



# Linux 文件系统

## 文件系统

### q 文件系统（File system）的各种定义

- u 文件系统是包括在一个磁盘（硬盘、光盘及其它存储设备）上的目录结构；一个磁盘设备可以包含一个或多个文件系统。
- u 文件系统是在一个磁盘（硬盘、光盘及其它存储设备）上组织文件的方法。
- u 文件系统是文件的数据结构或组织方法。
- u 文件系统是基于被划分的存储设备上的一种文件的命名、存储、组织及读取的方法。
- u 一个文件系统是有组织存储文件或数据的方法，目的是易于查询和存取。文件系统是基于一个存储设备，比如硬盘或光盘，并且包含文件物理位置的维护。

## 文件系统

### q 文件系统（File system）

- u 磁盘上存储文件的方法和数据结构，是操作系统组织、存取和保存信息的重要手段，每种操作系统都有自己的文件系统。
- u 文件系统包含的三个问题
  - l 从用户角度看，系统中的文件是怎么组织的
  - l 文件是怎样存储在存储设备上的
  - l 文件是如何操作的

## 一些基本概念

### u 存储介质

硬盘、光盘、软盘、Flash盘、磁带、网络存储设备等。

### u 磁盘的分割

主要是指硬盘；对于大硬盘，需要合理规划分区。硬盘的分割，Linux下可以用fdisk、parted等，Windows下可以使用磁盘管理工具。第三方软件有PQ magic等。

### u 文件系统的创建

一般也被称为格式化，通过一些初始化工具来进行。一般的情况下每中操作系统都有这方面的工具。

### u 文件系统的挂载（mount）

文件系统只有挂载后才能使用。Windows是自动挂载；Unix类的操作系统是通过mount命令实现，挂载时要有挂载点，通常是一个空置的目录。

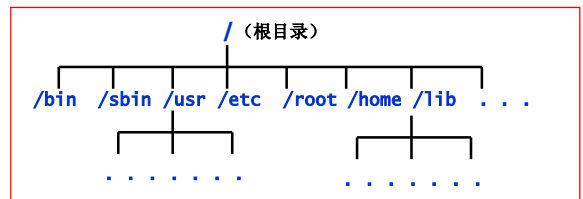
## 常见的文件系统

- u 常见的文件系统类型有
  - ü FAT16、FAT32、NTFS (Windows)
  - ü ext2、ext3、reiserfs、swap (Linux)
  - ü NFS（网络文件系统）
  - ü iso9660（光盘文件系统）
  - ü vfat（扩展的DOS分区，支持长文件名）
- u Linux支持多种文件系统
  - ü Linux目前几乎支持所有的Unix类的文件系统，如HFS、XFS、JFS、Minix fs及UFS等
  - ü Linux支持NFS文件系统
  - ü Linux也支持NTFS和vfat，但不支持NTFS的写操作

## Linux 文件系统结构

### q Linux 目录结构

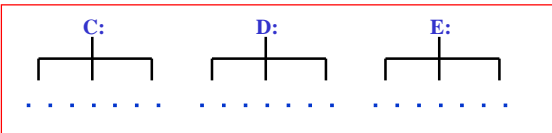
- u Linux文件系统是一个目录树的结构，文件系统结构从一个根目录开始，根目录下可以有任意多个文件和子目录，子目录中又可以有任意多个文件和子目录。
- u Linux的这种文件系统结构使得一个目录和它包含的文件/子目录之间形成一个父子关系。



## Linux 文件系统结构

### q Linux文件系统的结构与Windows的区别

- u Windows以分区为树根，若有多个分区，则形成多个树并列的情形：



- u Linux所有文件系统都安装在一个树根上。进行分区时必须先划分一个根分区，然后将其它的分区都挂载到这个根目录下。

## Linux 文件系统结构

- u Linux文件系统结构称为“树状结构”或“父子结构”。
- u Linux文件结构是以根目录“/”为起点，向下展开。

## 当前工作目录

### q 当前工作目录

- u 用户目前所处的目录
- u 用户登录后进入的目录通常是自己的主目录
- u 可用 `pwd` 命令查看用户的当前目录
- u 可用 `cd` 命令来切换目录
- u 一些特殊字符的特殊含义：
  - ü “.” 表示当前目录
  - ü “..” 表示当前目录的上一级目录（父目录）
  - ü “-” 表示用 `cd` 命令切换目录前所在的目录
  - ü “~” 表示用户主目录的绝对路径名

## 路径 (path)

### q 路径 (path)

- u 文件或目录在文件系统中所处的位置
- u 路径分为“绝对路径”和“相对路径”
- u 绝对路径：是指一个文件或目录从根目录开始的完整的路径。
- u 相对路径：是指一个文件或目录相当于当前工作目录的路径。
- u 任何不以 / 和 ~ 开始的路径均为相对路径。

## 路径 (path)

例：设用户当前所在的目录为

```
/home/jypan/course
```

则该目录的子目录 `linux` 下文件 `lect01.ppt` 的绝对路径为

```
/home/jypan/course/linux/lect01.ppt
```

相对路径为

```
./linux/lect01.ppt
```

## 文件类型与访问权限

- u 文件名：用户可以创建文件名最长为 255 个字符的文件，且可以使用几乎任何字符，但通常使用字母（大小写）、数字、下划线、句号等。

```
例：This_is_my_First_file.txt
```

文件命名时尽量少用特殊字符

- u Linux 下以点开头的文件为隐藏文件。
- u Linux 下的文件类型  
普通文件、目录文件、链接文件、设备文件

## 常见的普通文件

### u 普通文件

- l 压缩与归档文件  
`.gz .tar .tgz .zip .rar` 等
- l 文字与多媒体文件  
`.gif .jpg .png .html .ps .pdf .txt` 等
- l 编程与脚本文件  
`.c .cpp .h .o .f .so .sh` 等
- l 系统文件 `.cong .rpm` 等

注：查看一个文件的类型可以用 `file` 命令

```
file 文件名
```

## Linux 文件类型

### u 普通文件

### u 目录文件

### u 链接文件

- ü 其作用类似于 windows 下的快捷方式，它本身不包含内容，而是指向其它的文件或目录。
- ü 可用 `ln` 命令创建链接文件。

### u 设备文件

存放在 `/dev` 目录下。如：

```
hda、hdb、sda、sdb、fd0、lp1、...
```

## 文件的访问权限

### q 文件和目录的访问权限

- u 每个文件或目录都包含访问权限，这些权限决定了谁能访问和如何访问这些文件和目录。
- u 通过设定权限，可以用以下三种方式来限制访问权限：
  - l 只允许用户自己访问；
  - l 允许一个预先指定的用户组的用户访问；
  - l 允许系统内的所有用户访问；

## 文件的访问权限

- u 一个文件或目录可以有读、写和执行的权限。当一个文件或目录被创建时，系统会自动赋予文件所有者的读和写的权限，这样就允许文件所有者能够阅读和修改该文件。

- u 一个文件或目录可以被三种不同类型的用户访问：所有者、同组用户和其它用户。通常每个用户在创建时都被分配到一个特定的用户组，系统中存在许多不同的用户组。文件所有者可以通过修改文件的属性而授予其所在用户组的其他成员访问该文件的权限，同样也可以将自己的文件向系统内的所有用户开发。

## 文件的访问权限

q 查看文件和目录的属性: `ls -l`

```

Expanded shell output: *
total 32
-rw-r--r-- 1 root root 4096 3月 11:39 2000Data2.zip
-rw-r--r-- 1 root root 4096 3月 11:39 2000Data.zip
-rw-r--r-- 1 root root 4096 3月 11:39 backup
-rw-r--r-- 1 root root 4096 3月 11:39 blu
-rw-r--r-- 1 root root 4096 3月 11:39 Desktop
-rw-r--r-- 1 root root 4096 3月 11:39 Desktop

```

## 文件的访问权限

```

-rwxr-xr-x 3 jvace 100 4196 3月 11:39 blu

```

u 在显示的结果中, 第一个字段中的第一个字符是表示文件的类型, 具体含义为:

字符	代表的文件类型
-	普通文件
d	目录
l	链接文件
b, c	分别代表块设备和其它外围设备, 特殊类型文件
s, p	这些文件类型关系到系统的数据结构和管道, 很少见

## 文件的访问权限

```

-rwxr-xr-x 3 jvace 100 4196 3月 11:39 blu

```

n 在显示的结果中, 第一个字段的第 2~10 个字符是用来表示权限。这 9 个字符每 3 个一组, 组成 3 套权限控制, 第一套控制文件所有者的访问权限, 第二套控制所有者所在用户组的其他成员的访问权限, 第三套是控制系统内其他用户的访问权限。

## 文件的访问权限

```

d rwx r-x r-x

```

n 每套控制中三个字符按顺序排列, 所代表的含义如下:

r	<b>Read</b> (读取): 对文件而言, 具有阅读文件内容的权限, 对目录而言, 具有浏览目录内容的权限。
w	<b>Write</b> (写入): 对文件而言, 具有修改文件内容的权限, 对目录而言, 具有移动、删除目录内文件的权限。
x	<b>eXecute</b> (执行): 对文件而言, 具有执行文件的权限, 对目录而言, 具有进入目录的权限。
-	表示不具有该权限, 或取消该权限

## 访问权限举例

例: `-rwx-----`

## 访问权限举例

- `-rw-r--r--` 普通文件, 用户有读、写权限, 组用户和其他用户只有读权限。
- `drwx--x--x` 目录, 用户有读、写和执行权限, 组用户和其他用户只有执行权限。
- `drwx-----` 目录, 用户有读、写和执行权限, 组用户和其他用户不能访问。
- `-rwxr-xr-x` 普通文件, 用户有读、写和执行权限, 组用户和其他用户有读和执行权限。

每个用户都拥有自己的专属目录(主目录), 通常放置在 `/home` 目录下, 这些专属目录的默认权限通常为 `rwx-----`

## 修改访问权限

q 如何更改已有文件或目录的访问权限?

`chmod` 命令

q `chmod` 有两种用法:

- u 文字设定法  
使用字母和操作符表达式来修改或设定文件的访问权限
- u 数字设定法  
利用数字来设定文件的访问权限

## chmod 文字设定法

u 文字设定法

```

chmod [who] [+|-|=] [mode] 文件或目录名

```

l 操作对象 `who` 可是下述字母中的任一个或者它们的组合:

- u `u` 表示“用户 (`user`)”, 即文件或目录的所有者
- u `g` 表示“同组 (`group`) 用户”
- u `o` 表示“其他 (`others`) 用户”
- u `a` 表示“所有 (`all`) 用户”
- u 如果没有指定操作对象 `who`, 则缺省值为 `a`

## chmod 文字设定法

操作符号可以是：

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

访问权限可用下述字母的任意组合

- úr 可读
- úw 可写
- úx 可执行

文件或目录名

以空格分开的要改变权限的文件列表，支持通配符

## chmod 文字设定法举例

```
jypan@euler:~$ ls
total 700
-rw-r--r-- 1 jypan  cwd  700000  6月  7  13:29 2ndECPde.zip

jypan@euler:~$ chmod o+r 2ndECPde.zip
jypan@euler:~$ ls
total 700
-rw-r--r-- 1 jypan  cwd  700000  6月  7  13:29 2ndECPde.zip

jypan@euler:~$ chmod o+r 2ndECPde.zip
jypan@euler:~$ ls
total 700
-rw-rw-r-- 1 jypan  cwd  700000  6月  7  13:29 2ndECPde.zip
```

在一个命令行中可给出多个权限方式，其间用逗号隔开

例：上面的两此操作可以用下面一个命令完成

```
[jypan@euler jypan]$ chmod g+r,o+r 2ndECPde.zip
```

## chmod 文字设定法举例

```
chmod o-rx mydata/
```

```
chmod o=rwx mydata/
```

```
chmod go=r mydata/
```

```
chmod +x myname.txt
```

等价于：chmod a+x myname.txt

```
chmod +w myname.txt ?
```

等价于：chmod u+w myname.txt

## chmod 数字设定法

数字设定法

```
chmod num 文件或目录名
```

我们必须先了解如何用数字表示的属性：

- 0 表示没有权限
- 1 表示可执行权限
- 2 表示可写权限
- 4 表示可读权限

然后将其相加，即可得到一个表示访问权限的数字。

一个文件的访问权限可以用三个数字来表示，分别代表用户、同组用户和其它用户的访问权限。

## chmod 数字设定法

权限	对应数字
r	4
w	2
x	1
-	0

-rw-r--r-- ←→ 644

drwx--x--x ←→ 711

drwx----- ←→ 700

-rwxr-xr-x ←→ 755

## 数字设定法举例

```
chmod 644 myname.txt
```

设定文件 myname.txt 的属性为：-rw-r--r--

```
chmod 750 myname.txt
```

设定文件 myname.txt 的属性为：-rwxr-x---

```
chmod 700 mydata/
```

设定目录 mydata 的属性为：-rwx-----

## 文件系统的挂载

Linux 下使用 mount 命令挂载文件系统

可在命令行中用 mount 挂载 Linux 支持的任何文件系统，但通常需要有 root 权限。

```
mount [选项] [磁盘或分区的设备名] [挂载点]
```

常用选项

- t 指定载入文件系统的类型(fat32为vfat)
- r 以只读的模式载入设备
- w 以可读写模式载入设备
- a 载入在/etc/fstab中列出的所有设备
- o 可选项

## 文件系统的挂载

常见的磁盘分区设备名有：

- /dev/hda1 (第一个硬盘的第一个分区)
- /dev/hda5、/dev/hda6、...
- /dev/fd0 (软盘)
- /dev/hdc (光盘)
- /dev/sda1 (通常为移动硬盘的第一个分区)

挂载点是一个空目录（若不存在，请先建立一个空目录），mount 后该目录中的内容就是你载入的文件系统的内容。

## 挂载软盘

```
mount /dev/fd0 /media/floppy
```

u 若需更换软盘，必须先卸载原来挂载的软盘，然后取出软盘，放入新软盘，否则出错！

```
umount /dev/fd0 或 umount /media/floppy
```

- u 一个目录只能同时 **mount** 一个文件系统。
- u 挂载点可以是任意非空目录，**fedora** 缺省为 `/media/floppy` 如果该目录不存在，可自己创建一个。

## 挂载光盘

```
mount /dev/hdc /media/cdrom
```

u 若需更换或取出光盘，必须先卸载

```
umount /dev/hdc 或 umount /media/cdrom
```

u 挂载点可以是任意非空目录

```
mount /dev/hdc /home/jypan/mycd
```

u 查看一个光盘镜像文件(即.iso文件)，也需先挂载：

```
mount -o loop ./matlab7.iso ./mycd
```

## 挂载 Linux 分区

```
mount -t ext3 /dev/hda9 /opt
```

u 这里假设 `/dev/hda9` 是 `ext3` 的 Linux 分区

## 挂载 Windows 分区

u 假设 `/dev/hda5` 是 Windows 下的 fat 分区 (通常为 D: 盘)

```
mount -t vfat /dev/hda5 /dosd
```

u 假设 `/dev/hda6` 是 Windows 下的 NTFS 分区 (通常为 E: 盘)

```
mount -t ntfs /dev/hda6 /dose
```

注：挂载 `ntfs` 分区需要下载并安装与你机器上 `linux` 的内核版本相一致的 `linux-ntfs` 模块或 `ntfs-3g`。

## 挂载 NTFS 分区

u 使用 `linux-ntfs` 挂载 NTFS 分区

- l 适用 RedHat Enterprise 3、4、5
- l 查看 `linux` 的内核版本号：`uname -a`
- l 下载相应的 NTFS 的 rpm 包 <http://www.linux-ntfs.org>
- l 安装该模块 (超级用户身份)

```
rpm -Uvh Linux-ntfs的rpm包
```

l 挂载 NTFS 分区

```
mount -t ntfs /dev/hda6 /dose
```

## 挂载 NTFS 分区

u 使用 `ntfs-3g` 挂载 NTFS 分区

```
http://www.ntfs-3g.org/
```

- l 目前较新的 `Linux` 发行版都自带该软件
- l 也可以到软件主页下载并安装
- l 挂载 NTFS 分区

```
mount -t ntfs-3g /dev/hda6 /dose
```

## 显示 Windows 下的中文

q 为了能够在 `Linux` 下正确显示所挂载的文件系统 (特别是 Windows 分区、移动硬盘、U 盘等) 下的中文，挂载时需要加一些选项，如：

```
mount -t vfat /dev/hda5 -o utf8 /dosd
mount -t ntfs /dev/hda6 -o utf8 /dose
```

或

```
mount -t vfat /dev/hda5 -o iocharset=cp936 /dosd
mount -t ntfs /dev/hda6 -o nls=cp936 /dose
```

## 文件系统的卸载

q 使用 `umount` 命令卸载文件系统

```
umount 设备名或挂载点
```

```
umount /dev/hda5
```

或

```
umount /dosd
```

## 文件系统配置文件

### q /etc/fstab

**fstab** (file system table) 是一个纯文本文件，开机后，系统会自动搜索该文件中的内容，对列于该文件中的文件系统进行自动挂载。

q **fstab** 文件有几列组成，列与列之间用 **tab** 键隔开，一般格式如下：

<code>/dev/hda10</code>	<code>/opt</code>	<code>ext3</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
设备名	挂载点	文件系统类型	选项列表	dump时是否记录	fsck时的顺序

## fstab 选项参数

### u fstab 选项列表中的常见可选参数

参数	说明
<b>auto</b>	自动挂载，可用 <code>-a</code> 参数进行挂载
<b>async</b>	对此文件系统的所有 I/O 都必须异步完成
<b>defaults</b>	使用默认参数: <code>rw,suid,exec,auto,nouser,async</code>
<b>exec</b>	允许执行二进制程序
<b>user</b>	允许普通用户挂载此文件系统
<b>suid</b>	允许设置用户标识符或设置用户组标识符
<b>ro/rw</b>	read only/readable and writable
<code>noauto, sync, noexec, nouser, nosuid</code>	

## 一个 fstab 实例

```
# This file is edited by fstab-sync -- see 'man fstab-sync' for details
LABEL=/1 / ext3 defaults 1 1
none /dev/pts devpts gid=5,mode=620 0 0
none /dev/shm tmpfs defaults 0 0
LABEL=/home1 /home ext3 defaults 1 2
none /proc proc defaults 0 0
none /sys sysfs defaults 0 0
LABEL=SWAP swap swap defaults 0 0
#
/dev/hda2 /winnt ntfs noauto,ni s=cp936,ro,users 0 0
/dev/hda5 /dosd ntfs noauto,ni s=cp936,ro,users 0 0
/dev/hda6 /dose vfat noauto,i ocharset=cp936,rw 0 0
/dev/sda1 /usb1 vfat noauto,i ocharset=cp936,rw 0 0
```

## umask

q 创建新文件或新目录时，系统都会为它们指定默认（缺省）的访问权限，这个缺省的访问权限就由 **umask** 值来决定。

q **umask** 值是通过 **umask** 命令设置的

q 系统不允许用户在创建一个普通文件时就赋予它可执行权限，必须在创建后用 **chmod** 修改。目录则允许设定可执行权限。

q 查询当前 **umask** 的值: `umask`

q 修改当前 **umask** 的值: `umask 022`

## 常用 umask 值

### u 常见的umask值与新建文件/目录访问权限对应表

umask值	新建目录的访问权限	新建文件的访问权限
022	755	644
027	750	640
002	775	664
006	771	660
007	770	660

例: 可以在 `/etc/fstab` 的选项中加入 **umask** 值

```
/dev/hda10 /home ext3 noauto,umask=022,ro,users 0 0
```

## df、du、uname 和 cut

u **df**: 查看磁盘的使用情况以及文件系统被挂载的位置。

```
df -h
```

u **du**: 查看指定文件或目录的大小。

```
du -sh 目录名
```

u **uname**: 查看系统内核版本。

```
uname -a
```

u **cut**: `cut -c n1-n2 filename`

显示每行从开头算起第 **n1** 到 **n2** 个字符。

## fdisk 和 wc

u **fdisk**: Linux 下的分区工具。

常见用法: `fdisk -l` (需 **root** 权限)

u **wc**: 统计指定文件的行数、字数和字节数。

例: `wc filename`

```
wc -c filename (只显示字节数)
wc -w filename (只显示字数)
wc -l filename (只显示行数)
```

## head 和 tail

u **head**: 查看文件的开头部分，默认是开头 10 行。

```
head myname.txt (查看 myname.txt 的前 10 行)
head -20 myname.txt (查看前 20 行)
```

u **tail**: 查看文件的结尾部分，默认是最后 10 行。

```
tail myname.txt (查看 myname.txt 的最后 10 行)
tail -20 myname.txt (查看最后 20 行)
tail -f myname.txt
(实时动态查看最后 10 行，常用于跟踪日志文件等)
```