

磁盘与目录

磁盘

磁盘上的盘片被细分为多个大小相同的扇区，扇区是磁盘空间的基本单位，一般来说，一个扇区的大小为512字节。
磁盘中第一个扇区非常重要，其中存储了与磁盘正常使用相关的重要信息，分别为：主引导记录、磁盘分区表和魔数。

主引导记录（Master Boot Record）占用446字节，其中包含一段被称为引导加载流程（Boot Loader）的程序，计算机启动后，会到磁盘0扇面的0扇区去读取MBR中的内容，只有MBR中的程序正确无误，计算机才能正常开机。

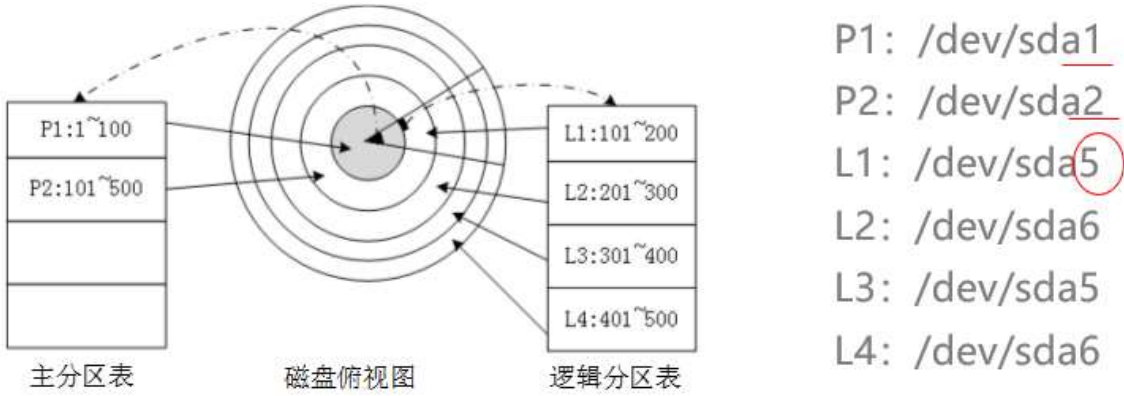
磁盘分区表（partition table）占用64个字节，其中记录整块磁盘的分区状态，每个分区的信息需要16个字节，因此磁盘分区表若只记录分区信息，便最多只能存储4个分区的分区信息。

魔数（Magic Number）占用2个字节，用来标识MBR是否有效。

Linux系统中遵循“一切皆文件”的思想，Linux下的设备也会被视为文件，硬盘作为设备的一种，其对应的文件被存储于系统的/dev目录下。若磁盘为SATA类型，则磁盘路径名为/dev/sda，如图所示的四个分区在/dev目录下对应的文件名如下所示：

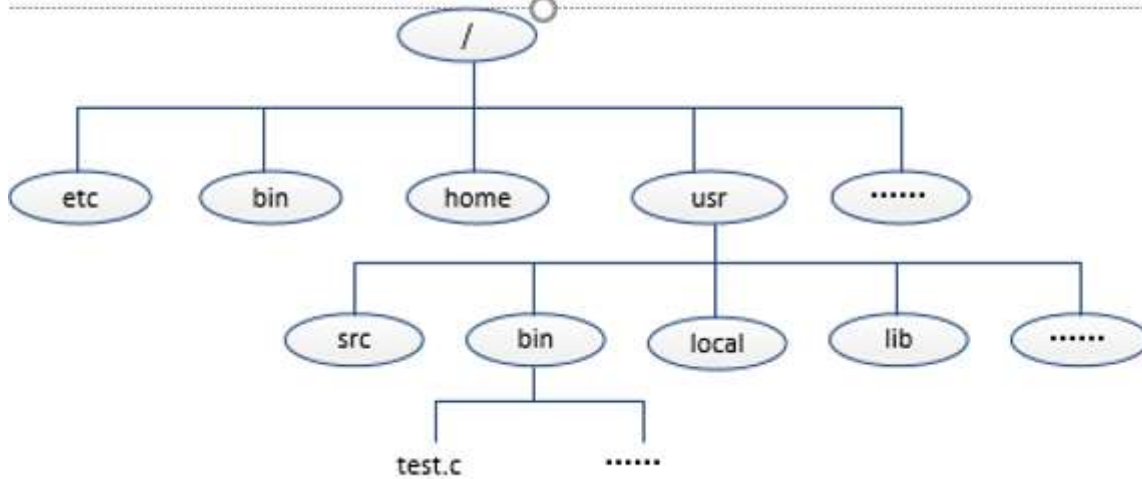
- P1: /dev/sda1
- P2: /dev/sda2
- P3: /dev/sda3
- P4: /dev/sda4

事实上，磁盘除可以划分出主分区外，还可以划分出一个扩展分区，而扩展分区可以继续划分。由扩展分区划分出的分区称为逻辑分区，而逻辑分区的分区信息会被存储在扩展分区中的第一个扇区。



目录

目录结构是磁盘等存储设备上文件的组织形式，主要体现在对文件和目录的组织方式上。Linux操作系统中只有一个树状结构，根目录“/”存在于所有目录和文件的路径中，是唯一的根结点。Linux文件系统目录结构如下图所示。



/: 根目录，只包含目录，不包含具体文件；

/bin: 存放可执行的文件，如常用命令ls、mkdir、rm等，都以二进制文件的形式存放在该目录中；

/dev: 存放设备文件，包括块设备文件（如磁盘对应文件）、字符设备文件（如键盘对应文件）等；

/root: 超级用户，即管理员的工作目录；

/home: 普通用户的工作目录，每个用户都有一个/home目录；

/opt: 存放附加的应用程序软件包；

/tmp: 存放临时文件，重启系统后该目录的文件不会被保存；每个用户都能创建该目录，但不能删除其它用户的/tmp目录；

/swap: 存放虚拟内存交换时所用文件；

/usr: 包含所有的用户程序（/usr/bin）、库文件（/usr/lib）、文档（/usr/share/doc）等，是占用空间最大的目录。

/lib: 主要存放动态链接共享库文件，类似于Windows中的.dll文件，该目录中的文件一般以.a、.dll、.so结尾（后缀不代表文件类型）；也会存放与内核模块相关的文件；

/boot: 存放操作系统启动时需要用到的文件，如内核文件、引导程序文件等；

/etc: 主要包含系统管理文件和配置文件；

/mnt: 存储挂载存储设备的挂载目录；

/proc: 存放系统内存的映射，可直接通过访问该目录来获取系统信息；

inode与dentry

Linux系统中将文件的属性与数据分开存储，文件中的数据存放的区域称为数据区，文件属性又被称为元数据，存放文件属性的区域称为元数据区，基于这种文件存储方式，Linux文件系统中定义了两个与文件相关的、至关重要的概念：索引结点和目录项。

(1)索引结点

索引结点（index node，简称inode）的实质是一个结构体，主要功能是保存文件的属性信息（如所有者、所属区、权限、文件大小、时间戳等），Linux系统中的每个文件都会被分配一个inode，当有文件创建时，系统会在inode表中获取一个空闲的inode，分配给这个文件。inode存储在inode表中，inode表存储inode和inode的编号（inumber），inode表在文件系统创建之时便被创建，因此文件系统中可存储的文件数量也在文件系统创建时已确定。

(2)目录项

Linux文件系统中的索引结点保存着文件的诸多属性信息，但并未保存文件的文件名。实际上，Linux系统中文件的文件名并不保存在文件中，而是保存在存放该文件的目录中。

Linux系统中定义了一个被称为目录项（dentry）的结构体，该结构体主要存储文件的文件名与inode编号，系统通过读取目录项中的文件名和文件的inumber，来判断文件是否存在于这个目录中。dentry中允许同一个inode对应不同的文件名，但不允许相同的文件名对应不同的inode。

Linux文件系统

ext -- > ext4

fdisk/mkf2es

fdisk

fdisk命令可以查看当前系统中的磁盘，以及磁盘中的分区情况，也可以用于磁盘分区。

命令格式如下：

fdisk [选项] [参数]

常用的选项为-l，该选项可以列出指定设备的分区表状况

mkf2es

mkf2es命令可为磁盘分区创建ext2、ext3文件系统。

命令格式如下：

fdisk [选项] [参数]

Mkfs命令亦可为磁盘分区创建系统，其常用选项为-t，参数一般为文件系统类型。

du/df

du

选项	说明
-a	显示所有目录以及目录中每个文件所占用的磁盘空间
-s	只显示目录及文件占用磁盘块的总和
-b/-k/-m/-g	以b/kb/mb/Gb为单位，显示目标占用磁盘块的总和
-x	以最初处理时的文件系统为准，跳过不同文件系统上的文件
-D	显示指定符号链接的源文件大小

df

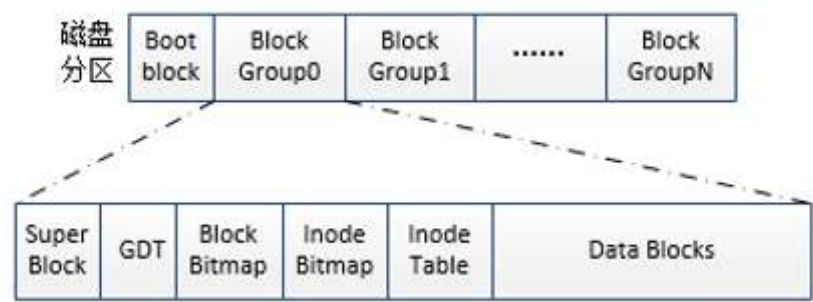
df的参数可以是文件，但打印的信息会是该文件所在文件系统磁盘的使用情况。在命令行输入命令df，会打印如下所示的信息：

```
[root@localhost itcast]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda2        18208184  5016648   12259952   30% /
tmpfs             502068       372     501696    1% /dev/shm
/dev/sda1         289293      34696    239237    13% /boot
```

ext2/ext3 文件系统

ext2

磁盘分区的第一个部分为启动块（Boot Block），启动块占用一个块空间，用来存储磁盘的分区信息和启动信息。图5-5中启动块之后是多个块组，每个块组中包含6部分，即：超级块（Super Block）、块组描述符（GDT）、块位图（Block Bitmap）、位图（Inode Bitmap）、位表（Inode Table）、数据块（Data Blocks）。



ext2文件系统的结构有以下几个优点：

- ①可有效防止磁盘碎片的产生，减少磁盘传送次数，降低系统消耗
- ②管理员可根据给定分区的大小，预计分区中存放的文件数，从而确定分区中inode的数量，保证磁盘空间利用率
- ③降低对存放于一个单独块组中的文件并行访问时磁盘的平均寻道时间
- ④支持快速符号链接

ext3

ext3是一个完全兼容ext2文件系统的日志文件系统，它在ext2的基础上，添加了一个被称为日志的块，专门来记录写入或修订文件时的步骤。

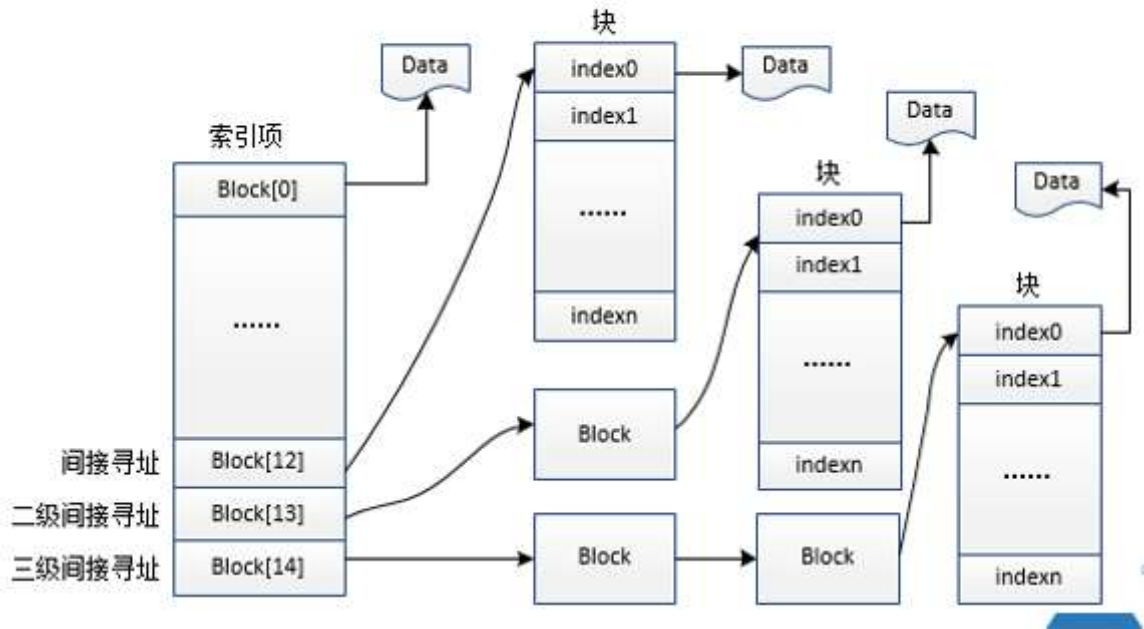
日志文件系统可以按照不同的方式进行工作，ext3中可通过对/etc/fstab文件中的data属性进行设置来修改文件系统的工作模式。日志文件系统的工作模式分为三种，其设置方式分别如下：

```
data=journal
data=ordered
data=writeback
```

数据块寻址

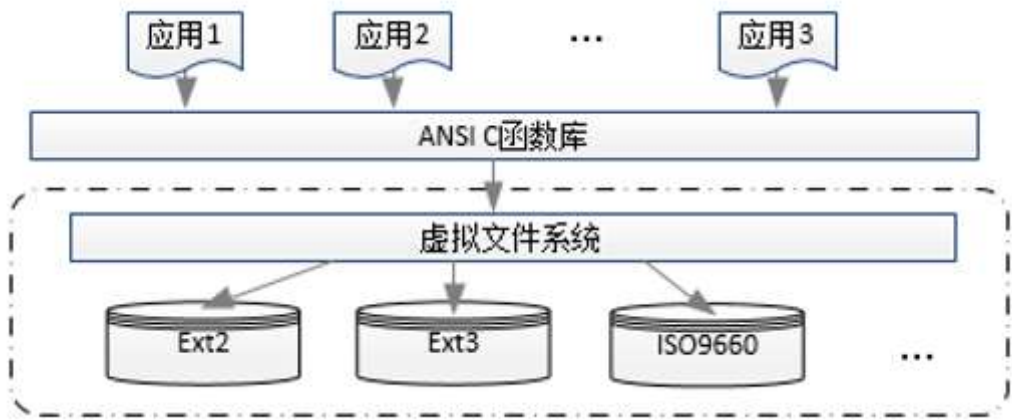
Linux文件系统中文件的属性信息和数据分开存放，系统通过文件inode中的索引项Block来查找文件数据。Block索引项是一个数组，包含15个索引项，其中：

- Block[0]~Block[11]是直接索引项；
- Block[12]为间接索引项；
- Block[13]为二级间接索引项；
- Block[14]为三级间接索引项。



虚拟文件系统

虚拟文件系统又称虚拟文件切换系统（Virtual Filesystem Switch），是操作系统中文件系统的虚拟层，其下才是具体的文件系统。虚拟文件系统的主要功能，是实现多种文件系统操作接口的统一，既能让上层的调用者使用同一套接口与底层的各种文件系统交互，又能对文件系统提供一个标准接口，使Linux系统能同时支持多种文件系统。虚拟文件系统与上层应用以及底层的各种文件系统之间的关系如图所示。



挂载

所谓挂载，是指将一个目录作为入口，将磁盘分区中的数据放置在以该目录为根结点的目录关系树中，这相当于为文件系统与磁盘进行了链接，指定了某个分区中文件系统访问的位置。

Linux系统中根目录是整个磁盘访问的基点，因此根目录必须要挂载到某个分区。

Linux系统中通过mount命令和unmount命令实现分区的挂载和卸载。

mount

Linux系统中可以使用mount命令将某个分区挂载到目录，mount命令常用的格式如下：

mount [选项] [参数] 设备 挂载点

mount命令常用的选项有两个，分别为-t和-o。

选项-t用于指定待挂载设备的文件系统类型，常见的类型如下：

光盘/光盘镜像：iso9660；

DOS fat16文件系统：msdos；

Windows 9x fat32文件系统：vfat；

Windows NT ntfs文件系统：ntfs；

Mount Windows文件网络共享：smbfs；

UNIX(LINUX)文件网络共享：nfs。

选项-o主要用来描述设备的挂载方式，常用的挂载方式如表所示。

方式	说明
loop	将一个文件视为硬盘分区挂载到系统
ro	read-only，采用只读的方式挂载设备（即系统只能对设备进行读操作）
rw	采用读写的方式挂载设备
iocharset	指定访问文件系统所用的字符集
remount	重新挂载

unmount

当需要挂载的分区只是一个移动存储设备（如移动硬盘）时，要进行的工作是在该设备与主机之间进行文件传输，那么在文件传输完毕之后，需要卸下该分区。Linux系统中卸下分区的命令是umount，该命令的格式如下：

umount [选项] [参数]

umount命令的参数通常为设备名与挂载点，即它可以通过设备名或挂载点来卸载分区。若以挂载点为参数，假设挂载点目录为/mnt，则使用的命令如下：

umount /mnt

通常以挂载点为参数卸载分区，因为以设备为参数时，可能会因设备正忙或无法响应，导致卸载失败。也可以为命令添加选项“-l”，该选项代表“lazy mount”，使用该选项时，系统会立刻从文件层次结构中卸载指定的设备，但在空闲时才清除文件系统的所有引用。

Linux文件类型

Linux系统中的文件也有类型之分，但Linux系统中的文件类型不以扩展名区分，虽文件也可能有扩展名，但这些扩展名只表示与程序的关联（如.tar、.gz等），不代表具体的文件类型。

“ls -l”命令打印的文件属性信息中的第一个字符，便代表文件的类型，该字符有7种取值，分别对应不同的文件：

- ①d：directory，目录文件；
- ②l：link，符号链接文件；
- ③s：socket，套接字文件；
- ④b：block，块设备文件；
- ⑤c：character，字符设备文件；
- ⑥p：pile，管道文件；
- ⑦-：不属于以上任一种文件的普通文件。

普通文件

目录文件

设备文件

Linux系统中将外部设备视为一个文件来管理，设备文件被保存在系统中的/dev目录下。将设备抽象为文件的好处是：应用程序可以使用与操作普通文件一样的方式，对设备文件执行打开、关闭和读写等操作。比如查看属性信息时，无论是普通文件还是设备，都可以使用“ls -l”命令。

使用“ls -l /dev”命令查看设备文件的详细信息

块设备文件

字符设备文件

特殊文件

管道文件

嵌套字文件

符号链接文件（软链接文件）

分为两种：硬链接文件和软链接文件，这两种文件都能实现windows中快捷方式的功能，但它们的实现方式不同。

Linux系统中创建链接文件的命令是ln，其命令格式如下：

ln [选项] 源文件 目标文件

ln命令的第一个参数为被链接的文件，即源文件的路径名；第二个参数为链接文件的路径名，指定链接文件的存储位置。

说明：

①当ln命令的选项缺省时，系统会创建一个硬链接文件。

②若搭配-s选项，则会创建一个软

注意：

（1）创建硬链接文件时，源路径中的对象不能是一个目录，因为硬链接文件与源文件inode相同，若创建的硬链接文件包含在源文件目录中时，会产生循环访问；软链接的inode与源文件不同，不受此限制。

（2）磁盘分区中的inode表是文件系统级别的，硬链接文件与源文件的inode相同，因此为文件创建硬链接时，硬链接文件可以在同一文件系统的不同目录中，但不能跨文件系统；而软链接文件与源文件inode不同，因此软链接文件可以跨文件系统。

（3）在创建硬链接文件时，文件的硬链接数会加一（可使用ls -l命令查看），若执行删除操作，只有在硬链接数为1时该文件才会真正被删除，其它时候只是删除文件路径名中目录项中的记录，并使文件硬链接数减1；创建软链接时不会增加被链接文件的链接次数。

（4）Linux系统中文件类型之一的符号链接文件只包含软链接文件，硬链接文件本质上是Linux系统中的普通文件。

文件操作

文件描述符

文件描述符（File Descriptor）是一个非负整数，它实质上是一个索引值，存储于由内核维护的该进程打开的文件描述符表中。

系统为每一个进程维护了一个文件描述符表（open file description table），用于存储进程打开文件的文件描述符，进程打开的普通文件其文件描述符从3开始。

存在于进程中的文件描述符是进程级别的，除此之外，内核也对所有打开的文件维护了一个文件描述符表，这是一个系统级的文件描述符表，该表又称打开文件表（open file table），表中的记录被称为打开文件句柄（open file handle）。

一个打开文件句柄中与一个已打开文件相关的信息如下所示：

当前文件偏移量；

打开文件时所用的状态标识；

文件访问模式；

与信号驱动相关的设置；

对该文件inode对象的引用；

文件类型和访问权限；

指向该文件所持有的锁列表的指针；

文件的各种属性信息。

文件I/O

补代码练习

open()

open()函数的功能是打开或创建一个文件，该函数存在于系统函数库fcntl.h中，其函数声明如下：

```
int open(const char *pathname,int flags[,mode_t mode]);
```

read()

read函数用于从已打开的设备或文件中读取数据，该函数存在于函数库unistd.h中，其函数声明如下：

```
ssize_t read(int fd, void *buf, size_t count);
```

①fd为从open()或creat()函数获取的文件描述符

②buf为缓冲区

③count为计划读取的字节数

write()

write()函数用于向已打开的设备或文件中写入数据，该函数存在于函数库unistd.h中，其函数声明如下：

```
ssize_t write(int fd, const void *buf, size_t count);
```

①fd为文件描述符

②buf为需要输出的缓冲区

③count为最大输出字节数。

lseek()

每个打开的文件都有一个当前文件偏移量（current file offset），该数值是一个非负整数，表示当前文件的读写位置，Linux系统中可以通过系统调用lseek()对该数值进行修改，lseek()函数位于函数库unistd.h中，其函数声明如下：

```
off_t lseek(int fd, off_t offset, int whence);
```

参数列表：

- ①fd为文件描述符
- ②offset用于对文件偏移量的设置，该参数值可正可负
- ③whence用于控制设置当前文件偏移量的方法

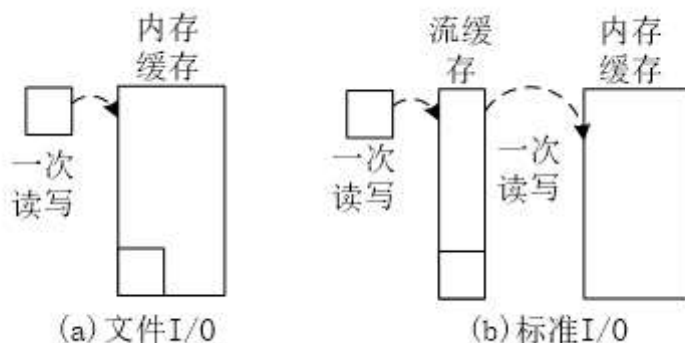
close()

打开的文件在操作结束后应该主动关闭，Linux系统调用中用于关闭文件的函数为close()函数，该函数的使用方法很简单，只要在函数中传入文件描述符，便可关闭文件。close()函数位于函数库unistd.h中，其声明如下：

```
int close(int fd);
```

若函数close()成功调用，则返回0，否则返回-1。

Linux系统调用中的文件I/O又被称为无缓存I/O，除此之外，在程序编写时我们还可以使用一种有缓存的I/O。有缓存的I/O又被称为标准I/O，是符合ANSI C标准的I/O处理，标准I/O有两个优点，一是执行系统调用read()和write()的次数较少；二是不依赖系统内核，可移植性强。



文件操作

①stat()函数

stat()函数用于获取文件的属性，该函数存在于函数库sys/stat.h中，其声明如下：

```
int stat(const char *path, struct stat *buf);
```

②access()函数

access()函数用于测试文件是否拥有某种权限，该函数存在于库函数unistd.h中，其声明如下：

```
int access(const char *pathname, int mode);
```

③chmod()函数

chmod()函数用于修改文件的访问权限，该函数存在于函数库sys/stat.h中，其函数声明如下：

```
int chmod(const char *path, mode_t mode);
```

④truncate()函数

truncate()函数用于修改文件大小，常用于扩展文件，其功能与lseek()类似，该函数存在于函数库sys/stat.h中，其函数声明如下：

```
int truncate(const char *path, off_t length);
```