度策略，这个策略具有公平的优点，因为每个请求都会得到处理，并且是按照接收到的顺序进行处理。
* SSTF(Shortest-seek-time First 最短寻道时间优先)：选择使磁头从当前位置开始移动最少的磁盘I/O请求，所以 SSTF 总是选择导致最小寻道时间的请求。总是选择最小寻找时间并不能保证平均寻找时间最小，但是能提供比 FCFS 算法更好的性能，会存在饥饿现象（会导致较远的I/O请求不能满足）。
* SCAN：SSTF+中途不回折，每个请求都有处理机会。SCAN 要求磁头仅仅沿一个方向移动，并在途中满足所有未完成的请求，直到它到达这个方向上的最后一个磁道，或者在这个方向上没有其他请求为止。由于磁头移动规律与电梯运行相似，SCAN 也被称为电梯算法。

SCAN 算法对最近扫描过的区域不公平，因此，它在访问局部性方面不如 FCFS 算法和 SSTF 算法好。

* C-SCAN：SCAN+直接移到另一端，两端请求都能很快处理。把扫描限定在一个方向，当访问到某个方向的最后一个磁道时，磁道返回磁盘相反方向磁道的末端，并再次开始扫描。其中“C”是Circular（环）的意思。
* LOOK(C-LOOK)：釆用SCAN算法和C-SCAN算法时磁头总是严格地遵循从盘面的一端到另一端，显然，在实际使用时还可以改进，**即磁头移动只需要到达最远端的一个请求即可返回，不需要到达磁盘端点**。这种形式的SCAN算法和C-SCAN算法称为LOOK和C-LOOK调度。这是因为它们在朝一个给定方向移动前会查看是否有请求。

## 文件系统

### 分区表

* MBR：支持最大卷为2 TB（Terabytes）并且每个磁盘最多有4个主分区（或3个主分区，1个扩展分区和无限制的逻辑驱动器）
* GPT：支持最大卷为18EB（Exabytes）并且每磁盘的分区数没有上限，只受到操作系统限制（由于分区表本身需要占用一定空间，最初规划硬盘分区时，留给分区表的空间决定了最多可以有多少个分区，IA-64版Windows限制最多有128个分区，这也是EFI标准规定的分区表的最小尺寸。另外，GPT分区磁盘有备份分区表来提高分区数据结构的完整性。

### RAID 技术

磁盘阵列（Redundant Arrays of Independent Disks，RAID），独立冗余磁盘阵列之。原理是利用数组方式来作磁盘组，配合数据分散排列的设计，提升数据的安全性。

### 常见文件系统

* Windows: FAT, FAT16, FAT32, NTFS
* Linux: ext2/3/4, btrfs, ZFS
* Mac OS X: HFS+

### Linux文件权限

Linux文件采用10个标志位来表示文件权限，如下所示：

-rw-r--r-- 1 skyline staff 20B 1 27 10:34 1.txt  
drwxr-xr-x 5 skyline staff 170B 12 23 19:01 ABTableViewCell

第一个字符一般用来区分文件和目录，其中：

* d：表示是一个目录，事实上在ext2fs中，目录是一个特殊的文件。
* －：表示这是一个普通的文件。
* l: 表示这是一个符号链接文件，实际上它指向另一个文件。
* b、c：分别表示区块设备和其他的外围设备，是特殊类型的文件。
* s、p：这些文件关系到系统的数据结构和管道，通常很少见到。

第2～10个字符当中的每3个为一组，左边三个字符表示所有者权限，中间3个字符表示与所有者同一组的用户的权限，右边3个字符是其他用户的权限。

这三个一组共9个字符，代表的意义如下：

* r(Read，读取)：对文件而言，具有读取文件内容的权限；对目录来说，具有浏览目录的权限
* w(Write,写入)：对文件而言，具有新增、修改文件内容的权限；对目录来说，具有删除、移动目录内文件的权限。
* x(eXecute，执行)：对文件而言，具有执行文件的权限；对目录来说该用户具有进入目录的权限。

权限的掩码可以使用十进制数字表示：

* 如果可读，权限是二进制的100，十进制是4；
* 如果可写，权限是二进制的010，十进制是2；
* 如果可运行，权限是二进制的001，十进制是1；

#### chmod命令

chmod命令非常重要，用于改变文件或目录的访问权限。用户用它控制文件或目录的访问权限。

该命令有两种用法。一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。

1. 文字设定法

chmod ［who］ ［+ | - | =］ ［mode］ 文件名

命令中各选项的含义为：

操作对象who可是下述字母中的任一个或者它们的组合：

* u 表示“用户（user）”，即文件或目录的所有者。
* g 表示“同组（group）用户”，即与文件属主有相同组ID的所有用户。
* o 表示“其他（others）用户”。
* a 表示“所有（all）用户”。它是系统默认值。

操作符号可以是：

* + 添加某个权限。
  + 取消某个权限。
* = 赋予给定权限并取消其他所有权限（如果有的话）。

设置mode所表示的权限可用下述字母的任意组合：

* r 可读。
* w 可写。
* x 可执行。
* X 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加x 属性。
* s 在文件执行时把进程的属主或组ID置为该文件的文件属主。方式“u＋s”设置文件的用户ID位，“g＋s”设置组ID位。
* t 保存程序的文本到交换设备上。
* u 与文件属主拥有一样的权限。
* g 与和文件属主同组的用户拥有一样的权限。
* o 与其他用户拥有一样的权限。

文件名：以空格分开的要改变权限的文件列表，支持通配符。

在一个命令行中可给出多个权限方式，其间用逗号隔开。例如：chmod g+r，o+r example 使同组和其他用户对文件example 有读权限。

1. 数字设定法

直接使用数字表示的权限来更改：

例： $ chmod 644 mm.txt

#### chgrp命令

功能：改变文件或目录所属的组。

语法：chgrp ［选项］ group filename

例：$ chgrp - R book /opt/local /book

改变/opt/local /book/及其子目录下的所有文件的属组为book。

#### chown命令

功能：更改某个文件或目录的属主和属组。这个命令也很常用。例如root用户把自己的一个文件拷贝给用户xu，为了让用户xu能够存取这个文件，root用户应该把这个文件的属主设为xu，否则，用户xu无法存取这个文件。

语法：chown ［选项］ 用户或组 文件

说明：chown将指定文件的拥有者改为指定的用户或组。用户可以是用户名或用户ID。组可以是组名或组ID。文件是以空格分开的要改变权限的文件列表，支持通配符。

例：把文件shiyan.c的所有者改为wang。  
  
 chown wang shiyan.c