



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象  
华清远见的企业理念是不仅要良心教育、做专业教育，更要做受人尊敬的职业教育。

# 《Linux 指令从初学到精通》

作者：华清远见

专业始于专注 卓识源于远见

## 第 3 章 文件管理

---

本章目标

---

UNIX 开发者 Ken Thompson 曾经说过，“在 UNIX/Linux 的世界里一切皆是文件”。这句话突出了 Linux 模式的简单性。对于 Linux 系统来说一切都是文件，控制台是一个文件，磁盘、光碟刻录机、调制解调器也都是文件。因此，对于 Linux 管理用户来说，熟练掌握文件管理命令是必需的。文件管理命令包括文件的创建、查看、搜索、修改，文件属性的查看、修改，以及对文件进行备份、整理、压缩等。

专业始于专注 卓识源于远见



# 3.1 aspell ( 检查文件的错误 )

(1) 频度等级: ☆☆

(2) 功能说明:

aspell 命令用于检查并修改文件中单词的拼写错误, 该命令只能检查或修改英文单词的错误。

(3) 语法:

```
aspell [选项] <命令>
```

(4) 所属包:

```
aspell
```

(5) 使用技巧:

aspell 命令不但能检查拼写错误, 还能纠正拼写错误。

需要使用 “--lang=<str>” 指定语言, 例如使用 “--lang=en” 指定检查英文拼写。

(6) 相关命令:

```
ispell、spell
```

(7) 参数:

aspell 的主要参数如表 3-1 所示。

表 3-1 aspell 参数表

参 数	含 义	参 数	含 义
选项		命令	
-? usage	显示简单的语法信息	--add   rem-dict-alias=<str>	建立目录别名
help	显示详细的帮助信息	--dict-dir=<str>	指定本地的主单词列表
-c   check <file>	检查指定文件中的拼写错误	--add   rem-filter=<str>	增加或删除一个过滤器
-a   pipe	采用兼容模式	--add   rem-filter-path=<str>	指定 aspell 查找过滤器的路径
list	从标准输入产生一个拼写错误的单词列表	--mode=<str>	指定过滤器模式
config <key>	打印当前某一个选项的值	-e,--mode=email	指定 E-mail 模式
[dump] dicts   filters   modes	列出有效的词典/过滤器/过滤器模式	-H,--mode=html	使用 HTML 模式
munch	产生可能的词根和词缀	-t,--mode=tex	指定文档模式
expand [1-4]	扩展词缀标志	-n,--mode=nroff	设置 nroff 文件模式
clean	清理单词列表使得每一行都成为一个有效的单词	-d,--lang=<str>	使用语言编码
-v   version	显示版本信息	-p,--personal=<str>	指定个人词典文件名
命令		--prefix=<str>	指定前缀目录
--conf=<str>	指定主配置文件	--repl=<str>	设置代替表文件名
--conf-dir=<str>	指定本地的主配置文件	--size=<str>	设置单词列表的大小
--data-dir=<str>	使用本地的语言数据库		

(8) 补充说明:

aspell 默认会使用 “/usr/lib/aspell/english.hash” 字典文件来检查文本文件。若在检查的文件中找到字典没有的词汇, aspell 会建议使用的词汇, 或是让用户将新的词汇加入到个人字典中。

(9) 典型实例:

例: 检查文件拼写错误。例如检查文件 test 中的英文单词的拼写错误, 可使用如下命令:

```
Aspell--lang=en--c test #检查 testfile 文件中英文单词的拼写错误, --lang=en 指定为英语, #-c 表示检查指定文件中是否存在拼写错误
```

得到如图 3-1 所示的结果。



图 3-1

此时，等待用户输入选择项，这里的可选择项有两组：数字组和字母组。数字组是参考修改选项，用于修改检查出的错误单词；字母组表示其他类型的操作，如忽略当前、忽略所有、取代、添加等操作。例如在此输入数字“1”，表示将该处错误的单词改为“around”，并跳转到下一个检查到的错误单词处，直至该文件检查完毕。如果此时选择字母“i”，则忽略当前错误的单词。

## 3.2 ar ( 建立、修改库或从档案文件中提取文件 )

(1) 频度等级：☆☆

(2) 功能说明：

ar 命令可以用来创建、修改库，也可以从库中提出单个模块。库是单独的文件，里面包含了按照特定的结构组织起来的其他一些文件（称做此库文件的 member）。原始文件的内容、模式、时间戳、属主、组等属性都保留在库文件中。

(3) 语法：

```
ar [-dmpqrtx] [cfosSuvV] [a<成员文件>] [b<成员文件>] [i<成员文件>] [档案文件] [成员文件]
```

(4) 所属包：

binutils

(5) 使用技巧：

ar 命令可把多个文件合并成一个归档文件进行备份，该归档文件中的所有文件的属性及权限保持不变。

ar 命令可在 C 语言开发过程中将目标文件打包成库文件。

(6) 相关命令：

nm

(7) 参数：

ar 的主要参数如表 3-2 所示。

表 3-2 ar 参数表

参 数	含 义	参 数	含 义		
指令参数	-d	删除档案文件中的成员文件	可选参数	c	建立档案文件
	-m	变更成员文件在档案文件中的次序		f	为避免过长的文件名不兼容其他系统的 ar 指令，可利用此参数，截掉要放入档案文件中过长的成员文件名称
	-p	显示档案文件中的成员文件内容，如果未指定 [成员文件] 参数，则显示档案文件中所有成员文件的内容		i<成员文件>	将文件插入档案文件中指定的成员文件之前（与 b 的作用相同）
	-q	将文件附加在档案文件末端		o	保留档案文件中文件的日期
	-r	将文件插入档案文件中		s	若档案文件中包含了对象模式，则可利

				用此参数建立档案文件的符号表。
	-t	显示档案文件中所包含的文件	S	不产生符号表
	-x	从档案文件中取出成员文件	u	只将日期较新的文件插入档案文件中
可选参数	a<成员文件>	将文件插入档案文件中指定的成员文件之后	v	程序执行时显示详细的信息
	b<成员文件>	将文件插入档案文件中指定的成员文件之前	V	显示版本信息

(8) 典型实例:

例 1: 使用 ar 命令将 testfile、testfile\_1、testfile\_2 三个文件加入到 testfile\_a 归档文件, 该文件本身并不存在, 使用 ar 命令新建。此时使用的命令为:

```
Ar-rv testfile_a testfile testfile_1 testfile_2
```

输出结果如下:

```
cmd@cmd-desktop:~$ ar -rv testfile_a testfile testfile_1 testfile_2 #建立归档文件
ar: creating testfile_a #生成的归档文件
a - testfile
a - testfile_1
a - testfile_2
```

例 2: ar 命令还可以用于查看成员文件的内容, 例如使用如下命令:

```
Ar-p testfile_a
```

由于为指定成员文件, 所以将显示全部成员文件的内容, 结果如下:

```
cmd@cmd-desktop:~$ ar -p testfile_a
#以下为第一个文件 testfile 的内容
Linux networks are becoming more and more common, but security is often an overlooked issue.
Unfortunately, in today's environment all networks are potential hacker targets, from top-secret
military research networks to small home LANs.
Linux Network Security focuses on securing Linux in a networked environment, where the security
of the entire network needs to be considered rather than just isolated machines. It uses a mix
of theory and practical techniques to teach administrators how to install and use security applications,
as well as how the applications work and why they are necessary.
#以下为第二个文件 testfile_1 的内容
test 30
test 30
test 30
Hello 95
Hello 95
Hello 95
Hello 95
Linux 85
Linux 85
#以下为第三个文件 testfile_2 的内容
Hello 2005
Linux 2009
test 2006
```

## 3.3 attr (设置 XFS 文件系统对象的扩展属性)

(1) 频度等级: ☆

(2) 功能说明:

设置 XFS 文件系统对象的扩展属性, XFS 包含一个名为“扩展属性”的优秀功能特性。扩展属性使用户定义的数据同文件系统对象相关联。例如, 用户如果有名为 mygraphic.png 的图像, 可以附加一个名为“thumbnail”的属性, 它包含该图像的一个小版本。普通文件 I/O 操作将看不到这个数据, 但任何程序都可以使用一个特殊的扩展属性 API 来访问它。扩展属性在某些方面类似 MacOS 系统中的“资源分支”。使用该命令可对 XFS 文件系统的扩展属性进行操作。

(3) 语法:

```
attr [-glLqrRs][-V attrvalue]
```

(4) 所属包:

```
attr
```

(5) 使用技巧:

当指定参数-s 时, 可以从标准输入读取一个属性值; 当设定参数-g 时, 可以读取一个属性值到标准输出。

(6) 相关命令:

```
ls
```

(7) 参数:

attr 主要参数如表 3-3 所示。

表 3-3 attr 参数表

参 数	含 义
-g	得到文件的属性值
-l	显示文件的属性列表
-L	若指定-L 参数, 并且对象是一个符号链接, 那么操作的是引用符号链接的对象的属性。若不指定这个参数, 那么操作的是符号链接本身
-q	不打印状态信息
-r	删除文件的属性值
-R	若指定-R 值, 并且处理过程用于适当的特权, 操作将发生在 root 用户的属性空间, 而不是一般用户的属性空间
-s	设置文件的属性值
-V attrvalue	可选选项, 设置属性值

(8) 典型实例:

例: 显示 XFS 文件系统的文件属性。

例如, 在进行 Linux 分区时, 可将某一分区格式化为 XFS 文件系统, 如将/tmp 挂载在 XFS 文件系统上。在命令行使用如下命令:

```
Attr-l /tmp/ #显示 XFS 文件系统的文件属性
```

该命令有如下输出结果:

```
cmd@cmd-desktop:~$cd /tmp #切换到/tmp 目录
cmd@cmd-desktop:~$ attr-l /tmp/ #显示 XFS 文件系统的文件属性
Attribute "selinux" has a 27 byte value for /tmp/
```

## 3.4 basename (显示目录或文件的基本名称)

(1) 频度等级: ☆

(2) 功能说明:

用于显示去掉路径之后纯粹的文件名。如果指定了文件扩展名, 则将文件扩展名也去掉。

(3) 语法:

```
basename [参数][文件]
```

(4) 所属包:

```
basename
```

(5) 使用技巧:

basename 命令在进行 shell 脚本程序设计时非常有用, 可以利用它得到去掉目录和扩展名之后纯粹的文件名, 以便在脚本程序中操作文件。

(6) 参数:

basename 的主要参数如表 3-4 所示。

表 3-4 basename 参数表

参 数	含 义
--help	显示帮助信息
--version	显示版本信息

(7) 典型实例:

例: 从文件名中去掉路径信息和扩展名, 只显示纯粹的文件名。例如去掉“/etc/apt/source.list”文件的路径信息和扩展名, 可使用如下命令:

```
basename /etc/apt/source.list #去掉路径信息和扩展名
```

该命令的输出结果如下:

```
cmd@cmd-desktop:~$ basename /etc/apt/source.list #去掉路径信息和扩展名
source #显示去掉路径信息和扩展名的文件名
```

## 3.5 bunzip2 (解压缩 bz2 文件)



(1) 频度等级: ☆☆

(2) 功能说明:

**bunzip2** 命令用来解压缩通过 **bzip2** 命令压缩过的、后缀名为 **.bz2** 的压缩文件。

(3) 语法:

```
bunzip2 [-fkLsvV][.bz2 压缩文件]
```

(4) 所属包:

```
bzip2
```

(5) 使用技巧:

使用该命令解压缩文件时, 如果系统内存不足, 则可以使用参数“-s”来降低程序运行时的内存使用量。

(6) 相关命令:

```
bzip2、bziprecover
```

(7) 参数:

**bunzip2** 的主要参数如表 3-3 所示。

表 3-5 bunzip2 参数表

参 数	含 义
-f 或--force	解压缩时, 若输出的文件与现有文件同名, 预设就不会覆盖现有的文件; 若要覆盖, 则需要使用此参数
-k 或--keep	在解压缩后, 预设会删除原来的压缩文件; 若要保留压缩文件, 则需要使用此参数
-s 或--small	降低程序执行时内存的使用量
-v 或--verbose	解压缩文件时, 显示详细的信息
-l,--license,-V 或--version	显示版本信息

(8) 补充说明:

**bunzip2** 实际上是 **bzip2** 的符号链接, **bunzip2** 与“**bzip2-d**”的执行效果相同。

(9) 使用实例:

例: 将 **testfile.bz2** 文件解压缩, 不删除原来的压缩文件, 同时显示详细信息, 可使用如下命令:

```
bunzip2-kv testfile.bz2 #解压缩打包文件
```

为了便于对比, 在解压缩前使用 **ls** 命令查看当前目录:

```
cmd@cmd-desktop:~$ ls #查看原来的目录结构
testfile.bz2
```

使用 **bunzip2** 命令之后, 当前目录中的文件为:

```
cmd@cmd-desktop:~$ bunzip2-vk testfile.bz2 #参数 v 表示显示详细信息, 参数 k 表示不删除原压缩文件
cmd@hdd-desktop:ls #查看解压后的文件目录
testfile testfile.bz2 #testfile 为解压后的文件
```





## 3.6 bzip2 ( 创建 bz2 格式的压缩文件 )

(1) 频度等级: ☆☆

(2) 功能说明:

bzip2 命令使用“Burrows-Wheeler block sorting text”压缩算法对文件进行压缩, 压缩后的后缀名为“.bz2”, 这种算法比一般压缩算法的压缩比率高。

(3) 语法:

```
bzip2 [-cdfhkLstVz][--repetitive-best][--repetitive-fast][-压缩等级][要压缩的文件]
```

(4) 所属包:

```
bzip2
```

(5) 使用技巧:

使用该命令解压缩文件时, 如果系统内存不足, 则可以使用参数“-s”来降低程序运行时的内存使用量。

使用参数“-d”可以对压缩过的.bz2 文件进行解压缩。

(6) 相关命令:

```
bunzip2、bzip2recover
```

(7) 参数:

bzip2 的主要参数如表 3-6 所示。

表 3-6 bzip2 参数表

参 数	含 义
-c 或--stdout	将压缩与解压缩的结果传送到标准输出
-d 或--decompress	执行解压缩
-f 或--force	bzip2 在压缩或解压缩时, 若输出文件与现有文件同名, 则预设不会覆盖现有文件。若要覆盖, 就需要使用此参数
-k 或--keep	bzip2 在压缩或解压缩后, 会删除原始的文件。若要保留原始文件, 就需要使用此参
-s 或--small	降低程序运行时内存的使用量
-t 或--test	测试.bz2 压缩文件的完整性
-v 或--verbose	压缩或解压缩文件时, 显示详细的信息
-z 或--compress	强制执行压缩
-L,--license, -V 或--version	显示版本信息
--repetitive-best	当文件中有重复出现的资料时, 可利用此参数提高压缩效果
--repetitive-fast	当文件中有重复出现的资料时, 可利用此参数加快执行速度
-压缩等级 (123456789)	指定压缩比率等级, 数值越大压缩率越大

(8) 补充说明:

bzip2 采用新的压缩演算法, 压缩效果比传统的 LZ77/LZ78 压缩演算法好。若没有加上任何参数, bzip2 压缩完文件后会产生 bz2 的压缩文件, 并删除原始的文件。

(9) 典型实例:

例 1: 把要压缩的文件 testfile 传递给 bzip2 命令, 完成压缩, bzip2 命令将文件压缩并添加“.bz2”后缀, 再删除源文件, 使用的命令为:

```
bzip2 testfile #使用 bzip2 命令压缩文件
```

此命令并没有任何的信息输出, 源文件 testfile 在压缩文件 testfile.bz2 生成后自动删除, 可以使用 ls 命令查看压缩前后的变化。例如, 在使用 bzip2 命令压缩前, 使用如下命令:

```
ls
```

得到以下输出结果:

```
cmd@cmd-desktop:~$ ls #使用 ls 前查看当前目录
testflie #待压缩文件
```

使用 bzip2 压缩后, 再使用 ls 命令查看如下所示。

```
cmd@cmd-desktop:~$ bzip2 testfile #使用 bzip2 命令压缩文件
```

```
cmd@cmd-desktop:~$ ls                #压缩后查看生成文件
testflie.bz2                          #生成后缀为.bz2的打包文件
```

可见，原来的 testfile 文件已经自动删除，生成的文件后缀名为 “.bz2”。

例 2：使用 bzip2 压缩多个文件。例如当前目录下有 testfile\_1、testfile\_2 两个文件，使用 bzip2 压缩文件，然后再解压缩生成的两个压缩文件。

为了对比，首先使用 ls 命令查看当前目录：

```
cmd@cmd-desktop:~$ ls                #使用ls前查看当前目录
testflie_1 testfile_2                #两个待压缩文件
```

使用 bzip2 压缩这两个文件，在当前目录输入如下命令：

```
bzip2 testfile_1 testfile_2          #使用bzip2命令压缩文件
```

同上例，再使用 ls 命令查看生成的压缩文件：

```
cmd@cmd-desktop:~$ ls                #压缩后查看生成文件
testflie_1.bz2 testfile_2.bz2        #生成两个后缀为.bz2的打包文件
```

接下来是使用“-d”参数解压缩文件，使用命令如下：

```
bzip2-d testfile_1.bz2 testfile_2.bz2 #解压缩打包文件
cmd@cmd-desktop:~$ ls                #压缩后查看生成文件
testflie_1 testfile_2                #生成两个解压缩文件
```

例 3：指定压缩率等级。例如先以默认的方式压缩 testfile 文件，再以压缩等级 1 压缩同一个文件，对比这两次压缩。

先以默认方式压缩，使用如下命令：

```
cmd@cmd-desktop:~$ bzip2 -v -k testfile
#“-v”表示显示详细信息，“-k”表示保留原始文件
testfile:1.537:1,5.204 bits/byte, 34.95% saved, 598 in, 389 out.
#其中：1.537:1表示压缩空间前后比；5.204表示每个字节将压缩成5.204个位；34.95%表示节省的空间比
#率；598 in表示压缩前文件的大小；389 out表示压缩后文件的大小
cmd@cmd-desktop:~$ bzip2 -v -1 -f testfile
#“-v”表示显示详细信息，“-f”表示覆盖原始文件，“-1”指定压缩等级1
testfile: 1.023:1,7.584 bits/byte, 25.36% saved, 598 in, 506 out.
#各个参数意义同上
```

可见，由于指定了较小的压缩等级，所以压缩空间较先前小。

## 3.7 bzip2recover (修复损坏的 bzip2 压缩文件)

(1) 频度等级：☆

(2) 功能说明：

bzip2recover 命令用来修复损坏的、后缀名为.bz2 的压缩文件。

(3) 语法：

```
bzip2recover [.bz2 压缩文件]
```

(4) 所属包：

```
bzip2
```

(5) 使用技巧：

bzip2recover 命令通常用于修复大的压缩文件。bzip2 命令以区块的方式压缩文件，每个区块都被看做独立的单元，所以，当某一区块损坏时，可利用 bzip2recover 命令尝试将文件中的对应区块隔离，这样就可以解压缩正常的区块。因此，对于很小的压缩文件，如果损坏，很难恢复。

(6) 相关命令：

```
bunzip2、bzip2
```

(7) 补充说明：

通常只适用在压缩文件很大的情况。



(8) 典型实例:

例: 直接将损坏的 bz2 文件传递给 bzip2recover 命令, 该命令将尝试修复此压缩文件。使用的命令如下:

```
bzip2recover testfile_bad.bz2 #修复损坏的“.bz2”文件
```

### 提示

此命令没有任何输出信息。



## 3.8 chattr (改变文件属性)

(1) 频度等级: ☆

(2) 功能说明:

chattr 命令用于改变文件的第 2 个扩展文件系统属性。

(3) 语法:

```
chattr [-RV][<版本编号>][+/-/=<属性>][文件或目录]
```

(4) 所属包:

e2fsprogs

(5) 使用技巧:

chattr 命令所修改的文件属性和 chmod 命令修改的文件属性是两个不同层次的, 前者是基于底层文件系统来设定的, 而 chmod 命令则是从用户的角度来设定的。

(6) 相关命令:

lsattr、mattrrib

(7) 参数:

chattr 的主要参数如表 3-7 所示。

表 3-7 chattr 参数表

参 数	含 义	参 数	含 义
-R	递归处理, 将指定目录下的所有文件及子目录一并处理	+<属性>	开启文件或目录的该项属性
-v <版本编号>	设置文件或目录版本	-<属性>	关闭文件或目录的该项属性
-V	显示指令执行过程	=<属性>	指定文件或目录的该项属性

(8) 补充说明:

这项指令可改变存放在 ext2 文件系统上的文件或目录属性, 这些属性共有以下 8 种模式, 如表 3-8 所示。

表 3-8 文件属性模式表

模 式	功能特点
a	Append only, 让文件或目录仅供附加用途。此项属性必须具有管理员权限才能设置或清除, 使文件或目录只能从附加模式中修改, 而不能直接编辑写入
A	Access time, 不更新文件或目录的最后存取时间。设置该属性, 可以减少硬盘访问次数, 对于笔记本电脑, 还能节省电能消耗
c	Compress, 将文件或目录压缩后存放。遇到设有该属性的文件或目录时, 系统内核会自动在写入数据时, 先把数据压缩, 然后存放在到硬盘, 读取数据时再自动解压缩文件
d	No dump, 将文件或目录排除在转存操作之外。转存操作是一种针对文件系统的全面性备份, 若文件或目录具备这个属性, 进行转存操作时, 则不会备份该文件或目录
i	Immutable, 不得任意更改文件或目录。此属性必须具有管理员权限才能操作, 文件或目录设置了该属性, 将无法删除或改名、编辑, 甚至无法链接这类文件或目录
s	Secure Delete, 保密性删除文件或目录。在删除设置了此类属性的文件或目录之前, 系统会用“00”(十六进制字符)字符填满原本存放文件或目录的块, 彻底清除数据, 不能恢复
S	Sync, 即时更新文件或目录。当硬盘写入数据时, 许多操作系统都会先把数据存入缓存, 避免频繁读写硬盘, 以提升整体性能。如果对文件或目录设置此属性, 则系统将不通过缓存, 直接写入目录或文件
u	Undelete, 预防意外删除。设置该属性, 其内容将会备份保存, 以备用户需要时恢复

(9) 典型实例:

例: 修改指定文件或目录的属性。

例如, 修改文件 testfile 的属性, 为其添加 a 属性, 可使用如下命令:

```
chattr +a testfile #为 testfile 文件添加 a 属性
```

为了对比, 在使用 chattr 命令之前, 可以使用 lsattr 命令查看原有的属性, 结果如下:

```
cmd@cmd-desktop:~$lsattr testfile #查看 testfile 属性
----- testfile
```

输出结果中, 属性都是减号, 表示还没有对此文件设置任何的 ext2 文件系统属性。然后使用 chattr 命令为 testfile 文件添加 a 属性, 再次查看文件属性, 结果如下:

```
cmd@cmd-desktop:~$chattr +a testfile #为 testfile 文件添加 a 属性, 此时无输出
cmd@cmd-desktop:~$lsattr testfile #再次查看 testfile 属性
----a----- testfile #testfile 文件属性已经修改
```

## 3.9 chgrp (更改文件或目录的所属群组)



(1) 频度等级: ☆☆

(2) 功能说明:

chgrp 命令用于变更文件或目录的所属群组, 设置方式采用群组名称或群组识别码。

(3) 语法:

```
chgrp [-cfhRv][--help][--version][所属群组][文件或目录...]
chgrp [-cfhRv][--help][--reference=<参考文件或目录>][--version][文件或目录...]
```

(4) 所属包:

chown

(5) 使用技巧:

如果大量文件需要改变所有权, 而这些文件又在同一目录下, 则可使用参数“-R”一次性完成修改; 如果修改的文件名有一定的规律, 则可以借助通配符来简化操作。

组名可以使用组 ID 代替。

指定的组用户必须是系统中存在的。

递归处理一个目录解引用(--dereference-R)的时候, 具体指明如何(-H 或-L)对遇见的链接文件处理。

目标文件或目录必须存在。引用的文件或目录也必须存在。

(6) 相关命令:

chown

(7) 参数:

chgrp 的主要参数如表 3-9 所示。

表 3-9 chgrp 参数表

参 数	含 义
-c 或--changes	效果类似“-v”参数, 但仅回报更改的部分
-f 或--quiet 或--silent	不显示错误信息
-h 或--no-dereference	只对符号链接的文件进行修改, 而不更改其他任何相关文件
-R 或--recursive	递归处理, 将指定目录下的所有文件及子目录一并处理
-v 或--verbose	显示指令执行过程
--reference=<参考文件或目录>	把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同
--help	显示帮助信息
--version	显示版本信息

(8) 补充说明:

在 UNIX 系统家族里, 文件或目录权限的掌控由拥有者及所属群组来管理。用户可以使用 chgrp 指令去变更文件与目录的所属群组。

(9) 典型实例:

例: 使用 `chgrp` 命令改变文件所属组属性。

例如修改文件 `testfile` 的文件属性, 使其所属组变为“root”组, 可以使用命令为:

```
chgrp -v 0 testfile #其中-v是显示详细的执行过程, 0是所属组ID为0, 即 root 组
```

使用该命令前, 先使用 `ls` 命令查看文件原有的属性:

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
```

```
-rw-r--r-- 1 cmd cmd 598 2009-09-01 13:57 testfile
```

#从左至右依次表示文件类型及权限、链接数、所有者、所属组、大小、最后访问时间、文件名

使用 `chgrp` 命令之后, 再次查看属性, 结果如下:

```
cmd@cmd-desktop:~$chgrp -v 0 testfile #改变 testfile 文件属性
```

```
changed group of `testfile' to 0 #改变的详细过程
```

```
cmd@cmd-desktop:~$ls-l testfile #查看改变后的属性
```

```
-rw-r--r-- 1 cmd root 598 2009-09-01 14:00 testfile #属性结果, 所属组改为 root
```

从上面可以看出, 文件“testfile”所属的组已经被改成“root”组, 通常不使用“-v”参数, 这里是为了让读者能详细地了解修改的过程, 实际应用中读者可以根据情况决定是否使用。



## 3.10 chmod (更改文件或目录的访问权限)

(1) 频度等级: ☆☆☆

(2) 功能说明:

`chmod` 命令用于变更文件或目录的访问权限, Linux 系统提供了字符方式和八进制方式改变文件的访问权限。

(3) 语法:

```
chmod [-cfRv][--help][--version][<权限范围>+/-/=<权限设置...>][文件或目录...]
```

```
chmod [-cfRv][--help][--version][数字代号][文件或目录...]
```

```
chmod [-cfRv][--help][--reference=<参考文件或目录>][--version][文件或目录...]
```

(4) 所属包:

```
coreutils
```

(5) 使用技巧:

`chmod` 命令不能改变符号链接的权限, 当使用该命令改变符号链接的权限时, 实际改变的是符号链接所指向的文件的权限。

如果使用该命令的参数“-R”进行递归操作, `chmod` 命令将忽略所遇到的符号链接。

root 用户可以对具有执行权限(x)的文件设置 `suid` 权限, 此时运行此文件的用户将临时具有超级用户的权限。这种方式使权限的设置更加灵活, 在许多情况下, 这也被认为是一个安全隐患。另外, 普通用户是不能给 root 用户的文件设置 `suid` 权限的。

`suid` 和 `sgid` 权限仅对可执行文件有作用, 对于不可执行文件是没有意义的, 在使用“`ls-l`”命令显示不可执行文件的 `suid` 和 `sgid` 权限时, 对应的权限位显示为大写的“S”。

当权限模式采用 3 个数字表示时, `suid` 和 `sgid` 权限将被忽略, 当使用 4 个数字表示权限模式时, 第一个数字表示的是特殊权限位。

目录的读权限表示可以用 `ls` 命令列出目录内容, 目录的执行权限表示可以利用 `cd` 命令进入目录, 目录的写权限表示可以在目录下创建文件或新的子目录。

(6) 相关命令:

```
Chown、chgrp
```

(7) 参数:

`chmod` 的主要参数如表 3-10 所示。

表 3-10 chmod 参数表

参 数	含 义
-c 或--changes	效果类似“-v”参数，但只回报更改的部分
-f 或--quiet 或--silent	不显示错误信息
-R 或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-v 或--verbose	显示指令执行过程
--reference=<参考文件或目录>	把指定文件或目录的权限全部设成和参考文件或目录的权限相同
<权限范围>+<权限设置>	开启权限范围的文件或目录的该项权限设置
<权限范围>-<权限设置>	关闭权限范围的文件或目录的该项权限设置
<权限范围>=<权限设置>	指定权限范围的文件或目录的该项权限设置
--help	显示帮助信息
--version	显示版本信息

#### (8) 补充说明:

在 UNIX 系统家族里，文件或目录权限的控制分别以读取、写入、执行 3 种一般权限来区分，还有 3 种特殊权限可供运用，拥有者与所属群组也具有管理权限。用户可使用 `chmod` 指令去变更文件与目录的权限，设置方式采用文字或数字代号。符号链接权限无法变更，如果用户对符号链接修改权限，其改变会作用在被链接的原始文件。

权限范围的表示法如表 3-11 所示。

表 3-11 权限范围表示法列表

选 项	功 能
u	User, 即文件或目录的拥有者
g	Group, 即文件或目录的所属群组
o	Other, 除了文件或目录拥有者或所属群组之外, 其他用户皆属于这个范围
a	All, 即全部的用户, 包含拥有者、所属群组, 以及其他用户
[+-=]	“+”表示增加权限, “-”表示取消权限, “=”表示设定唯一权限

有关权限代号的部分如表 3-12 所示。

表 3-12 权限代号列表

选 项	功 能
r	读取权限, 数字代号为“4”
w	写入权限, 数字代号为“2”
x	执行或切换权限, 数字代号为“1”
X	表示该文件是个子目录或者该文件已经被设定为可执行
s	设定 suid 和 sgid 权限, 只能和“u”、“g”连用, 例如“u+s”“g-s”
t	粘附位。在 POSIX 标准中没有提及粘附位, 它最初是指可执行文件在第一次运行时, 在交换空间上保留程序文件副本, 以备下次运行时提高运行速度, 现在已经很少使用。如果目录设置了粘附位, 那么只有文件和目录的所有者可以删除该目录下的文件。粘附位一般适用于类似于“/tmp”这样有基本写权限的目录。如果文件或目录没有可执行权限, 则粘附位设置后将在对应位置显示“T”, 这将使粘附位失去其功能
u	设置与文件拥有者具有相同的权限
g	设置与文件所属组具有相同的权限
o	设置与其他用户具有相同的权限
-	不具有任何权限, 数字代号为“0”

数字标记法表示的权限模式为 4 个八进制数，每个数由位权位 4、2、1 的 3 个八进制数相加得到，如果对应的数字被省略，则将在在此位置的数字默认设置为 0。数字所代表的权限的含义：0 表示没有权限；在第一个八进制数中 1 表示粘附位，2 表示 sgid 权限，4 表示 suid 权限，这 3 个数字相加所得到的数字即最终权限。第 2-4 个八进制中 1 表示可执行权限，2 表示可写权限，4 表示可读权限，这 3 个数字相加得到的数字即最终权限。第 2 位数字代表文件所有者 (u) 的权限。第 3 位数字表示文件所属组的用户 (g) 的权限。第 4 位数字代表其他所有用户 (o) 的权限。

#### (9) 典型实例:

例 1: 使用“+”、“-”修改文件的属性。

例如想修改文件 `testfile` 的属性，增加所有者执行权限，组内增加写权限，其他用户减去读权限，可以使用如下命令：

```
chmod u+x,g+w,o-r testfile #增加所有者执行权限, 组内增加写权限, 其他用户减去读权限
```

为了使读者更加清楚，先是要通过 `ls` 命令查看原始属性，`testfile` 的原始属性如下：



```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rw-r--r-- 1 cmd cmd 598 2009-09-01 14:57 testfile #所有者可读、可写，组内可读，其他用户可读
然后使用 chmod 命令修改文件属性，之后再查看属性，便于对比，结果如下：
```

```
cmd@cmd-desktop:~$chmod u+x,g+w,o-r testfile #增加所有者执行权限，组内增加写权限，#其他用户减
去读权限
```

```
cmd@cmd-desktop:~$ls-l testfile #查看修改后的属性
-rwxrw---- 1 cmd cmd 598 2009-09-01 15:00 testfile #所有者可读、可写、可执行，组内可读、#可写，
其他用户不可读、不可写、不可执行
```

例 2：为可执行文件加上特殊权限位 `suid`。

例如为文件 `testfile` 添加特殊权限位，可使用如下命令：

```
chmod u+s testfile #为文件 testfile 添加特殊权限位
```

同上，使用该命令前后调用 `ls` 命令查看属性变化，得到结果如下：

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rwxr--r-- 1 cmd cmd 598 2009-09-01 15:03 testfile #所有者可读、可写可执行，组内可读，其#他用户
可读
```

```
cmd@cmd-desktop:~$ chmod u+s testfile #为文件 testfile 添加特殊权限位
```

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rwsr--r-- 1 cmd cmd 598 2009-09-01 15:20 testfile #所有者具有可读、可写特殊权限，组内
#可读，其他用户可读
```

例 3：使用数字方式修改特殊权限位 `suid`、`sgid`，以及粘附位。

例如为文件 `testfile` 添加特殊权限位 `suid`、`sgid`，以及粘附位，可使用如下命令：

```
chmod 7755 testfile #为文件 testfile 添加特殊权限位 suid、sgid，以及粘附位
```

同上，使用该命令前后调用 `ls` 命令查看属性变化，得到结果如下：

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rwxr-xr-- 1 cmd cmd 598 2009-09-01 15:03 testfile #所有者可读、可写，组内可读、可执行，#其他用
户可读
```

```
cmd@cmd-desktop:~$ chmod 7755 testfile #为文件 testfile 添加特殊权限位
```

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rwsr-sr-t 1 cmd cmd 598 2009-09-01 15:20 testfile #所有者具有可读、可写特殊权限
```

## 3.11 chown (更改文件或目录的拥有者或所属组)

(1) 频度等级：☆☆

(2) 功能说明：

UNIX/Linux 是多人多作业系统，所有的文件皆有拥有者。利用 `chown` 可以将文件的拥有者加以改变。

(3) 语法：

```
chown [-cfhRv][--dereference][--help][--version][拥有者.<所属群组>][文件或目录..] 或
chown [-chfRv][--dereference][--help][--version][.所属群组][文件或目录...] 或
chown [-cfhRv][--dereference][--help][--reference=<参考文件或目录>][--version][文件或目录...]
```

(4) 所属包：

```
chgrp
```

(5) 使用技巧：

批量修改同目录下的文件的所有者和组信息，可使用参数“`-R`”。如果被修改文件命名有一定的规则，则可使用通配符。

命令指定的用户名和组信息可以使用用户 ID 和组 ID 来代替。

(6) 相关命令：

```
chgrp
```

(7) 参数：

`chown` 主要参数如表 3-13 所示。

表 3-13 chown 参数表

参数	含义
-c 或--changes	效果类似“-v”参数，但仅回报更改的部分
-f 或--quite 或--silent	不显示错误信息
-h 或--no-dereference	只对符号链接的文件做修改，而不更改其他任何相关文件
-R 或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-v 或--version	显示指令执行过程
--dereference	效果和“-h”参数相同
--reference=<参考文件或目录>	把指定文件或目录的拥有者与所属群组全部设成和参考文件或目录的拥有者与所属群组相同
--help	显示帮助信息
--version	显示版本信息

## (8) 补充说明:

在 UNIX 系统家族里，文件或目录权限的掌控由拥有者及所属群组来管理。用户可以使用 `chown` 指令去变更文件与目录的拥有者或所属群组，设置方式采用用户名或用户识别码，设置群组则用群组名称或群组识别码。

## (9) 典型实例:

例 1: 改变文件所有者。

例如修改文件 `testfile` 的文件所有者为 `root`，可使用如下命令：

```
chown-v root testfile #修改文件所有者为 root，-v 参数为显示详细过程
```

为了使读者更加清楚，先通过 `ls` 命令查看原始属性，`testfile` 的原始属性如下：

```
cmd@cmd-desktop:~$ls-l testfile #查看原有属性
-rw-r--r-- 1 cmd cmd 598 2009-09-01 15:05 testfile #所有者为 cmd，所属组为 cmd
```

使用 `chown` 命令修改所属组，再次查看属性，结果如下：

```
cmd@cmd-desktop:~$chown-v root testfile
changed ownership of 'testfile' to 'root'
cmd@cmd-desktop:~$ls-l testfile #查看修改后的属性
-rw-r-r-- 1 root cmd 598 2009-09-01 15:07 testfile #文件所有者改为 root
```

例 2: 同时修改文件所有者和所属组。如果想同时修改文件所有者和所属组，可使用如下命令：

```
chown-v root:root testfile #修改文件所有者为 root，所属组为 root，-v 参数为显示详细过程
```

使用 `chown` 命令修改所属组和所有者，查看属性，结果如下：

```
cmd@cmd-desktop:~$chown-v root testfile
changed ownership of 'testfile' to root:root
cmd@cmd-desktop:~$ls-l testfile #查看修改后的属性
-rw-r-r-- 1 root root 598 2009-09-01 15:10 testfile #文件所有者改为 root，所属组改为 root
```

## 3.12 cksum ( 检查文件的 CRC )



(1) 频度等级: ☆☆

(2) 功能说明:

`cksum` 命令用于确保文件从一个系统传输到另一个系统的过程中不被损坏。这个测试要求校验和在源系统中被计算出来，在目的系统中又被计算一次，两个数字比较，如果校验和相等，则该文件被认为是被正确传输了。当需要检查文件或目录是否被改动过时就要用到 `cksum` 命令。通过将一目录或文件的校验和与它以前的校验和相比较，就能判断该文件是否被改动过。

(3) 语法:

```
cksum [--help][--version][文件...]
```

(4) 所属包:

```
coreutils
```

(5) 使用技巧:

CRC 是一种排错检查方式，该演算法的标准由 CCITT 指定，至少可检测到 99.998% 的已知错误。

(6) 相关命令:

```
Sum、md5sum
```

(7) 参数:



cksum 主要参数如表 3-14 所示。

表 3-14 cksum 参数表

参 数	含 义
--help	显示帮助信息
--version	显示版本信息

(8) 补充说明:

指定文件交由 cksum 演算, 它会回报计算结果, 供用户核对文件是否正确无误。若不指定任何文件名或是所给予的文件名为“-”, 则 cksum 指令会从标准输入设备读取数据。

(9) 典型实例:

例: 计算文件的 CRC 校验码, 并检查文件的完整性。

例如, 使用 cksum 命令计算文件 testfile 的完整性。命令如下:

```
cksum testfile      #计算 testfile 的 CRC 校验码
```

输出结果如下:

```
cmd@cmd-desktop:~$cksum testfile      #计算 testfile 的 CRC 校验码
1862500430 56 testfile                 #其中, 1862500430 为校验码, 56 为字节数
```

## 提示

文件中任何字符的改变都将改变计算后 CRC 校验码的值。

# 3.13 cmp ( 比较两个文件是否有差异 )



(1) 频度等级: ☆☆

(2) 功能说明:

比较两个文件是否有差异。当相互比较的两个文件完全一样时, 则该指令不会显示任何信息。若发现有差异, 预设会显示出第一个不同之处的字符和列数编号。若不指定任何文件名称或是所给予的文件名为“-”, 则 cmp 指令会从标准输入设备读取数据。

(3) 语法:

```
cmp [-clsv][--i <字符数目>][--help][第一个文件][第二个文件]
```

(4) 所属包:

diff

(5) 使用技巧:

要获取 cmp 命令的返回值, 可以在执行完 cmp 命令之后, 使用“echo \$?”命令。

(6) 相关命令:

diff

(7) 参数:

cmp 主要参数如表 3-15 所示。

表 3-15 cmp 参数表

参 数	含 义
-c 或 --print-chars	除了标明差异处的十进制字码之外, 同时还显示该字符所对应字符
-i<字符数目>或 --ignore-initial=<字符数目>	指定一个数目
-l 或 --verbose	标识出所有不一样的地方
-s 或 --quiet 或 --silent	对于内容不同的文件, 不显示任何信息, 只返回 cmp 命令的退出状态, 退出状态的含义如下 <ul style="list-style-type: none"> <li>● 0: 两个文件相同</li> <li>● 1: 两个文件不同</li> <li>● &gt;1: 发生了一个错误</li> </ul>
--help	显示帮助信息

-v 或--version

显示版本信息

(8) 典型实例:

 例 1: 使用 `cmp` 命令比较两个文件的差异。

 例如比较文件 `testfile` 和文件 `testfile1` 两个文件, 可使用如下命令:

```
cmp testfile testfile1 #比较两个文件
```

 首先查看两个文件的内容, 使用 `cat` 命令, 结果如下:

```
cmd@cmd-desktop:~$ cat testfile #查看第一个源文件的内容
```

```
wangwen 80
```

```
liuqu 77
```

```
madaoming 61
```

```
dongzhijiang 88
```

```
cmd@cmd-desktop:~$ cat testfile1 #查看第二个源文件的内容
```

```
wangwei 85
```

```
liuqi 92
```

```
madaoming 65
```

```
dongzhijiang 70
```

 使用 `cmp` 命令比较, 可得到如下结果:

```
cmd@cmd-desktop:~$ cmp testfile testfile1 #比较两个文件
```

```
testfile testfile1 有差异: 第 7 字节, 第 1 行 #比较结果, 该命令只能显示第一比较结果
```

例 2: 列出两文件的所有差异处。

 例如接上例, 详细列出文件 `testfile` 和文件 `testfile1` 的差异, 可使用如下命令:

```
cmp-l testfile testfile2 #详细列出文件 testfile 和文件 testfile1 的差异
```

使用该命令后有如下输出:

```
hdd@hdd-desktop:~$ cmp -l testfile testfile1
```

```
7 156 151 #显示结果
```

```
10 60 65 #其中第一列表示差异处所在编码, 这里表示两文件的第 7、10、16、
```

```
16 165 151 #18、19、32、47、48 字节有差异, 第二列表示第一个文件与第二
```

```
18 67 71 #个文件的差异字节列表, 第三列表示第二个文件与第一个文件的差
```

```
19 67 62 #异字节的列表, 均以十进制码表示
```

```
32 61 65
```

```
47 70 67
```

```
48 70 60
```

## 3.14 cp (复制文件或目录)



(1) 频度等级: ☆☆☆

(2) 功能说明:

复制文件或目录。`cp` 命令可以将单个文件复制到一个具体的文件名或一个已经存在的目录下, 也可以复制多个文件到一个指定的目录中 (此时要求目标文件参数必须是一个已经存在的目录名)。

(3) 语法:

```
cp [-abdflpPrRsuvx][-S <备份字尾字符串>][-V <备份方式>][--help][--spares=<使用时机>][--version][源文件或目录][目标文件或目录][目的目录]
```

(4) 所属包:

```
coreutils
```

(5) 使用技巧:

使用 `cp` 命令且使用 “`-i`” 参数复制文件时, 在覆盖已经存在的目标文件时会给出提示信息, 以保证重要文件不会被覆盖。

如果要复制的文件是多个文件的列表, 则目标文件应该是一个目录名, 且已经存在。

如果磁盘空间紧张，可以使用 `cp` 命令的“-l”、“-s”参数，为源文件创建硬链接或符号链接，以节省磁盘空间。

使用“-l”参数为源文件创建硬链接时要求源文件和目标文件必须在同一文件系统之下。

(6) 相关命令：

`dd`

(7) 参数：

`cp` 主要参数如表 3-16 所示。

表 3-16 `cp` 参数表

参 数	含 义
-a 或--archive	此参数的效果和同时指定“-dpR”参数相同
-b 或--backup	覆盖目标文件之前的备份，备份文件会在字尾加上一个备份字符串
-d 或--no-dereference	当复制符号链接时，把目标文件或目录也建立为符号链接，并指向与源文件或目录链接的原始文件或目录
-f 或--force	强行复制文件或目录，不论目标文件或目录是否已存在
-i 或--interactive	覆盖已有文件之前先询问用户
-l 或--link	对源文件建立硬链接，而非复制文件
-p 或--preserve	保留源文件或目录的属性
-P 或--parents	保留源文件或目录的路径
-r	递归处理，将指定目录下的文件与子目录一并处理
-R 或--recursive	递归处理，将指定目录下的所有文件与子目录一并处理
-s 或--symbolic-link	对源文件建立符号链接，而非复制文件
-S< 备份字尾字符串 > 或 --suffix=<备份字尾字符串>	用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的备份字尾字符串是符号“~”
-u 或--update	使用这项参数后只会在源文件的更改时间较目标文件更新时，或是名称相互对应的目标文件并不存在时，才复制文件
-v 或--verbose	显示指令执行过程
-V< 备份方式 > 或 --version-control=<备份方式>	用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，这字符串不仅可用“-S”参数变更，当使用“-V”参数指定不同备份方式时，也会产生不同字尾的备份字符串
-x 或--one-file-system	复制的文件或目录存放的文件系统，必须与 <code>cp</code> 指令执行时所处的文件系统相同，否则不予复制
--sparse=<使用时机>	设置保存稀疏文件的时机
--help	显示帮助信息
--version	显示版本信息

(8) 补充说明：

`cp` 指令用于复制文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到该目录中。若同时指定多个文件或目录，而最后的目的地并非是一个已存在的目录，则会出现错误信息。

(9) 典型实例：

例 1：复制文件和目录。

在默认情况下，不带参数复制目录将输出错误信息，例如在终端中输入如下命令：

```
cmd@cmd-desktop:~$ cp test/ test1 #默认情况下使用 cp 命令复制目录将出错
将输出以下错误信息：
```

```
cp: 略过目录 “test/”
```

因此，如果想要复制目录，必须使用参数“-R”，具体命令如下：

```
cp-R test/ test1
```

先在使用 `cp` 命令复制目录前使用 `dir` 命令查看当前目录结构，如下所示：

```
cmd@cmd-desktop:~$ dir #查看原始目录结构
examples.desktop testfile testfile1 testfile_1 模板 图片 音乐
test testfile1 公共的 视频 文档 桌面
```

使用 `cp` 命令复制目录，该命令执行后没有任何输出，再次使用 `dir` 命令查看当前目录结构，结果如下：

```
cmd@cmd-desktop:~$ cp -R test/ test1 #使用 cp 命令复制目录
```

```
cmd@cmd-desktop:~$ dir #查看复制后的目录结构
examples.desktop test1 testfile1 公共的 视频 文档 桌面
test testfile testfile_1 模板 图片 音乐
```

例 2: 将文件 testfile 改名为 testfile1, 如果 testfile1 文件已经存在, 可使用“-f”参数, 使用强行复制模式, 在终端中输入如下命令:

```
cp-f testfile testfile1 #将文件 testfile 强行改名为 testfile1
```

同上, 使用 cp 命令前后调用 dir 查看目录结构, 输出结果如下:

```
cmd@cmd-desktop:~$dir #查看改名前的目录结构
examples.desktop testfile testfile_1 模板 图片 音乐
test testfile1 公共的 视频 文档 桌面
cmd@cmd-desktop:~$ cp-f testfile testfile1 #将文件 testfile 强行改名为 testfile1
cmd@cmd-desktop:~$dir #可见 testfile 被强行改名为 testfile1
examples.desktop testfile1 testfile_1 模板 图片 音乐
test 公共的 视频 文档 桌面
```

例 3: 复制多个文件。

例如, 要将同目录下的文件 testfile1、testfile\_1, 以及目录 test1 复制到目录 test 下, 可使用命令:

```
cp-R testfile1 testfile_1 test1/* test #复制多个目录和文件到其他目录
```

该命令使用前后比较如下:

```
cmd@cmd-desktop:~$dir #查看复制前的目录结构
examples.desktop testfile1 testfile_1 模板 图片 音乐
test 公共的 视频 文档 桌面
cmd@cmd-desktop:~$ cp-R testfile1 testfile_1 test1/* test #复制多个目录和文件到其他目录
cmd@cmd-desktop:~$dir test/ #查看复制后的目录结构
testfile1 testfile_1 test1 #可见文件 testfile1、testfile_1, 以及目录 test1
```

#复制到目录 test 下

## 3.15 cpio (将文件备份、归档或归档文件解压缩)

(1) 频度等级: ☆☆

(2) 功能说明:

cpio 是用来建立, 还原归档文件的工具程序, 它可以加入, 解开 cpio 或 tar 归档文件内的文件。

(3) 语法:

```
cpio [-0aABckLovV][ -C <输入/输出大小>][ -F <归档文件>][ -H <备份格式>][ -O <归档文件>][ --block-size=<
区块大小>][ --force-local][ --help][ --quiet][ --version]
cpio [-bBcdfikmrsStuvV][ -C <输入/输出大小>][ -E <范本文件>][ -F <归档文件>][ -H <备份格式>][ -I <归
档文件 >][ -M <回传信息 >][ -R <拥有者 >[:/.>< 所属群组 >][ --block-size=< 区块大
小>][ --force-local][ --help][ --no-absolute-filenames][ --no-preserve-owner][ --only-verify-crc][ -
-quiet][ --sparse] [ --version][ 范本样式...]
cpio [-0adkiLmpuvV][ -R <拥有者 >[:/.>< 所属群组 >][ --help][ --no-preserve-owner][ --quiet]
[ --sparse][ --version][ 目的目录]
```

(4) 所属包:

```
cpio
```

(5) 使用技巧:

使用 cpio 命令进行备份时, 只有损坏部分不能被访问, 其他部分能够正常访问。

使用 cpio 命令的 copy-out 模式备份时, 归档包中包含了文件的所有者、时间属性、访问许可等信息。

(6) 相关命令:

```
tar
```

(7) 参数:

cpio 主要参数如表 3-17 所示。

表 3-17 cpio 参数表

参 数	含 义
-0 或--null	接受新增列控制字符，通常配合 find 指令的“-print0”参数使用
-a 或--reset-access-time	重新设置文件的存取时间
-A 或--append	附加到已存在的归档文件中，且这个归档文件必须存放在磁盘上，而不能放置于磁带机里
-b 或--swap	此参数的效果和同时指定“-sS”参数相同
-B	将输入/输出的区块大小改成 5210 Bytes
-c	使用旧 ASCII 备份格式
-C<输入/输出大小> 或 --io-size=<区块大小>	设置输入/输出的区块大小，单位是 Byte
-d 或--make-directories	如有需要 cpio 会自行建立目录
-E<范本文件> 或 --pattern-file=<范本文件>	指定范本文件，其内含有一个或多个范本样式，让 cpio 解开符合范本条件的文件，格式为每列一个范本样式
-f 或--nonmatching	解开所有不符合范本条件的文件
-F<归档文件>或--file=<归档文件>	指定归档文件的名称，用来取代标准输入或输出，也能借此通过网络使用另一台主机的保存设备存取归档文件
-H<备份格式>	指定备份时要使用的文件格式
-i 或--extract	执行 copy-in 模式，还原归档文件
-l<归档文件>	指定归档文件的名称，用来取代标准输入，也能借此通过网络使用另一台主机的保存设备读取归档文件
-k	此参数将忽略不予处理，仅负责解决 cpio 不同版本间的兼容性问题
-l 或--link	以硬链接的方式取代复制文件，可在 copy-pass 模式下运用
-L 或--dereference	不建立符号链接，直接复制该链接所指向的原始文件
-m 或 preserve-modification-time	不去更换文件的更改时间
-M<回传信息>或--message=<回传信息>	设置更换保存媒体的信息
-n 或--numeric-uid-gid	使用“-tv”参数列出归档文件的内容时，若再加上参数“-n”，则会以用户识别码和群组识别码替代拥有者和群组名称列出文件清单
-o 或--create	执行 copy-out 模式，建立归档文件
-O<归档文件>	指定归档文件的名称，用来取代标准输出，也能借此通过网络使用另一台主机的保存设备存放归档文件
-p 或--pass-through	执行 copy-pass 模式，略过备份步骤，直接将文件复制到目的目录
-r 或--rename	当有文件名称需要更改时，采用互动模式
-R<拥有者><:/.><所属群组> 或 ---owner<拥有者 00><:/.><所属群组>	在执行 copy-in 模式还原归档文件或 copy-pass 模式复制文件时，可指定这些备份复制文件的拥有者与所属群组
-s 或--swap-bytes	交换每对字节的内容
-S 或--swap-halfwords	交换每半个字节的内容
-t 或--list	将输入的内容呈现出来
-u 或--unconditional	置换所有文件，不论日期时间的新旧与否，皆不予询问而直接覆盖
-v 或--verbose	详细显示指令的执行过程
-V 或--dot	执行指令时，在每个文件的执行程序前面加上“.”号
--block-size=<区块大小>	设置输入/输出的区块大小，假如设置数值为 5，则区块大小为 2500，若设置成 10，则区块大小为 5120，依次类推
--force-local	强制将归档文件存放在本地主机
--no-absolute-filenames	使用相对路径建立文件名称
--no-preserve-owner	不保留文件的拥有者，谁解开了归档文件，那些文件就归谁所有
-only-verify-crc	当归档文件采用 CRC 备份格式时，可使用这项参数检查归档文件内的每个文件是否正确无误
--quiet	不显示复制了多少区块
--sparse	倘若一个文件内含大量的连续 0 字节，则将此文件存成稀疏文件
--help	显示帮助信息
--version	显示版本信息

#### (8) 典型实例：

例 1：使用 cpio 命令生成打包文件。



例如，使用 `cpio` 命令的参数“-o”，结合管道输出重定向，对当前文件打包备份，使用命令为：

```
ls | cpio-o > file_bak #将当前目录下的所有文件备份到归档文件 file_bak 中
```

使用该命令输出如下：

```
cmd@cmd-desktop:~$ ls | cpio-o > file_bak #将当前目录下的所有文件备份到归档文件 file_bak 中
3 块 #输出打包后文件的大小
```

该例中，将 `ls` 命令输出的文件列表通过管道传递给 `cpio` 命令，后者将文件打包，再重定向到 `file_bak` 文件。如果目标文件分布在各个目录中，可以结合 `find` 命令先查找，再打包。

例 2：结合 `find` 命令查找打包文件。例如查找所有 `passwd` 文件并打包，可使用如下命令：

```
find / -name passwd | cpio: o > passwd_bak #查找所有 passwd 文件并打包
```

该命令的输出结果如下：

```
cmd@cmd-desktop:~$ find / -name passwd | cpio: o > passwd_bak #查找所有 passwd 文件并打包
/etc/pam.d/passwd:truncating inode number #查找到的文件信息
/etc/passwd:truncating inode number
/usr/bin/passwd:truncating inode number
/usr/share/doc/passwd:truncating inode number
/usr/share/lintian/overrides/passwd:truncating inode number
86 块 #归档文件大小
```

例 3：显示生成的归档文件的文件列表。例如想查看上例中生成的归档文件 `passwd_bak` 文件列表，使用“-t”参数输出结果如下：

```
cmd@cmd-desktop:~$ cpio -t < passwd_bak #查看 passwd_bak 文件列表
/etc/pam.d/passwd #显示归档文件的文件列表
/etc/passwd
/usr/bin/passwd
/usr/share/doc/passwd
/usr/share/lintian/overrides/passwd
86 块 #归档文件大小
```

## 3.16 diff ( 比较文件的差异 )



(1) 频度等级：☆☆

(2) 功能说明：

比较文件的差异，并将文件的差异显示到标准输出。

(3) 语法：

```
diff [-abBcdefHilnNpPqrstTuvwy][-<行数>][-C <行数>][-D <巨集名称>][-I <字符或字符串>][-S <文件>][-W <宽度>][-x <文件或目录>][-X <文件>][--help][--left-column] [--suppress-革 common-line][文件或目录 1][文件或目录 2]
```

(4) 所属包：

```
diffutils
```

(5) 使用技巧：

如果不指定文件名，或者文件名为“-”，则 `diff` 命令从标准输入读取内容。

`diff` 命令不但可以比较两个文本文件，还可以比较两个二进制文件。在默认情况下，`diff` 命令比较的是两个文件，但比较二进制文件时，使用“-a”参数。

(6) 相关命令：

```
cmp、diffstat、diff3
```

(7) 参数：

`diff` 主要参数如表 3-18 所示。

表 3-18 diff 参数表



参 数	含 义
-<行数>	指定要显示多少行的文本。此参数必须与“-c”或“-u”参数一起使用
-a 或--text	比较二进制文件。diff 默认只会逐行比较文本文件
-b 或--ignore-space-change	不检查空格字符的不同
-B 或--ignore-blank-lines	不检查空白行
-c	显示全部内文，并标出不同之处
-C<行数>或--context<行数>	与执行“-c<行数>”指令相同
-d 或--minimal	使用不同的演算法，以较小的单位来作比较
-D<巨集名称>或 ifdef<巨集名称>	此参数的输出格式可用于前置处理器巨集
-e 或--ed	此参数的输出格式可用于 ed 的 script 文件
-f 或--forward-ed	输出的格式类似 ed 的 script 文件，但按照原来文件的顺序来显示不同处
-H 或--speed-large-files	比较大文件时，使用此参数可加快速度
-l<字符或字符串> 或 --ignore-matching-lines<字符或 00 字符串>	若两个文件在某几行有所不同，而这几行同时都包含了选项中指定的字符或字符串，则不显示这两个文件的差异
-i 或--ignore-case	不检查大小写的不同
-l 或--paginate	将结果交由 pr 程序来分页
-n 或--rcs	将比较结果以 RCS 的格式来显示
-N 或--new-file	在比较目录时，若文件 A 仅出现在某个目录中，预设会显示 Only in 目录：文件 A 若使用“-N”参数，则 diff 会将文件 A 与一个空白的文件比较
-p	若比较的文件为 C 语言的程序码文件，则显示差异所在的函数名称
-P 或--unidirectional-new-file	与-N 类似，但只有当第二个目录包含了一个第一个目录所没有的文件时，才会将这个文件与空白的文件作比较
-q 或--brief	仅显示有无差异，不显示详细的信息
-r 或--recursive	比较子目录中的文件
-s 或--report-identical-files	若没有发现任何差异，就仍然显示信息
-S<文件>或--starting-file<文件>	在比较目录时，从指定的文件开始比较
-t 或--expand-tabs	在输出时，将 tab 字符展开
-T 或--initial-tab	在每行前面加上 tab 字符以便对齐
-u,-U<列数>或--unified=<列数>	以合并的方式来显示文件内容的不同
-v 或--version	显示版本信息
-w 或--ignore-all-space	忽略全部的空格字符
-W<宽度>或--width<宽度>	在使用-y 参数时，指定栏宽
-x<文件名或目录> 或 --exclude<文件名或目录>	不比较选项中所指定的文件或目录
-y 或--side-by-side	用户可以将文件或目录类型存成文本文件，然后在=<文件>中指定此文本文件
--help	显示帮助信息
--left-column	在使用-y 参数时，若两个文件某一行内容相同，则仅在左侧的字段显示该内容
--suppress-common-lines	在使用-y 参数时，仅显示不同之处

#### (8) 补充说明：

diff 以逐行的方式比较文本文件的异同处。如果指定比较目录，diff 会比较目录中相同文件名的文件，但不会比较其子目录。

#### (9) 典型实例：

例 1：比较两个文件是否相同。

例如比较文件 testfile 和 testfile1 是否相同，使用“-q”参数，不显示差异的详细内容，具体命令如下：

```
diff -q testfile testfile1 #比较两个文件是否存在差异
```

该命令的输出如下：

```
cmd@cmd-desktop:~$ diff -q testfile testfile1 #比较两个文件是否存在差异
文件 testfile 和 testfile1 不同           #显示比较结果
```

例 2：接上例，比较两个文件并显示其差异，不使用参数“-q”，此时除了输出比较结果外，还将输出详细的比较内容。

文件 testfile 和文件 testfile1 中的内容分别如下：

```
cmd@cmd-desktop:~$ cat testfile      #查看第一个源文件的内容
wangwei 85
liuqi 90
madaoming 65
dongzhijiang 70
cmd@cmd-desktop:~$ cat testfile1    #查看第二个源文件的内容
wangwei 85
liuqi 92
madaoming 77
dongzhijiang 70
```

比较输出结果如下：

```
cmd@cmd-desktop:~$ diff testfile testfile1  #比较并输出结果
2,3c2,3                                     #不同的行数
< liuqi 90                                 #不同行中 testfile 的内容
< madaoming 65
---                                         #分隔线
> liuqi 92                                 #不同行中 testfile1 的内容
> madaoming 77
```

## 3.17 diff3 ( 比较三个文件的不同 )



(1) 频度等级：☆

(2) 功能说明：

比较三个文件，将不同文本的范围输出。如果输出以“====”表示开头，表示三个文件各不相同；以“====1”开头，表示第一个文件不同；以此类推，以“====2”开头，表示第二个文件不同，以“====3”开头，表示第三个文件不同。

(3) 语法：

```
diff3 [参数][文件名]
```

(4) 所属包：

```
diffutils
```

(5) 使用技巧：

diff3 命令用于比较三个文件的差异，diff 命令用于比较两个文件的差异。

(6) 参数：

diff3 的主要参数如表 3-19 所示。

表 3-19 diff3 参数表

参 数	含 义
-a	将所有的文件都看做文本文件

(7) 典型实例：

例：比较三个文本文件的不同。

例如要比较当前目录下三个文本文件 testfile1、testfile2、testfile3 的不同，可使用命令如下：

```
diff3 testfile1 testfile2 testfile3      #比较三个文本文件的不同
```

使用该命令前，可先使用 cat 命令查看这三个文件的内容，以便于对比，得到三个文件分别为：

```
cmd@cmd-desktop:~$cat testfile1        #查看 testfile1 文件的内容
this is the first file!
cmd@cmd-desktop:~$cat testfile2        #查看 testfile2 文件的内容
this is the second file!
```

```
cmd@cmd-desktop:~$cat testfile3 #查看 testfile3 文件的内容
this is the third file!
```

使用 diff3 命令比较三个文件之后，得到如下输出结果：

```
cmd@cmd-desktop:~$ diff3 testfile1 testfile2 testfile3 #比较三个文本文件的不同
==== #三个文件各不相同
1:1c #第一个文件共一行
  this is the first file! #第一个文件的内容
2:1c #第二个文件共一行
  this is the second file! #第二个文件的内容
3:1c #第三个文件共一行
  this is the third file! #第三个文件的内容
```

## 3.18 diffstat ( 根据 diff 的比较结果，显示统计信息 )

(1) 频度等级：☆

(2) 功能说明：

根据 diff 的比较结果，显示所有的相关属性，如包括文件的删除、插入、修改等操作的统计信息。

(3) 语法：

```
diff [-cehkouwV][-n <文件名长度>][-p <文件名长度>]
```

(4) 所属包：

```
diffutils
```

(5) 使用技巧：

可以使用管道“|”将输出的结果直接送到 diffstat 命令显示统计信息。

(6) 相关命令：

```
diff
```

(7) 参数：

diffstat 主要参数如表 3-20 所示。

表 3-20 diffstat 参数表

参 数	含 义	参 数	含 义
-c	在输出的文件前面加上#，将该行视为注释	-p<文件名长度>	与-n 参数相同，但此处的<文件名长度>包括了文件的路径
-e	将标准错误输出到指定文件	-u	不再显示中将文件排序
-h	显示帮助信息	-v	显示执行过程
-k	在输出中禁止文件名合并	-V	显示版本信息
-n<文件名长度>	指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名	-w	指定输出时字段的宽度
-o	将标准（正确）输出到指定文件		

(8) 补充说明：

diffstat 读取 diff 的输出结果，然后统计各文件的插入、删除、修改等差异计量。

(9) 典型实例：

例：比较两个目录的差别，并输出统计信息。

例如要比较 test 和 file 两个目录的不同，并输出统计信息，可使用如下命令：

```
diff test file | diffstat #比较目录 test 和目录 file 的不同，并结合 diffstat 输出统计信息
首先，查看这两个目录下的文件结构，以及这两个目录中文件的内容，结果如下：
```

```
hdd@hdd-desktop:~$ ls test/ file/ #查看 test/、file/目录下的文件
file/ #file/目录下的文件
testfile
test/ #test/目录下的文件
```

```
testfile
hdd@hdd-desktop:~$ cat test/testfile          #查看 test/testfile 的内容
Linux is a free Unix-type operating system originally created by Linus Torvalds with the
assistance of developers around the world. Developed under the GNU General Public License , the
source code for Linux is freely available to everyone.
Click on the link below to find out more about the operating system that is causing a revolution
in the world of computers.
hdd@hdd-desktop:~$ cat file/testfile          #查看 file/testfile 的内容
Linux is a free Unix-type operating system originally created by Linus Torvalds with the
assistance of developers around the world. Developed under the GNU General Public License , the
source code for Linux is freely available to everyone.
```

该命令执行后输出结果如下:

```
hdd@hdd-desktop:~$ diff test/ file/ | diffstat  #比较这两个目录, 并输出统计信息
testfile |      2 +-                          #统计信息
1 file changed, 1 insertion(+), 1 deletion(-)
```



## 3.19 dump ( 文件系统备份 )

(1) 频度等级: ☆

(2) 功能说明:

**dump** 为备份工具程序, 可将目录或整个文件系统备份至指定的设备, 或备份成一个大文件。

(3) 语法:

```
dump[-cnu][[-0123456789]][-b <区块大小>][-B <区块数目>][-d <密度>][-f <设备名称>][-h <层级>][-s <
磁带长度>][-T <日期>][<目录或文件系统>]
dump [-wW]
```

(4) 所属包:

```
dump
```

(5) 使用技巧:

指定生成的备份文件名或目标设备文件, 可使用参数“-f”。

第一次备份文件时, 备份层级为 0, 以后使用增量备份, 以缩短备份时间并节省空间。

(6) 相关命令:

```
restore
```

(7) 参数:

**dump** 主要参数如表 3-21 所示。

表 3-21 dump 参数表

参 数	含 义
-0123456789	指定备份的层级
-b<区块大小>	指定区块的大小, 单位为 KB
-B<区块数目>	指定备份卷册的区块数目
-c	修改备份磁带预设的密度与容量
-d<密度>	设置磁带的密度, 单位为 BPI
-f<设备名称>	指定备份文件或目标设备文件 (将文件系统备份到对应的设备)
-h<层级>	当备份层级等于或大于指定的层级时, 将不备份用户标识为“nodump”的文件
-n	当备份工作需要管理员介入时, 向所有“operator”群组中的使用者发出通知
-s<磁带长度>	备份磁带的长度, 单位为英尺
-T<日期>	指定开始备份的时间与日期
-u	备份完毕后, 在“/etc/dumpdates”中记录备份的文件系统、层级、日期与时间等

-w	与“-W”类似，但仅显示需要备份的文件
-W	显示需要备份的文件及其最后一次备份的层级、时间与日期

(8) 典型实例：

例 1：使用 `dump` 命令备份指定目录。

例如使用“-f”参数备份指定的“/etc”目录，使用如下命令：

```
dump-f etc.bak /etc/ #备份/etc/目录下所有文件
```

将输出以下信息：

```
cmd@cmd-desktop:~$ dump -f etc.bak /etc/ #备份/etc/目录下所有文件
DUMP: Date of this level dump: Sun Sep 20 14:47:35 2009 #备份输出信息
DUMP: Dumping /dev/sdal (/ (dir etc)) to etc.bak
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 14472 blocks.
DUMP: Volume 1 started with block 1 at: Sun Sep 20 14:47:36 2009
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing etc.bak
DUMP: Volume 1 completed at: Sun Sep 20 14:47:38 2009
DUMP: Volume 1 14870 blocks (14.52MB)
DUMP: Volume 1 took 0:00:02
DUMP: Volume 1 transfer rate: 7435 kB/s
DUMP: 14870 blocks (14.52MB) on 1 volume(s)
DUMP: finished in 2 seconds, throughput 7435 kBytes/sec
DUMP: Date of this level dump: Sun Sep 20 14:47:35 2009
DUMP: Date this dump completed: Sun Sep 20 14:47:38 2009
DUMP: Average transfer rate: 7435 kB/s
DUMP: DUMP IS DONE
```

使用 `file` 命令查看文件信息，可得到如下结果：

```
cmd@cmd-desktop:~$ file etc.bak #查看备份文件信息
etc.bak: new-fs dump file (little endian), This dump Sun Sep 20 14:47:35 2009, Previous dump
Thu Jan 1 08:00:00 1970, Volume 1, Level zero, type: tape header, Label none, Filesystem / (dir
etc), Device /dev/sdal, Host hdd-desktop, Flags 3#备份文件的详细信息
```

例 2：使用 `dump` 命令备份整个根文件系统。使用该命令可以将整个根文件系统备份到指定的外设，使用的命令如下：

```
dump -0f /dev/nst0 / #将整个根文件系统备份到 SCSI 设备/dev/nst0
```

该命令将整个根文件系统备份到外设“/dev/nst0”上。

## 3.20 fsck ( 检查并修复 Linux 文件系统 )



(1) 频度等级：☆☆

(2) 功能说明：

检查文件系统的一致性并且以交互方式修复文件系统。在出现系统故障之后，总是运行 `fsck` 命令。矫正的动作也许会导致某些数据丢失。对于每个一致性的矫正，默认的操作是等待操作员输入“yes”或者“no”。如果对于已经受到影响的文件系统操作员没有写的许可，那么无论实际响应是什么，`fsck` 命令默认的动作都是“no”。

(3) 语法：



```
fsck [-aANPrRsTV][-t <文件系统类型>][文件系统...]
```

(4) 所属包:

```
e2fsprogs
```

(5) 使用技巧:

fsck 仅仅是一个前端工具，它必须通过“-t”参数来调用相应的文件系统检查器完成文件系统的检查和修复工作。

fsck 命令的返回状态值可以反映命令的运行状况，返回值及其含义如表 3-22 所示。

表 3-22 fsck 返回值及其含义列表

返回值	含 义	返回值	含 义
0	无错误	8	操作错误
1	发生文件系统错误	16	应用或语法错误
2	发生文件系统错误，系统应该重新启动	32	e2fsck 命令被用户取消
4	文件系统不正常退出错误	128	共享库错误

(6) 相关命令:

```
e2fsck
```

(7) 参数:

fsck 主要参数如表 3-23 所示。

表 3-23 fsck 参数表

参 数	含 义
-a	自动修复文件系统，不询问任何问题
-A	依照“/etc/fstab”配置文件的内容，检查文件内所列的全部文件系统
-C	显示文件系统检查器的完成进度
-n	对系统的提问自动回答 n
-N	不执行指令，仅列出实际执行会进行的动作
-P	当搭配“-A”参数使用时，则会同时检查所有的文件系统
-r	采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式
-R	当搭配“-A”参数使用时，则会略过目录的文件系统不予检查
-s	依序执行检查作业，而非同时执行
-t<文件系统类型>	指定要检查的文件系统类型
-T	执行 fsck 指令时，不显示标题信息
-V	显示指令执行过程
-y	对系统的提问自动回答 y

(8) 补充说明:

当文件系统发生错误时，可用 fsck 指令尝试加以修复。

对于一个已经安装好了的文件系统，fsck 命令不会做出矫正。

fsck 命令可以在一个已经安装好了的文件系统中运行，但不进行修复。当文件系统安装完毕之后，也许会返回不准确的错误消息。

(9) 典型实例:

例：检查文件系统。

例如使用 fsck 命令的“-t”参数指明要检查的文件系统的类型，具体命令为：

```
fsck-t ext3 /dev/sda1 #检查 ext3 类型的分区/dev/sad1
```

此时输出如下信息，需要用户确认，输入“yes”之后按回车键。

```
cmd@cmd-desktop:~$ fsck -t ext3 /dev/sda1 #检查 ext3 类型的分区/dev/sad1
fsck 1.41.3 (12-Oct-2008) #fsck 版本
e2fsck 1.41.3 (12-Oct-2008) #e2fsck 版本
/dev/sda1 is mounted. #已经加载的文件系统
#强制检查已经加载的文件系统，是一种不推荐使用的方式。除非管理员可以确定当前没有任何程序正在读写要检查的文件系统，否则必须先卸载文件系统后，再进行检查
WARNING!!! Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.
```



```
Do you really want to continue (y/n)?          #等待用户确认, 输入 yes 或 no
继续输出如下信息:
```

```
/dev/sda1: recovering journal                #检查结果
/dev/sda1: clean, 177826/1001712 files, 1141903/4006201 blocks
```

## 3.21 fsck.ext2(检查 ext2 文件系统并尝试修复文件的错误)

(1) 频度等级: ☆

(2) 功能说明:

fsck.ext2 命令用于检查 ext2 文件系统并尝试修复错误, 该命令只能对 ext2 文件系统使用。

(3) 语法:

```
fsck.ext2[-acdfFnprsStvVy][-b <分区第一个磁区地址>][-B <区块大小>][-C <反叙述器>][-I <inode 缓冲
区块数>][-l/L <损坏区块文件>][-P <处理 inode 大小>][<外围设备代号>]
```

(4) 所属包:

e2fsprogs

(5) 使用技巧:

当 ext2 文件系统发生错误时, 可尝试用 fsck.ext2 指令加以修复。

(6) 相关命令:

fsck、fsck.ext2、fsck.ext3、fsck.minix

(7) 参数:

fsck.ext2 主要参数如表 3-24 所示。

表 3-24 fsck.ext2 参数表

参 数	含 义
-a	自动修复文件系统, 不询问任何问题
-b<分区第一个磁区地址>	指定分区的第一个磁区的起始地址, 也就是 Super Block
-B<区块大小>	设置该分区每个区块的大小
-c	检查指定的文件系统内是否存在损坏的区块
-C<反叙述器>	指定反叙述器, fsck.ext2 指令会把全部的执行过程, 都交由其逆向叙述, 便于排错或监控程序执行的情形
-d	详细显示指令执行过程, 便于排错或分析程序执行的情形
-f	强制对该文件系统进行全面检查, 即使该文件系统在概略检查下没有问题
-F	检查文件系统之前, 先清理该保存设备块区内的数据
-I<inode 缓冲区块数>	设置要检查的文件系统中 inode 缓冲区的区块数目
-l<损坏区块文件>	把文件所列出的区块视为损坏区块并将其标识出来, 避免应用程序使用该区块
-L<损坏区块文件>	此参数的效果和指定“-l”参数类似, 但在参考损坏区块文件标识损坏区块之前, 会先将原来标识成损坏区块者清除, 即全部重新设置, 而不仅仅是加入新的损坏区块标识
-n	把要检查的文件系统设为只读, 并关闭互动模式, 否决所有询问的问题
-p	此参数的效果和指定“-a”参数相同
-P<处理 inode 大小>	设置 fsck.ext2 指令所能处理的 inode 大小
-r	此参数将忽略不予处理, 仅负责解决兼容性的问题
-s	检查文件系统时, 交换每对字节的内容
-S	此参数的效果和指定“-s”参数类似, 但不论该文件系统是否已是标准位顺序, 一律交换每对字节的内容
-t	显示 fsck.ext2 指令的时序信息
-v	详细显示指令执行过程
-V	显示版本信息
-y	关闭互动模式, 且同意所有询问的问题

(8) 典型实例:

例: 检查“/dev/sda1”设备。

使用该命令可以检查“/dev/sda1”设备，具体的命令如下：

```
fsck.ext2-y /dev/sda1 #检查“/dev/sda1”设备
```

输出结果如下：

```
hdd@hdd-desktop:~$ sudo fsck.ext2 -y /dev/sda1
#检查“/dev/sda1”设备，同意所有询问信息

e2fsck 1.41.3 (12-Oct-2008) #e2fsck 版本信息
/dev/sda1 is mounted.

WARNING!!! Running e2fsck on a mounted filesystem may cause #警告信息
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes #确认操作

/dev/sda1: clean, 106871/1001712 files, 590781/4006201 blocks #设备检查结果
```

## 3.22 fsck.ext3(检查 ext3 文件系统并尝试修复文件的错误)

(1) 频度等级：☆

(2) 功能说明：

fsck.ext3 命令用于检查 ext3 文件系统，并尝试修复文件的错误。

(3) 语法：

```
fsck.ext3 [-acdfFnprsStvVy][-b<第一个扇区位置>][-B<块的大小>][-C<反叙述器>][-I<inode 缓冲块数>][-l<损坏区块文件>][-L<损坏区块文件>][-P<处理 inode 大小>]
```

(4) 所属包：

```
e2fsprogs
```

(5) 使用技巧：

该命令仅对 ext3 文件系统有效。

(6) 相关命令：

```
fsck、fsck.ext2、fsck.minix
```

(7) 参数：

fsck.ext3 的主要参数如表 3-25 所示。

表 3-25 fsck.ext3 参数表

参 数	含 义
-a	自动修复文件系统，不询问任何问题
-b<第一个扇区位置>	指定分区的第一个磁区的起始地址，也就是 Super Block
-B<区块大小>	设置该分区每个区块的大小
-c	检查指定的文件系统，并将其表明
-C<反叙述器>	指定反叙述器，并执行逆向描述
-d	详细显示指令执行过程，便于排错或分析程序执行的情形
-f	强制对该文件系统进行全面检查，即使该文件系统在概略检查下没有问题
-F	检查文件系统之前，先清理该保存设备区块内的数据
-I<inode 缓冲区块数>	设置在要检查的文件系统中 inode 缓冲区的区块数目
-l<损坏区块文件>	把文件所列出的区块视为损坏区块并将其标识出来，避免应用程序使用该区块
-L<损坏区块文件>	此参数的效果和指定“-l”参数类似，但在参考损坏区块文件标识损坏区块之前，会先将原来标识成损坏区块者清除，即全部重新设置，而不仅仅是加入新的损坏区块标识
-n	把要检查的文件系统设为只读，并关闭互动模式，否决所有询问的问题
-p	此参数的效果和指定“-a”参数相同，自动修复文件系统
-P<处理 inode 大小>	设置 fsck.ext3 指令所能处理的 inode 大小

-r	此参数将忽略不予处理，仅负责解决兼容性的问题
-s	检查文件系统时，交换每对字节的内容
-S	此参数的效果和指定“-s”参数类似，但不论该文件系统是否已是标准位顺序，一律交换每对字节的内容
-t	显示 fsck.ext3 指令的时序信息
-v	详细显示指令执行过程
-V	显示版本信息
-y	关闭互动模式，且同意所有询问的问题

(8) 补充说明：

该命令与 fsck.ext2 命令的用法类似，可对损坏的文件系统加以修复。

(9) 典型实例：

参见命令 fsck.ext2。

## 3.23 fsck.minix( 检查 minix 文件系统并尝试修复文件的错误 )

(1) 频度等级：☆

(2) 功能说明：

fsck.minix 命令用于检查文件系统并尝试修复文件的错误，该命令只对 minix 文件系统有效。

(3) 语法：

```
fsck.minix [-afmrsv][外围设备代号]
```

(4) 所属包：

e2fsprogs

(5) 使用技巧：

该命令仅对 minix 文件系统有效，且一般 Fedora 系统没有该命令。

(6) 相关命令：

fsck、fsck.ext2、fsck.ext3

(7) 参数：

fsck.minix 主要参数如表 3-26 所示。

表 3-26 fsck.minix 参数表

参 数	含 义
-a	自动修复文件系统，不询问任何问题
-f	强制对该文件系统进行完整检查，即使该文件系统在概略检查下没有问题
-l	列出所有文件名称
-m	使用类似 MINIX 操作系统的警告信息
-r	采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式
-s	显示该分区第一个磁区的相关信息
-v	显示指令执行过程

(8) 补充说明：

当 minix 文件系统发生错误时，可尝试用 fsck.minix 命令修复。

(9) 典型实例：

例：以互动的方式检查指定的 minix 文件系统的正确性。使用的命令如下：

```
fsck.minix-r /dev/sdc1 #以互动的方式检查指定的 minix 文件系统的正确性
```

该命令尝试修复文件错误，一般没有输出结果。

# 3.24 file ( 识别文件类型 )



(1) 频度等级: ☆☆☆

(2) 功能说明:

file 命令用于识别文件类型。该命令对文件的检查分为 3 个过程, 分别为文件系统检查、魔幻数检查和语言检查。

(3) 语法:

```
file [-beLvz][[-f <名称文件>][[-m <魔法数字文件>...]][文件或目录...]
```

(4) 所属包:

file

(5) 使用技巧:

由环境变量 MAGIC 设置 file 命令默认的魔幻数文件, file 命令默认的魔幻数文件是“/usr/share/magic”。

file 命令的文件系统检查是建立在对 stat 系统调用结果的分析上的, 程序分析文件是否为空, 或者是否是某种特殊文件, 对于所有可在现有 Linux 系统上使用的文件类型, 比如 socket 文件、动态链接文件、命名管道文件等, 只要它在系统头文件“sys/stat.h”中已经定义过, 就可以被检查到。

(6) 相关命令:

od

(7) 参数:

file 主要参数如表 3-27 所示。

表 3-27 file 参数表

参 数	含 义
-b	列出识别结果, 不显示文件名称
-c	详细显示指令执行过程, 便于排错或分析程序执行的情形
-f<名称文件>	指定名称文件, 其内容有一个或多个文件名称, 让 file 依序识别这些文件, 格式为每列一个文件名称
-L	直接显示符号链接所指向的文件的类别
-n	在检查完一个文件时就强制刷新标准输出, 仅在检查一组文件时有效, 一般在将 file 命令输出的文件类型输出到管道时采用此选项
-m<魔法数字文件>	指定魔法数字文件
-v	显示版本信息
-z	尝试去解读压缩文件的内容

(8) 补充说明:

通过 file 命令我们得以识别该文件的类型。file 命令会尝试识别所指定的文件类型, 如果是文本文件, 会告诉用户是哪一类文本文件, 如 ASCII、English、International 等; 如果是可执行文件, 会告诉用户是哪一类可执行文件, 如 ELF、Bourne-Again Shell Script、Bourne Shell Script 等; 若为符号链接, 则会显示该链接所指向的文件。其他诸如目录、特殊文件 (Block、Character Socket、Pipe 等)、图形文件 (JPEG、GIF、BMP、PCX 等)、备份文件或压缩文件 (TAR、GZIP、ZIP 等) 等, 只要该文件按类型在魔法数字文件内有记录数据, file 命令就会返回其类型的相关信息。

(9) 典型实例:

例 1: 确定单个文件的类型。

例如查看文件“/sbin/ifconfig”的文件类型, 只需将文件传递给 file 命令目录, 即可显示文件的详细类型, 具体命令如下:

```
file /sbin/ifconfig #查看 ifconfig 的文件类型
```

该命令将有如下输出结果:

```
cmd@cmd-desktop:~$ file /sbin/ifconfig
/sbin/ifconfig: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
```

可见，该文件是一个 32 位的二进制可执行文件程序。

例 2：查看多个文件的类型。

查看多个文件的类型可使用“-f”参数，也可以在命令行中指定多个文件。例如使用如下命令查看多个文件的类型：

```
file /bin/c* #查看/bin下所有以c开头的文件
```

将有如下输出结果：

```
cmd@cmd-desktop:~$ file /bin/c* #查看/bin下所有以c开头的文件
#从左至右为文件名称、文件类型为32位elf二进制文件、使用动态库、去除调试信息
/bin/cat: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/chgrp: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/chmod: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/chown: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/chvt: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/cp: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
/bin/cpio: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
dynamically linked (uses shared libs), stripped
```

## 3.25 ftp (传输文件)



(1) 频度等级：☆☆☆

(2) 功能说明：

ftp 命令用于设置文件系统相关功能。

(3) 语法：

```
ftp [-dignv][主机名称或IP地址]
```

(4) 所属包：

```
ftp
```

(5) 使用技巧：

互联网上有很多向所有用户开放的 FTP 服务器，称为匿名 FTP 服务器，这种服务器登录的时候，使用“anonymous”账户，用任意的电子邮件作为密码。通常匿名 FTP 服务器不允许上传文件。

Linux 发行版的厂商都设置了匿名 FTP 服务器，允许用户免费下载 Linux 操作系统。

(6) 相关命令：

```
ftpcount、ftpsht、ftpwho
```

(7) 参数：

ftp 主要参数如表 3-28 所示。

表 3-28 ftp 参数表

参 数	含 义	参 数	含 义
-d	详细显示指令执行过程，便于排错或分析程序执行的情形	-n	不使用自动登录
-i	关闭互动模式，不询问任何问题	-v	显示指令执行过程
-g	关闭本地主机文件名称，支持特殊字符的扩充特性		

(8) 补充说明：

ftp 是 ARPANet 的标准文件传输协议，该网络就是现在 Internet 的前身。



ftp 命令使用内置的命令实现客户端和服务端间的操作，内置命令具体如表 3-29 所示。

表 3-29 ftp 内置命令列表

内置命令	功能
!	从 ftp 子系统退出到外壳
?	显示 ftp 命令说明。?与 help 相同 格式: ? [command] 说明: [command]指定需要帮助的命令名称。如果没有指定 command, ftp 将显示全部命令的列表
append	使用当前文件类型设置将本地文件附加到远程计算机上的文件 格式: append local-file [remote-file] 说明: local-file 指定要添加的本地文件
remote-file	指定要添加 local-file 的远程计算机上的文件。如果省略了 remote-file, 本地文件名将被用做远程文件名
ascii	将文件传送类型设置为默认的 ASCII 说明: FTP 支持两种文件传送类型, ASCII 码和二进制图像。在传送文本文件时应该使用 ASCII
bell	切换响铃以在每个文件传送命令完成后响铃。在默认情况下, 铃声是关闭的
binary (或 bi)	将文件传送类型设置为二进制
bye (或 by)	结束与远程计算机的 FTP 会话并退出 ftp
cd	更改远程计算机上的工作目录 格式: cd remote-directory 说明: remote-directory 指定要更改的远程计算机上的目录
close	结束与远程服务器的 FTP 会话并返回命令解释程序
debug	切换调试。当调试打开时, 发送到远程计算机的每个命令都打印, 前面是字符串 ">”。在默认情况下, 调试是关闭的
delete	删除远程计算机上的文件 格式: delete remote-file 说明: remote-file 指定要删除的文件
dir	显示远程目录文件和子目录列表 格式: dir [remote-directory] [local-file] 说明: remote-directory 指定要查看其列表的目录。如果没有指定目录, 将使用远程计算机中的当前工作目录。Local-file 指定要存储列表的本地文件。如果没有指定, 输出将显示在屏幕上
disconnect	从远程计算机断开, 保留 ftp 提示
get	使用当前文件转换类型将远程文件复制到本地计算机 格式: get remote-file [local-file] 说明: remote-file 指定要复制的远程文件
local-file	指定要在本地计算机上使用的名称。如果没有指定, 文件将命名为 remote-file
glob	切换文件名组合。组合允许在内部文件或路径名中使用通配符 (*和?)。在默认情况下, 组合是打开的
hash	切换已传输的每个数据块的数字签名 (#) 打印。数据块的大小是 2048 字节。在默认情况下, 散列符号打印是关闭的
help	显示 ftp 命令说明 格式: help [command] 说明: command 指定需要有关说明的命令名称。如果没有指定 command, ftp 将显示全部命令的列表
lcd	更改本地计算机上的工作目录。在默认情况下, 工作目录是启动 ftp 的目录 格式: lcd [directory] 说明: directory 指定要更改的本地计算机上的目录。如果没有指定 directory, 将显示本地计算机中当前的工作目录
literal	将参数逐字发送到远程 FTP 服务器。将返回单个的 FTP 回复代码 格式: literal argument [...] 说明: argument 指定要发送到 FTP 服务器的参数
ls	显示远程目录文件和子目录的缩写列表 格式: ls [remote-directory] [local-file] 说明: remote-directory 指定要查看其列表的目录。如果没有指定目录, 将使用远程计算机中的当前工作目录。local-file 指定要存储列表的本地文件。如果没有指定, 输出将显示在屏幕上
mdelete	删除远程计算机上的文件 格式: mdelete remote-files [...] 说明: remote-files 指定要删除的远程文件
mdir	显示远程目录文件和子目录列表。可以使用 mdir 指定多个文件 格式: mdir remote-files [...] local-file 说明: remote-files 指定要查看列表的目录。必须指定 remote-files。请输入 "-" 使用远程计算机上的当前工作目录
local-file	指定要还原列表的本地文件。请输入 "-" 在屏幕上显示列表

mget	使用当前文件传送类型将远程文件复制到本地计算机 格式: mget remote-files [...] 说明: remote-files 指定要复制到本地计算机的远程文件
mkdir	创建远程目录 格式: mkdir directory 说明: directory 指定新的远程目录的名称
mls	显示远程目录文件和子目录的缩写列表 格式: mls remote-files [...] local-file 说明: remote-files 指定要查看列表的文件。必须指定 remote-files; 请输入 "-" 使用远程计算机上的当前工作目录
(local-file)	指定要存储列表的本地文件。请输入 "-" 以在屏幕上显示列表
mput	使用当前文件传送类型将本地文件复制到远程计算机上 格式: mput local-files [...] 说明: local-files 指定要复制到远程计算机的本地文件
open	与指定的 FTP 服务器链接 格式: open computer [port] 说明: computer 指定要链接的远程计算机。可以通过 IP 地址或计算机名称指定计算机 (DNS 或主机文件必须可用)。如果自动登录打开 (默认), ftp 还将尝试自动将用户登录到 FTP 服务器 port 指定用来联系 FTP 服务器的端口号
prompt	切换提示。如果关闭提示时 mget 及 mput 传送所有文件, ftp 在多文件传送过程中将提示允许用户有选择地检索或存储文件。在默认情况下, 提示是打开的
put	使用当前文件传送类型将本地文件复制到远程计算机上 格式: put local-file [remote-file] 说明: local-file 指定要复制的本地文件。remote-file 指定要在远程计算机上使用的名称。如果没有指定, 文件将命名为 local-file
pwd	显示远程计算机上的当前目录
quit	结束与远程计算机的 FTP 会话并退出 ftp
quote	将参数逐字发送到远程 FTP 服务器中。将返回单个的 FTP 回复代码。quote 与 literal 相同 格式: quote argument [...] 说明: argument 指定要发送到 FTP 服务器的参数
recv	使用当前文件传送类型将远程文件复制到本地计算机。recv 与 get 相同 格式: recv remote-file [local-file] 说明: remote-file 指定要复制的远程文件
local-file	指定要在本地计算机上使用的名称。如果没有指定, 文件将命名为 remote-file
remotehelp	显示远程命令帮助 格式: remotehelp [command] 说明: command 指定需要帮助的命令的名称。如果没有指定 command, ftp 将显示全部远程命令的列表
rename	重命名远程文件 格式: rename filename newfilename 说明: filename 指定要重命名的文件。 newfilename 指定新的文件名
rmdir	删除远程目录 格式: rmdir directory 说明: directory 指定要删除的远程目录的名称
send	使用当前文件传送类型将本地文件复制到远程计算机上。send 与 put 相同 格式: send local-file [remote-file] 说明: local-file 指定要复制的本地文件。 remote-file 指定要在远程计算机上使用的名称。如果没有指定, 文件将命名为 local-file
status	显示 FTP 链接和切换的当前状态
trace	切换数据包跟踪。trace 在运行 ftp 命令时显示每个数据包的路由
type	设置或显示文件传送类型 格式: type [type-name] 说明: type-name 指定文件传送类型。默认设置为 ASCII。如果没有指定 type-name, 将显示当前的类型
user	指定远程计算机的用户 格式: user username [password] [account] 说明: user-name 指定登录到远程计算机所使用的用户名。password 指定 user-name 的密码。如果没有指定 (必须指定), ftp 会提示输入密码
account	指定登录到远程计算机所使用的账户。如果没有指定 account (需要指定), ftp 会提示输入账户
verbose	切换 verbose 模式。如果打开, 将显示所有 ftp 响应。在文件传送完成后, 将同时显示与传送效率有关的统计信息。在默认情况下, verbose 是打开的

### (9) 典型实例:

例 1: 使用 ftp 命令登录匿名 FTP 服务器。

例如使用 ftp 命令匿名登录 ftp.kernel.org 服务器，该服务是 Linux 内核的官方服务器，可以使用如下命令：

```
ftp ftp.kernel.org #发起链接请求
链接上服务器之后返回信息如下，此时要求输入用户名和密码，匿名用户输入“anonymous”。
cmd@cmd-desktop:~$ ftp ftp.kernel.org #发起链接请求
Connected to pub.all.kernel.org.
220 Welcome to ftp.kernel.org.
Name (ftp.kernel.org:hdd): anonymous #输入用户名，匿名用户输入 anonymous
331 Please specify the password. #输入密码，任意邮箱名，且不显示
Password:
230- Welcome to the #服务器欢迎信息
230-
230- LINUX KERNEL ARCHIVES
230- ftp.kernel.org
230-
230- "Much more than just kernels"
230-
230- IF YOU'RE ACCESSING THIS SITE VIA A WEB BROWSER
230- PLEASE USE THE HTTP URL BELOW INSTEAD!
230-
230-----> If you are looking for mirror sites, please go <----
230-----> to mirrors.kernel.org instead <----
230-
230-This site is provided as a public service by the Linux Kernel
230-Organization, a California nonprofit corporation. Bandwidth is
230-provided by The Internet Software Consortium, Inc. Our servers are
230-located in San Francisco and Palo Alto, California; Corvallis, Oregon;
230-Amsterdam, Netherlands and Umeå, Sweden; use in violation of any
230-applicable laws strictly prohibited.
230-
230-Due to U.S. Exports Regulations, all cryptographic software on this
230-site is subject to the following legal notice:
230-
230- This site includes publicly available encryption source code
230- which, together with object code resulting from the compiling of
230- publicly available source code, may be exported from the United
230- States under License Exception "TSU" pursuant to 15 C.F.R. Section
230- 740.13(e).
230-
230-This legal notice applies to cryptographic software only. Please see
230-the Bureau of Industry and Security (http://www.bis.doc.gov/) for more
230-information about current U.S. regulations.
230-
230-Neither the Linux Kernel Organization, nor its sponsors make any
230-guarantees, explicit or implicit, about the contents of this site.
230-Use at your own risk.
230-
230-This site is accessible via the following mechanisms:
230-
230-FTP ftp://ftp.kernel.org/pub/
230-HTTP http://www.kernel.org/pub/
```

```

230-RSYNC          rsync://rsync.kernel.org/pub/
230-
230-NFS and SMB/CIFS are no longer available.
230-
230-For comments on this site, please contact <ftpadmin@kernel.org>.
230-Please do not use this address for questions that are not related to
230-the operation of this site. Please see our homepage at
230-http://www.kernel.org/ for links to Linux documentation resources.
230-
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
    
```

#进入 ftp 命令提示符

此时，用户已经与服务器建立了匿名链接，“ftp>”是 ftp 命令提示符，使用命令列表中的 ftp 命令可以进行相应的操作。同时，用户也可以使用 help 命令查看帮助信息，输入 help 之后，终端将输出如下信息：

```

ftp> help
Commands may be abbreviated.  Commands are:

!      debug  mdir    qc      send
$      dir    mget    sendportsite
account disconnect mkdir  put    size
append exit    mls     pwd     status
ascii  form   mode   quit    struct
bell   get    modtime quote   system
binary glob   mput   recv    sunique
bye    hash   newer  reget   tenex
case   help  nmap   rstatus tick
cd     idle  nlist  rhelp   trace
cdup   image ntrans rename  type
chmod  lcd   open   reset   user
close  ls    prompt restart umask
cr     macdef passive rmdir   verbose
delete mdelete proxy  runique ?
ftp>
    
```

#使用 help 命令查看帮助信息

#命令列表

#回到 ftp 命令提示符

例 2 使用 FTP 命令。

例如可以使用“cd”命令切换在 FTP 服务器中的当前目录，使用“ls”命令列出当前目录内容，结果如下：

```

ftp> pwd
257 "/"
ftp> cd /pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrws---  2 536      536      71 Sep 23   2008 RCS
-r--r--r--  1 536      536     1912 Aug 05   2007 README
-r--r--r--  1 536      536     578 Mar 18   2003 README_ABOUT_BZ2_FILES
drwxrwsr-x  7 536      536     93 Jul 22   2005 dist
-r--r--r--  1 536      536    2322 Sep 23   2008 index.html
drwxrwsr-x 10 536      536     120 Feb 12   2009 linux
drwxr-x---  2 536      536     6 Oct 27   1998 lost+found
    
```

#查看当前路径

#当前路径为根目录

#转到/pub 目录

#切换目录成功提示

#列出当前路径下文件列表

#返回信息

#返回列表

```
drwxrwsr-x   3 536   536   18 Sep 23 2008 media
drwxrwsr-x  19 536   536   4096 Jun 02 2008 scm
drwxrwsr-x   3 536   536   59 Nov 05 2003 site
drwxrwsr-x  13 536   536   154 Aug 22 18:03 software
drwxr-sr-x   3 536   536   22 Apr 30 2008 tools
226 Directory send OK.          #列表完成
```

例 3: 使用“get”命令下载文件。

例如下载“/pub”目录下的 README 文件, 使用如下命令:

```
get README          #下载 README 文件
```

下载结果如下:

```
ftp> get README          #下载 README 文件
local: README remote: README      #开始下载指定文件
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for README (1912 bytes).
226 File send OK.
1912 bytes received in 1.12 secs (1.7 KB/s)      #下载完成
```

下载的文件保存在当前目录中。

如果想离开 FTP 服务器可以使用 quit 命令, 返回的离开信息如下:

```
ftp> quit
221 Goodbye.
```

## 3.26 ftpcount (显示当前以 FTP 登录的用户人数)

(1) 频度等级: ☆

(2) 功能说明:

ftpcount 命令用于显示目前以 FTP 登录的用户人数, 同时能够显示 FTP 服务器设定的登录人数限制。

(3) 语法:

```
ftpcount
```

(4) 所属包:

```
proftpd
```

(5) 使用技巧:

执行这项指令可知用 FTP 登录系统的人数, 以及 FTP 登录人数的上限。

(6) 相关命令:

```
ftp、ftpsht、ftpwho
```

(7) 参数:

ftpcount 的主要参数如表 3-30 所示。

表 3-30 ftpcount 参数表

参 数	含 义
-f<设定文件>	指定设定文件所在路径
-h,--help	显示帮助信息

(8) 补充说明:

该命令只有部分 Linux 发行版本 (如 Mandrake Linux) 才有, 属于 wu-ftp, 可另行安装。

(9) 典型实例:

例: 查询当前 FTP 用户的人数。

ftpcount 可以直接查询 FTP 服务器上用户的人数, 可直接使用如下命令:

```
ftpcount          #查询当前 FTP 用户的人数
```

该命令有如下输出结果:



```
cmd@cmd-desktop:~$ ftpcount #查询当前 FTP 用户的人数
Master proftpd process 2085:
Service class - 6 user#当前共 6 个用户登录到服务器
```

## 3.27 ftpshut ( 在指定的时间关闭 FTP 服务器 )

(1) 频度等级: ☆

(2) 功能说明:

ftpshut 命令用于在指定的时间关闭 FTP 服务器。

(3) 语法:

```
ftpshut [-d<分钟>][-l<分钟>][关闭时间][“警告信息”]
```

(4) 所属包:

```
proftpd
```

(5) 使用技巧:

利用该命令可以在指定的时间关闭 FTP 服务器, 属于 wu-ftp, 可另行安装。

(6) 相关命令:

```
ftp、ftpshut、ftpwho
```

(7) 参数:

ftpshut 主要参数如表 3-31 所示。

表 3-31 ftpshut 参数表

参 数	含 义
-d<分钟>	切断所有 FTP 连线时间
-l<分钟>	停止接受 FTP 登录的时间

(8) 补充说明:

本指令提供系统管理者在设置的时间关闭 FTP 服务器, 且能在关闭之前发出警告信息通知用户。关闭时间若设置后为 “none”, 则会马上关闭服务器。如果采用 “+30” 的方式来设置表示服务器在 30 分钟之后关闭。以此类推, 假设使用 “1130” 的格式则代表服务器会在每日的 11 时 30 分关闭 (时间格式为 24 小时制)。FTP 服务器关闭后, 在 “/etc” 目录下会产生一个名称为 “shutmsg” 的文件, 把它删除后即可再度启动 FTP 服务器的功能。

(9) 典型实例:

例: 在晚上 11:00 关闭 FTP 服务器, 并在关闭前 5 分钟拒绝新的 FTP 登录, 前 3 分钟关闭所有 ftp 的连接, 且给出警告信息, 可使用如下命令:

```
ftpshut-d 3 -l 5 1100 “Server will be shutdown at 23:00:00” #23:00 关闭 FTP 服务器, 关闭前 5 分钟拒绝新的 FTP 登录, 前 3 分钟关闭所有 FTP 的连接, 且给出警告信息
```

该命令没有输出, 而关闭服务器。

## 3.28 ftpwho ( 显示当前所有以 FTP 登录的用户信息 )

(1) 频度等级: ☆

(2) 功能说明:

ftpwho 命令用于显示目前所有以 FTP 登录的用户信息。

(3) 语法:

```
ftpwho
```

(4) 所属包:

```
proftpd
```

(5) 使用技巧:

执行这项指令可得知目前用 FTP 登录系统的用户有哪些人, 以及他们正在进行哪些操作。

(6) 相关命令:

ftp、ftpsht、ftpcount

(7) 参数:

ftpwho 的主要参数如表 3-32 所示。

表 3-32 ftpwho 参数表

参 数	含 义
-v	显示版本信息

(8) 补充说明:

该命令只有部分 Linux 发行版本 (如 Mandrake Linux) 才有, 属于 wu-ftp, 可另行安装。

(9) 典型实例:

例: 查询当前有哪些用户正在登录 FTP 服务器, 可直接使用如下命令:

```
ftpwho #查询当前有哪些用户正在登录 FTP 服务器
```

该命令有如下输出结果:

```
cmd@cmd-desktop:~$ ftpwho #查询当前正在登录 FTP 服务器的用户
standalone FTP daemon[2085]:
  3547 wyw [1m20s] 1m25s(idle)
Service class - 1 user#当前有一个用户登录 FTP 服务器
```

## 3.29 find ( 查找文件或目录 )



(1) 频度等级: ☆☆☆

(2) 功能说明:

查找文件或目录, 并可对查找到的文件执行指定的操作。find 命令查找是从指定目录向下递归搜索各个子目录, 找到满足查找条件的文件后将其显示在标准输出中, 并对其进行指定的操作。

(3) 语法:

```
find [目录...][-amin <分钟>][-anewer <参考文件或目录>][-atime <24 小时数>][-cmin <分钟>][-cnewer <参考文件或目录>][-ctime <24 小时数>][-daystart][-depth][-empty][-exec <执行指令>][-false][-fls <列表文件>][-follow][-fprint <列表文件>][-fprint0 <列表文件>][-fprintf <列表文件><输出格式>][-fstype <文件系统类型>][-gid <群组识别码>][-group <群组名称>][-help] [-ilname <范本样式>][-iname <范本样式>][-inum <inode 编号>][-ipath <范本样式>][-iregex <范本样式>][-links <链接数目>][-lname <范本样式>][-ls][-maxdepth <目录层级>][-mindepth <目录层级>][-mmin <分钟>][-mount][-mtime <24 小时数>][-name <范本样式>][-newer <参考文件或目录>][-nogroup] [noleaf][-nouser][-ok <执行指令>][-path <范本样式>][-perm <权限数值>] [-print][-print0] [-printf <输出格式>][-prune][-regex <范本样式>][-size <文件大小>][-true][-type <文件类型>][-uid <用户识别码>][-used <日数>][-user <拥有者名称>][-version][-xdev][-xtype <文件类型>]
```

(4) 所属包:

```
findutils
```

(5) 使用技巧:

不带参数使用该命令可以打印当前目录下的所有内容。

find 命令提供的查找条件可以是逻辑运算符与 (and)、或 (or)、非 (not) 等组成的复合条件。其中参数“-a”表示逻辑与 (是默认参数), 表示只有当所有给定的条件都满足时, 才符合查找条件; 参数“-o”表示逻辑或, 表示只要有一个条件满足, 就符合查找条件; 参数“!”表示逻辑非, 它表示与所给定的条件相反的文件。

find 命令还可以对查找到的文件进行相关的操作。

参数“-exec”的含义是将所有匹配的文件全部传输给“exec”去执行。但是，有些系统对该参数传递的命令长度有限制。因此，使用该命令时，如果查找文件过多，可能出现“参数太长”或“参数溢出”等错误。此时，可使用 xarg 命令分批处理文件。

(6) 相关命令：

locate、updatedb

(7) 参数：

find 主要参数如表 3-33 所示。

表 3-33 find 参数表

参 数	含 义
-amin<分钟>	查找在指定时间曾被存取过的文件或目录，单位以分钟计算
-anewer<参考文件或目录>	查找其存取时间较指定文件或目录的存取时间更接近现在的文件或目录
-atime<24 小时数>	查找在指定时间内曾被存取过的文件或目录，单位以 24 小时计算
-cmin<分钟>	查找在指定时间之时被更改的文件或目录
-cnewer<参考文件或目录>	查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录
-ctime<24 小时数>	查找在指定时间被更改的文件或目录，单位以 24 小时计算
-daystart	从本日开始计算时间
-depth	从指定目录下最深层的子目录开始查找
-empty	寻找文件大小为 0Byte 的文件，或目录下没有任何子目录或文件的空目录
-exec<执行指令>{} \;	如果 find 指令的回传值为 True，就执行该指令。{} 表示将查找到的文件作为命令参数；命令最后必须是固定字符“\”。另外，使用该命令时，会提示用户是否执行该命令
-false	将 find 指令的回传值皆设为 False
-fls<列表文件>	此参数的效果和指定“-ls”参数类似，但会把结果保存为指定的列表文件
-follow	排除符号链接
-fprint<列表文件>	此参数的效果和指定“-print”参数类似，但会把结果保存成指定的列表文件
-fprint0<列表文件>	此参数的效果和指定“-print0”参数类似，但会把结果保存成指定的列表文件
-fprintf<列表文件><输出格式>	此参数的效果和指定“-printf”参数类似，但会把结果保存成指定的列表文件
-fstype<文件系统类型>	只查找该文件系统类型下的文件或目录
-gid<群组识别码>	查找符合指定的群组识别码的文件或目录
-group<群组名称>	查找符合指定的群组名称的文件或目录
-help 或 -help	显示帮助信息
-iname<范本样式>	此参数的效果和指定“-lname”参数类似，但忽略字符大小写的差别
-iname<范本样式>	此参数的效果和指定“-name”参数类似，但忽略字符大小写的差别
-inum<inode 编号>	查找符合指定的 inode 编号的文件或目录
-ipath<范本样式>	此参数的效果和指定“-lpath”参数类似，但忽略字符大小写的差别
-iregex<范本样式>	此参数的效果和指定“-regex”参数类似，但忽略字符大小写的差别
-links<链接数目>	查找符合指定的硬链接数目的文件或目录
-lname<范本样式>	指定字符串作为寻找符号链接的范本样式
-ls	如果 find 指令的回传值为 True，就将文件或目录名称列出到标准输出
-maxdepth<目录层级>	设置最大目录层级
-mindepth<目录层级>	设置最小目录层级
-mmin<分钟>	查找在指定时间曾被更改过的文件或目录，单位以分钟计算
-mount	此参数的效果和指定“-xdev”相同
-mtime<24 小时数>	查找在指定时间曾被更改过的文件或目录，单位以 24 小时计算
-name<范本样式>	指定字符串作为查找文件或目录的范本样式
-newer<参考文件或目录>	查找更改时间较指定文件或目录的更改时间更接近现在的文件或目录
-nogroup	找出不属于本地主机群组识别码的文件或目录
-noleaf	不去考虑目录至少需要两个硬链接存在
-nouser	找出不属于本地主机用户识别码的文件或目录
-ok<执行指令>	此参数的效果和指定“-exec”参数类似，但在执行指令之前会先询问用户，若回答“y”或“Y”，则放弃执行指令
-path<范本样式>	指定字符串作为查找目录的范本样式，范例样式可以使用通配符“*”、“?”、“[]”

-perm<权限数值>	查找符合指定的权限数值的文件或目录
-print	如果 find 指令的回传值为 True，就将文件或目录名称列出到标准输出格式为每列一个名称，每个名称之前皆有“/”字符串
-print0	如果 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。格式为全部的名称皆在同一行
-printf<输出格式>	如果 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。格式可以自行指定
-prune	不指定字符串作为查找文件或目录的范本样式
-regex<范本样式>	指定字符串作为查找文件或目录的范本样式
-size<文件大小>	查找符合指定的文件大小的文件
-l	将 find 指令的回传值皆设为 True
-type<文件类型>	只查找符合指定文件类型的文件。支持的文件类型有如下几种 <ul style="list-style-type: none"> <li>● b: 块设备文件</li> <li>● c: 字符设备文件</li> <li>● d: 目录文件</li> <li>● p: 命名管道 (FIFO)</li> <li>● f: 普通文件</li> <li>● l: 符号链接文件 (symbolic links)</li> <li>● s: socket 文件</li> </ul>
-uid<用户识别码>	查找符合指定用户识别码的文件或目录
-used<日数>	查找文件或目录被更改之后，在指定时间曾被存取过的文件或目录，单位以日计算
-user<拥有者名称>	查找符合指定拥有者名称的文件或目录
-version 或 --version	显示版本信息
-xdev	将范围局限在先执行的文件系统中
-xtype<文件类型>	此参数的效果和指定“-type”参数类似，差别在于它针对符号链接检查

#### (8) 补充说明:

find 指令用于查找符合条件的文件。任何位于参数之前的字符串都将被视为要查找的目录。

#### (9) 典型实例:

例 1: 直接在终端中使用 find 命令，可以列出当前目录下所有的子目录及文件（包括隐藏文件）。

例如想查看当前目录下的所有子目录和文件，可使用如下命令:

```
find #查看当前目录下的子目录和文件
```

将得到如下结果:

```
cmd@cmd-desktop:~/test$ find #查看当前目录下的子目录和文件
. #当前目录下所有文件及子目录列表
./testfile1
./file
./testfile2
./testfile
```

例 2: 查找指定时间内修改过的文件。

例如查找当前目录下所有在两个小时之内修改过的文件，可使用如下命令:

```
find-atime -2 #当前目录下所有在两个小时之内修改过的文件
```

查找结果如下:

```
cmd@cmd-desktop:~/test$ find-atime -2 #查看当前目录下所有在两个小时之内修改过的文件
./testfile1 #查找到的文件列表
./testfile2
./testfile
```

例 3: 查找指定字符串类型的文件。

例如查找“/etc/cups”目录下以“.conf”为扩展名的文件，可以使用如下命令:

```
find /etc/cups -name *.conf #查找“/etc/cups”目录下以“.conf”为扩展名的文件
```

查找结果如下:

```
cmd@cmd-desktop:~$ find /etc/cups -name *.conf #查找“/etc/cups”目录下以“.conf”为扩展名的文件
```

```
/etc/cups/pdftops.conf #查找到的文件列表
/etc/cups/acroread.conf
/etc/cups/cupsd.conf
/etc/cups/snmp.conf
```

例 4: 操作当前查找到的文件, 可使用参数“-exec”。例如将查找到的文件权限修改, 可使用如下命令:

```
find-exec chown-R root:root {} \; #将查找到的文件权限修改
```

使用该命令前, 查看文件属性如下:

```
cmd@cmd-desktop:~/test$ ls-l #查看文件原来的所有者和所属组
总用量 12
-rw-r--r-- 1 cmd cmd 0 2009-09-21 09:46 file#文件所有者和所属组分别为 cmd、cmd
-rw-r--r-- 1 cmd cmd 14 2009-09-21 09:44 testfile
-rw-r--r-- 1 cmd cmd 13 2009-09-21 09:45 testfile1
-rw-r--r-- 1 cmd cmd 26 2009-09-21 09:45 testfile2
```

修改属性后的结果如下:

```
cmd@cmd-desktop:~/test$ sudo find -exec chown -R root:root {} \; #查找并修改文件所有者和所属组
```

```
cmd@cmd-desktop:~/test$ ls-l #再次查看文件属性
总用量 12
-rw-r--r-- 1 root root 0 2009-09-21 09:46 file #文件所有者和所属组分别为 root、root
-rw-r--r-- 1 root root 14 2009-09-21 09:44 testfile
-rw-r--r-- 1 root root 13 2009-09-21 09:45 testfile1
-rw-r--r-- 1 root root 26 2009-09-21 09:45 testfile2
```

## 3.30 gpg (文件的签字、检查、加密和解密)

(1) 频度等级: ☆

(2) 功能说明:

gpg (GNU Privacy Guard) 是用来加密数据与制作证书的一套工具, 其作用与 PGP 类似, 是为文件签字、检查文件及对文件加密或解密。

(3) 语法:

```
gpg [选项] [文件名]
```

(4) 所属包:

```
gnupg
```

(5) 使用技巧:

gpg 支持的算法如下:

- 支持的公钥有 RSA、RSA-E、RSA-S、ELG-E、DSA。
- 支持的对称加密算法有 3DES、CAST5、BLOWFISH、AES、AES192、AES256、TWOFISH。
- 支持的散列有 MD5、SHA1、RIPEMD160、SHA256、SHA384、SHA512。
- 对压缩文件的支持包括不压缩、ZIP、ZLIB、BZIP2。

(6) 相关命令:

```
openssl
```

(7) 参数:

gpg 的主要参数如表 3-34 所示。

表 3-34 gpg 参数表

参 数	含 义	参 数	含 义
命令 -s, --sign [文件名]	生成一份签字	命令 --refresh-keys	从公钥服务器中更新所有的本地密钥



--clearsign [文件名]	生成一份明文签字	选项	--import	导入/合并密钥
-b, --detach-sign	生成一份分离的签字		--card-status	打印卡状态
-e, --encrypt	加密数据		--card-edit	更改卡上的数据
-c, --symmetric	仅使用对称加密		--change-pin	更改卡的 PIN
-d, --decrypt	解密数据 (默认)		--update-trustdb	更新信任度数据库
--verify	验证签字		--print-md 算法 [文件]	使用指定的散列算法打印报文散列值
--list-keys	列出密钥		-a, --armor	输出经 ASCII 封装
--list-signs	列出密钥和签字		-r, --recipient 某甲	为收件者“某甲”加密
--check-signs	列出并检查密钥签字		-u, --local-user	使用当前用户标识来签字或解密
--fingerprint	列出密钥和指纹		-z N	设定压缩等级为 N (0 表示不压缩)
-K, --list-secret-keys	列出私钥		--textmode	使用标准的文本模式
--gen-key	生成一副新的密钥对		-o, --output	指定输出文件
--delete-keys	从公钥钥匙环里删除密钥		-v, --verbose	详细模式
--delete-secret-keys	从私钥钥匙环里删除密钥		-n, --dry-run	不做任何改变
--sign-key	为某把密钥添加签字		-i, --interactive	覆盖前先询问
--lsign-key	为某把密钥添加本地签字		--openpgp	行为严格遵循 OpenPGP 定义
--edit-key	编辑某把密钥或为其添加签字		--pgp2	生成与 PGP 2.x 兼容的报文
--gen-revoke	生成一份吊销证书		-se -r Bob [文件名]	为 Bob 这个收件人签字及加密
--export	导出密钥		--clearsign [文件名]	做出明文签字
--send-keys	把密钥导出到某个公钥服务器上		--detach-sign [文件名]	做出分离式签字
--recv-keys	从公钥服务器上导入密钥	--list-keys [某甲]	显示密钥	
--search-keys	在公钥服务器上搜寻密钥	--fingerprint [某甲]	显示指纹	

#### (8) 补充说明:

gpg 命令为二进制程序, XFOCUS 未对其进行检查。

#### (9) 典型实例:

例: 生成密钥。

例如想生成一个用来发送加密数据和进行解密数据的密钥。执行 gpg 命令, 在主目录下创建一个 .gnupg 子目录, 具体命令如下:

```
gpg --gen-key #生成密钥
```

在该子目录里面有一个 gpg.conf 的配置文件, 它里面是 gpg 工具的各种配置选项及其默认设置值。得到如下结果:

```
cmd@cmd-desktop:~$ gpg --gen-key #生成密钥
gpg (GnuPG) 1.2.4; Copyright (C) 2003 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.
Please select what kind of key you want: #选择密钥类型
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair. #选择密钥长度
    minimum keysize is 768 bits
    default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024) 768
Requested keysize is 768 bits
Please specify how long the key should be valid #选择密钥有效期, 0 代表没有期限
```



`gunzip` 命令用于解压由 `gzip` 命令压缩的打包文件。`gunzip` 是个使用广泛的解压程序，它用于解开被 `gzip` 压缩过的文件，这些压缩文件预设最后的扩展名为 “.gz”。

(3) 语法：

```
gunzip [-acfhlLnNqrtvV][-S<压缩字尾字符串>][文件...] 或 gunzip [-acfhlLnNqrtvV][-S<压缩字尾字符串>][目录]
```

(4) 所属包：

```
gzip
```

(5) 使用技巧：

`gunzip` 命令可以解压缩由 `gzip`、`zip`、`compress`、`compress-H` 及 `pack` 命令生成的压缩文件。对于由 `gzip` 生成的压缩文件，如果压缩文件中只有一个文件，并且是以 “deflation” 方式压缩的，可以由 `gunzip` 解压。

`gunzip` 命令可以自动识别压缩文件的压缩格式，但仅限于 “.tgz”、“.tar.gz”、“.tar.Z” 三种格式。对于这三种以外的压缩格式，需要使用 “-S” 参数指明压缩文件的扩展名。

在大部分 Linux 的发行版本中，`gunzip` 命令是 `gzip` 命令的符号链接或硬链接。

(6) 相关命令：

```
gzip
```

(7) 参数：

`gunzip` 主要参数如表 3-35 所示。

表 3-35 `gunzip` 参数表

参 数	含 义
-a 或 --ascii	使用 ASCII 文字模式
-c 或 --stdout 或 --to-stdout	把解压后的文件输出到标准输出设备
-f 或 --force	强行解开压缩文件，不理睬文件名称或硬链接是否存在，以及该文件是否为符号链接
-h 或 --help	显示帮助信息
-l 或 --list	列出压缩文件的相关信息
-L 或 --license	显示版本与版权信息
-n 或 --no-name	解压缩时，若压缩文件内含有原来的文件名称及时间戳记，则将其忽略不予处理
-N 或 --name	解压缩时，若压缩文件内含有原来的文件名称及时间戳记，则将其回存到解压的文件中
-q 或 --quiet	不显示警告信息
-r 或 --recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-S<压缩字尾字符串> 或 --suffix<压缩字尾字符串>	更改压缩字尾字符串
-t 或 --test	测试压缩文件是否正确无误
-v 或 --verbose	显示指令执行过程
-V 或 --version	显示版本信息

(8) 补充说明：

事实上 `gunzip` 就是 `gzip` 的硬链接，因此不论是压缩或解压缩，都可通过 `gzip` 指令单独完成。

(9) 典型实例：

例：解压缩由 `gzip` 命令压缩的文件。

例如解压缩由 `gzip` 压缩而来的压缩文件 `testfile.gz`，可使用如下命令：

```
gunzip-v testfile.gz #解压缩 gz 文件
```

该命令将输出如下信息：

```
cmd@cmd-desktop:~/test$gunzip -v testfile.gz      #解压缩打包文件
testfile.gz:  -14.3% -- replaced with testfile    #从左至右为原文件名、压缩比、生成解压#缩名
```

## 3.32 gzexe ( 压缩可执行文件 )



(1) 频度等级：☆

(2) 功能说明:

压缩可执行文件。该命令可以压缩可执行程序, 并且在执行此程序时能自动完成解压缩。

(3) 语法:

```
gzexe [-d][执行文件...]
```

(4) 所属包:

gzip

(5) 使用技巧:

如果可执行文件设置了 suid 权限, 则不能使用 gzexe 来压缩。

该命令适合在磁盘空间较低的情况下使用。

程序在使用该命令后生成新的应用程序, 该应用程序是一个 bash 脚本程序。它的前半部分是用于启动 gzexe 命令的 shell 脚本, 后半部分是压缩后的二进制文件的内容。

(6) 相关命令:

zforce、gzip

(7) 参数:

gzexe 只有“-d”一个参数, 如表 3-36 所示。

表 3-36 gzexe 参数表

参 数	含 义
-d	解开压缩文件

(8) 补充说明:

gzexe 是用来压缩执行文件的程序。当执行被压缩过的执行文件时, 该文件会自动解压然后继续执行, 和使用一般的执行文件相同。

(9) 典型实例:

例: 压缩可执行程序。

例如压缩可执行程序“/bin/ip”, 将可执行程序传递给 gzexe 命令后, 即可压缩可执行程序。具体命令如下:

```
gzexe /bin/ip #压缩可执行程序 ip, 源文件将改为 ip~, 压缩后的文件被命名为 ip
```

输出信息为:

```
cmd@cmd-desktop: $ gzexe /bin/ip #压缩可执行程序 ip
/bin/ip: 58.3% #分别为文件名、压缩比率
```

使用 file 命令查看文件属性, 得到如下信息:

```
cmd@cmd-desktop:~$ file /bin/ip #查看文件属性
ip: Bourne-Again shell script text executable #压缩后 ip 属性
```

从输出可以看出, 压缩后的 ip 文件类型是 bash 脚本程序。

## 3.33 gzip ( 压缩/解压缩文件 )



(1) 频度等级: ☆☆☆

(2) 功能说明:

gzip 命令主要用于压缩文件。GZIP 最早由 Jean-loup Gailly 和 Mark Adler 创建, 用于 UNIX 系统的文件压缩。我们在 Linux 中经常会用到后缀为.gz 的文件, 它们就是 GZIP 格式的。现在它已经成为 Internet 上使用非常普遍的一种数据压缩格式, 或者说是一种文件格式。HTTP 协议上的 GZIP 编码是一种用来改进 Web 应用程序性能的技术。大流量的 Web 站点常常使用 GZIP 压缩技术来让用户感受更快的速度。

(3) 语法:

```
gzip [ -acdfhlLnNqr tvV][ -S <压缩字尾字符串>][ -<压缩效率>][ - -best / fast ] [文件...]
gzip [ -acdfhlLnNqr tvV][ -S <压缩字尾字符串>][ -<压缩效率>][ --best/fast ][目录]
```

(4) 所属包:

gzip

(5) 使用技巧:

使用“-d”参数可以是对压缩文件进行解压缩。

(6) 相关命令:

gunzip、zcat

(7) 参数:

gzip 主要参数如表 3-37 所示。

表 3-37 gzip 参数表

参 数	含 义
-a 或--ascii	使用 ASCII 文字模式
-c 或--stdout 或--to-stdout	把压缩后的文件输出到标准输出设备，不去更改原始文件
-d 或--decompress 或--uncompress	解开压缩文件
-f 或--force	强行压缩文件。不理睬文件名称或硬链接是否存在，以及该文件是否为符号链接
-l 或--list	列出压缩文件的相关信息
-L 或--license	显示版本与版权信息
-n 或--no-name	压缩文件时，不保存原来的文件名称及时间戳记
-N 或--name	压缩文件时，保存原来的文件名称及时间戳记
-q 或--quiet	不显示警告信息
-r 或--recursive	递归处理，将指定目录下的所有文件及子目录一并处理
-S<压缩字尾字符串> 或 ---suffix<压缩字尾字符串>	更改压缩字尾字符串
-t 或--test	测试压缩文件是否正确无误
-v 或--verbose	显示指令执行过程
<压缩效率>	压缩效率是一个介于 1-9 的数值，预设值为“6”，指定越大的数值，压缩效率就会越高
--best	此参数的效果和指定“-9”参数相同
--fast	此参数的效果和指定“-1”参数相同

(8) 补充说明:

gzip 是一个使用广泛的压缩程序，文件经它压缩过后，其名称后面会多出“.gz”的扩展名。

(9) 典型实例:

例 1: 使用 gzip 命令压缩文件。

例如使用 gzip 命令压缩当前目录下的 testfile 文件，可使用如下命令:

```
gzip -v testfile #使用 gzip 命令压缩当前目录下 testfile 文件，参数 -v 表示显示详细信息
```

压缩结果如下:

```
cmd@cmd-desktop:~/test$ gzip -v testfile #使用 gzip 命令压缩当前目录下 testfile 文件
testfile: -14.3% -- replaced with testfile.gz #从左至右为原文件名、压缩比、生成压缩名
```

例 2: 查看压缩文件的详细信息。

例如查看上例中生成的压缩文件的详细信息，可使用如下命令:

```
gzip -l testfile.gz #查看压缩文件的详细信息
```

查看的结果如下:

```
cmd@cmd-desktop:~/test$ gzip -l testfile.gz #查看压缩文件详细信息
      compressed      uncompressed  ratio uncompressed_name
            43              14    -14.3%      testfile
#压缩文件的详细信息，包括压缩文件大小、压缩比率、解压后文件名称等信息
```

例 3: 解压缩文件。

例如将上例中生成的压缩文件解压缩，可使用参数“-d”，具体命令如下:

```
gzip -v -d testfile.gz #解压缩打包文件
```

使用上述命令将有如下输出:

```
cmd@cmd-desktop:~/test$ gzip -v -d testfile.gz #解压缩打包文件，-v 表示显示详细过程
testfile.gz: -14.3% -- replaced with testfile
```



#从左至右为原文件名、压缩比、生成压缩名

# 3.34 indent (调整 C 源代码文件的格式)



(1) 频度等级: ☆

(2) 功能说明:

indent 可辨识 C 的原始代码文件并格式化, 以方便程序设计师阅读。

(3) 语法:

```
indent [参数][源文件]
```

```
indent [参数][源文件][-o 目标文件]
```

(4) 所属包:

```
indent
```

(5) 使用技巧:

该命令常用于 shell 脚本中。

(6) 参数:

indent 主要参数如表 3-38 所示。

表 3-38 indent 参数表

参 数	含 义
-bad 或--blank-lines-after-declarations	在声明区段后加上空白行
-bap 或--blank-lines-after-procedures	在程序后加上空白行
-bbb 或--blank-lines-after-block-comments	在注释区段后加上空白行
-bc 或--blank-lines-after-commas	在声明区段中, 若出现逗号即换行
-bl 或--braces-after-if-line	If (或是 else、for 等) 与后面执行区段的“{”不同行, 且“}”自成一
-bli<缩排格数>或--brace-indent<缩排格数>	设置{ }缩排的格数
-br 或--braces-on-if-line	If (或是 else、for 等) 与后面执行区段的“{”不同行, 且“}”自成一
-bs 或--blank-before-sizeof	在 sizeof 之后空一格
-c<栏数>或--comment-indentation<栏数>	将注释置于程序码右侧指定的字段
-cd<栏数>	将注释置于声明右侧指定的字段
或--declaration-comment-column<栏数>	
-cdb 或--comment-delimiters-on-blank-lines	注释符号自成一
-ce 或--cuddle-else	将 else 置于“}”(if 执行区段的结尾)之后
-ci<缩排格数>	叙述过长需要换行时, 指定换行后缩排的格数
或--continuation-indentation<缩排格数>	
-cli<缩排格数>或--case-indentation<缩排格数>	使用 case 时, switch 缩排的格数
-cp<栏数>或--else-endif-column<栏数>	将注释置于 else 与 elseif 叙述右侧指定的字段
-cs 或--space-after-cast	在 cast 之后空一格
-d<缩排格数>或--line-comments-indentation<缩排格数>	针对不是放在程序码右侧的注释, 设置其缩排格数
-di<栏数>或--declaration-indentation<栏数>	将声明区段的变量置于指定的字段
-fcl 或--format-first-column-comments	针对放在每行最前端的注释, 设置格式
-fca 或--format-all-comments	设置所有注释的格式
-gnu 或--gnu-style	指定使用 GNU 的格式, 此为预设值
-i<格数>或--indent-level<格数>	设置缩排的格数
-ip<格数>或--parameter-indentation<格数>	设置参数的缩排格数
-kr 或--k-and-r-style	指定使用 Kernighan&Ritchie 的格式
-lp 或--continue-at-parentheses	如果叙述过长需要换行, 且叙述中包含了括弧, 则将括弧中的每行起始字段内容垂直对齐排列
-nbad 或--no-blank-lines-after-declarations	在声明区段后不要加上空白行
-nbap 或--no-blank-lines-after-procedures	在程序后不要加上空白行
-nbbb 或--no-blank-lines-after-block-comments	在注释区段后不要加上空白行

-nbc 或--no-blank-lines-after-commas	在声明区段中，即使出现逗号，仍旧不要换行
-ncdb 或--no-comment-delimiters-on-blank-lines	注释符号不要自成一行为
-nce 或--dont-cuddle-else	不要将 else 置于 “}” 之后
-ncs 或--no-space-after-casts	不要在 cast 之后空一格
-nfc1 或--dont-format-first-column-comments	不要格式化放在每行最前端的注释
-nfca 或--dont-format-comments	不要格式化任何的注释
-nip 或--no-parameter-indentation	参数不要缩排
-nlp 或--dont-line-up-parentheses	当叙述过长需要换行，且叙述中包含了括弧时，不用将括弧中的每行起始段垂直对齐排列
-npcs 或--no-space-after-function-call-names	在调用的函数名称之后，不要加上空格
-npro 或--ignore-profile	不要读取 indent 的配置文件 .indent.pro
-npsl 或--dont-break-procedure-type	程序类型与程序名称放在同一行
-nsc 或--dont-star-comments	注释左侧不要加上星号 (*)
-nsob 或--leave-optional-semicolon	不用处理多余的空白行
-nss 或--dont-space-special-semicolon	若 for 或 while 区段仅有一行，在分号前不加上空格
-nv 或--no-verbosity	不显示详细的信息
-orig 或--original	使用 Berkeley 的格式
-pcs 或--space-after-procedure-calls	在调用的函数名称与 “{” 之间加上空格
-psl 或--procnames-start-lines	程序类型置于程序名称的前一行
-sc 或--start-left-side-of-comments	在每行注释左侧加上星号 (*)
-sob 或--swallow-optional-blank-lines	删除多余的空白行
-ss 或--space-special-semicolon	若 for 或 swile 区段仅有一行，在分号前加上空格
-st 或--standard-output	将结果显示在标准输出设备
-T	数据类型名称缩排
-ts<格数>或--tab-size<格数>	设置 tab 的长度
-v 或--verbose	执行时显示详细的信息

### (7) 典型实例：

例：调整 C 源码文件格式。例如，想调整 C 源文件 file.c，可直接使用如下命令：

```
indent file.c #调整 C 源码文件格式
```

使用该命令前，先用 cat 命令查看 file.c 的内容，调整之后再次查看，结果如下：

```
hdd@hdd-desktop:~$ cat file.c #查看 file.c 的原格式
#include <stdio.h>
void main ()
{printf ("hello,this is a linux file!"); //print message
}
hdd@hdd-desktop:~$ indent file.c #调整格式
hdd@hdd-desktop:~$ cat file.c #再次查看
#include <stdio.h> #C 源文件使用缩进格式显示
void
main ()
{
printf ("hello,this is a linux file!"); //print message
}
```

## 3.35 ln ( 链接文件或目录 )



(1) 频度等级：☆☆☆

(2) 功能说明：

ln 命令为某一个文件在另外一个位置建立一个同步的链接。

(3) 语法：

```
ln [-bdfinsv][-S <字尾备份字符串>][-v <备份方式>][--help][--version][源文件或目录][目标文件或目录]
ln [-bdfinsv][-S <字尾备份字符串>][-v <备份方式>][--help][--version][源文件或目录...] [目的目录]
```

(4) 所属包:

coreutils

(5) 使用技巧:

ln 指令用于链接文件或目录，如果同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则会把前面指定的所有文件或目录复制到该目录中。

(6) 相关命令:

Cp、ln -s、symlinks

(7) 参数:

ln 主要参数如表 3-39 所示。

表 3-39 ln 参数表

参 数	含 义
-b 或--backup	覆盖目标文件之前的备份
-d 或-F 或--directory	建立目录的硬链接
-f 或--force	强行建立文件或目录的链接，不论文件或目录是否存在
-i 或--interactive	覆盖已有文件之前先询问用户
-n 或--no-dereference	把符号链接的目的目录视为一般文件
-s 或--symbolic	对源文件建立符号链接，而非硬链接
-S<字尾备份字符串>或--suffix=<字尾备份字符串>	用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的字尾备份字符串是符号“~”，可通过“-S”参数来改变它
-v 或--verbose	显示指令执行过程
-V<备份方式>或--version-control=<备份方式>	用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，这个字符串不仅可用“-S”参数变更，当使用“-V”参数<备份方式>指定不同备份方式时，也会产生不同字尾的备份字符串

(8) 补充说明:

若同时指定多个文件或目录，且最后的目的地并非是一个已存在的目录，则会出现错误信息。

(9) 典型实例:

例 1: 创建硬链接。

ln 命令不使用参数时创建的是硬链接。例如，为当前目录下 testfile 文件创建名为 test 的硬链接可使用如下命令:

```
ln testfile test #为源文件 testfile 创建硬链接文件 test
```

该命令没有任何输出信息。

例 2: 为目录创建硬链接。

由于该命令不能直接为目录创建硬链接，因此要为目录创建硬链接时，可使用参数“-s”、例如，要为当前目录下的子目录 test 创建硬链接，可以使用如下命令:

```
ln -s test testln #为当前目录下的子目录 test 创建硬链接
```

## 3.36 locate ( 查找文件 )



(1) 频度等级: ☆

(2) 功能说明:

查找文件。它的工作机制是先建立一个包含系统内所有文件名称及路径的 locate 数据库，然后在查找文件时可以直接从这个数据库中查找，而不需要再遍历整个文件系统的目录，这极大提高了查询速度。

(3) 语法:

```
locate [-d<数据库文件>][--help][--version][范本样式...]
```

(4) 所属包:

slocate

(5) 使用技巧:

首次运行 locate 命令之前, 应先使用 updatedb 命令创建数据库, 否则 locate 命令无法运行。

(6) 相关命令:

whereis、updatedb

(7) 参数:

locate 主要参数如表 3-40 所示。

表 3-40 locate 参数表

参 数	含 义
-d<数据库文件>或--database=<数据库文件>	设置 locate 指令使用的数据库。locate 指令预设的数据库在/var/lib/slocate 目录中, 文件名为 slocate.db, 操作时可使用这个参数另行指定
-u	创建或更新 locate 数据库, 功能与“updatedb”相同
-i	在查找文件时, 忽略文件名中字符的大小写差异
-q	静默模式, 不显示错误信息
-n<数字>	指定最多输出的、查找到的结果数目
-r<规则表达式>或--regexp=<规则表达式>	使用基本的 POSIX 规则表达式进行查找匹配

(8) 补充说明:

locate 指令用于查找符合条件的文件, 它会在保存文件与目录名称的数据库中查找符合范本样式条件的文件或目录。

(9) 典型实例:

例: 快速定位文件路径。

使用 locate 不指定路径命令, 就可以显示详细位置信息, 例如在命令行中输入如下命令:

```
locate fdisk #显示文件名中含有 fdisk 关键字的文件的 路径信息
```

输出信息如下:

```
cmd@cmd-desktop:~$ locate fdisk #显示文件名中含有 fdisk 关键字的文件的 路径信息
/sbin/cfdisk #搜索到的文件路径列表
/sbin/fdisk
/sbin/sfdisk
/usr/include/grub/ieee1275/ofdisk.h
/usr/share/doc/util-linux/README.cfdisk
/usr/share/doc/util-linux/README.fdisk.gz
/usr/share/doc/util-linux/examples/sfdisk.examples.gz
/usr/share/man/man8/cfdisk.8.gz
/usr/share/man/man8/fdisk.8.gz
/usr/share/man/man8/sfdisk.8.gz
```

## 提 示

如果数据库长时间没有更新, 可能会输出类似的系统警告信息, “arning:locate: warning:database/var/lib/slocate/slocate.db is more than 8 days old”, 表示 locate 数据库已经有 8 天没有更新, 可能使查询到的结果不准确。此时需要通过“updatedb”命令来更新 locate 数据库。

## 3.37 lsattr (显示文件属性)



(1) 频度等级: ☆☆

(2) 功能说明:

lsattr 命令显示关于给定设备或设备类型的属性信息。

(3) 语法:

```
lsattr [-adlrVv][文件或目录...]
```

(4) 所属包:

coreutils

(5) 使用技巧:

lsattr 命令显示文件系统属性与 ls 显示的 UNIX 文件系统属性是两个不同的概念。lsattr 实现的属性是文件系统的物理属性, 而 ls 显示的文件属性是操作系统进行管理文件系统的逻辑属性。

(6) 相关命令:

chattr

(7) 参数:

lsattr 主要参数如表 3-41 所示。

表 3-41 lsattr 参数表

参 数	含 义
-a	显示所有文件和目录, 包括以"."为名称开头字符的文件、当前目录"."和上层目录".."
-d	显示目录名称, 而不是它的内容
-l	此参数目前没有任何作用
-R	递归处理, 将指定目录下的所有文件及子目录一并处理
-v	显示文件或目录版本
-V	显示版本信息

(8) 补充说明:

用 chattr 命令改变文件或目录的属性, 可执行 lsattr 指令查询其属性。

(9) 典型实例:

例: 显示指定目录的文件系统属性。

例如直接指定目录/usr/local, 显示该目录的文件系统属性, 使用命令如下:

```
lsattr /usr/local #显示目录的文件系统属性
```

输出如下信息:

```
cmd@cmd-desktop:~$ lsattr /usr/local #显示目录的文件系统属性
-----e- /usr/local/src #/usr/local 目录下文件系统属性列表, 这些文件设
#置了“e”属性, 即可执行
-----e- /usr/local/include
-----e- /usr/local/games
-----e- /usr/local/bin
-----e- /usr/local/share
-----e- /usr/local/lib
-----e- /usr/local/etc
-----e- /usr/local/sbin
```

## 3.38 mattrib ( MS-DOS 文件的属性操作 )

(1) 频度等级: ☆

(2) 功能说明:

mattrib 命令用于增加、删除、修改或显示 MS-DOS 文件的属性。

(3) 语法:

```
mattrib [+/-][ahrs][-/X][文件]
mattrib [-/X][文件]
```

(4) 所属包:

mtools

(5) 使用技巧:

mattrib 命令的功能和 Linux 的 lsattr 和 chattr 命令相似, 不仅可以查看文件属性, 还可以修改文件的属性, 但是 MS-DOS 下的文件属性比 Linux 下少得多。

(6) 相关命令:



mdel、mdir、mmount、mtools、mzip

(7) 参数:

mattrib 主要参数如表 3-42 所示。

表 3-42 mattrib 参数表

参 数	含 义
+/-	“+”代表开启指定的文件属性，“-”代表关闭指定的文件属性
a	备份属性。使备份程序用来判断文件是否已经备份过
h	隐藏属性
r	只读属性
s	系统属性
-/	列出指定目录及子目录下所有文件的属性
-X	显示文件属性时，中间不输入空格

(8) 补充说明:

mattrib 为 mtools 工具指令，模拟 MS-DOS 的 attrib 指令，可变更 MS-DOS 文件的属性。

(9) 典型实例:

例：显示 A 盘中所有文件的属性。

例如 A 盘中包含目录和文件，可使用参数“-/”递归显示指定目录及子目录下所有文件的属性，具体命令如下：

```
mattrib-/ a:/*          #显示 A 盘中所有文件的属性
mattrib -/ A:\*
```

在 MS-DOS 文件系统中，文件的分隔符为“\”，由于在 Linux 系统下 DOS 文件系统执行相应的操作，所以所有使用 Linux 系统下文件的分隔符“/”同样能执行成功。

该命令执行后的结果如下：

```
cmd@cmd-desktop:~$ mattrib-/ a:/*          #显示 A 盘中所有文件的属性
A  HR      A:/BACKUP          #递归显示文件属性：存档、隐藏、只读
          A:/FILE.DOC
A          A:/AUTORUN.INF      #存档文件
A  H       A:/AUTORUN.BAT    #存档、隐藏文件
          S  R      A:/ EXPLORER.EXE #系统文件、只读
```

## 3.39 mc ( 菜单式的文件管理程序 )



(1) 频度等级：☆

(2) 功能说明:

mc 命令提供一个菜单式的文件管理程序来管理文件。

(3) 语法:

```
mc [-abcdfhkPstuUVx][ -C <参数>][ -l <文件>][ -v <文件>][目录]
```

(4) 所属包:

```
mc
```

(5) 使用技巧:

执行 mc 之后，将会看到菜单式的文件管理程序，共分成 4 个部分。

(6) 相关命令:

```
mcedit
```

(7) 参数:

mc 主要参数如表 3-43 所示。

表 3-43 mc 参数表

参 数	含 义	参 数	含 义
-----	-----	-----	-----

-a	当 mc 程序画线时不使用绘图字符画线	-P	程序结束时，列出最后的工作目录
-b	使用单色模式显示	-s	用慢速的终端机模式显示，在这种模式下将减少大量的绘图及文字显示
-c	使用彩色模式显示	-t	使用 TEMPCAP 变量设置终端机，而不使用预设置
-C<参数>	指定显示的颜色	-u	不用目前的 shell 程序
-d	不使用鼠标	-U	使用目前的 shell 程序
-f	显示 mc 函数库所在的目录	-v<文件>	使用 mc 的内部编辑器来显示指定的文件
-k	重设 softkeys 为预设置	-x	指定以 xterm 模式显示
-l<文件>	在指定文件中保存 ftpfs 对话框的内容		

(8) 补充说明:

该命令只在部分发行版本中存在，可选择安装。

(9) 典型实例:

例：直接在命令提示符下输入 mc 命令，即可打开菜单式文件管理器，如图 3-2 所示。

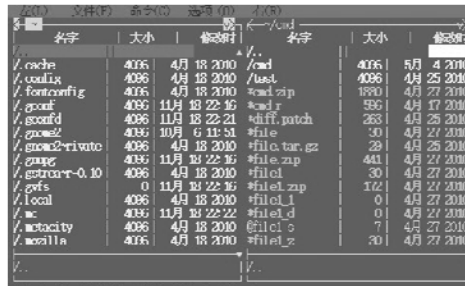


图 3-2

## 3.40 mcopy (复制 MS-DOS 文件)



(1) 频度等级: ☆

(2) 功能说明:

mcopy 命令用于复制 MS-DOS 文件系统下的文件。

(3) 语法:

```
mcopy [-bnmpQt/][源文件][目标文件或目录]
```

(4) 所属包:

mttools

(5) 使用技巧:

复制文件时，目标目录中可能存在同名文件，为了防止同名文件被覆盖，不建议使用“-o”和“-n”参数。

(6) 相关命令:

mttools、mdel

(7) 参数:

mcopy 主要参数如表 3-44 所示。

表 3-44 mcopy 参数表

参 数	含 义	参 数	含 义
-b	批处理模式	-Q	复制多个文件时，若发生错误，则立即结束程序
-n	覆盖其他文件时，不需进行确认而直接覆盖	-t	转换文本文件
-m	将源文件修改时间设为目标文件的修改时间	-/	复制子目录及其中的所有文件
-p	将源文件属性设为目标文件的属性		

(8) 补充说明:

`mcopy` 为 `mtools` 工具指令，可在 MS-DOS 文件系统中复制文件或是在 MS-DOS 与 Linux 的文件系统之间复制文件。`mcopy` 会依据 MS-DOS 磁盘驱动器代号出现的位置来判断如何复制文件。

(9) 典型实例：

例：将 A 盘上的所有文件复制到当前工作目录。

当复制的内容包括子目录和文件时，必须使用参数 “-/” 递归操作，因此该命令为：

```
mcopy -/ A:\*           #将 A 盘上的所有文件复制到当前工作目录
```

执行该命令前先使用 `mdir` 命令查看原来的目录结构，执行 `mcopy` 之后可使用 `ls` 命令查看复制之后 Linux 系统中的文件结构，结果如下：

```
cmd@cmd-desktop:~$ mdir -/ a:\*           #查看 A 盘中的文件
Volume in drive A has no label           #加载信息
Volume Serial Number is 13D2~055C

Directory for A:/                         #以下为目录信息
#文件名      目录      大小      修改时间
./TEST       <DIR>           2009-09-23  16:59
AUTORUN.INF  265           2009-09-23  16:53
AUTORUN.BAT  43            2009-09-23  16:56
          3 files           308 bytes           #统计总大小
          724 325 bytes free       #剩余空间

cmd@cmd-desktop:~$ mcopy -/ A:\*         #将 A 盘上的所有文件复制到当前工作目录
cmd@cmd-desktop:~$ ls
TEST      AUTORUN.INF      AUTORUN.BAT #A 盘中的内容复制到 Linux 文件系统结构中
```

## 3.41 mdel (删除 MS-DOS 系统文件)



(1) 频度等级：☆

(2) 功能说明：

`mdel` 命令用于删除 MS-DOS 文件系统下的文件。

(3) 语法：

```
mdel [文件...]
```

(4) 所属包：

```
mtools
```

(5) 使用技巧：

`mdel` 命令在删除目录或文件夹时，默认将目录下的所有内容全部删除，所以用户在删除目录时应确认目录下所有文件都是需要删除的，这点与 Linux 命令 `rm` 略有不同。

`mdel` 只能删除 MS-DOS 系统文件，当试图删除一个 Linux 系统下的文件时，将会提示 “`model:File xxx not found`”，产生错误。

(6) 相关命令：

```
mdir、mcopy
```

(7) 补充说明：

`mdel` 为 `mtools` 工具指令，模拟 MS-DOS 的 `del` 指令，可删除 MS-DOS 文件系统中的文件。

(8) 典型实例：

例：删除 A 盘中文件。

例如，使用 `mdel` 命令删除 `autorun.inf` 文件，可直接使用如下命令：

```
mdel autorun.inf      #删除 autorun.inf 文件
```

使用该命令前，先查看该目录下的目录结构，结果如下：

```
cmd@cmd-desktop:~$ mdir -/ a:\*           #查看 A 盘中的文件
```

```

Volume in drive A has no label           #加载信息
Volume Serial Number is 13D2~055C

Directory for A:/                        #以下为目录信息
#文件名      目录      大小      修改时间
./TEST       <DIR>           2009-09-23  16:59
AUTORUN.INF      265      2009-09-23  16:53
AUTORUN.BAT     43       2009-09-23  16:56
              3 files      308 bytes      #统计总大小
              724 325 bytes free #剩余空间
    
```

输入 `mdel` 命令，再次查看，有如下输出结果：

```

cmd@cmd-desktop:~$ mdel autorun.inf     #删除 autorun.inf 文件
cmd@cmd-desktop:~$ mdir -/ a:\*        #再次查看 A 盘中的文件
Volume in drive A has no label         #加载信息
Volume Serial Number is 13D2~055C

Directory for A:/                      #以下为目录信息
#文件名      目录      大小      修改时间
./TEST       <DIR>           2009-09-23  16:59
AUTORUN.BAT  43       2009-09-23  16:56
              2 files      43 bytes      #统计总大小
              724 590 bytes free #剩余空间
    
```

## 3.42 mdir (显示 MS-DOS 目录)



(1) 频度等级：☆

(2) 功能说明：

`mdir` 命令用于显示 MS-DOS 文件系统下的目录。

(3) 语法：

```
mdir [-afwx/][目录]
```

(4) 所属包：

mtools

(5) 使用技巧：

`mdir` 命令只能查看 MS-DOS 文件系统的目录，查看 Linux 文件系统将出错。

(6) 相关命令：

`mcopy`、`mdel`

(7) 参数：

`mdir` 主要参数如表 3-45 所示。

表 3-45 mdir 参数表

参 数	含 义
-/	显示目录下所有子目录与文件
-a	显示隐藏文件
-f	不显示磁盘所剩余的可用空间
-w	仅显示目录或文件名称，并以横排方式呈现，以便一次能显示较多的目录或文件
-x	仅显示目录下所有子目录与文件的完整路径，不显示其他信息

(8) 补充说明：

`mdir` 为 `mtools` 工具指令，模拟 MS-DOS 的 `dir` 指令，可显示 MS-DOS 文件系统中的目录内容。

(9) 典型实例：

例：显示 A 盘的目录。

如果目录中包含子目录和文件，可使用“-/”参数显示目录，具体命令如下：

```
mdir -/ a:\* #查看 A 盘中的文件
```

该命令将有如下输出信息：

```
cmd@cmd-desktop:~$ mdir -/ a:\* #查看 A 盘中的文件
Volume in drive A has no label #加载信息
Volume Serial Number is 13D2~055C

Directory for A:/ #以下为目录信息
#文件名 目录 大小 修改时间
./TEST <DIR> 2009-09-23 16:59
AUTORUN.INF 265 2009-09-23 16:53
AUTORUN.BAT 43 2009-09-23 16:56
3 files 308 bytes #统计总大小
724 325 bytes free #剩余空间
```



## 3.43 mktemp ( 建立临时文件 )

(1) 频度等级：☆

(2) 功能说明：

建立暂存文件（提供给 shell 脚本程序使用）。该命令创建的临时文件往往文件名比较特别，这样可以防止被猜解。

(3) 语法：

```
mktemp [-qu][文件名参数]
```

(4) 所属包：

```
mktemp
```

(5) 使用技巧：

在进行 shell 脚本程序设计时，经常需要生成临时文件，使用 mktemp 命令即可满足这样的操作，且能保证文件的安全性。

(6) 相关命令：

```
mdir、mtools
```

(7) 参数：

mktemp 主要参数如表 3-46 所示。

表 3-46 mktemp 参数表

参 数	含 义
-q	执行时若发生错误，不会显示任何信息
-u	暂存文件会在 mktemp 结束前先行删除
-V	显示版本信息
[文件名参数]	文件名参数必须是以“自订名称.XXXXXX”的格式

(8) 典型实例：

例：使用 mktemp 命令生成临时文件时，文件名参数应当以“文件名.XXXX”的形式给出，mktemp 会根据文件名参数建立一个临时文件。在命令行提示符输入如下命令：

```
mktemp tmp.xxxx #生成临时文件
```

使用该命令后，可使用 dir 或 ls 看当前目录，得到如下结果：

```
cmd@cmd-desktop:~$ mktemp tmp.xxxx #生成临时文件
cmd@cmd-desktop:~$ dir #查看当前目录
file test testfile testfile1 tmp.3847 #生成了 tmp.3847
```



由此可见，生成的临时文件为 tmp.3847，其中，文件名参数中的“XXXX”被4个随机产生的字符所取代。

## 3.44 mmove (在 MS-DOS 文件系统中移动文件、目录或更改名称)



(1) 频度等级：☆

(2) 功能说明：

mmove 命令用于在 MS-DOS 文件系统中移动文件、目录或更改名称。

(3) 语法：

```
mmove [源文件或目录...][目标文件或目录]
```

(4) 所属包：

```
mtools
```

(5) 使用技巧：

与 MS-DOS 下的 move 命令不同的是，mmove 可以移动子目录。文件或目录只能在同一个文件系统中移动，数据既不能从 MS-DOS 移动到 Linux，也不能从 Linux 移动到 MS-DOS。

如果命令参数中忽略了目标文件或目标目录的渠道卷标，将会默认与原文件具有相同的卷标。如果所有的参数都忽略了驱动器卷标，系统将默认为驱动器 A（软驱）。

(6) 相关命令：

```
mdel、mdir
```

(7) 补充说明：

mmove 为 mtools 工具指令，模拟 MS-DOS 的 move 指令，可在 MS-DOS 文件系统中移动现有的文件或目录，或是更改现有文件或目录的名称。

(8) 典型实例：

例：移动 A 盘中的文件。例如移动 A 盘中的 autorun.bat 文件到 test 目录，可使用如下命令：

```
mmove autorun.bat test #移动 autorun.bat 文件到 test 目录
```

使用该命令前先查看原来的目录，得到如下结果：

```
cmd@cmd-desktop:~$ mdir -/ a:\* #查看 A 盘中的文件
Volume in drive A has no label #加载信息
Volume Serial Number is 13D2~055C
```

```
Directory for A:/ #以下为目录信息
#文件名 目录 大小 修改时间
./TEST <DIR> 2009-09-23 16:59
AUTORUN.INF 265 2009-09-23 16:53
AUTORUN.BAT 43 2009-09-23 16:56
3 files 308 bytes #统计总大小
724 325 bytes free #剩余空间
```

使用 mmove 命令，并再次查看，结果如下：

```
cmd@cmd-desktop:~$ mmove autorun.bat test #移动 autorun.bat 文件到 test 目录
cmd@cmd-desktop:~$ mdir -/ a:\* #再次查看 A 盘中的文件
Volume in drive A has no label #加载信息
Volume Serial Number is 13D2~055C
Directory for A:/ #以下为目录信息
#文件名 目录 大小 修改时间
./TEST <DIR> 2009-09-23 16:59
AUTORUN.INF 265 2009-09-23 16:53
```

```

2 files                265 bytes      #统计总大小
                    724 683 bytes free  #剩余空间
cmd@cmd-desktop:~$ mdir -/ a:\test\*  #再次查看 test 目录中文件
AUTORUN.BAT           43            2009-09-23  16:56
                    1 files        43 bytes      #统计总大小
    
```

可见，autorun.bat 文件已经移动到 test 目录。



## 3.45 mread ( 复制 MS-DOS 文件 )

(1) 频度等级: ☆

(2) 功能说明:

mread 命令将 MS-DOS 文件复制到 Linux/UNIX 的目录中。

(3) 语法:

```
mread [MS-DOS 文件...][Linux 文件或目录]
```

(4) 所属包:

```
mtools
```

(5) 使用技巧:

mread 为 mtools 工具指令，可将 MS-DOS 文件复制到 Linux 的文件系统中。

(6) 相关命令:

```
mdel、dcopy、mren、mdir、mtools
```

(7) 补充说明:

这个指令现在已经不常用了，一般都使用 mcopy 指令来代替。

(8) 典型实例:

例：将 A 盘上的所有文件复制到当前工作目录。

可使用 mread 命令复制 A 盘下的所有内容，包括子目录和文件，具体命令为：

```
mread A:\* ./          #将 A 盘上的所有文件复制到当前工作目录
```

执行该命令前先使用 mdir 命令查看原来的目录结构，执行 mread 之后可使用 ls 命令查看复制之后 Linux 系统中的文件结构，结果如下：

```

cmd@cmd-desktop:~$ mdir -/ a:\*      #查看 A 盘中的文件
Volume in drive A has no label       #加载信息
Volume Serial Number is 13D2~055C
Directory for A:/                    #以下为目录信息
#文件名      目录      大小      修改时间
./TEST      <DIR>      2009-09-23  16:59
AUTORUN.INF 265        2009-09-23  16:53
AUTORUN.BAT 43         2009-09-23  16:56
          3 files      308 bytes      #统计总大小
          724 325 bytes free #剩余空间
cmd@cmd-desktop:~$ mread A:\* ./    #将 A 盘上的所有文件复制到当前工作目录
cmd@cmd-desktop:~$ ls
TEST  AUTORUN.INF  AUTORUN.BAT
    
```

#A 盘中的内容复制到 Linux 文件系统结构中

## 3.46 mren ( 更改 MS-DOS 文件或目录的名称，或移动文件或目录 )



(1) 频度等级: ☆

(2) 功能说明:

mren 命令用于更改 MS-DOS 文件或目录的名称, 或是移动文件或目录。

(3) 语法:

```
mren [源文件或目录...][目标文件或目录]
```

(4) 所属包:

```
mtools
```

(5) 使用技巧:

源文件必须是磁盘上已经存在的文件, 若忽略盘符及路径, 则表示当前盘及当前目录的文件。

新文件名是所要更换的文件名称。新文件名称前不可以加同源文件不同的盘符及路径, 因为该命令只能对同一盘上的文件更换文件名称。

(6) 相关命令:

```
mdel, mmove, mtools
```

(7) 补充说明:

mren 为 MS-DOS 工具指令, 模拟 MS-DOS 的 ren 指令, 可更改 MS-DOS 文件或目录名称。

(8) 典型实例:

例: 将 A 盘下的文件重命名。

例如, 将 A 盘下的文件 autorun.inf 重命名为 auto.inf, 可直接使用 mren 命令。

```
mren a:\autorun.inf auto.inf #将文件 autorun.inf 重命名为 auto.inf
```

使用该命令前后使用 mdir 命令查看并对比, 得到如下结果:

```
cmd@cmd-desktop:~$ mdiskpart -a a: #查看 A 盘中的文件
Volume in drive A has no label #加载信息
Volume Serial Number is 13D2~055C
Directory for A:/ #以下为目录信息
#文件名 目录 大小 修改时间
./TEST <DIR> 2009-09-23 16:59
AUTORUN.INF 265 2009-09-23 16:53
AUTORUN.BAT 43 2009-09-23 16:56
3 files 308 bytes #统计总大小
724 325 bytes free #剩余空间

cmd@cmd-desktop:~$ mren a:\autorun.inf auto.inf #将文件 autorun.inf 重命名为 auto.inf
cmd@cmd-desktop:~$ mdiskpart -a a: #再次查看 A 盘中文件
Volume in drive A has no label #加载信息
Volume Serial Number is 13D2~055C
Directory for A:/ #以下为目录信息
#文件名 目录 大小 修改时间
./TEST <DIR> 2009-09-23 16:59
AUTO.INF 265 2009-09-23 18:14 #文件名被改为 auto.inf, 修改时间改为当前系统时间
AUTORUN.BAT 43 2009-09-23 16:56
3 files 308 bytes #统计总大小
724 325 bytes free #剩余空间
```

## 3.47 mtype (显示 MS-DOS 文件的内容)



(1) 频度等级: ☆

(2) 功能说明:

mtype 命令用于显示 MS-DOS 文件的内容, 该命令模拟 MS-DOS 下的 type 命令。

(3) 语法:

```
mtype[参数] [文件...]  
mtype [参数][文件,>PRN]
```

(4) 所属包:

```
mtools
```

(5) 使用技巧:

当文件较长时, 一屏显示不完, 可以利用 “`mtype [盘符: ][路径]<文件名>|MORE`” 分屏显示。其中, MORE 为分屏显示命令, 使用该命令, 当满屏时会暂停, 按任意键将会继续显示, 这和 Linux 系统下查看较大文件文件基本一致。

(6) 相关命令:

```
mdel, mdir, mtoos
```

(7) 补充说明:

该命令一次只能显示一个文件的内容, 不能使用通配符。

(8) 典型实例:

例: 查看文件的内容。

例如查看 A 盘下的文本文件 test.txt, 可直接在终端中使用如下命令:

```
mtype test.txt #查看 test.txt 文本文件
```

该命令的输出结果如下:

```
cmd@cmd-desktop:~$ mtype test.txt #查看 test.txt 文本文件  
hello,this is a MS_DOS test file!
```

## 提示

若查看的文件内容较大时(超过一屏), 可以连续滚动屏幕, 此时可以借助 “MORE” 命令分屏显示。

# 3.48 mtools (显示 mtools 支持的命令)



(1) 频度等级: ☆

(2) 功能说明:

mtools 命令用于显示 mtools 支持的指令。mtools 实际上是一个命令集合, 是 DOS 文件系统的工具程序, 它可以模拟许多 MS-DOS 命令, 使用起来非常方便。使用权限是所有用户。Linux 系统提供了一组被称为 mtools 的可移植工具, 可以让用户轻松地从一个标准的 MS-DOS 软盘上读、写文件和目录。它们对于在 MS-DOS 和 Linux 环境之间交换文件非常有用。

(3) 语法:

```
mtools
```

(4) 所属包:

```
mtools
```

(5) 使用技巧:

MS-DOS 文件系统工具软件包可模拟很多 MS-DOS 命令, 这些命令都是 mtools 命令的符号链接, 它们的语法与 MS-DOS 下的语法相同。

(6) 相关命令:

```
mtoolstest
```

## (7) 参数:

mtools 主要参数如表 3-47 所示。

表 3-47 mtools 参数表

参 数	含 义
-a	长文件名重复时自动更改目标文件的长文件名
-A	短文件名重复但长文件名不同时自动更改目标文件的短文件名
-o	长文件名重复时, 将目标文件覆盖现有的文件
-O	短文件名重复但长文件名不同时, 将目标文件覆盖现有的文件
-r	长文件名重复时, 要求用户更改目标文件的长文件名
-R	短文件名重复但长文件名不同时, 要求用户更改目标文件的短文件名
-s	长文件名重复时, 不处理该目标文件
-S	短文件名重复但长文件名不同时, 不处理该目标文件
-v	执行时显示详细的说明
-V	显示版本信息

## (8) 典型实例:

例: 显示 mtools 软件包所支持的 MS-DOS 命令。

在命令提示符中直接输入 mtools, 可显示其所支持的 MS-DOS 命令, 如下所示:

```
hdd@hdd-desktop:~$ mtools           #显示所支持的 MS-DOS 命令
Supported commands:                 #命令列表
mattrib, mbadblocks, mcat, mcd, mclasserace, mcopy, mdel, mdeltree
mdir, mdoctorfat, mdu, mformat, minfo, mlabel, mmcd, mmount
mpartition, mrd, mread, mmove, mren, mshowfat, mtoolstest, mtype
mwrite, mzip
```

## 3.49 mtoolstest ( 测试 mtools 的相关设置 )



(1) 频度等级: ☆

(2) 功能说明:

mtoolstest 命令用于测试并显示 mtools 的相关设置。mtoolstest 为 mtools 的工具指令, 可读取与分析 mtools 的配置文件, 并在屏幕上显示结果。

(3) 语法:

```
mtoolstest
```

(4) 所属包:

```
mtools
```

(5) 使用技巧:

mtools 软件包的配置信息保存在 “/etc/mtools/conf” 中。

(6) 相关命令:

```
mtools
```

(7) 典型实例:

例: 在命令行中直接输入 mtoolstest, 即可显示 mtools 软件包当前的配置信息, 结果如下:

```
hdd@hdd-desktop:~$ mtoolstest           #显示 mtools 软件包当前的配置信息
drive J:                                 #mtools 软件包当前的配置信息列表
#fn=0 mode=0 builtin
file="/dev/sdb4" fat_bits=16
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
mformat_only
drive Z:
```



```
#fn=0 mode=0 builtin
file="/dev/sdb4" fat_bits=16
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
mformat_only

drive X:
#fn=0 mode=0 builtin
file="$DISPLAY" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0

drive A:
#fn=2 mode=128 defined in /etc/mtools.conf
file="/dev/fd0" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
exclusive

drive B:
#fn=2 mode=128 defined in /etc/mtools.conf
file="/dev/fd1" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0
exclusive

drive M:
#fn=2 mode=0 defined in /etc/mtools.conf
file="/var/lib/dosemu/hdimage.first" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x80
partition=1

drive N:
#fn=2 mode=0 defined in /etc/mtools.conf
file="/var/lib/dosemu/fdimage" fat_bits=0
tracks=0 heads=0 sectors=0 hidden=0
offset=0x0
partition=0

mtools_fat_compatibility=0
mtools_skip_check=0
mtools_lower_case=0
```

## 3.50

mv (移动或更名现有的文件或目录)



(1) 频度等级: ☆☆☆

(2) 功能说明:

`mv` 命令用于移动或更名现有的文件或目录。

(3) 语法:

```
mv [-bfuiuv][--help][--version][-S <附加字尾>][-V <方法>][源文件或目录][目标文件或目录]
```

(4) 所属包:

coreutils

(5) 使用技巧:

一般“-i 或--interactive”参数使用较多, 因为该参数能防止错误地将重要文件覆盖。

`mv` 命令还能用于为文件更名。

(6) 相关命令:

rename

(7) 参数:

`mv` 主要参数如表 3-48 所示。

表 3-48 mv 参数表

参 数	含 义
--backup=<备份方式>	移动文件时, 为其创建一个备份。备份方式有如下几种 <ul style="list-style-type: none"> <li>● none 和 off 表示关闭 <code>mv</code> 命令的备份功能</li> <li>● number 和 t 表示为指定文件追加一个数字扩展名的备份</li> <li>● existing 和 nil 表示如果数字扩展名存在, 则覆盖</li> <li>● simpe 和 never 总是仅做简单备份</li> </ul>
-b	移动文件时, 为存在的目标文件创建一个备份
-f 或--force	若目标文件或目录与现有的文件或目录重复, 则直接覆盖现有的文件或目录。此参数慎用
-i 或--interactive	覆盖前先询问用户
-S<附加字尾>或--suffix=<附加字尾>	与-b 参数一并使用, 可指定备份文件所要附加的字尾
-u 或--update	在移动或更改文件名时, 若目标文件已存在, 且其文件日期比源文件新, 则不覆盖目标文件
-v 或--verbose	执行时显示详细的信息
-V=<方法> 或 --version-control=<方法>	与-b 参数一起使用, 可指定备份的方法

(8) 典型实例:

例 1: 为文件改名。

例如想为当前目录下的文件 `testfile` 改名为 `testfile_bak`, 可使用如下命令:

```
mv testfile testfile_bak #将 testfile 改名为 testfile_bak
```

为了对比效果, 使用该命令前使用 `ls` 命令查看当前文件名, 得到如下信息:

```
cmd@cmd-desktop:~$ls #查看原文件名
testfil #原文件名为 testfiel
```

使用 `mv` 命令改名后, 当前目录下的文件名为:

```
cmd@cmd-desktop:~$mv testfile testfile_bak #将 testfile 改名为 testfile_bak
cmd@cmd-desktop:~$ls #查看修改后文件名
testfile_bak #文件名被修改为 testfile_bak
```

例 2: 移动文件。

`mv` 命令最主要的功能是移动文件, 该命令不仅可以移动单个文件, 还可以使用通配符移动多个文件到指定目录。例如使用 `mv` 命令移动 `file1`、`file2`、`file3`、`file4`、`file5`、`file6` 这 6 个文件到当前子目录 `test`, 可使用如下命令:

```
mv file[1-6] test #将文件 file1、file2、file3、file4、file5、file6 移动到 test 子目录
```

使用该命令前先查看当前目录结构, 得到信息如下:

```
cmd@cmd-desktop:~$ls #查看当前目录结构
file1 file2 file3 file4 file5 file6 #当前目录结构
```

使用 `mv` 命令, 移动这 6 个文件, 再次查看得到如下结果:

```
cmd@cmd-desktop:~$ls #查看当前目录结构
```

```

#当前目录为空
cmd@cmd-desktop:~$ls test #查看 test 子目录
file1 file2 file3 file4 file5 file6 # file1 file2 file3 file4 file5 file6 移动到
#test 子目录中
    
```



## 3.51 ncftp (传输文件)

(1) 频度等级: ☆☆☆

(2) 功能说明:

ncftp 命令用于传输文件。该命令被设计用来取代传统的 ftp 命令,它在实现 ftp 命令功能的同时,对各功能也进行了很大的改进。

(3) 语法:

```
ncftp [主机或 IP 地址]
```

(4) 所属包:

```
ncftp
```

(5) 使用技巧:

当不指定用户名时,ncftp 命令会自动尝试使用匿名账户 anonymous 去连接远程 FTP 服务器,不需要用户输入账号和密码。

(6) 相关命令:

```
ftp
```

(7) 参数:

ncftp 主要参数如表 3-49 所示。

表 3-49 ncftp 参数表

参 数	含 义
-u <用户>	指定登录服务器用户,禁止匿名用户 anonymous
-p <密码>	设置用户的登录密码
-P <端口>	指定 FTP 服务器端口,默认端口为 21

(8) 补充说明:

ftp 让用户得以下载存放于服务器主机的文件,也能将文件上传到远端主机放置。ncftp 是文字模式 FTP 程序的佼佼者,它具有多种特色,包括显示传输速率、下载进度、自动续传、标注书签、可通过防火墙和代理服务器等。

(9) 典型实例:

例 1: 使用 ncftp 命令匿名连接 FTP 服务器。

例如想匿名连接 ftp.kernel.org 服务器,同时不想输入 anonymous 等匿名用户名,可直接使用 ncftp 命令:

```
ncftp ftp.kernel.org #匿名连接 ftp.kernel.org 服务器
```

得到如下信息:

```

cmd@cmd-desktop:~$ ncftp ftp.kernel.org #匿名连接 ftp.kernel.org 服务器
NcFTP 3.2.1 (Jul 29, 2007) by Mike Gleason (http://www.NcFTP.com/contact/).
#ncftp 版权、版本等信息

Copyright (c) 1992-2005 by Mike Gleason.
All rights reserved.

Connecting to 149.20.20.133... #连接服务器
Welcome to ftp.kernel.org.
Logging in... #匿名登录

Welcome to the #欢迎信息
    
```

LINUX KERNEL ARCHIVES

ftp.kernel.org

"Much more than just kernels"

IF YOU'RE ACCESSING THIS SITE VIA A WEB BROWSER

PLEASE USE THE HTTP URL BELOW INSTEAD!

```
---->      If you are looking for mirror sites, please go      <----  
---->      to mirrors.kernel.org instead                        <----
```

This site is provided as a public service by the Linux Kernel Organization, a California nonprofit corporation. Bandwidth is provided by The Internet Software Consortium, Inc. Our servers are located in San Francisco and Palo Alto, California; Corvallis, Oregon; Amsterdam, Netherlands and Umeå, Sweden; use in violation of any applicable laws strictly prohibited.

Due to U.S. Exports Regulations, all cryptographic software on this site is subject to the following legal notice:

This site includes publicly available encryption source code which, together with object code resulting from the compiling of publicly available source code, may be exported from the United States under License Exception "TSU" pursuant to 15 C.F.R. Section 740.13(e).

This legal notice applies to cryptographic software only. Please see the Bureau of Industry and Security (<http://www.bis.doc.gov/>) for more information about current U.S. regulations.

Neither the Linux Kernel Organization, nor its sponsors make any guarantees, explicit or implicit, about the contents of this site. Use at your own risk.

This site is accessible via the following mechanisms:

```
FTP      ftp://ftp.kernel.org/pub/  
HTTP     http://www.kernel.org/pub/  
RSYNC    rsync://rsync.kernel.org/pub/
```

NFS and SMB/CIFS are no longer available.

For comments on this site, please contact <[ftpadm@kernel.org](mailto:ftpadm@kernel.org)>. Please do not use this address for questions that are not related to the operation of this site. Please see our homepage at <http://www.kernel.org/> for links to Linux documentation resources. Login successful.

Logged in to ftp.kernel.org.

```
ncftp / >
```

#命令提示符行

## 提示

ncftp 的命令提示符为“ncftp />”，而不是 ftp 中的“ftp />”。

例 2: 使用 ncftp 命令操作、下载文件。

ncftp 的命令基本上与 ftp 相同, 例如可以使用 “cd” 命令切换在 FTP 服务器中的当前目录, 使用 “ls” 命令列出当前目录内容, 使用 “get” 命令下载 “/pub” 目录下的 README 文件、使用 “quit” 离开 ncftp 等。操作结果如下:

```
ncftp / > pwd #查看当前路径
ftp://ftp.kernel.org #当前路径为根目录
ncftp / > ls #查看当前目录列表
bin/ for_mirrors_only/ pub/
dev/ lib/ usr@
etc/ lost+found/ welcome.msg@
ncftp / > cd pub #切换目录到 pub 子目录
Directory successfully changed.
ncftp /pub > ls #查看 pub 的目录列表
dist/ media/ scm/
index.html RCS/ site/
linux/ README software/
lost+found/ README_ABOUT_BZ2_FILES tools/
ncftp /pub > get README #下载 README 文件
README: 1.87 KB 10.39 KB/s
ncftp /pub > quit #离开 ncftp
```

与 ftp 不同的是, ncftp 此时会提示用户是否将 FTP 服务器保存为书签, 以便于下次登录, 用户可以进行自定义书签名等操作, 如下所示:

```
You have not saved a bookmark for this site. #离开提示信息

Would you like to save a bookmark to:
ftp://ftp.kernel.org/pub/

Save? (yes/no) yes #确认是否保存
Enter a name for this bookmark, or hit enter for "kernel": kernel #输入书签名
Bookmark "kernel" saved.
```

## 3.52 ncftpget ( 下载文件 )



(1) 频度等级: ☆

(2) 功能说明:

ncftpget 命令用于从远端 FTP 服务器通过 FTP 协议下载文件。

(3) 语法:

```
ncftpget [-aAbEFRTvVzZ][-B size][-d XX][-DD][-f filename][-j count][-p passwd][-P port][-r num][-t time][-u username]
```

(4) 所属包:

```
ncftp
```

(5) 使用技巧:

该命令相当于 ncftp 的简化版。

(6) 相关命令:

```
ncftp, ncftpbatch, ncftpls, ncftpput
```

(7) 参数:

ncftpget 的主要参数如表 3-50 所示。

表 3-50 ncftpget 参数表



参 数	含 义	参 数	含 义
-a	以 ASCII 方式传输数据	-P port	指定 FTP 端口
-A	添加附件	-r num	设置重试的次数限制
-b	在后台运行	-R	递归模式，对目录及所有的子目录进行操作
-B size	将 TCP/IP 套接口缓冲区大小设置为 size 字节	-t time	指定操作超时时间，以秒为单位
-d XX	指定记录文件，以便于调试	-T	在递归模式中，不使用 TAR 的格式下载
-DD	成功下载文件以后将文件从远端主机上删除	-u username	指定登录用户名，默认使用 anonymous 匿名登录
-E	使用普通连接根式 (PORT)	-v	显示下载进度，这是默认选项
-f filename	从指定文件中读取主机、用户名和密码等信息	-V	不显示下载进度
-F	使用被动连接模式 (PASV)，这是默认选项	-z	继续上次未完成的下载，这是默认选项
-j count	除了提供用户名和密码以外，还提供一个账号供登录使用	-Z	不继续上次未完成的下载
-p passwd	指定登录所使用的密码		

(8) 补充说明：

使用该命令时可以直接输入相关的参数，执行下载文件的操作。

(9) 典型实例：

例：从 FTP 服务器上下载文件。

例如从 ftp.ncftpget.com 下载文件 ncftp.tar.Z 到当前目录，已知该文件在 FTP 上的位置，可直接使用如下命令：

```
cmd@cmd-desktop:~$ ncftpget ftp://ftp.ncftpget.com/pub/ncftp/ncftp.tar.Z #从指定 FTP
#站点下载指定文件 ncftp.tar.Z
```

## 3.53 ncftpput ( 上传文件 )



(1) 频度等级：☆

(2) 功能说明：

ncftpput 命令用于从远端 FTP 服务器通过 FTP 协议上传文件。

(3) 语法：

```
ncftpput [-aAbEFRTvVzZ][-B size][-d XX][-DD][-f filename][-j count][-p passwd][-P port][-r
num][-t time][-u username]
```

(4) 所属包：

ncftp

(5) 使用技巧：

该命令相当于 ncftp 的简化版。

(6) 相关命令：

ncftp、ncftpbatch、ncftpls、ncftpget

(7) 参数：

ncftpput 的主要参数如表 3-51 所示。

表 3-51 ncftpput 参数表

参 数	含 义	参 数	含 义
-a	以 ASCII 方式传输数据	-p passwd	指定登录所使用的密码
-A	添加附件	-P port	指定 FTP 端口
-b	在后台运行	-r num	设置重试的次数限制
-B size	将 TCP/IP 套接口缓冲区大小设置为 size 字节	-R	递归模式，对目录及所有的子目录进行操作
-d XX	指定记录文件，以便于调试	-t time	指定操作超时时间，以秒为单位
-DD	成功上传文件以后将文件从远端主机上删除	-T	在递归模式中，不使用 TAR 的格式上传
-E	使用普通连接根式 (PORT)	-u username	指定登录用户名，默认使用 anonymous 匿名登录

-f filename	从指定文件中读取主机、用户名和密码等信息	-v	显示上传进度，这是默认选项
-F	使用被动连接模式（PASV），这是默认选项	-V	不显示上传进度
-j count	除了提供用户名和密码以外，还提供一个账号供登录使用	-z	继续上次未完成的上传，这是默认选项

(8) 补充说明：

使用该命令时可以直接输入相关的参数，执行上传文件的操作。

(9) 典型实例：

例：上传文件到 FTP 服务器。

例如上传文件到 ftp.ncftpget.com 上，并建立以 .tmp 为后缀的临时文件，可直接使用如下命令：

```
cmd@cmd-desktop:~$ ncftpput -S .tmp ftp://ftp.ncftpget.com/pub/incoming/ README
#将当前目录下的文件 README 上传到服务器的 /pub/incoming 目录下，建立以 .tmp 为后缀的临时文件
```

## 3.54 ncftpls (显示远程文件列表)



(1) 频度等级：☆

(2) 功能说明：

ncftpls 命令用于显示远程 FTP 服务器文件列表。

(3) 语法：

```
ncftpls [参数][ftp 地址]
```

(4) 所属包：

```
ncftp
```

(5) 使用技巧：

该命令相当于 ncftp 的简化版。

(6) 相关命令：

```
ncftp、ncftpbatch、ncftpls、ncftpput
```

(7) 参数：

ncftpget 的主要参数如表 3-52 所示。

表 3-52 ncftpls 参数表

参 数	含 义
-j count	除了提供用户名和密码以外，还提供一个账号供登录使用
-p passwd	指定登录所使用的密码
-P port	指定 FTP 端口
-u username	指定登录用户名，默认使用 anonymous 匿名登录
-v	显示版本信息

(8) 补充说明：

使用该命令时可以直接输入相关的参数。

(9) 典型实例：

例：显示远程 FTP 服务器文件列表。

例如显示 218.28.188.228 文件列表，可直接使用如下命令：

```
ncftpls-u whizer 218.28.188.228:211 #显示 218.28.188.228 文件列表
```

输出结果如下：

```
cmd@cmd-desktop:~$ ncftpls-u whizer 218.28.188.228:211 #显示 218.28.188.228 文件列表
Password:***** #输入密码
web/ #远程文件列表
db/
log/
```

# 3.55 paste (合并文件的行)



(1) 频度等级: ☆

(2) 功能说明:

paste 命令用于合并文件的行。该命令从在命令行上指定的文件中读取输入。如果出现“-”（减号）作为文件名，此命令从标准输入中读取。此命令连接给定文件中的行并把结果行写到标准输出中。

在默认情况下，paste 命令把每个文件当做栏，并用制表符水平链接它们（并行合并）。可以把 paste 命令看做 cat 命令（垂直连接，也就是一个接一个）的相对命令。

(3) 语法:

```
paste [-s][-d <间隔字符>][--help][--version][文件...]
```

(4) 所属包:

coreutils

(5) 使用技巧:

在默认情况下，paste 命令的合并结果会传送到标准输出，可以使用输出重定向将其保存到文件。该命令一般供管理员使用，完成信息的组合。

(6) 相关命令:

join

(7) 参数:

paste 主要参数如表 3-53 所示。

表 3-53 paste 参数表

参 数	含 义
-d<间隔字符>或--delimiters=<间隔字符>	用指定的间隔字符取代跳格字符
-s 或--serial	串行进行而非平行处理
--help	显示帮助信息
--version	显示版本信息

(8) 补充说明:

paste 指令会把每个文件以列对列的方式，一列列地加以合并。

(9) 典型实例:

例 1: 合并多个文件。

例如合并 file、testfile、testfile1 三个文件，可使用如下命令:

```
paste file testfile testfile1 #合并三个文件
```

合并前，使用 cat 命令查看这三个文件原来的内容，以便于对比。得到三个文件的内容分别为:

```
cmd@cmd-desktop:~$ cat file #file 文件的内容
wangyangming 1685
sunzhongshan 1926
huanggai 932
zhixi 1664
cmd@cmd-desktop:~$ cat testfile #testfile 文件的内容
wangyangming 20
sunzhongshan 15
huanggai 49
zhixi 77
cmd@cmd-desktop:~$ cat testfile1 #testfile1 文件的内容
wangyangming 89
sunzhongshan 76
huanggai 56
zhixi 73
```

使用 `paste` 命令合并三个文件得到如下输出结果：

```
cmd@cmd-desktop:~$ paste file testfile testfile1 #合并三个文件
#三个文件合并后输出到标准输出
Wangyangming    1685    wangyangming    20      wangyangming    89
Sunzhongshan    1926    sunzhongshan    15      sunzhongshan    76
huanggai        932     huanggai        49      huanggai        56
zhixi           1664    zhixi           77      zhixi           73
```

## 提示

`paste` 命令只是将文件合并显示，并没有改变源文件的内容。

例 2：合并单个文件中的多行。

`paste` 命令的参数“-s”的作用是将单个文件中的多行合并到一行，并显示在标准输出上。例如将 `testfile` 文件中的四行合并到一行输出到终端，可使用如下命令：

```
paste-s testfile #将 testfile 中的多行合并一行显示
输出结果如下：
```

```
cmd@cmd-desktop:~$ paste -s testfile
wangyangming    20    sunzhongshan    15    huanggai    49    zhixi    77
```

同样，该参数也只是将 `testfile` 文件的内容调整显示方式，并不改变原文件的内容，此时使用 `cat` 命令查看 `testfile` 可得到如下内容：

```
cmd@cmd-desktop:~$ cat testfile #查看 testfile 文件的内容
wangyangming    20
sunzhongshan    15
huanggai        49
zhixi           77
```

由此可知，`testfile` 中的内容并没有改变。

## 3.56 patch ( 修补文件 )



(1) 频度等级：☆☆

(2) 功能说明：

`patch` 命令用于修补文件。该命令读取如何更改文件的源文件指示信息，然后应用这些更改。源文件包含由 `diff` 命令产生的差别列表（或者 `diff` 列表）。差别列表是比较两个文件和构建关于如何纠正差别指示信息的结果。在默认情况下，`patch` 命令使用从标准输入读入源文件，但是使用“-i”标志和 `PatchFile` 变量可以覆盖此设置。

(3) 语法：

```
patch [-bceEfnNRstTuvZ][-B <备份字首字符串>][-d <工作目录>][-D <标识符号>][-F <鉴别列数>][-g <控制数值>][-i <修补文件>][-o <输出文件>][-p <剥离层级>][-r <拒绝文件>][-V <备份方式>][-Y <备份字首字符串>][-z <备份字尾字符串>][--backup-if -mismatch][--binary] [--help] 此
[--nobackup-if-mismatch][--verbose][原始文件 <修补文件>] 或 path [-p <剥离层级>] < [修补文件]
```

(4) 所属包：

```
patch
```

(5) 使用技巧：

`patch` 指令让用户利用设置修补文件的方式，修改、更新原始文件。倘若一次仅修改一个文件，可直接在指令列中下达指令依序执行。如果配合修补文件的方式，则能一次修补大批文件，这也是 Linux 系统核心的升级方法之一。

(6) 相关命令：

```
diff
```

## (7) 参数:

patch 主要参数如表 3-54 所示。

表 3-54 patch 参数表

参 数	含 义
-b 或--backup	备份每一个原始文件
-B<备份字首字符串>或 --prefix=<备份字首字符串>	设置文件备份时, 附加在文件名称前面的字首字符串, 该字符串可以是路径名称
-c 或--context	把修补数据解译成关联性的差异
-d<工作目录>或--directory=<工作目录>	设置工作目录
-D<标识符号>或--ifdef=<标识符号>	用指定的符号把改变的地方标识出来
-e 或--ed	把修补数据解译成“ed”命令可用的叙述文件
-E 或--remove-empty-files	若修补过后输出的文件内容是一片空白, 则移除该文件
-f 或--force	此参数的效果和指定“-t”参数类似, 但会假设修补数据的版本为新版本
-F<监别列数>或--fuzz=<监别列数>	设置监别列数的最大值
-g<控制数值>或--get=<控制数值>	设置以 RSC 或 SCCS 控制修补作业
-i<修补文件>或--input=<修补文件>	读取指定的修补文件
-l 或--ignore-whitespace	忽略修补数据与输入数据的跳格、空格字符
-n 或--normal	把修补数据解译成一般性的差异
-N 或--forward	忽略修补的数据较原始文件的版本更旧, 或该版本的修补数据已使用过
-o<输出文件>或--output=<输出文件>	设置输出文件的名称, 修补过的文件会以该名称存放
-p<剥离层级>或--strip=<剥离层级>	设置要剥离几层路径名称
-f<拒绝文件>或 --reject-file=<拒绝文件>	设置保存拒绝修补相关信息的文件名称, 预设的文件名称为.rej
-r 或--reverse	假设修补数据是由新旧文件交换位置而产生
-s 或--quiet 或--silent	不显示指令执行过程, 除非发生错误
-t 或--batch	自动略过错误, 不询问任何问题
-T 或--set-time	此参数的效果和指定“-Z”参数类似, 但以本地时间为主
-u 或--unified	把修补数据解译成一致化的差异
-V<备份方式>或 --version-control=<备份方式>	用“-b”参数备份目标文件后, 备份文件的字尾会被加上一个备份字符串, 这个字符串不仅可用“-z”参数变更, 当使用“-V”参数指定不同备份方式时, 也会产生不同字尾的备份字符串
-Y<备份字首字符串>或--basename-prefix=<备份字首字符串>	设置文件备份时, 附加在文件基本名称开头的字首字符串
-z<备份字尾字符串>或--suffix=<备份字尾字符串>	此参数的效果和指定“-B”参数类似, 差别在于修补作业使用的路径与文件名若为“src/linux/fs/super.c”, 加上“backup/”字符串后, 文件“super.c”会备份于“/src/linux/fsbackup”目录里
-Z 或--set-utc	更改修补过的文件, 存取时间设为 UTC
--backup-if-mismatch	在修补数据不完全吻合, 且没有刻意指定要备份文件时, 备份文件
--binary	以二进制模式读写数据, 而不通过标准输出设备
--nobackup-if-mismatch	在修补数据不完全吻合, 且没有刻意指定要备份文件时, 不要备份文件
--verbose	详细显示命令的执行过程

## (9) 典型实例:

例: 升级文件。

例如要将文件 testfile1 升级, 补丁为 testfile.patch, 该文件为 testfile1 升级前和修改后文件通过 diff 命令生成的补丁文件。使用 patch 为 testfile1 打包的命令为:

```
patch-p0 testfile1 testfile.patch #使用补丁程序升级文件
```

使用该命令前先查看 testfile1 的内容。首先将修改的文件与旧文件使用 diff 命令比较生成补丁文件, 使用 patch 后即可升级 testfile1 文件, 过程如下:

```
cmd@cmd-desktop:~$ cat testfile1 #查看 testfile1 的内容
Hello,This is the first file!
cmd@cmd-desktop:~$ cat testfile2 #查看 testfile2 的内容
Hello,This is the second file!
hdd@hdd-desktop:~$ diff testfile1 testfile2 #比较两个文件
1c1
< Hello,This is the first file!
---
```



```
> Hello,This is the second file!
hdd@hdd-desktop:~$ diff testfile1 testfile2 >testfile.patch
#将比较结果保存到 testfile.patch 文件
hdd@hdd-desktop:~$ cat testfile.patch
#查看补丁包的内容
lcl
< Hello,This is the first file!
---
> Hello,This is the second file!
hdd@hdd-desktop:~$ patch testfile1 < testfile.patch
#使用补丁包升级 testfile1 文件
patching file testfile1
hdd@hdd-desktop:~$ cat testfile1
#再次查看 testfile1 的内容
Hello,This is the second file!
#testfile1 文件被修改为和 testfile2 一样的内容
```

## 3.57 rcp ( 远程复制文件或目录 )



(1) 频度等级: ☆

(2) 功能说明:

rcp 代表“remote file copy (远程文件复制)”，该命令用于在计算机之间复制文件。

(3) 语法:

```
rcp [-pr][源文件或目录][目标文件或目录]
rcp [-pr][源文件或目录...][目标文件]
```

(4) 所属包:

rsh

(5) 使用技巧:

rcp 命令和 rsh、rlogin 命令同属于 R-系列工具。rlogin 可以不建立信任机制，登录用户直接给出口令；rsh 只需远程主机建立信任机制，而 rcp 则需要双方建立信任机制，这是由于 rcp 需要从远程主机复制文件到本地或将文件由本地复制到远程主机，所以需要双方建立信任机制，否则将出现“permission denied”的错误提示信息。

(6) 相关命令:

scp

(7) 参数:

rcp 主要参数如表 3-55 所示。

表 3-55 rcp 参数表

参 数	含 义
-p	保留源文件或目录的属性，包括拥有者、所属群组、权限与时间
-r	递归处理，将指定目录下的文件与子目录一并处理

(8) 补充说明:

rcp 指令用于复制远端文件或目录，如果同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到该目录中。

(9) 典型实例:

例：复制远程主机文件到本机。

假设本地主机当前账户为 cmd，远程主机账户为 cmd，要将远程主机（kernel.org）宿主目录下的文件 testfile 复制到本地，可使用如下任一命令：

```
rcp cmd@kernel.org:./testfile testfile #复制远程文件到本地
rcp cmd@kernel.org:home/cmd/testfile testfile
rcp kernel.org:./testfile testfile #要求当前登录账户 cmd 登录到远程主机
```

该命令没有返回信息，在 test 目录可以看到复制下来的文件 testfile。

## 3.58 rm ( 删除文件或目录 )



(1) 频度等级: ☆☆☆

(2) 功能说明:

**rm** 命令用于删除文件或目录。用户可以用 **rm** 命令删除不需要的文件。该命令的功能是删除一个目录中的一个或多个文件或目录, 它也可以将某个目录及其下的所有文件及子目录全部删除。对于链接文件, 只是断开了链接, 源文件保持不变。

(3) 语法:

```
rm [-dfrv][--help][--version][文件或目录...]
```

(4) 所属包:

coreutils

(5) 使用技巧:

使用“-i”参数能防止误删除, 该参数会在删除文件前给出提示信息且让用户确认删除, 使用该参数可提高系统安全性。

使用参数“-f”可以在不给出任何警告信息的情况下删除文件或目录, 因此用户应慎用。

要删除指定目录, 必须使用参数“-r”或“-R”。

(6) 相关命令:

rmdir、mv

(7) 参数:

**rm** 主要参数如表 3-56 所示。

表 3-56 rm 参数表

参 数	含 义
-d 或--directory	直接把要删除的目录的硬链接数据变成 0, 删除该目录
-f 或--force	强制删除文件或目录, 不给出任何确认删除信息, 并且忽略不存在的文件, 该参数慎用
-i 或--interactive	删除已有文件或目录之前先询问用户
-r 或-R 或--recursive	递归处理, 将指定目录下的所有文件及子目录一起处理
-v 或--verbose	显示指令执行过程

(8) 典型实例:

**例 1:** 删除目录或文件。**rm** 命令不能直接删除目录, 需要使用参数“-R”递归删除目录及后面的内容, 直接删除将产生错误信息。

例如要删除当前目录下的 **test** 子目录, 可使用如下命令:

```
rm-R test/          #删除当前目录下的 test 子目录
```

删除前, 使用 **ls** 查看当前目录结构:

```
cmd@cmd-desktop:~$ ls          #查看当前目录结构
examples.desktop  README  file    test    testfile1  testfile
```

删除的结果如下:

```
cmd@cmd-desktop:~$ rm test/    #直接删除将产生错误
rm: 无法删除 " test/" : 是一个目录      #提示不能直接删除目录
cmd@cmd-desktop:~$ rm -R test/    #使用参数-R 删除目录
cmd@cmd-desktop:~$ ls          #再次查看当前目录
examples.desktop  README  file    testfile1  testfile  #test 子目录已被删除
```

**例 2:** 使用交互式模式删除当前目录中所有扩展名为“.txt”的文件, 并且在删除前询问用户是否删除, 使用命令如下:

```
rm-iv *.txt        #使用交互式模式删除当前目录中所有扩展名为 ".txt" 的文件
```

该命令的输出结果如下:

```
hdd@hdd-desktop:~$ ls          #查看删除前的目录结构
a.txt  d.txt    testfile1.txt  testfile4.txt  模板  文档
b.txt  Examples testfile2.txt  testfile.txt   视频  音乐
c.txt  test     testfile3.txt  公共的        图片  桌面
hdd@hdd-desktop:~$ rm -iv *.txt #使用交互式模式删除当前目录中所有扩展名为
# ".txt" 的文件
```

```

rm: 是否删除 普通空文件 “a.txt” ? y          #删除 “a.txt” 前询问, 等待用户确认
已删除 “a.txt”                               #删除结果
rm: 是否删除 普通空文件 “b.txt” ? n
rm: 是否删除 普通空文件 “c.txt” ? y
已删除 “c.txt”
rm: 是否删除 普通空文件 “d.txt” ? y
已删除 “d.txt”
rm: 是否删除 普通空文件 “testfile1.txt” ? n
rm: 是否删除 普通空文件 “testfile2.txt” ? y
已删除 “testfile2.txt”
rm: 是否删除 普通空文件 “testfile3.txt” ? y
已删除 “testfile3.txt”
rm: 是否删除 普通空文件 “testfile4.txt” ? n
rm: 是否删除 普通空文件 “testfile.txt” ? y
已删除 “testfile.txt”
hdd@hdd-desktop:~$ y
bash: y: 找不到命令
hdd@hdd-desktop:~$ ls          #使用 rm 删除命令后, 再次查看目录结构, 确认删除的文件已经不存在
b.txt      test          testfile4.txt  模板  图片  音乐
Examples  testfile1.txt  公共的      视频  文档  桌面
    
```

## 3.59 scp ( 远程复制文件 )



(1) 频度等级: ☆

(2) 功能说明:

scp 代表 “Secure Copy”, 用来进行远程文件复制, 使用 ssh1 传输数据, 并且和 ssh1 使用相同的认证方式, 提供相同的安全保证。与 rcp 不同的是, 在需要的时候 scp 会要求你输入密码。

(3) 语法:

```

scp [选项][文件 1][文件 2]
scp [选项][文件 1 文件 2...文件 n][目标目录文件]
    
```

(4) 所属包:

scp

(5) 使用技巧:

scp 是 SSH 服务提供安全主机间文件复制的实现接口, 与 ssh、sftp 命令一样, 只要安装了 OPENSSSH 软件, 即可实现。

(6) 相关命令:

rcp、sftp

(7) 参数:

scp 主要参数如表 3-57 所示。

表 3-57 scp 参数表

参 数	含 义
-1	强制使用版本 1 的 SSH 通信协议
-2	强制使用版本 2 的 SSH 通信协议
-C	以压缩方式传输文件
-o <ssh 参数>	以 ssh-config(5)的格式传输参数给 SSH 服务, 比如-o port 指定 SSH 服务的监听端口。SSH 参数及其他合法值, 可以参考 ssh-config(5)手册
-p<端口>	指定服务器端 SSH 服务监听端口, SSH 服务默认的监听端口为 23。若服务器修改其默认监听端口, 必须在命令行给出监听端口, 否则将不能成功和服务器建立连接
-l<带宽限制>	限制文件传输协议的带宽, 以 Kb/s 为单位
-P	复制文件的时候, 保持源文件的修改时间、访问时间属性及访问属性

-q scp	默认显示文件复制的进度，该参数将关闭文件传输进度
-r	若源文件为目录，则递归复制该目录的内容
-v	开启冗长模式

(8) 补充说明:

scp 命令与 rcp 命令功能相似，不同的是，scp 对传输的数据进行了加密，提高了文件传输的安全性。

(9) 典型实例:

例: 复制远程主机文件到本机，并复制本地文件到远程主机。

假设本地主机当前账户为 cmd，远程主机账户为 cmd，要将远程主机 (kernel.org) 宿主目录下的文件 testfile 复制到本地，可使用如下任一命令:

```
scp cmd@kernel.org:./testfile testfile      #复制远程文件到本地
scp cmd@kernel.org:home/cmd/testfile testfile
scp kernel.org:./testfile testfile          #要求当前登录账户 cmd 登录到远程主机
```

该命令没有返回信息，在 test 目录可以看到复制下来的文件 testfile。



## 3.60 slocate ( 查找文件或目录 )

(1) 频度等级: ☆☆

(2) 功能说明:

slocate 命令用于查找文件或目录。

(3) 语法:

```
slocate [-u][--help][--version][-d <目录>][查找的文件]
```

(4) 所属包:

```
slocate
```

(5) 使用技巧:

slocate 命令是 GNU locate 命令的安全增强版，在大多数 Linux 发行版本中，locate 命令实际上是 slocate 命令的符号链接。

(6) 相关命令:

```
locate
```

(7) 参数:

slocate 参数参考 locate 命令。

(8) 补充说明:

slocate 本身具有一个数据库，里面存放了系统中文件与目录的相关信息。

(9) 典型实例:

例: 快速定位文件路径。

同 locate 一样，使用 slocate 不指定路径命令，可以显示详细位置信息，例如在命令行中输入如下命令:

```
slocate fdisk      #显示文件名中含有 fdisk 关键字的文件的的路径信息
```

输出信息如下:

```
cmd@cmd-desktop:~$ slocate fdisk #显示文件名中含有 fdisk 关键字的文件的的路径信息
/sbin/cfdisk      #搜索到的文件路径列表
/sbin/fdisk
/sbin/sfdisk
/usr/include/grub/ieee1275/ofdisk.h
/usr/share/doc/util-linux/README.cfdisk
/usr/share/doc/util-linux/README.fdisk.gz
/usr/share/doc/util-linux/examples/sfdisk.examples.gz
/usr/share/man/man8/cfdisk.8.gz
/usr/share/man/man8/fdisk.8.gz
```



## 3.61 split (切割文件)

(1) 频度等级: ☆☆

(2) 功能说明:

split 命令用于切割文件。该命令将大文件分割成较小的文件, 默认情况下每 1000 行分割成一个小文件。

(3) 语法:

```
split [--help][--version][-<行数>][-b <字节>][-C <字节>][-l <行数>][要切割的文件][输出文件名]
```

(4) 所属包:

coreutils

(5) 使用技巧:

特别大的文件很难用文本编辑器直接打开, 可以先使用 split 命令将其分割, 然后再用文本编辑器打开。

(6) 相关命令:

csplit

(7) 参数:

split 主要参数如表 3-58 所示。

表 3-58 split 参数表

参 数	含 义
-<行数>或-l<行数>	指定每多少行切成一个小文件
-b<字节>	指定每多少字节切成一个小文件
-C<字节>	与“-b”参数类似, 但切割时尽量维持每行的完整性
[输出文件名]	设置切割后文件的前置文件名, split 会自动在前置文件名后加上编号
--help	显示帮助信息
--version	显示版本信息

(8) 典型实例:

例: 分割大文件。

split 命令将大文件分割成小文件。例如, 将 README 文件每六行分割成一个文件, 可以使用如下命令:

```
split -6 README #将 README 文件每六行分割成一个文件
```

首先使用 cat 命令查看 README 文件内容如下:

```
cmd@cmd-desktop:~$ cat README
Welcome to the
LINUX KERNEL ARCHIVES
ftp.kernel.org
"Much more than just kernels"
IF YOU'RE ACCESSING THIS SITE VIA A WEB BROWSER
PLEASE USE THE HTTP URL BELOW INSTEAD!
----> If you are looking for mirror sites, please go <----
----> to mirrors.kernel.org instead <----
This site is provided as a public service by the Linux Kernel
Organization, a California nonprofit corporation. Bandwidth is
provided by The Internet Software Consortium, Inc. Our servers are
located in San Francisco and Palo Alto, California; Corvallis, Oregon;
Amsterdam, Netherlands and Umeå, Sweden; use in violation of any
applicable laws strictly prohibited.
Due to U.S. Exports Regulations, all cryptographic software on this
site is subject to the following legal notice:
```



This site includes publicly available encryption source code which, together with object code resulting from the compiling of publicly available source code, may be exported from the United States under License Exception "TSU" pursuant to 15 C.F.R. Section 740.13(e).

This legal notice applies to cryptographic software only. Please see the Bureau of Industry and Security (<http://www.bis.doc.gov/>) for more information about current U.S. regulations.

Neither the Linux Kernel Organization, nor its sponsors make any guarantees, explicit or implicit, about the contents of this site.

Use at your own risk.

This site is accessible via the following mechanisms:

```
FTP          ftp://ftp.kernel.org/pub/
HTTP        http://www.kernel.org/pub/
RSYNC      rsync://rsync.kernel.org/pub/
```

NFS and SMB/CIFS are no longer available.

For comments on this site, please contact <[ftpadmin@kernel.org](mailto:ftpadmin@kernel.org)>.

Please do not use this address for questions that are not related to the operation of this site. Please see our homepage at <http://www.kernel.org/> for links to Linux documentation resources.

使用 `split` 命令分割文件，该命令没有输出结果，如下所示：

```
cmd@cmd-desktop:~$ split -6 README
cmd@cmd-desktop:~$
```

如果使用 `ls` 命令查看当前目录可得到如下目录结构：

```
README xaa xad xag xab xae xah xac xaf xai
```

由此可以看到，`split` 将原来的大文件 `README` 分割成多个以“x”开头的新的的小文件。事实上，这些小文件中，每个文件都只有 6 行内容。

## 3.62 tar ( 压缩/解压缩文件 )



(1) 频度等级：☆☆☆

(2) 功能说明：

`tar` 命令用于备份文件。该命令是 UNIX 的一个 shell 命令，可以为多个指定文件创建一个档案文件，也可以从一个档案文件中解压缩出文件。`tar` 档案文件的扩展名为“`.tar`”。`tar` 包中的文件并不是压缩文件，而是所有文件集成的一个文件。

`tar` 这个名字源自自在磁带上备份文件的时代。现在 `tar` 档案文件多用于 UNIX 系统中的文件传输。Windows 系统中比较常用的压缩工具为 WinZip，它也可以用来解压缩 `tar` 档案文件。Tarball 是 `tar` 档案文件的专业术语，意思是“把一捆文件粘成一个沥青球”。

(3) 语法：

```
tar[-ABcdgGhiklmMoOpPrRsStuUvwWxzZ][-b <区块数目>][-C <目的目录>][-f <备份文件>][-F <Script 文件>][-K <文件>][-L <媒体容量>][-N <日期时间>][-T <范本文件>][-V <卷册名称>][-X <范本文件>][-<设备编号><存储密度>][--after-date=<日期时间>][--atime-preserve] [--backup=<备份方式>][--checkpoint][--concatenate][--confirmation][--delete][--exclude=<范本样式>][--force-local][--group=<群组名称>][--help][--ignore-failed-read][--new-volume-script=<Script 文件>][--newer-mtime][--no-recursion][--null][--numeric-owner] [--owner=<用户名>] [--posix] [--preserve][--preserve-order][--preserve-permissions][--record-size=<区块数目>] [--recursive-unlink][--remove-files][--rsh-command=<执行指令>][--same-owner][--suffix=<备份字尾字符串>][--totals][--use-compress-program=<执行指令>][--version][--volno-file=<编号文件>][文件或目录...]
```

(4) 所属包：

tar

(5) 使用技巧:

该命令是一个打包程序,最好结合压缩工具对其进行压缩,以节省磁盘空间,在使用他人命令打包文件的同时可以使用“-z”、“-Z”等参数压缩打包文件。

使用 tar 命令的“-r”参数向归档文件尾部追加文件时,要求归档文件没有被压缩,否则不能实现“-r”参数的功能。

(6) 相关命令:

Compress、cpio、gzip

(7) 参数:

tar 主要参数如表 3-59 所示。

表 3-59 tar 参数表

参 数	含 义
-A 或--catenate	新增文件到已存在的备份文件
-b<区块数目>或--blocking-factor=<区块数目>	设置每笔记录的区块数目,每个区块大小为 12B
-B 或--read-full-records	读取数据时重设区块大小
-c 或--create	建立新的备份文件
-C<目的目录>或--directory=<目的目录>	切换到指定的目录
-d 或--diff 或--compare	对比备份文件内和文件系统上的文件的差异
-f<备份文件>或--file=<备份文件>	指定备份文件
-F<Script 文件>或--info-script=<Script 文件>	每次更换磁带时,就执行指定的 Script 文件
-g 或--listed-incremental	处理 GNU 格式的大量备份
-G 或--incremental	处理旧的 GNU 格式的大量备份
-h 或--dereference	不建立符号链接,直接复制该链接所指向的原始文件
-i 或--ignore-zeros	忽略备份文件中的 0 Byte 区块,也就是 EOF
-k 或--keep-old-files	解开备份文件时,不覆盖已有的文件
-K<文件>或--starting-file=<文件>	从指定的文件开始还原
-l 或--one-file-system	复制的文件或目录存放的文件系统,必须与 tar 指令执行时所处的文件系统相同,否则不予复制
-L<媒体容量>或--tape-length=<媒体容量>	设置存放媒体的容量,单位以 1024 B 计算
-m 或--modification-time	还原文件时,不变更文件的更改时间
-M 或--multi-volume	在建立、还原备份文件或列出其中的内容时,采用多卷册模式
-N<日期格式>或--newer=<日期时间>	只将较指定日期更新的文件保存到备份文件中
-o 或--old-archive 或--portability	将资料写入备份文件时使用 V7 格式
-O 或--stdout	把从备份文件中还原的文件输出到标准输出设备
-p 或--same-permissions	用原来的文件权限还原文件
-P 或--absolute-names	文件名使用绝对名称,不移除文件名称前的“/”号
-r 或--append	新增文件到已存在的备份文件的结尾部分
-R 或--block-number	列出每个信息在备份文件中的区块编号
-s 或--same-order	还原文件的顺序和备份文件内的存放顺序相同
-S 或--sparse	若一个文件内含大量的连续 0 字节,则将此文件存成稀疏文件
-t 或--list	列出备份文件的内容
-T<范本文件>或--files-from=<范本文件>	指定范本文件,其内含有一个或多个范本样式,让 tar 解开或建立符合设置条件的文件
-u 或--update	仅置换较备份文件内的文件更新的文件
-U 或--unlink-first	解开压缩文件还原文件之前,先解除文件的链接
-v 或--verbose	显示指令执行过程
-V<卷册名称>或--label=<卷册名称>	建立使用指定的卷册名称的备份文件
-w 或--interactive	遇到问题时先询问用户
-W 或--verify	写入备份文件后,确认文件正确无误

-x 或--extract 或--get	从备份文件中还原文件
-X<范本文件>或--exclude-from=<范本文件>	指定范本文件，其内含有一个或多个范本样式，让 tar 排除符合设置条件的文件
-z 或--gzip 或--ungzip	通过 gzip 指令处理备份文件
-Z 或--compress 或--uncompress	通过 compress 指令处理备份文件
-<设备编号><存储密度>	设置备份用的外围设备编号及存放数据的密度
--after-date=<日期时间>	此参数的效果和指定“-N”参数相同
--atime-preserve	不变更文件的存取时间
--backup=<备份方式>或--backup	移除文件前先进行备份
--checkpoint	读取备份文件时列出目录名称
--concatenate	此参数的效果和指定“-A”参数相同
--confirmation	此参数的效果和指定“-w”参数相同
--delete	从备份文件中删除指定的文件
--exclude=<范本样式>	排除符合范本样式的文件
--group=<群组名称>	把加入设备文件中文件的所属群组设成指定的群组
--ignore-failed-read	忽略数据读取错误，不中断程序的执行
--new-volume-script=<Script 文件>	此参数的效果和指定“-F”参数相同
--newer-mtime	只保存更改过的文件
--no-recursion	不做递归处理，也就是指定目录下的所有文件及子目录不予处理
--null	从 null 设备读取文件名称
--numeric-owner	以用户识别码及群组识别码取代用户名称和群组名称
--owner=<用户名>	把加入备份文件中的文件的拥有者设成指定的用户
--posix	将数据写入备份文件时使用 POSIX 格式
--preserve	此参数的效果和指定“-ps”参数相同
--preserve-order	此参数的效果和指定“-A”参数相同
--preserve-permissions	此参数的效果和指定“-p”参数相同
--record-size=<区块数目>	此参数的效果和指定“-b”参数相同
--recursive-unlink	解开压缩文件还原目录之前，先解除整个目录下所有文件的链接
--remove-files	文件加入备份文件后，就将其删除
--rsh-command=<执行指令>	设置要在远端主机上执行的指令，以取代 rsh 指令
--same-owner	尝试以相同的文件拥有者还原文件
--suffix=<备份字尾字符串>	移除文件前先进行备份
--totals	备份文件建立后，列出文件大小
--use-compress-program=<执行指令>	通过指定的指令处理备份文件
--volno-file=<编号文件>	使用指定文件内的编号取代预设的卷册编号

#### (8) 补充说明：

tar 是用来建立、还原备份文件的工具程序，它可以加入、解开备份文件内的文件。

#### (9) 典型实例：

##### 例 1：打包备份文件。

tar 的主要作用之一是创建打包文件，当该命令用于打包文件时，需配合使用参数“-c”，结合参数“-z”可将文件包压缩成 gzip 格式。例如，要将当前目录下的 test 文件夹打包并压缩成 gzip 格式，可使用如下命令：

```
tar-cvzf test.tar.gz test/ #将当前目录下的 test 文件夹打包并压缩成 gzip 格式，其中-c
#表示打包文件，-v 表示显示详细过程，-z 表示使用 gzip 压缩，-f 表示指定文件
```

该命令将会有如下输出结果：

```
cmd@cmd-desktop:~$ tar -cvzf test.tar.gz test/ #将当前目录下的 test 文件夹打包并压缩成 gzip 格式
test/ #压缩文件列表
test/testfile1
test/xag
```

```
test/README
test/xaa
test/xad
test/xac
test/file
test/xai
test/xaf
test/testfile
test/xab
test/xae
test/xah
```

例 2: 解压缩打包文件。

`tar` 的另外一个重要功能是解压缩打包文件, 当该命令用于解压缩文件时, 需配合使用参数“-x”, 结合参数“-z”可将以 `gzip` 压缩的文件格式解压。例如, 要将当前目录下的打包文件 `test.tar.gz` 解压, 可使用如下命令:

```
tar-xvzf test.tar.gz #将当前目录下的 test.tar.gz 文件解压, 其中-x 表示解压文件, -v
#表示显示详细过程, -z 表示使用 gzip 解压, -f 表示指定文件
```

该命令将会有如下输出结果:

```
cmd@cmd-desktop:~$ tar -xvzf test.tar.gz #将当前目录下的 test.tar.gz 文件解压
test/ #解压的文件列表
test/testfile1
test/xag
test/README
test/xaa
test/xad
test/xac
test/file
test/xai
test/xaf
test/testfile
test/xab
test/xae
test/xah
```

## 3.63 tee (读取标准输入的数据并输出成文件)

(1) 频度等级: ☆

(2) 功能说明:

`tee` 命令用于读取标准输入的数据, 并将其内容输出成文件。若用户同时需要一个文件的多个副本, 可以创建 (`cat` 或 `vi`) 该文件, 然后把该文件复制 (`cp`) 多个副本。但是利用 `tee` 命令可以一次实现该功能, `tee` 一方面接收从标准输入设备读入数据、打印一行到标准输出设备, 同时可以保存为多个文件, 这样一条命令就可以实现上述的系列操作。

(3) 语法:

```
tee [-ai][--help][--version][文件...]
```

(4) 所属包:

```
coreutils
```

(5) 使用技巧:

使用“-a”参数可以将输入的内容追加到指定文件尾部。

(6) 参数:

tee 主要参数如表 3-60 所示。

表 3-60 tee 参数表

参 数	含 义	参 数	含 义
-a 或--append	附加到已有文件的后面,而非覆盖它	--help	显示帮助信息
-i 或--ignore-interrupts	忽略中断信号	--version	显示版本信息

(7) 补充说明:

tee 指令会从标准输入设备读取数据,将其内容输出到标准输出设备,同时保存成文件。

tee 命令存在缓存,每 1024 个字节输出一次,若从管道接收输入数据,应该是缓冲满,才将数据转存到指定的文件中;若文件内容不到 1024 个字节,则接收完从标准输入设备读入数据,刷新一次缓冲,转存数据到指定文件中。

(8) 典型实例:

例:复制内容到多个文件。

例如,直接向 tee 命令传递 3 个新文件名,可使用如下命令:

```
tee file1 file2 file3 #直接向 tee 命令传递 file1、file2、file3 文件名
```

输出信息如下:

```
hello #输入字符串
hello #输出字符串,反馈输出
```

上述命令之后,会新建 3 个具有相同内容的文件,且文件名分别为 file1、file2、file3,内容为“hello”。

## 3.64 tftp (传输文件)



(1) 频度等级: ☆

(2) 功能说明:

tftp 命令用于传输文件,该命令是简单文件传输协议的一个客户端软件的实现,用来和 TFTP 服务器联机操作。

(3) 语法:

```
tftp [主机名称或 IP 地址]
```

(4) 所属包:

```
tftp
```

(5) 使用技巧:

tftp 命令在联机模式下以“tftp>”为提示符。

使用 tftp 命令上传文件时,要求上传的文件在 TFTP 服务器已经存在,即 tftp 命令只能覆盖服务器上的文件。如果该文件不存在,则提示“Error code 1:File not found”错误信息。

tftp 命令使用 UDP 协议传送文件,适合于网络质量较好的场合。

tftp 和服务器联机后的内置命令操作如表 3-61 所示。

表 3-61 tftp 操作命令表

操 作	功 能	操 作	功 能
connect	链接远程 TFTP 服务器	trace	打开数据包跟踪模式
mode	设置文件传输模式	binary	使用二进制传输模式,与“mode binary”等价
put	上传文件到 TFTP 服务器	ascii	使用网络 ASCII 模式,与“mode ascii”等价
get	从 TFTP 服务器下载文件	rexmt	指定数据包的传输超时时间
quit	退出 TFTP 命令	timeout	设置总共的重传超时时间
verbose	打开冗长模式	?	显示帮助信息,与 help 等价

(6) 相关命令:

```
ftp
```



(7) 参数:

tftp 主要参数如表 3-62 所示。

表 3-62 tftp 参数表

参 数	含 义
-c <命令>	指定和服务器联机后要执行的命令
-m <模式>	指定文件的传输模式, 如 ascii、binary
-v	显示命令的详细执行过程
-V	显示版本信息

(8) 补充说明:

tftp 让用户得以下载存放于远端主机的文件, 也能将文件上传到远端主机放置。tftp 是简单的文字模式 FTP 程序, 它所使用的指令和 ftp 类似。

(9) 典型实例:

例: 连接远程服务器并下载文件。

例如连接远程服务器“218.28.188.288”, 然后使用 put 命令下载其中根目录下的文件“README”, 可使用命令如下:

```
tftp 218.28.188.288          #连接远程服务器
链接服务器之后可进行相应的操作, 具体如下:

hdd@hdd-desktop:~$ tftp 218.28.188.228          #连接远程服务器
tftp> ?                                          #使用?, 参考帮助
Commands may be abbreviated.  Commands are:    #帮助命令列表
connect connect to remote tftp
mode      set file transfer mode
put       send file
get       receive file
quit      exit tftp
verbose   toggle verbose mode
trace     toggle packet tracing
status    show current status
binary    set mode to octet
ascii     set mode to netascii
rexmt     set per-packet retransmission timeout
timeout   set total retransmission timeout
?         print help information
tftp>get README                          #远程下载 README 文件
getting from 218.28.188.288 to /home/cmd
Recived 168236 bytes in 1.5 seconds[112157 bit/s]
tftp>quit                                  #离开 tftp
```

## 3.65 tmpwatch ( 删除临时文件 )



(1) 频度等级: ☆

(2) 功能说明:

tmpwatch 命令用于删除暂存的临时文件, 以节省磁盘空间。该命令的工作前提是将不常用的文件集中到一个目录中, 然后利用该命令对该目录进行时间设置, 这样 tmpwatch 命令将会自动检测该目录下的文件是否过期, 若过期将自动删除。

(3) 语法:

```
tmpwatch [-afqv][--test][超期时间][目录...]
```

(4) 所属包:

tmpwatch

(5) 使用技巧:

执行 tmpwatch 指令可删除不必要的暂存文件, 可以设置文件超期时间, 单位以小时计算。

(6) 相关命令:

rm

(7) 参数:

tmpwatch 主要参数如表 3-63 所示。

表 3-63 tmpwatch 参数表

参 数	含 义
-a 或--all	删除任何类型的文件
-c 或--ctime	通过节点的最后改变时间判断文件是否过期
-d--nodirs	不删除目录, 即使目录为空也不删除
-d--nosymlinks	不删除符号链接
-f 或--force	强制删除文件或目录, 其效果类似 rm 指令的“-f”参数
-m 或--mtime	通过文件的最后修改时间判断文件是否过期
-M	通过目录的最后修改时间判断文件是否过期
-q 或--quiet	不显示指令执行过程
--test	仅仅测试性地删除文件, 不是真正删除
-v 或--verbose	详细显示指令执行过程

(8) 补充说明:

运行该命令需要管理员 (root) 权限, 在普通用户下运行该命令将会提示该命令并未找到。

(9) 典型实例:

例: 删除/tmp 目录中超过一天未使用的文件。具体命令为:

```
tmpwatch 24 /tmp/ #删除/tmp 目录中超过一天未使用的文件
```

该命令需要 root 权限, 因此在使用 tmpwatch 命令前应该使用 su 命令切换用户, 最后有如下输出:

```
hdd@hdd-desktop:~$su #切换到 root 用户
口令: #输入密码
hdd@hdd-desktop:~# tmpwatch-v 24 /tmp/ #删除/tmp 目录中超过一天未使用的文件, -v 表示显示详细过程
removing directctmp/orbit-tom if not empty
```



## 3.66 touch ( 改变文件或目录时间 )

(1) 频度等级: ☆☆

(2) 功能说明:

touch 命令用于改变文件或目录时间。文件的时间属性包括文件的最后访问时间、最后修改时间, 以及最后在磁盘上修改的时间, 命令 stat 显示结果能显示文件对应的三个时间属性; 文件的时间属性记录了用户对文件的所有操作时间, 而 touch 命令可以对这些时间值进行改变, 但前提条件是用户对该文件具有写权限。另外, touch 命令只能改变文件的最后访问时间和最后修改时间。

(3) 语法:

```
touch [-acfm][ -d <日期时间> ][ -r <参考文件或目录> ][ -t <日期时间> ][ --help ][ --version ] [ 文件或目录... ]
touch [-acfm][ --help ][ --version ] [ 日期时间 ] [ 文件或目录... ]
```

(4) 所属包:

coreutils

(5) 使用技巧:

该命令搭配适当的 shell 通配符可以批量创建空文件。

使用该命令可以创建原来不存在的空文件，新建的空文件的最后访问时间和最后修改时间均为当前系统时间。

如果不使用“-r”、“-t”参数，则文件的时间属性将被改为当前的系统时间。

(6) 相关命令：

chattr

(7) 参数：

touch 主要参数如表 3-64 所示。

表 3-64 touch 参数表

参 数	含 义
-a 或--time=atime 或--time=access 或--time=use	只更改存取时间
-c 或--no-create	不建立任何文件
-d<时间日期>	使用指定的日期时间，而非现在的时间
-f	此参数将忽略不予处理，仅负责解决 BSD 版本 touch 指令的兼容性问题
-m 或--time=mtime 或--time=modify	只更改变动时间
-r<参考文件或目录>或--reference<参考文件或目录>	把指定文件或目录的日期时间，设成和参考文件或目录的日期时间相同
-t<日期时间>	使用指定的日期时间，而非现在的时间。时间格式为 MMDDhhmm[[CC]YY][.ss]。从左至右分别表示月、日、小时、分钟、世纪、年、秒

(8) 典型实例：

例 1：修改文件的时间属性。

例如，要修改 testfile 文件的时间属性为当前系统时间，可使用如下命令

```
touch testfile          #修改 testfile 时间属性为当前系统时间
```

首先使用 ls 命令查看 testfile 文件的属性如下所示：

```
cmd@cmd-desktop:~$ ls -l testfile          #查看文件的时间属性
-rw-r--r-- 1 hdd hdd 55 2009-09-21 16:09 testfile#原来文件的修改时间为 16:09
```

使用 touch 命令修改文件属性，并再次查看该文件的时间属性：

```
cmd@cmd-desktop:~$ touch testfile          #修改文件时间属性为当前系统时间
cmd@cmd-desktop:~$ ls -l testfile          #查看文件的时间属性
-rw-r--r-- 1 hdd hdd 55 2009-09-21 19:53 testfile#修改后文件的时间属性为当前系统时间
```

例 2：创建新的空文件。

使用 touch 命令时，如果该文件不存在，则会创建一个内容为空的新文件，例如使用如下命令：

```
touch testfile3        #创建一个名为 testfile3 的新空白文件
```

首先使用 ls 命令查看当前目录结构，结果如下：

```
cmd@cmd-desktop:~$ ls          #查看当前目录的文件结构
test testfile testfile1 testfile2    #当前目录下并不存在 testfile3 文件
```

使用 touch 命令创建 testfile3 文件后，再次查看目录结构，信息如下：

```
cmd@cmd-desktop:~$ touch testfile3    #创建一个名为 testfile3 的新空白文件
cmd@cmd-desktop:~$ ls          #再次查看当前目录的文件结构
test testfile testfile1 testfile2 testfile3    #在当前目录创建了 testfile3 文件
```

## 3.67 tree (以树状显示目录结构)



(1) 频度等级：☆☆

(2) 功能说明：

tree 命令用于以树状图列出目录的内容，该命令支持递归目录，并能够产生色彩化的输出。

(3) 语法：

```
tree [-aACdDfGglnNpqstux][-I <范本样式>][--P <范本样式>][目录...]
```

(4) 所属包：

tree

(5) 使用技巧:

查看 tree 命令的详细用法, 可使用命令 “info tree | more”。

(6) 相关命令:

ls

(7) 参数:

tree 主要参数如表 3-65 所示。

表 3-65 tree 参数表

参 数	含 义
-a	显示所有文件和目录
-A	使用 ANSI 绘图字符显示树状图, 而非以 ASCII 字符组合
-C	给文件和目录清单加上色彩, 便于区分各种类型
-d	显示目录名称而非内容
-D	列出文件或目录的更改时间
-f	在每个文件或目录之前, 显示完整的相对路径名称
-F	在执行文件、目录、Socket、符号链接、管道名称时各自加上 “*”、“/”、“=”、“@”、“ ”号
-g	列出文件或目录的所属群组名称, 没有对应的名称时, 显示群组识别码
-i	不以阶梯状列出文件或目录名称
-I<范本样式>	不显示符合范本样式的文件或目录名称
-l	如果遇到性质为符号链接的目录, 就直接列出该链接所指向的原始目录
-n	不在文件和目录清单加上色彩
-N	直接列出文件和目录名称, 包括控制字符
-p	列出权限标识
-P<范本样式>	只显示符合范本样式的文件或目录名称
-q	用 “?” 号取代控制字符, 列出文件和目录名称
-s	列出文件或目录大小
-t	用文件和目录的更改时间排序
-u	列出文件或目录的拥有者名称, 没有对应的名称时, 显示用户识别码
-x	将范围局限在现行的文件系统中, 若指定目录下的某些子目录存放于另一个文件系统中, 则将孩子目录排除在寻找范围之外

(8) 补充说明:

执行 tree 指令, 它会列出指定目录下的所有文件, 包括子目录里的文件。

(9) 典型实例:

例: 以树状图列出当前目录结构。可直接使用如下命令:

tree #以树状图列出当前目录结构

该命令有如下输出结果:

```
hdd@hdd-desktop:~$ tree #以树状图列出当前目录结构
. #当前目录结构
|-- README
|-- examples.desktop
|-- file
|-- file.new
|-- index.htm
|-- test
|   |-- README
|   |-- file
|   |-- testfile
|   |-- testfile1
|   |-- xaa
|   |-- xab
|   |-- xac
|   |-- xad
|   |-- xae
|   |-- xaf
```

```

|-- xag
|-- xah
`-- xai
|-- test.tar.gz
|-- test.zip
|-- testfile
|-- testfile.new
|-- testfile.patch
|-- testfile1
|-- testfile2
|-- testfile3
|-- xaa
|-- xab
|-- xac
|-- xad
|-- xae
|-- xaf
|-- xag
|-- xah
|-- xai
|-- \345\205\254\345\205\261\347\232\204
|-- \345\233\276\347\211\207
|-- 075b5c2bb1628c1a5343c10a.jpg
|-- 0c978fe989ac787e799757095719d3c4.jpg
|-- 20050726194826866443.jpg
|-- 20061113171548785122.jpg
|-- 2007102221576687.jpg
|-- 39.jpg
|-- 434887ec4340916a78f0559a.jpg
|-- 498da016ac02fb2bc93d6d08.jpg
|-- 7b284f5a0f854da2f3bf90b204149a34.jpg
|-- 9196c030d342a68d5edf0e98.jpg
|-- a56c5a90de15c8a9a977a4cc.jpg
|-- c74f62167c9d2b244a90a79e.jpg
`-- img13.jpg
|-- \346\226\207\346\241\243
|-- \346\241\214\351\235\242
|-- \346\250\241\346\235\277
|-- \350\247\206\351\242\221
`-- \351\237\263\344\271\220
    
```

8 directories, 48 files

#统计信息, 该目录共 8 个子目录, 48 个文件

## 3.68 umask ( 指定在建立文件时默认的权限掩码 )

(1) 频度等级: ☆☆

(2) 功能说明:

**umask** 命令用于指定在建立文件时预设的权限掩码。当新文件被创建时, 其最初的权限由文件创建掩码决定。用户每次注册进入系统时, **umask** 命令都被执行, 并自动设置掩码改变默认值, 新的权限将会把旧的覆盖。

(3) 语法:

```
umask [-S][权限掩码]
```

(4) 所属包:

```
bash
```

(5) 使用技巧:

**umask** 命令可用来设定权限掩码。

(6) 相关命令:

```
chmod
```

(7) 参数:



umask 主要参数如表 3-66 所示。

表 3-66 umask 参数表

参 数	含 义
-S	以文字的方式来表示权限掩码

(8) 补充说明:

权限掩码是由 3 个八进制的数字所组成的, 将现有的存取权限减掉权限掩码后, 即可产生建立文件时预设的权限。

(9) 典型实例:

例: 查看当前权限掩码。

单独使用 umask 命令和带参数 “-p” 使用都能显示当前的权限掩码。例如单独使用 umask 命令, 可得到如下输出信息:

```
cmd@cmd-desktop:~$umask
0022
```

当前的权限掩码为 0022。然而新创建的目录, 权限一般应该是 755 (计算机内部按与运算求得权限), 但是对于新创建的普通文件, 权限并不是 755。因为 umask 掩码对于普通文件的执行权限不起作用, 要排除执行权限, 因此新建文件的权限应该为 644。

使用参数 “-p” 显示权限掩码, 具体命令为:

```
umask-p #显示当前权限掩码
```

输出如下信息:

```
umask 0022
```

例 2: 设置权限掩码。

使用八进制设置权限掩码。例如设置当前权限掩码为 044, 可使用如下命令:

```
umask 044 #设置权限掩码为 044
```

再次查看权限掩码, 得到如下信息:

```
cmd@cmd-desktop:~$umask 044 #设置权限掩码为 044
cmd@cmd-desktop:~$umask-p #再次查看权限掩码
umask 044 #权限掩码修改为 044
```

## 3.69 whereis ( 查找文件 )



(1) 频度等级: ☆☆☆

(2) 功能说明:

whereis 命令用于找到程序的源、二进制文件或手册。

(3) 语法:

```
whereis [-bfmsu][[-B <目录>...][[-M <目录>...][[-S <目录>...][文件...]]
```

(4) 所属包:

```
util-linux
```

(5) 使用技巧:

whereis 命令只能用于查找二进制文件、源代码文件和 man 手册页, 一般文件的定位需使用 locate 命令。

(6) 相关命令:

```
find、locate、slocate、which
```

(7) 参数:

whereis 主要参数如表 3-67 所示。

表 3-67 whereis 参数表

参 数	含 义	参 数	含 义
-b	只查找二进制文件	-M<目录>	只在设置的目录下查找说明文件
-B<目录>	只在设置的目录下查找二进制文件	-s	只查找原始代码文件
-f	不显示文件名前的路径名称	-S<目录>	只在设置的目录下查找原始代码文件
-m	只查找说明文件	-u	查找不包含指定类型的文件

(8) 补充说明:

**whereis** 指令会在特定目录中查找符合条件的文件。这些文件应属于原始代码、二进制文件，或是帮助文件。

(9) 典型实例:

例: 定位 **bash** 命令及其 **man** 手册的位置。

例如, 要查询 **bash** 命令的位置, 可直接使用 **whereis** 命令, 如下所示:

```
whereis bash #显示 bash 命令的程序路径和 man 手册页路径
```

输出信息如下:

```
cmd@cmd-desktop:~$ whereis bash #显示 bash 命令的程序路径和 man 手册页路径
bash: /bin/bash /etc/bash.bashrc /usr/share/man/man1/bash.1.gz
#从左至右分别为查询的程序名、bash 路径、bash 的 man 手册页路径
```

例 2: 单独定位二进制文件或帮助文件。

例如, 要单独查询 **bash** 命令的二进制文件或帮助文件, 可使用如下命令:

```
whereis -b bash #显示 bash 命令的二进制程序
whereis -m bash #显示 bash 命令的帮助文件
```

输出信息如下:

```
hdd@hdd-desktop:~$ whereis -b bash #显示 bash 命令的二进制程序
bash: /bin/bash /etc/bash.bashrc /usr/share/bash # bash 命令的二进制程序的地址
hdd@hdd-desktop:~$ whereis -m bash #显示 bash 命令的帮助文件
bash: /usr/share/man/man1/bash.1.gz #bash 命令的帮助文件地址
```

## 3.70 which ( 查找文件 )



(1) 频度等级: ☆☆

(2) 功能说明:

**which** 命令用于查找文件。相对于其他命令, **which** 命令是在更小范围内查找文件, 只在环境变量中查找指定的文件。

(3) 语法:

```
which [文件...]
```

(4) 所属包:

```
which
```

(5) 使用技巧:

该命令只显示命令本身的绝对路径, 要显示命令的源代码文件和 **man** 手册文件的路径等信息, 需要使用 **whereis** 命令。

**which** 命令是基于环境变量 **PATH** 查找目录的, 如果 **PATH** 设置有问题, 则可能出现目录找不到的提示。

(6) 相关命令:

```
whereis
```

(7) 参数:

**which** 主要参数如表 3-68 所示。

表 3-68 which 参数表

参 数	含 义
-a 或--all	显示查找到的所有文件的路径信息，不是仅显示第一个
-n<文件名长度>	指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名
-p<文件名长度>	与-n 参数相同，但此处的<文件名长度>包括了文件的路径
--read-functions	从标准输入设备读取 shell 函数定义，将查找到的函数显示到标准输出设备
--skip-tilde	跳过环境变量 PATH 中以波浪线开头的目录
--skip-dot	跳过环境变量 PATH 中以点开头的目录
-w	指定输出时字段的宽度
--help	显示帮助信息
-V 或--version	显示版本信息

(8) 补充说明:

which 指令会在环境变量 \$PATH 设置的目录中查找符合条件的文件。

which 命令一般用于安装新的软件时，检查路径设置是否成功。若成功将会显示该执行文件的全部路径，否则将显示 \$PATH 的内容。

which 对文件名是全部匹配的，而 locate 是部分匹配的。比如，“which test” 查找的是环境变量中的文件 test，而“locate test” 是查找所有包含字符串“test”的文件。

(9) 典型实例:

例：显示命令的绝对路径。

例如，要查询 bash 的绝对路径，可使用如下命令：

```
which bash #查询 bash 的绝对路径
```

此时有如下输出信息：

```
cmd@cmd-desktop:~$ which bash #显示 bash 命令的绝对路径
/bin/bash # bash 可执行程序的绝对路径
```

## 3.71 unzip ( 解压缩 zip 文件 )



(1) 频度等级：☆☆☆

(2) 功能说明:

unzip 命令用于解压缩 zip 文件，该命令可以在 MS Windows 下使用压缩软件 winzip 压缩的文件在 Linux 中解压缩。

(3) 语法:

```
unzip[-cflptuvz][-agCjLMnoqsVX][-P <密码>][.zip 文件][文件][-d <目录>][-x <文件>]
unzip [-Z]
```

(4) 所属包:

```
unzip
```

(5) 使用技巧:

使用该命令的“-d”参数可以指定文件解压后的存放目录。

(6) 相关命令:

```
zip、zipinfo
```

(7) 参数:

unzip 主要参数如表 3-69 所示。

表 3-69 unzip 参数表

参 数	含 义
-c	将解压缩的结果显示到屏幕上，并对字符做适当的转换
-f	更新现有的文件
-l	显示压缩文件内所包含的文件
-p	与-c 参数类似，会将解压缩的结果显示到屏幕上，但不会执行任何的转换

-t	检查压缩文件是否正确
-u	与-f 参数类似，但是除了更新现有的文件外，也会将压缩文件中的其他文件解压缩到目录中
-v	执行时显示详细的信息
-z	仅显示压缩文件的备注文字
-a	对文本文件进行必要的字符转换
-b	不要对文本文件进行字符转换
-C	压缩文件中的文件名称区分大小写
-j	不处理压缩文件中原有的目录路径
-L	将压缩文件中的全部文件名改为小写
-M	将输出结果送到 more 程序处理
-n	解压缩时不要覆盖原有的文件
-o	不必先询问用户，unzip 执行后覆盖原有文件
-P<密码>	使用 zip 的密码选项
-q	执行时不显示任何信息
-s	将文件名中的空白字符转换为底线字符
-V	保留 VMS 的文件版本信息
-X	解压缩时同时回存文件原来的 UID/GID
[.zip 文件]	指定.zip 压缩文件
[文件]	指定要处理.zip 压缩文件中的哪些文件
-d<目录>	指定文件解压缩后要存储的目录
-x<文件>	指定不要处理.zip 压缩文件中的哪些文件
-Z	等于执行 zipinfo 指令

(8) 补充说明:

unzip 为 “.zip” 压缩文件的解压缩程序。

(9) 典型实例:

例 1: 显示压缩文件的文件列表。

例如，使用 unzip 命令的“-l”参数，查看压缩包中的详细列表，具体命令如下:

```
unzip -l test.zip #查看压缩包中的详细列表
```

返回结果如下:

```
cmd@cmd-desktop:~$ unzip -l test.zip #查看压缩包中的详细列表
Archive: test.zip #文件名
  Length      Date    Time    Name      #详细列表
-----
   61  2009-09-21 16:09   file
   55  2009-09-21 19:53  testfile
   54  2009-09-21 16:10  testfile1
   54  2009-09-21 16:10  testfile2
   54  2009-09-21 16:10  testfile3
-----
  278                                #统计信息
                    5 files
```

例 2: 解压缩 zip 文件。例如使用参数“-v”解压缩当前目录下的 test.zip 文件，可使用如下命令:

```
Unzip-v test.zip #解压缩 test.zip 文件
```

返回结果如下:

```
cmd@cmd-desktop:~$ unzip -v test.zip #解压缩文件
Archive: test.zip #文件名
  Length Method      Size  Cmpr   Date    Time    CRC-32   Name #解压缩详细过程
-----
   61  Defl:N        59    3%   2009-09-21 16:09 d3a9c5f5  file
   55  Defl:N        54    2%   2009-09-21 19:53 2d421bea  testfile
   54  Defl:N        53    2%   2009-09-21 16:10 4424f91a  testfile1
   54  Defl:N        53    2%   2009-09-21 16:10 4424f91a  testfile2
   54  Defl:N        53    2%   2009-09-21 16:10 4424f91a  testfile3
-----
  278                                #统
计信息                    272  2%                                5 files
```

# 3.72 uucico ( UUCP 文件传输服务程序 )



(1) 频度等级: ☆

(2) 功能说明:

uucico 命令的作用是启动 UUCP 文件传输服务, 完成数据的发送和接收。

(3) 语法:

```
uucico[-cCDefqvwz][-i<类型>][-I<文件>][-p<链接端口号码>][-r1][-s<主机>][-S<主机>][-u<用户>][-x<类型>][--help]
```

(4) 所属包:

uucp

(5) 使用技巧:

uucico 有两种工作模式: 主动模式和附属模式。当在主动模式下时, uucico 会调用远端主机; 在附属模式下时, uucico 则接受远端主机的调用。

(6) 相关命令:

uucp

(7) 参数:

uucico 主要参数如表 3-70 所示。

参 数	含 义
-c 或 --quiet	当不执行任何工作时, 不要更改记录文件的内容及更新目前的状态
-C 或 --ifwork	当有工作要执行时, 调用-s 或-S 参数所指定的主机
-D 或 --nodetach	不要与控制终端机离线
-e 或 --loop	在附属模式下执行, 并且出现要求登录的提示画面
-f 或 --force	当执行错误时, 不等待任何时间即重新调用主机
-i<类型>或--stdin<类型>	当使用到标准输入设备时, 指定连接端口的类型
-I<文件>--config<文件>	指定使用的配置文件
-l 或 --prompt	出现要求登录的提示画面
-p<链接端口号码>或--port<链接端口号码>	指定连接端口号码
-q 或 --quiet	不要启动 uuxqt 服务程序
-r0 或 --slave	以附属模式启动
-r1 或 --master	以主动模式启动
-s<主机>或--system<主机>	调用指定的主机
-u<用户>或--login<用户>	指定登录的用户账号, 而不允许输入任意的登录账号
-w 或 --wait	在主动模式下, 当执行调用动作时, 则出现要求登录的提示画面
-x<类型>或-X<类型>或 outgoing-debug<类型>	启动指定的排错模式
-z 或 --try-next	当执行不成功时, 尝试下一个选择而不结束程序

(8) 补充说明:

uucico 是用来处理 uucp 或 uux 命令传送到队列的文件传输工具。

(9) 典型实例:

例: 使用主动模式启动 uucico 服务。在命令提示符下直接输入如下命令:

```
uucico-r1 #主动模式启动 uucico 服务
```

## 提 示

该命令一般没有输出。

# 3.73 uucp ( 在 UNIX 系统之间传送文件 )





(1) 频度等级: ☆

(2) 功能说明:

UUCP 代表“Unix-to-Unix Copy”，它是 UNIX 系统的一项功能，允许计算机之间以存储-转发方式交换 E-mail 和消息。在 Internet 兴起之前它是 UNIX 系统之间连网的主要方式。uucp 命令用于在 UNIX 系统之间传送文件。

(3) 语法:

```
uucp [-cCdfjmrRtvW] [-g<等级>] [-I<配置文件>] [-n<用户>] [-x<类型>] [--help][...来源][目的]
```

(4) 所属包:

```
uucp
```

(5) 使用技巧:

uucp 使用 UUCP 协议，主要的功能为传送文件。

(6) 相关命令:

```
uucico
```

(7) 参数:

uucp 主要参数如表 3-71 所示。

表 3-71 uucp 参数表

参 数	含 义
-c 或--nocopy	不用将文件复制到缓冲区
-C 或--copy	将文件复制到缓冲区
-d 或--directories	在传送文件时，自动在[目的]中建立必要的目录
-f 或--nodirectories	在传送文件时，若需要在[目的]中建立目录，则放弃执行该作业
-g<等级>或--grade<等级>	指定文件传送作业的优先顺序
-I<配置文件>或--config<配置文件>	指定 uucp 配置文件
-j 或--jobid	显示作业编号
-m 或--mail	作业结束后，以电子邮件报告作业是否顺利完成
-n<用户>或--notify<用户>	作业结束后，以电子邮件向指定的用户报告作业是否顺利完成
-r 或--nouucico	不要立即启动 uucico 服务程序，仅将作业传送到队列中，待稍后再执行
-R 或--recursive	若[来源]为目录，则将整个目录及子目录复制到[目的]
-t 或--uuto	将最后一个参数视为“主机名!用户”
-W 或--noexpand	不要将目前所在的目录加入路径
-x<类型>或--debug<类型>	启动指定的排错模式
[...来源]	指定源文件或路径
[目的]	指定目标文件或路径

(8) 补充说明:

UUCP 是在 UNIX 系统之间，通过序列线来连线的协议。

(9) 典型实例:

例：将 temp/目录下所有文件传送到远程主机 localhost 的 uucp 公共目录下的 Public/目录下。在命令行中输入如下命令:

```
uucp-d-R temp localhost ~/Public/ #将 temp/目录下所有文件传送到远程主机
```

该命令通常没有输出。

## 3.74 uupick ( 处理传送进来的文件 )



(1) 频度等级: ☆

(2) 功能说明:

uupick 命令用于传输并处理 uuto 命令所发出的文件。

(3) 语法:

```
uupick [-v][-I<配置文件>][-s<主机>][-x<层级>][--help]
```

(4) 所属包:

```
uucp
```

(5) 相关命令:

```
uucico, uucp
```

(6) 参数:

uupick 主要参数如表 3-72 所示。

表 3-72 uupick 参数表

参 数	含 义
-I<配置文件>或--config<配置文件>	指定配置文件
-s<主机>或--system<主机>	处理由指定主机传送过来的文件

(7) 补充说明:

当其他主机通过 uucp 命令将文件传送进来时, 可利用 uupick 指令取出这些文件。

(8) 典型实例:

例: 处理由主机 localhost 传送过来的文件。在命令行直接输入如下命令:

```
uupick-s localhost #处理由主机 localhost 传送过来的文件
```

该命令通常没有输出。

## 3.75 uuto ( 传送文件至 UUCP 主机 )



(1) 频度等级: ☆

(2) 功能说明:

uuto 命令用于将文件传送到远端的 UUCP 主机。

(3) 语法:

```
uuto [文件][目的]
```

(4) 所属包:

```
uucp
```

(5) 使用技巧:

uuto 为 script 文件, 它实际上会执行 uucp, 用来将文件传送到远端 UUCP 主机, 并在完成工作后, 以邮件通知远端主机上的用户。

(6) 相关命令:

```
uucp, uucico, uupick
```

(7) 参数:

uuto 相关参数请参考 uucp 指令。

(8) 典型实例:

例: 将文件传送到远程 UUCP 主机 localhost 的 tmp 目录, 在命令提示符中直接输入如下命令:

```
uuto ./testfile localhost/tmp #将文件传送到远程 UUCP 主机 localhost 的 tmp 目录
```

该命令通常没有输出。

## 3.76 zip ( 压缩文件 )



(1) 频度等级: ☆☆

(2) 功能说明:

zip 命令用于压缩文件。经该命令压缩后的文件的扩展名为“.zip”，目前所有主流操作系统都支持这种格式的压缩文件。

(3) 语法:

```
zip [-AcDdFfGghjJKLlmoqrSTuvVwXyz$][-b <工作目录>][-ll][-n <字尾字符串>][-t <日期时间>][-<压缩效率>][压缩文件][文件...][-i <范本样式>][-x <范本样式>]
```

(4) 所属包:

zip

(5) 使用技巧:

为了和 MS-DOS 兼容，可以加上“-k”参数。

(6) 相关命令:

unzip、zipinfo

(7) 参数:

zip 主要参数如表 3-73 所示。

表 3-73 zip 参数表

参 数	含 义
-a	将系统使用的 EBCDIC 文件转换为 ASCII 文件
-A	调整可执行的自动解压文件
-b<工作目录>	指定暂时存放文件的目录
-c	为每个被压缩的文件加上注释
-d	从压缩文件内删除指定的文件
-D	在压缩文件内不建立目录名称
-f	此参数的效果和指定“-u”参数类似，但不仅更新已有文件，如果某些文件原本不存在于压缩文件内，使用本参数会一并将其加入压缩文件中
-F	尝试修复已损坏的压缩文件
-g	将文件压缩后附加在已有的压缩文件之后，而非另行建立新的压缩文件
-i<范本样式>	只压缩符合条件的文件
-J	删除压缩文件前面不必要的数据
-k	使用 MS-DOS 兼容格式的文件名称
-l	压缩文件时，把 LF 字符置换成 LF+CR 字符
-ll	压缩文件时，把 LF+CR 字符置换成 LF 字符
-L	显示版权信息
-m	将文件压缩并加入压缩文件后，删除原始文件，即把文件移到压缩文件中
-n<字尾字符串>	不压缩具有特定字尾字符串的文件
-o	以压缩文件内拥有最新更改时间的文件为准，将压缩文件的更改时间设成和该文件相同
-q	不显示指令执行过程
-r	递归处理，将指定目录下的所有文件和子目录一并处理
-S	包含系统和隐藏文件
-t<日期时间>	把压缩文件的日期设成指定的日期
-T	检查备份文件内的每个文件是否正确无误
-u	更换较新的文件到压缩文件内
-v	显示指令执行过程或显示版本信息
-V	保存 VMS 操作系统的文件属性
-w	在文件名称里加入版本编号，本参数仅在 VMS 操作系统下有效
-x<范本样式>	压缩时排除符合条件的文件
-X	不保存额外的文件属性
-y	直接保存符号链接，而非该链接所指向的文件，本参数仅在 UNIX 之类的系统下有效
-z	为压缩文件加上注释
-\$	保存第一个被压缩文件所在磁盘的卷册名称
-<压缩效率>	压缩效率是一个介于 1-9 的数值

(8) 补充说明:

zip 是一个使用广泛的压缩程序, 文件经它压缩后会另外产生具有 “.zip” 扩展名的压缩文件。

(9) 典型实例:

例: 压缩多个文件。

例如要将当前目录下 file、testfile、testfile1、testfile2、testfile3 压缩到一个归档文件, 并命名为 test.zip, 可使用如下命令:

```
zip test file testfile testfile1 testfile2 testfile3
#将当前目录下 file、testfile、testfile1、testfile2、testfile3 压缩
```

输出信息如下:

```
cmd@cmd-desktop:~$ zip test file testfile testfile1 testfile2 testfile3 #压缩五个文件
adding: file (deflated 3%) #显示第一个文件进度、压缩比
adding: testfile (deflated 2%) #显示第二个文件进度、压缩比
adding: testfile1 (deflated 2%) #显示第三个文件进度、压缩比
adding: testfile2 (deflated 2%) #显示第四个文件进度、压缩比
adding: testfile3 (deflated 2%) #显示第五个文件进度、压缩比
```

## 3.77 zipinfo (列出 zip 压缩文件的详细信息)

(1) 频度等级: ☆

(2) 功能说明:

zipinfo 命令用于列出压缩文件的详细信息。在使用 unzip 解压文件时, 只解压归档文件中的某个特殊文件, 而不是解压缩全部文件。unzip 命令可以在命令行指明用户需要解压缩的文件, 当用户要求了解文件的内容时, 可利用 zipinfo 获得该信息。

(3) 语法:

```
zipinfo [-l2hlmMstTvz][压缩文件][文件...][-x <范本样式>]
```

(4) 所属包:

```
unzip
```

(5) 使用技巧:

使用该命令可以显示压缩包中的文件数、压缩字节与压缩比, 可以显示被压缩文件在压缩前的详细信息。

(6) 相关命令:

```
zip
```

(7) 参数:

zipinfo 主要参数如表 3-74 所示。

表 3-74 zipinfo 参数表

参 数	含 义
-l	只列出文件名称
-2	此参数的效果和指定“-l”参数类似, 但可搭配“-h”, “-t”和“-z”参数使用
-h	只列出压缩文件的文件名称
-l	此参数的效果和指定“-m”参数类似, 但会列出原始文件的大小而非每个文件的压缩率
-m	此参数的效果和指定“-s”参数类似, 但大多会列出每个文件的压缩率
-M	若信息内容超过一个画面, 则采用类似 more 指令的方式列出信息
-s	用类似执行“ls-l”指令的效果列出压缩文件内容
-t	只列出压缩文件内所包含的文件数目, 压缩前后的文件大小及压缩率
-T	将压缩文件内每个文件的日期时间用年、月、日、时、分、秒的顺序列出
-v	详细显示压缩文件内每一个文件的信息

-x<范本样式>	不列出符合条件的文件的信息
-z	如果压缩文件内含有注释，就将注释显示出来

(8) 补充说明:

执行 zipinfo 指令可得知 Zip 压缩文件的详细信息。

虽然 zipinfo 和 “unzip-Z” 的功能一样，但是前者在现实压缩文件信息时，更具有多样性。

和 unzip 命令一样，要处理的指定文件必须在压缩归档文件中存在。另外，该命令不能同时显示两个压缩文件的信息。

(9) 典型实例:

例：查看压缩文件的详细信息。

例如，用 zipinfo 命令查看 zip 命令生成的 test.zip 压缩文件，可使用如下命令：

```
zipinfo test.zip #查看 test.zip 文件的信息
```

输出信息如下：

```
cmd@cmd-desktop:~$ zipinfo test.zip #查看 test.zip 文件的信息
Archive: test.zip #文件名
Zip file size: 1012 bytes, number of entries: 5 #文件大小
-rw-r--r-- 3.0 unx 61 tx defN 09-Sep-21 16:09 file #单个文件属性
-rw-r--r-- 3.0 unx 55 tx defN 09-Sep-21 19:53 testfile
-rw-r--r-- 3.0 unx 54 tx defN 09-Sep-21 16:10 testfile1
-rw-r--r-- 3.0 unx 54 tx defN 09-Sep-21 16:10 testfile2
-rw-r--r-- 3.0 unx 54 tx defN 09-Sep-21 16:10 testfile3
5 files, 278 bytes uncompressed, 272 bytes compressed: 2.2% #文件统计信息
```

## 联系方式

集团官网: [www.hqyj.com](http://www.hqyj.com)

嵌入式学院: [www.embedu.org](http://www.embedu.org)

移动互联网学院: [www.3g-edu.org](http://www.3g-edu.org)

企业学院: [www.farsight.com.cn](http://www.farsight.com.cn)

物联网学院: [www.topsight.cn](http://www.topsight.cn)

研发中心: [dev.hqyj.com](http://dev.hqyj.com)

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-25590506

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

广州地址：广州市天河区中山大道 268 号天河广场 3 层，电话：020-28916067