

第2章 文件系统

本章内容

- Linux 文件
- 文件系统
- 设备文件
- 文件及文件系统管理
- 文件操作shell命令

1. Linux文件

- ❑ Linux系统的文件命名方式与Windows系统不同。
- ❑ 系统目录结构以“ /” 为根目录，形成一棵树状结构。
- ❑ 和DOS一样，“.”和“..”分别用来表示“当前目录”和“父目录”，因此也不能作为文件名。
- ❑ “/”表示目录树的根或者路径名的分隔符，不能出现在文件名中。

文件名

- Linux系统区分英文字符的大小写。
- 例如，helloworld、HelloWorld和Helloworld表示三个不同的文件。
- 用户登录名和密码也区分大小写。
- Linux目录命名和文件命名的规则相同，除非有特别的原因，否则用户创建的文件和目录名通常使用小写字符。
- Linux系统下的文件名长度最多可到256个字符。
- 通常文件名的字符包括：字母、数字、“.”（点）、“_”（下划线）和“-”（连字符）。

转意字符

- Linux系统中有一些转意字符。
- 转意字符：被系统借用，表示某种特殊含义的字符。
 - “?”，问号，代表任意一个字符的通配符
 - “*”，星号，代表任意多个字符的通配符
 - “ ”（空格）、“\$”、“&”、括号等等。
 - 在文件名中尽量避免使用这些字符。

文件名规则

- Linux中的文件和目录，可以给它们取任何名字，但是必须遵守以下规则：
 - 除了 “/” 外，所有的字符都可以使用；
 - 有些字符最好不用，如空格符、制表符、退格符和其它转意字符；
 - 避免使用 “.” 作为普通文件名的第一个字符，因为在Linux下以 “.” 开头的文件是属于隐藏文件。

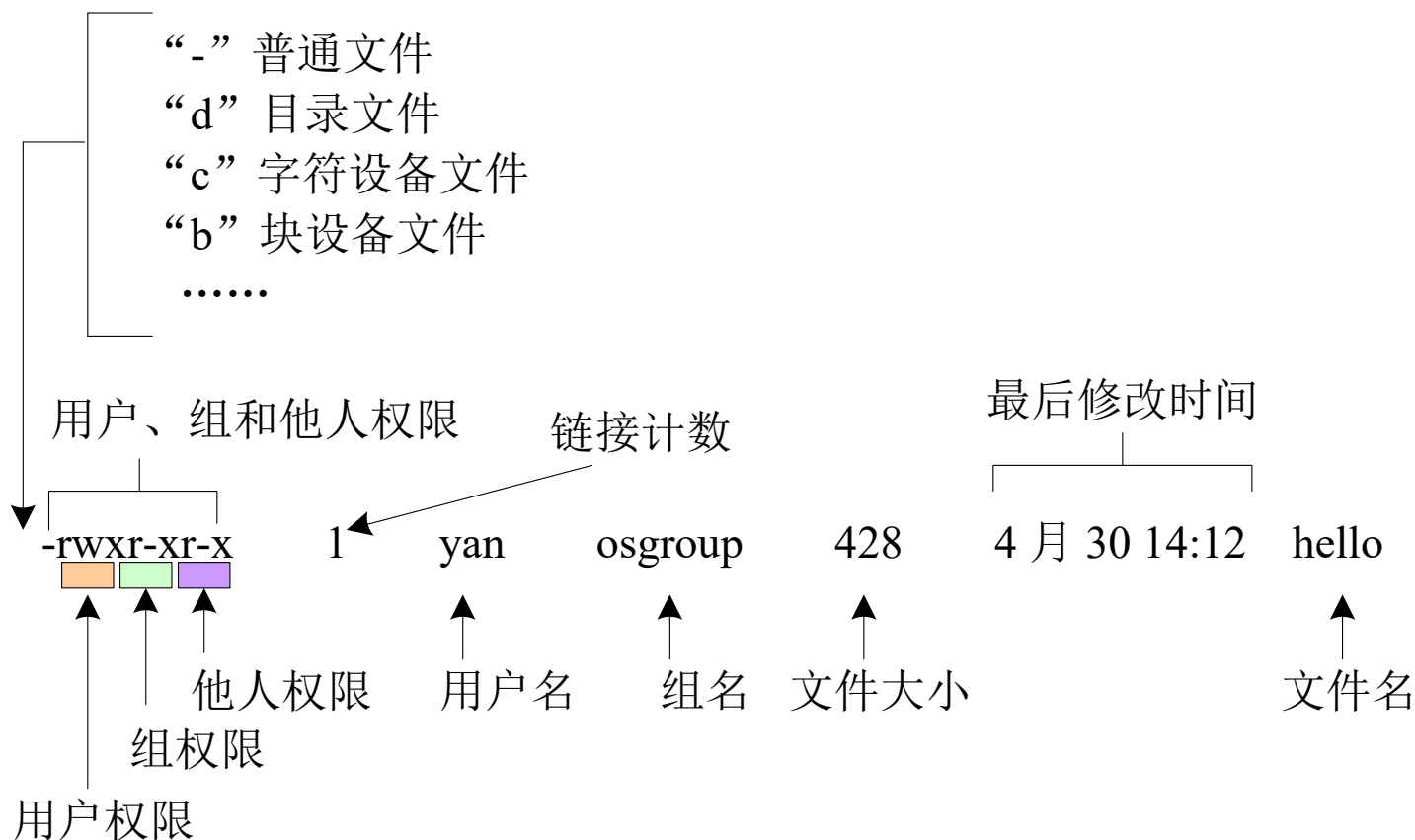
文件类型和属性

- Linux文件类型在文件的属性字段中第1位标志，共有8种类型。当需要对特殊文件进行I/O操作时，该操作将调用相应的设备驱动程序。

文件类型	标志	说明
普通文件	-	可以是各种文件，如文本文件、图片、可执行文件等
目录文件	d	包含文件列表的特殊文件
块设备文件	b	与块设备交互的接口文件，如硬盘、软盘、光盘、移动存储设备等
字符设备文件	c	与字符设备交互的接口文件，如打印机、声音设备等
符号链接文件	l	建立指向普通文件或目录的链接文件
硬链接文件	-	只能建立指向普通文件的链接文件
套接字	s	用于进程之间通信，一般用于网络通信
有名管道	p	用于进程之间通信

文件类型和属性查看与解释

- 例如，在shell下执行 “ls -l” 命令，可以查看文件的属性。



Linux系统目录结构

- Linux系统中文件以“/”为根目录，形成一棵树状结构，系统的目录结构根据不同的发行版略有不同，以下为主要的和最基本的目录：
 - /bin : binary的缩写，存放用户经常使用的shell命令
 - /boot : 存放启动Linux使用的一些核心文件
 - /dev : device的缩写，存放所有外部设备
 - /home: 系统默认的用户主目录，每个用户在该目录下对应一个子目录
 - /lib : 存放系统最基本的动态链接共享库
 - /lost+found: 一般为空，当系统不正常关机时，会存放一些中间文件
 - /mnt: 此目录为空，用于临时安装其它文件系统的目录
 - /proc : 系统内存的映像，即内容不在硬盘上而在内存里，也称虚拟目录
 - /root : 系统管理员的登录目录
 - /sbin : 存放系统管理员使用的管理程序
 - /tmp : 存放临时文件
 - /usr : 系统和应用程序文件几乎都存放在这个目录下
 - /var : 为保持/usr的稳定，经常被修改的文件放在这里，如系统日志文件

2. 文件系统

- 文件系统是操作系统的核心部分，提供了分层结构的目录和文件。
- 文件系统指文件存在的物理空间。
- 在Linux系统中，每个分区都是一个文件系统，都有自己的目录层次结构。

文件系统分类

- Linux中常见的文件系统及其特点：
 - Ext2：专为Linux设计。可用于标准的块设备如硬盘，也可用于软盘等移动存储设备。
 - Ext3：Ext2的日志文件系统，采用独立的日志文件跟踪磁盘内容的变化，在写入文件内容的同时写入文件的元数据，每次修改文件的元数据，都要先向日志的数据结构中登记相应的条目，避免系统非正常关机引起文件系统中某些文件的元数据被破坏。

文件系统分类（续）

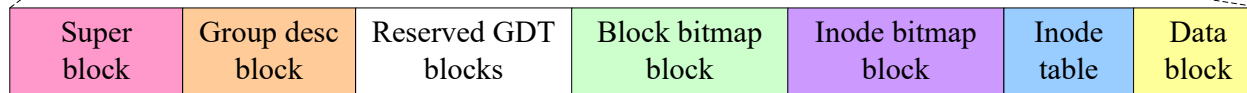
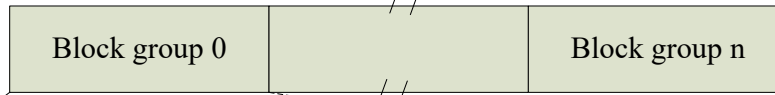
- Ext4: Linux内核自2.6.28开始正式支持文件系统Ext4。Ext3和Ext2的差别仅局限于日志系统，但是Ext4修改了文件系统的大部分重要数据结构，比如文件数据的存储方式，可以提供更佳的性能和可靠性。
- iso9660: CD-ROM的标准文件系统。
- swap: 用于Linux磁盘交换分区的特殊文件系统。
- /proc: 一个伪文件系统，不占用外存空间。以文件系统的方式为访问系统内核数据的操作提供接口，使得用户和应用程序可以通过proc得到系统的信息，并可以改变内核的某些参数。

Ext2与Ext3

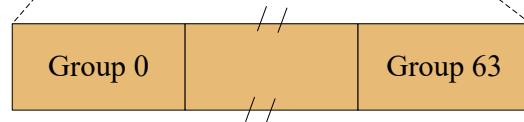
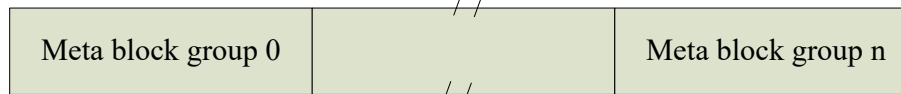
- 如果分区采用Ext2/Ext3文件系统，那么每个分区开头都会有一个引导块，用来保存引导信息，为了管理方便，文件系统将剩余空间划分为一个个块组，分别为：超级块、块组描述符、数据块位图、索引节点位图、索引节点表、数据块。
- Ext2/Ext3文件系统与其它经典的Unix/Linux文件系统一样，采用了直接、间接、二级间接和三级间接块的形式来定位磁盘中的数据块。

Ext3与Ext4

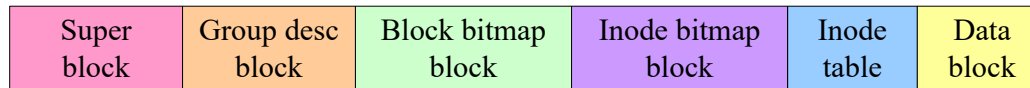
Ext3



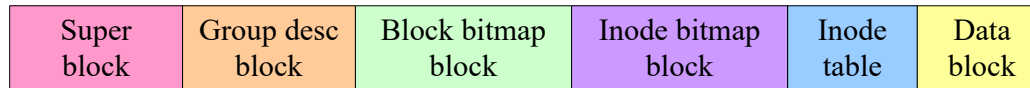
Ext4



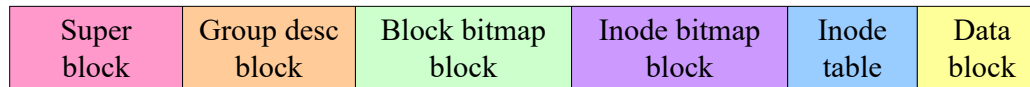
Group 0



Group 1



Group 63



硬链接与符号链接

- Linux文件链接分两种，一种被称为硬链接，另一种被称为符号链接。
 - 创建硬链接的shell命令：ln -d
 - 注意：默认情况下（即ln命令不带任何选项），ln命令产生硬链接。
 - 创建符号链接的shell命令：ln -s

硬链接与符号链接（续）

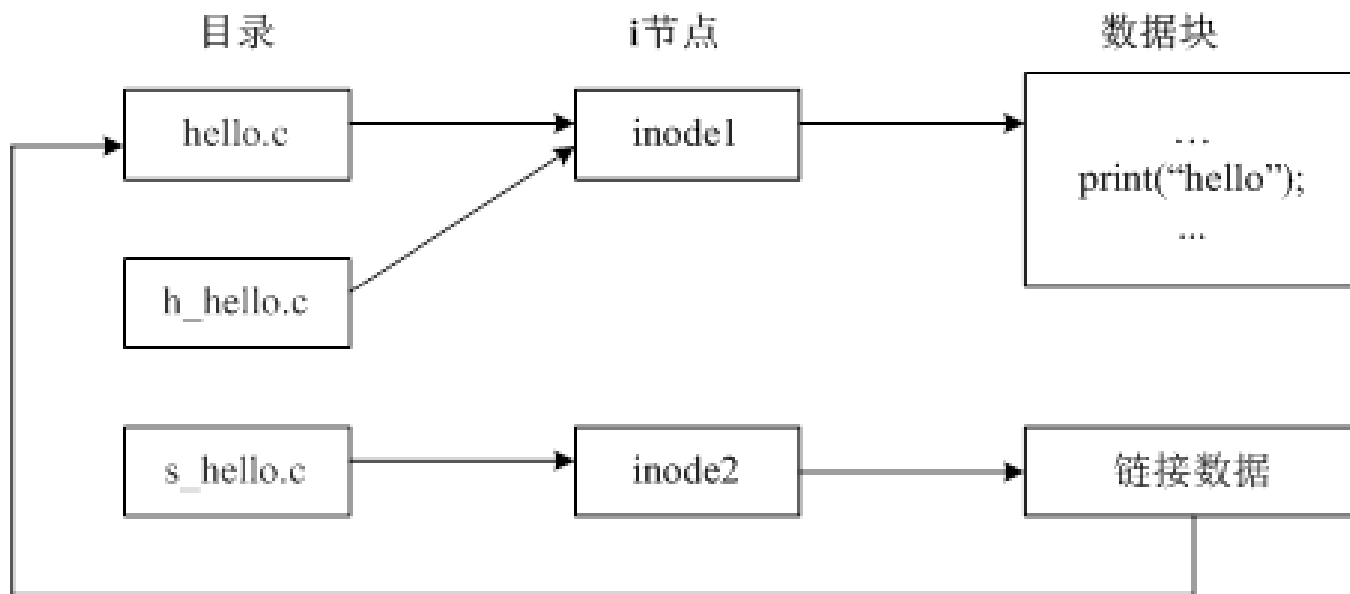
- 硬链接：多个文件名指向同一个i节点。
- 硬链接的作用是允许一个文件拥有多个有效路径名。只删除一个链接并不影响i节点本身和其它的链接，只有当最后一个链接被删除后，文件的数据块及目录的链接才会被释放。也就是说，文件真正删除的条件是与之相关的所有硬链接文件均被删除。
- 符号链接：也叫软连接。
- 软链接文件类似于Windows的快捷方式，它实际上是一个特殊的文件。在符号链接中，文件内容实际上是一个文本文件，其中包含所链接的文件的位置信息。

示例

- 假设有文件file1和file2，创建file1的硬链接h_file1，创建file2的符号链接s_file2。
- 当删掉源文件file1时，那么磁盘上file1对应的数据文件是不会被删除的，因为i节点上记录了该文件的硬链接数，只有硬链接数是0的时候，删除文件名的时候，对应的数据在磁盘上才会删除。

硬链接与符号链接示意图

- 当删除源文件file2时，s_file2将无法使用，因为它链接的文件已经不存在。
- 下图描述了文件hello.c、硬链接h_hello.c和符号链接s_hello.c之间的关系。



3. 设备文件

- Linux系统中，所有的设备都是以文件的形式存在，所有的文件都是在以“/”目录为根的一颗“大”目录树中进行管理。
- 存储设备包括USB、光盘、软盘等，在需要访问时，必须将这些设备对应的“小”目录挂载到“大”目录树中；访问结束后需要从“大”目录树中卸载掉。

设备文件挂载

- 挂载对象的文件系统类型
- 系统文件/`/proc/filesystems`中含有当前系统所支持的文件系统类型。如：
 - `vfat`是指Windows常用的FAT32文件系统，它兼容FAT16文件系统；
 - `Ext4`是目前Linux系统常用的文件系统。如果在文件/`/proc/filesystems`中看不到`vfat`文件系统，可以通过“`modprobe vfat`”命令加载该模块。

挂载对象的设备名称

- /dev/hda1中, “hd” 代表IDE硬盘, 对于每个硬盘设备, 可以划分多个分区, 通常主分区有1个, 其它的为逻辑分区, hda1表示第一个IDE硬盘的第一个分区。
- /dev/sda1中, “sd” 代表SCSI接口的硬盘。sda1表示第一个SCSI硬盘的第一个分区。usb存储设备通常也被模拟为SCSI设备来驱动。
- /dev/fd0中, “fd” 代表软盘。fd0表示第一个软盘, fd1表示第二个软盘。

挂载对象的挂载点

- Linux系统中有一个/mnt目录，专门用来做挂载点目录。在执行mount命令时，首先查看挂载点目录是否存在，如果不存在则需要创建该目录，否则mount命令无法正常执行。
- mount [选项] <挂载设备名称> <挂载点>

示例

- `mount -t vfat /dev/fd0 /mnt/floppy`
- 解释：把软驱设备fd0挂载到目录/mnt/floppy目录中，使用的文件系统类型为vfat，即Fat32或Fat16。

- `mount /dev/sdb /mnt/usb`
- 解释：把usb设备挂载到目录/mnt/usb目录中

设备文件卸载

- ❑ `umount` 卸载
- ❑ `umount` [挂载点或设备名]

示例

- ❑ `umount /mnt/usb`
- ❑ 解释：卸载之前挂载的usb设备

- ❑ `umount /dev/sdb`
- ❑ 解释：卸载usb设备/dev/sdb

- ❑ 注意：如果挂载的设备正在使用，那么执行该命令会出现“device busy”的错误信息。在卸载设备前，必须先从挂载点如/mnt/usb中退出。

设备文件自动挂载配置

- mount和umount命令都是通过手动方式加载和卸载设备文件。
- 系统提供自动方式来管理设备文件，其方法是：修改配置文件/etc/fstab。
- 例如，当一台机器中同时安装了Linux和Windows两个操作系统时，用户可以在启动Linux系统时自动挂载Windows的硬盘分区。

设备文件自动挂载配置（续）

- /etc/fstab文件的每一行为一个设备文件的记录，格式如下：

`<fs_spec> <fs_file> <fs_vfstype> <fs_mntops> <fs_freq> <fs_passno>`

- fs_spec: 设备名称。
- fs_file: 挂载点。
- fs_vfstype: 文件系统类型名称。
- fs_mntops: 挂载时需要的参数，如ro表示以只读模式加载该文件系统，rw表示以读写模式加载该文件系统，iocharset表示指定输出字符编码集。
- fs_freq: 定义“dump”命令对文件系统进行转储的频率，若不需要转储就将该字段设置为0。
- fs_passno: 被fsck命令用来决定在重启时检查文件系统的顺序。

示例

□ /etc/fstab文件示例:

```
/dev/hda1    /                    Ext4      defaults          1 1
/dev/cdrom    /mnt/cdrom          iso9660    noauto,owner,ro   0 0
```

- 保存配置文件，重启系统后即可生效。生效后，系统将会自动加载hda1分区和cdrom。

几种特殊的设备文件

- 字符设备是指每次与系统传输1个字符的设备，字符设备在实现时，大多不使用缓存器。
 - 例如，键盘是一个字符设备。
- 块设备是指与系统间用块的方式移动数据的设备，这些设备节点通常代表可寻址设备。
 - 如硬盘、CD-ROM就是块设备。
- 伪设备是指没有对应物理设备的设备，操作系统运用它们提供多种功能。
 - 经常使用到的伪设备包括：见下页。

伪设备

(1) 标准输入输出设备

- ❑ 执行shell命令时通常会自动打开三个标准文件：
 - 标准输入文件stdin
 - 标准输出文件stdout
 - 标准错误输出文件stderr
- ❑ 进程从标准输入文件中得到输入数据，将正确输出数据输出到标准输出文件，将错误信息送到标准错误文件。
- ❑ 标准输入stdin文件描述符为0，标准输出stdout文件描述符为1，标准错误stderr文件描述符为2。

伪设备（续）

(2) dev/null设备

- ❑ 这是一个空设备，丢弃一切写入其中的数据。
- ❑ 空设备通常被用于丢弃不需要的输出流。

`1>/dev/null 2>&1`

伪设备（续）

(3) dev/zero设备

- 这是一个特殊的文件，当读它的时候，它会提供无限的空字符。
- 一个典型用法是用它提供的字符流来覆盖数据。
- 另一个常见用法是产生一个特定大小的空白文件。

示例

□ `$ shell命令 1>/dev/null 2>&1`

□ 解释：

- `1>/dev/null`：首先表示标准输出重定向到空设备文件，也就是不输出任何信息到终端，说白了就是不显示任何信息。
- `2>&1`：接着，标准错误输出重定向到标准输出，因为之前标准输出已经重定向到了空设备文件，所以标准错误输出也重定向到空设备文件。

示例

- ❑ `dd if=/dev/zero of=testzero count=1024 bs=1024`
 - 1024+0 records in
 - 1024+0 records out
 - 1048576 bytes (1.0 MB) copied, 0.0107194 seconds, 97.8 MB/s
- ❑ 解释：
 - 创建一个大小为1M文件，该文件一个块是1024字节，一共是1024块（刚好1M），用/dev/zero文件内容填充它，输出为testzero文件。

4. 文件操作shell命令

- 大部分shell命令都是针对文件的。
- 文件操作shell命令
 - 文件内容显示
 - 文件查找
 - 文件系统管理
 - 文件权限

文件内容显示

□ more 文件名

- `$ more file1`
- 分页显示文件内容

□ less 文件名

- `$ less file1`
- 与more功能相近，只是从浏览习惯和显示方式上不同

□ cat 文件名

- `$ cat file1`
- `$ cat file1 file2`
- 顺序显示文件内容，cat命令有合并文件的功能

文件内容显示

□ nl 文件名

- `$ nl file1`
- 带行号显示文件内容

□ tac 文件名

- `$ tac file1`
- 逆序显示文件内容

□ head [-n number]

- `$ head -10 file1`
- 从顶部往下显示指定的行数

文件内容显示

❑ tail [-n number]

- `$ tail -20 file1`
- 从底部往上显示指定的行数，例如显示log文件中最后几行或几十行最新记录

❑ od [-t TYPE]

- `$ od file1`
- 以二进制的方式读取文件内容

❑ clear

- `$ clear`
- 清屏

示例

- ❑ `[root@主机名]# cat -n /etc/profile`
- ❑ 解释：对/etc目录中的profile文件的所有行（包括空白行）进行编号并输出显示。

- ❑ `[root@主机名]# cat -b file1 file2 >> file3`
- ❑ 解释：把 file1和file2 的档案内容加上行号（空白行不加）之后将内容附加到 file3中。

文件查询

- `find <指定目录> <指定条件> <指定动作>`
- `$ find . -name 'my*'`
 - 搜索当前目录（含子目录）中所有文件名以my开头的文件。
- `$ find . -name 'my*' -ls`
 - 搜索当前目录中，所有以my开头的文件，并显示它们的详细信息。
- `$ find . -type f -mmin -10`
 - 搜索当前目录中，所有过去10分钟中更新过的普通文件。

文件查询（续）

- locate
- `$ locate ~/m`
- 解释：搜索用户主目录下，所有以m开头的文件。
- locate命令其实是"find -name"的另一种写法，但是要比后者快得多，原因在于它不搜索具体目录，而是搜索/var/lib/locatedb数据库。
- Linux系统自动创建这个数据库，并且每天自动更新一次。为了避免查不到最新变动过的文件，可以使用updatedb命令，手动更新数据库。

文件查询（续）

□ which命令名

- `$ which grep`
- 在PATH变量指定的路径中，搜索某个系统命令的位置，并且返回第一个搜索结果。

□ type命令名

- `$ type cd`
- 解释：系统会提示，cd是shell的自带命令（build-in）。用来区分某个命令是由shell自带，还是由shell外部的独立二进制文件提供。如果一个命令是外部命令，那么使用-p参数，会显示该命令的路径，相当于which命令。

文件及文件系统管理

- cp在当前目录下复制文件或目录(加-r参数)。
- mv移动文件或者目录。
- rm删除文件或目录，-rf表示强制级联删除文件。
- vi进入后是命令模式，按"a"键或者shift+"i"键进入编辑模式；在编辑模式按"Esc"键退回到命令模式，此时使用dd可以删除当前行，dw删除当前单词，o在当前光标所在位置插入一行，O在当前光标所在位置的下一行插入一行。在命令模式输入"ZZ"存盘退出，退回到命令行窗口。

文件及文件系统管理（续）

□ mkdir创建目录

- `$ mfs -t Ext3 /dev/sda6`
- 将sda6分区格式化为Ext3格式

□ ls

- `$ ls -l` 列出各个子目录和文件的相关信息

□ 压缩和解压缩文件

- gzip压缩和解压缩文件，gz格式
- zip压缩和解压缩文件，zip格式
- bzip2压缩和解压缩文件，bzip2格式
- tar归档、压缩文件，tar格式

示例

- ❑ `tar -cvf myjpg.tar *.jpg`
- ❑ 解释：将目录里所有jpg文件打包成myjpg.tar。
- ❑ `tar -xvf myjpg.tar`
- ❑ 解释：解压tar包。
- ❑ `tar -czf jpg.tar.gz *.jpg`
- ❑ 解释：将目录里所有jpg文件打包成myjpg.tar后，并且将其用gzip压缩，生成一个gzip压缩过的包，命名为myjpg.tar.gz。
- ❑ `tar -xzvf myjpg.tar.gz`
- ❑ 解释：解压myjpg.tar.gz。

更改文件访问权限

□ chmod

chmod [选项] <mode> <file>...

mode: [ugoa][[+ -=][rwxX]...][, ...]

mode也可以用数字来表示权限:

chmod abc file

□ 下一章将有详细介绍!

几个具有杀伤力的命令

- 命令一： `rm -rf /`
- `rm`的“-r”递归删除和“-f”强制删除是很危险的选项，即使日常操作，也会遇到误删文件的情况。
- 类似的命令还有：
 - `rm -rf .` 删除当前目录下的所有文件
 - `rm -rf *` 删除当前目录下的所有文件
 - `rm -rf *` or `rm -rf *.*` 删除当前目录下的所有文件
 - `rm -rf ~ / &` 删除根分区和主目录，即使你不是root，主目录还是不能幸免。

几个具有杀伤力的命令（续）

- ❑ 命令二： `mv /home/yourhomedirectory/* /dev/null`
- ❑ 解释：这条命令无疑会让系统抓狂，你的主目录会再也看不到。

几个具有杀伤力的命令（续）

- 命令三： `mkfs.xxxx`
- 解释： `xxxx`可以是 `vfat`、`Ext2`、`Ext3`、`bfs`等，`mkfs`是格式化命令，运行时会擦除你的硬盘分区。

几个具有杀伤力的命令（续）

- ❑ 命令四：`dd if=/dev/zero of=/dev/sda`
- ❑ 解释：全部硬盘清零。
- ❑ `dd`是强大的IO输入输出定向工具，如果使用不当，破坏性很大，不仅仅是当前分区，当前系统，有的时候是整个硬盘。

几个具有杀伤力的命令 (续)

□ 类似命令还有:

- `dd if=/dev/sda of=/dev/sdb`
 - 用第一块硬盘的内容覆盖第二块的内容
- `dd if=something of=/dev/sda`
 - 往硬盘上写垃圾数据
- `any_command > /dev/sda`
 - 将用随意的数据破坏硬盘

本章小结

- ❑ Linux系统中，所有的目录、设备等都以文件形式存在。
- ❑ 文件类型在文件的属性字段中标志。
- ❑ 文件命名有严格的规范，且区分大小写。整个系统中的文件以“/”为根目录，形成一棵树状结构。
- ❑ Linux文件系统从Ext2、Ext3发展到Ext4，能够提供更佳的性能和可靠性。
- ❑ 设备以文件的形式存在，访问需要通过挂载、卸载来完成。
- ❑ 用户对文件及文件系统的操作都通过shell命令来完成，包括文件的新建、查询、删除、拷贝、移动、压缩，文件内容的查看，文件的链接等。
- ❑ 不过有几个具有杀伤力的命令在使用时必须谨慎。