

1. Linux系统中常用命令解析	2
1.1 cat-连接内容（输出文件内容）	3
1.2 cp-复制文件或目录	5
1.3 curl-下载工具	6
1.4 echo-字符串输出	12
1.5 export-设置或显示环境变量	14
1.6 file-识别文件类型	16
1.7 find-查找文件并执行指定操作	18
1.8 grep-查找文件里符合条件的字符串	20
1.9 gzip-压缩文件	22
1.10 ifconfig-显示或设置网络设备	24
1.11 ifdown-禁用指定的网络接口	26
1.12 locate-查找文档或目录	27
1.13 ls-显示目录结构	29
1.14 mkdir-创建目录	31
1.15 mv-移动文件或文件夹	32
1.16 netstat-打印网络系统的状态信息	33
1.17 ping-检测主机网络	37
1.18 pwd-显示工作目录	39
1.19 rmdir-删除空目录	40
1.20 rm-删除文件或目录	41
1.21 route-路由设置	43
1.22 scp命令-复制文件和目录	45
1.23 shutdown-关闭/重启计算机	48
1.24 touch-创建文件或修改文件时间	50
1.25 traceroute-显示数据包到主机间的路径	52
1.26 useradd-建立用户账号	54
1.27 which-显示命令的绝对路径	56
1.28 kill-删除执行中的程序或工作	57
1.29 cmp-比较两个文件是否存在差异	59
1.30 read-从标准输入读取数值	61

Linux系统中常用命令解析

此页面下为Linux系统中常用命令的学习资料。



每个命令添加为一个子页面，并在子页面中给出命令的具体说明。

- cat-连接内容（输出文件内容）
- cp-复制文件或目录
- curl-下载工具
- echo-字符串输出
- export-设置或显示环境变量
- file-识别文件类型
- find-查找文件并执行指定操作
- grep-查找文件里符合条件的字符串
- gzip-压缩文件
- ifconfig-显示或设置网络设备
- ifdown-禁用指定的网络接口
- locate-查找文档或目录
- ls-显示目录结构
- mkdir-创建目录
- mv-移动文件或文件夹
- netstat-打印网络系统的状态信息
- ping-检测主机网络
- pwd-显示工作目录
- rmdir-删除空目录
- rm-删除文件或目录
- route-路由设置
- scp命令-复制文件和目录
- shutdown-关闭/重启计算机
- touch-创建文件或修改文件时间
- traceroute-显示数据包到主机间的路径
- useradd-建立用户账号
- which-显示命令的绝对路径
- kill-删除执行中的程序或工作
- cmp-比较两个文件是否存在差异
- read-从标准输入读取数值

cat-连接内容（输出文件内容）

- 1. 功能说明
- 2. 使用权限
- 3. 命令格式与参数
 - 参数说明
- 4. 实例
 - 4.1 在屏幕上打印文件内容
 - 4.2 输出文件内容
 - 4.3 附加文件内容
 - 4.4 清空文件内容
 - 4.5 制作镜像文件

1. 功能说明

cat命令的英文全拼是concatenate，用于连接文件并打印到标准输出设备上。

concatenate

adj. 连在一起的；连锁的
v. 使连接（连续，衔接）起来；连锁；串级

2. 使用权限

所有使用者

3. 命令格式与参数

```
cat [-AbeEnstTuv] [--help] [--version] fileName
```

参数说明

参数简写	参数全写	解释
-n	--number	由 1 开始对所有输出的行数编号
-b	--number-nonblank	和 -n 相似，只不过对于空白行不编号
-s	--squeeze-blank	当遇到有连续两行以上的空白行，就代换为一行的空白行
-v	--show-nonprinting	使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外
-E	--show-ends	在每行结束处显示 \$
-T	--show-tabs	将 TAB 字符显示为 ^I
-A	--show-all	等价于 -vET
-c		等价于 "-vE" 选项
-t		等价于 "-vI" 选项

4. 实例

4.1 在屏幕上打印文件内容

e.g. 在屏幕上打印textfile的全部文档内容

```
$ cat textfile
```

4.2 输出文件内容

e.g. 把 textfile1 的文档内容加上行号后输入 textfile2 文档里

```
$ cat -n textfile1 > textfile2
```

4.3 附加文件内容

e.g. 把 textfile1 和 textfile2 的文档内容加上行号（空白行不加）之后将内容附加到 textfile3 文档里

```
$ cat -b textfile1 textfile2 >> textfile3
```

4.4 清空文件内容

e.g. 清空 /etc/test.txt 文档内容

```
$ cat /dev/null > /etc/test.txt
```

4.5 制作镜像文件

e.g. 制作软盘的镜像文件，将软盘放好后输入

```
$ cat /dev/fd0 > OUTFILE
```

e.g. 相反的，如果想把 image file 写到软盘，输入

```
$ cat IMG_FILE > /dev/fd0
```

注释:

- OUTFILE 指输出的镜像文件名；
- IMG_FILE 指镜像文件；
- 若从镜像文件写回 device 时，device 容量需与相当；
- 通常用制作开机磁片。

cp-复制文件或目录

- 1.功能说明
- 2.命令格式和参数
- 3.常见用法实例
 - 3.1 将文件复制到新目录

1.功能说明

Linux **cp**（英文全拼：copy file）命令主要用于复制文件或目录。

2.命令格式和参数

```
cp [options] source dest
#
cp [options] source... directory
```

参数说明

参数	说明
-a	此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于dpR参数组合。
-d	复制时保留链接。这里所说的链接相当于Windows系统中的快捷方式。
-f	覆盖已经存在的目标文件而不给出提示。
-i	与-f选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖，回答"y"时目标文件将被覆盖。
-l	不复制文件，只是生成链接文件。
-p	除复制文件的内容外，还把修改时间和访问权限也复制到新文件中。
-r 或 -R	若给出的源文件是一个目录文件，此时将复制该目录下所有的子目录和文件。
-s	不进行真正的复制操作，仅为源文件创建符号连接。（与 ln -s 指令的功能相同）
-u	当目标文件不存在或者源文件比目标文件新时才执行复制操作，否则不进行复制
-S	在备份文件时，用指定的后缀“SUFFIX”代替文件名的默认后缀
-b	覆盖已存在的目标文件前将目标文件备份
-v	详细显示指令执行的操作

3.常见用法实例

3.1 将文件复制到新目录

e.g. 使用指令 cp 将当前目录 **test/** 下的所有文件复制到新目录 **newtest** 下，输入如下命令：

```
$ cp -r test/ newtest
```

curl-下载工具

- 1、功能说明
- 2、命令格式与参数
 - 选项
- 3、常见用法实例
 - 3.1 文件下载
 - 3.2 断点续传
 - 3.3 使用curl设置参照页字符串 断点续传
 - 3.4 用curl设置cookies
 - 3.5 用curl设置用户代理字符串
 - 3.6 curl的带宽控制和下载配额
 - 3.7 用curl进行认证
 - 3.8 只打印响应头部信息

1、功能说明

curl命令是一个利用URL规则在命令行下工作的文件传输工具。它支持文件的上传和下载，所以是综合传输工具，但按传统，习惯称curl为下载工具。作为一款强力工具，curl支持包括HTTP、HTTPS、[ftp](#)等众多协议，还支持POST、cookies、认证、从指定偏移处下载部分文件、用户代理字符串、限速、文件大小、进度条等特征。做网页处理流程和数据检索自动化，curl可以祝一臂之力。

2、命令格式与参数

shell
curl () ()

选项

-a/--append	上传文件时，附加到目标文件
-A/--user-agent <string>	设置用户代理发送给服务器
-anyauth	可以使用“任何”身份验证方法
-b/--cookie <name=string/ file>	cookie字符串或文件读取位置
--basic	使用HTTP基本验证
-B/--use-ascii	使用ASCII / 文本传输
-c/--cookie-jar <file>	操作结束后把cookie写入到这个文件中
-C/--continue-at <offset>	断点续转
-d/--data <data>	HTTP POST方式传送数据
--data-ascii <data>	以ascii的方式post数据
--data-binary <data>	以二进制的方式post数据
--negotiate	使用HTTP身份验证
--digest	使用数字身份验证
--disable-eprt	禁止使用EPRT或LPRT
--disable-epsv	禁止使用EPSV
-D/--dump-header <file>	把header信息写入到该文件中
--egd-file <file>	为随机数据(SSL)设置EGD socket路径

--tcp-nodelay	使用TCP_NODELAY选项
-e/--referer	来源网址
-E/--cert <cert[:passwd]>	客户端证书文件和密码 (SSL)
--cert-type <type>	证书文件类型 (DER/PEM/ENG) (SSL)
--key <key>	私钥文件名 (SSL)
--key-type <type>	私钥文件类型 (DER/PEM/ENG) (SSL)
--pass <pass>	私钥密码 (SSL)
--engine <eng>	加密引擎使用 (SSL). "--engine list" for list
--cacert <file>	CA证书 (SSL)
--capath <directory>	CA目录 (made using c_rehash) to verify peer against (SSL)
--ciphers <list>	SSL密码
--compressed	要求返回是压缩的形势 (using deflate or gzip)
--connect-timeout <seconds>	设置最大请求时间
--create-dirs	建立本地目录的目录层次结构
--crlf	上传是把LF转变成CRLF
-f/--fail	连接失败时不显示http错误
--ftp-create-dirs	如果远程目录不存在, 创建远程目录
--ftp-method [multicwd/nocwd/singlecwd]	控制CWD的使用
--ftp-pasv	使用 PASV/EPSV 代替端口
--ftp-skip-pasv-ip	使用PASV的时候,忽略该IP地址
--ftp-ssl	尝试用 SSL/TLS 来进行ftp数据传输
--ftp-ssl-reqd	要求用 SSL/TLS 来进行ftp数据传输
-F/--form <name=content>	模拟http表单提交数据
--form-string <name=string>	模拟http表单提交数据
-g/--globoff	禁用网址序列和范围使用{}和[]
-G/--get	以get的方式来发送数据
-H/--header <line>	自定义头信息传递给服务器
--ignore-content-length	忽略的HTTP头信息的长度
-i/--include	输出时包括protocol头信息
-I/--head	只显示请求头信息
-j/--junk-session-cookies	读取文件进忽略session cookie
--interface <interface>	使用指定网络接口/地址
--krb4 <level>	使用指定安全级别的krb4
-k/--insecure	允许不使用证书到SSL站点
-K/--config	指定的配置文件读取
-l/--list-only	列出ftp目录下的文件名称
--limit-rate <rate>	设置传输速度
--local-port<NUM>	强制使用本地端口号

-m/--max-time <seconds>	设置最大传输时间
--max-redirs <num>	设置最大读取的目录数
--max-filesize <bytes>	设置最大下载的文件总量
-M/--manual	显示全手动
-n/--netrc	从netrc文件中读取用户名和密码
--netrc-optional	使用 .netrc 或者 URL来覆盖-n
--ntlm	使用 HTTP NTLM 身份验证
-N/--no-buffer	禁用缓冲输出
-o/--output	把输出写到该文件中
-O/--remote-name	把输出写到该文件中，保留远程文件的文件名
-p/--proxytunnel	使用HTTP代理
--proxy-anyauth	选择任一代理身份验证方法
--proxy-basic	在代理上使用基本身份验证
--proxy-digest	在代理上使用数字身份验证
--proxy-ntlm	在代理上使用ntlm身份验证
-P/--ftp-port <address>	使用端口地址，而不是使用PASV
-q	作为第一个参数，关闭 .curlrc
-Q/--quote <cmd>	文件传输前，发送命令到服务器
-r/--range <range>	检索来自HTTP/1.1或FTP服务器字节范围
--range-file	读取（SSL）的随机文件
-R/--remote-time	在本地生成文件时，保留远程文件时间
--retry <num>	传输出现问题时，重试的次数
--retry-delay <seconds>	传输出现问题时，设置重试间隔时间
--retry-max-time <seconds>	传输出现问题时，设置最大重试时间
-s/--silent	静默模式。不输出任何东西
-S/--show-error	显示错误
--socks4 <host[:port]>	用socks4代理给定主机和端口
--socks5 <host[:port]>	用socks5代理给定主机和端口
--stderr <file>	
-t/--telnet-option <OPT=val>	Telnet选项设置
--trace <file>	对指定文件进行debug
--trace-ascii <file>	Like --跟踪但没有hex输出
--trace-time	跟踪/详细输出时，添加时间戳
-T/--upload-file <file>	上传文件
--url <URL>	Spet URL to work with
-u/--user <user[:password]>	设置服务器的用户和密码
-U/--proxy-user <user[:password]>	设置代理用户名和密码
-w/--write-out [format]	什么输出完成后

-x/--proxy <host[:port]>	在给定的端口上使用HTTP代理
-X/--request <command>	指定什么命令
-y/--speed-time	放弃限速所要的时间，默认为30
-Y/--speed-limit	停止传输速度的限制，速度时间

3、常见用法实例

3.1 文件下载

curl命令可以用来执行下载、发送各种HTTP请求，指定HTTP头部等操作。如果系统没有**curl**可以使用**yum install curl**安装，也可以下载安装。**curl**是将下载文件输出到stdout，将进度信息输出到stderr，不显示进度信息使用**--silent**选项。

```
shell
curl URL --silent
```

这条命令是将下载文件输出到终端，所有下载的数据都被写入到stdout。

使用选项**-O**将下载的数据写入到文件，必须使用文件的绝对地址

```
shell
curl http://man.linuxde.net/text.iso --silent -O
```

选项**-o**将下载数据写入到指定名称的文件中，并使用**--progress**显示进度条

```
shell
curl http://man.linuxde.net/test.iso -o filename.iso --progress
##### 100.0%
```

3.2 断点续传

curl能够从特定的文件偏移处继续下载，它可以通过指定一个偏移量来下载部分文件

```
shell
curl URL/File -C
```

3.3 使用curl设置参照页字符串 断点续传

参照页是位于HTTP头部中的一个字符串，用来表示用户是从哪个页面到达当前页面的，如果用户点击网页A中的某个连接，那么用户就会跳转到B网页，网页B头部的参照页字符串就包含网页A的URL。

使用**--referer**选项指定参照页字符串

shell

```
curl --referer http://www.google.com http://man.linuxde.net
```

3.4 用curl设置cookies

使用--cookie "COKKIES" 选项来指定cookie，多个cookie使用分号分隔

shell

```
curl http://man.linuxde.net --cookie "user=root;pass=123456"
```

将cookie另存为一个文件，使用--cookie-jar选项

shell

```
curl URL --cookie-jar cookie_file
```

3.5 用curl设置用户代理字符串

有些网站访问会提示只能使用IE浏览器来访问，这是因为这些网站设置了检查用户代理，可以使用curl把用户代理设置为IE，这样就可以访问了。使用--user-agent或者-A选项

shell

```
curl URL --user-agent "Mozilla/5.0"  
curl URL -A "Mozilla/5.0"
```

其他HTTP头部信息也可以使用curl来发送，使用-H"头部信息" 传递多个头部信息

shell

```
curl -H "Host:man.linuxde.net" -H "accept-language:zh-cn" URL
```

3.6 curl的带宽控制和下载配额

使用--limit-rate限制curl的下载速度

shell

```
curl URL --limit-rate 50k
```

命令中用k（千字节）和m（兆字节）指定下载速度限制。

使用--max-filesize指定可下载的最大文件大小

shell

```
curl URL --max-filesize bytes
```

如果文件大小超出限制，命令则返回一个非0退出码，如果命令正常则返回0。

3.7 用curl进行认证

使用curl选项 -u 可以完成HTTP或者FTP的认证，可以指定密码，也可以不指定密码在后续操作中输入密码

shell

```
curl -u user:pwd http://man.linuxde.net  
curl -u user http://man.linuxde.net
```

3.8 只打印响应头部信息

通过-I或者-head可以只打印出HTTP头部信息

shell

```
[root@localhost text]# curl -I http://man.linuxde.net  
HTTP/1.1 200 OK  
Server: nginx/1.2.5  
date: Mon, 10 Dec 2012 09:24:34 GMT  
Content-Type: text/html; charset=UTF-8  
Connection: keep-alive  
Vary: Accept-Encoding  
X-Pingback: http://man.linuxde.net/xmlrpc.php
```

echo-字符串输出

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 显示普通字符串
 - 3.2 显示转义字符
 - 3.3 显示换行
 - 3.4 显示不换行
 - 3.5 显示结果定向至文件
 - 3.6 原样输出字符串，不进行转义或取变量（用单引号）
 - 3.7 显示命令执行结果

1、功能说明

Shell 的 echo 指令与 PHP 的 echo 指令类似，都是用于字符串的输出。

2、命令格式与参数

Shell

```
# echo string
```

3、常见用法实例

3.1 显示普通字符串

Shell

```
# echo "It is a test"
```

这里的双引号完全可以省略，以下命令与上面实例效果一致：

Shell

```
# echo It is a test
```

3.2 显示转义字符

Shell

```
# echo "\"It is a test\""  
"It is a test"
```

同样的，双引号也可以省略。

3.3 显示换行

Shell

```
# echo -e "OK! \n"
OK
```

3.4 显示不换行

Shell

```
# echo -e "OK! \c" # -e \c
echo "It is a test"
```

输出的结果为：

Shell

```
OK! It is a test
```

3.5 显示结果定向至文件

Shell

```
# echo "It is a test" > myfile
```

3.6 原样输出字符串，不进行转义或取变量（用单引号）

Shell

```
# echo '$name\''
$name\"
```

3.7 显示命令执行结果

Shell

```
# echo `date`
2020 10 22 14:50:51 +08
```

注意： 这里使用的是反引号`，而不是单引号'。

结果将显示当前日期

export-设置或显示环境变量

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 列出所有的环境变量
 - 3.2 定义环境变量并查看
 - 3.2 导入函数并查看

1、功能说明

在 shell 中执行程序时，shell 会提供一组环境变量。**export** 可新增，修改或删除环境变量，供后续执行的程序使用。**export** 进行的操作仅在本次登录有效。

2、命令格式与参数

```
export [-fnp][[]=[ ]]
```

参数	说明
-f	表示“变量名称”为函数名称
-n	删除指定的变量
-p	列出所有的环境变量
-pf	列出导入的函数
-nf	删除指定的函数

3、常见用法实例

3.1 列出所有的环境变量

```
$ export -p
declare -x CLUTTER_IM_MODULE="ibus"
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GJS_DEBUG_OUTPUT="stderr"
declare -x GJS_DEBUG_TOPICS="JS ERROR;JS LOG"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/297041bd_3f9f_4212_95b9_973b64ac66d0"
```

3.2 定义环境变量并查看

```
$ export ABC=7 //ABC
$ export -p //
declare -x ABC="7"
declare -x CLUTTER_IM_MODULE="ibus"
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GJS_DEBUG_OUTPUT="stderr"
declare -x GJS_DEBUG_TOPICS="JS ERROR;JS LOG"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/297041bd_3f9f_4212_95b9_973b64ac66d0"
```

3.2 导入函数并查看

```
$ function func_1(){ echo "123"; } //
$ export -f func_1 //
$ export -pf //
func_1 ()
{
    echo "123"
}
declare -fx func_1
```

file-识别文件类型

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 显示文件类型
 - 3.2 显示符号链接的文件类型

1、功能说明

通过file指令，我们得以辨识该文件的类型。

2、命令格式与参数

Shell

```
# file [-bcLvz][-f <>][-m <>...][...]
```

参数	说明
-b	列出辨识结果时，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	指定名称文件，其内容有一个或多个文件名称时，让file依序辨识这些文件，格式为每列一个文件名称。
-L	直接显示符号连接所指向的文件的类别。
-v	显示版本信息。
-z	尝试去解读压缩文件的内容。

3、常见用法实例

3.1 显示文件类型

Shell

```
[root@localhost ~]# file install.log
install.log: UTF-8 Unicode text

[root@localhost ~]# file -b install.log      <==
UTF-8 Unicode text

[root@localhost ~]# file -i install.log      <== MIME
install.log: text/plain; charset=utf-8

[root@localhost ~]# file -b -i install.log
text/plain; charset=utf-8
```

3.2 显示符号链接的文件类型

Shell

```
[root@localhost ~]# ls -l /var/mail  
lrwxrwxrwx 1 root root 10 08-13 00:11 /var/mail -> spool/mail
```

```
[root@localhost ~]# file /var/mail  
/var/mail: symbolic link to `spool/mail'
```

```
[root@localhost ~]# file -L /var/mail  
/var/mail: directory
```

```
[root@localhost ~]# file /var/spool/mail  
/var/spool/mail: directory
```

```
[root@localhost ~]# file -L /var/spool/mail  
/var/spool/mail: directory
```

find-查找文件并执行指定操作

- 1. 功能说明
- 2. 使用权限
- 3. 命令格式与参数
 - 参数说明
 - 4. 实例
 - 4.1 列出目前目录及其子目录下所有延伸档名是 c 的文件
 - 4.2 列出目前目录其下子目录中所有一般文件
 - 4.3 将目前目录及其子目录下所有最近 20 天内更新过的文件列出
 - 4.4 查找前目录中文件属主具有读、写权限，并且文件所属组的用户和其他用户具有读权限的文件
 - 4.5 为了查找系统中所有文件长度为0的普通文件，并列出它们的完整路径

1. 功能说明

Linux find 命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。

如果使用该命令时，不设置任何参数，则 find 命令将在当前目录下查找子目录与文件。

并且将查找到的子目录和文件全部进行显示。

2. 使用权限

所有使用者

3. 命令格式与参数

1	<pre>find path - option [-pri nt][-exec-ok command]{} \;</pre>
---	---

参数说明

find 根据下列规则判断 path 和 expression，在命令列上第一个 - () , ! 之前的部份为 path，之后的是 expression。如果 path 是空字符串则使用目前路径，如果 expression 是空字符串则使用 -print 为预设 expression。

expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。

参数	说明
-mount, -xdev	只检查和指定目录在同一个文件系统下的文件，避免列出其它文件系统下的文件
-amin n	在过去 n 分钟内被读取过
-anewer file	比文件 file 更晚被读取过的文件
-atime n	在过去n天内被读取过的文件
-cmin n	在过去 n 分钟内被修改过
-cnewer file	比文件 file 更新的文件
-ctime n	在过去n天内被修改过的文件
-empty	空的文件-gid n or -group name: gid 是 n 或是 group 名称是 name
-ipath p, -path p	路径名称符合 p 的文件，ipath 会忽略大小写
-name name, -iname name	文件名称符合 name 的文件。iname 会忽略大小写
-size n	文件大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w 是二个位元组

-type c	文件类型是 c 的文件
	d: 目录
	c: 字型装置文件
	b: 区块装置文件
	p: 具名贮列
	f: 一般文件
	l: 符号连结
	s: socket

注释: -pid n : process id 是 n 的文件

你可以使用 () 将运算式分隔, 并使用下列运算。

exp1 -and exp2

! expr

-not expr

exp1 -or exp2

exp1, exp2

4. 实例

4.1 列出目前目录及其子目录下所有延伸档名是 c 的文件

```
1 # find . -name "*.c"
```

4.2 列出目前目录其下子目录中所有一般文件

```
1 # find . -type f
```

4.3 将目前目录及其子目录下所有最近 20 天内更新过的文件列出

```
1 # find . -ctime -20
```

4.4 查找前目录中文件属主具有读、写权限, 并且文件所属组的用户和其他用户具有读权限的文件

```
1 # find . -type f -perm 644 -exec ls -l {} \;
```

4.5 为了查找系统中所有文件长度为0的普通文件, 并列出它们的完整路径

```
1 # find / -type f -size 0 -exec ls -l {} \;
```

grep-查找文件里符合条件的字符串

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 查找后缀有 "xxx"字样的文件中包含"xxx"字符串的文件
 - 3.2 以递归的方式查找符合条件的文件
 - 3.3反向查找

1、功能说明

grep 指令用于查找内容包含指定的范本样式的文件。

如果发现某文件的内容符合所指定的范本样式，预设 grep 指令会把含有范本样式的那一行显示出来。若不指定任何文件名称，或是所给予的文件名为 -，则 grep 指令会从标准输入设备读取数据。

2、命令格式与参数

```
shell

grep [-abcEFgHiILnqrsvVwxy][-A<>][-B<>][-C<>][-d<>][-e<>][-f<>][--help][...]
```

参数	说明
-i 或 --ignore-case	不区分大小写
-h 或 --no-filename	在显示符合样式的那一行之前，不标示该行所属的文件名称。
-l 或 --file-with-matches	列出文件内容符合指定的样式的文件名称
-n 或 --line-number	显示匹配行及行号
-s 或 --no-messages	不显示错误信息
-v 或 --invert-match	显示不包含匹配文本的所有行。
-l 或 --file-with-matches	列出文件内容符合指定的样式的文件名称
-x 或 --line-regexp	只显示全列符合的列。
-w 或 --word-regexp	只显示全字符符合的列

3、常见用法实例

3.1 查找后缀有 "xxx"字样的文件中包含"xxx"字符串的文件

```
在当前目录中，查找后缀有 file 字样的文件中包含 test 字符串的文件，并打印出该字符串的行。

#grep test *file
//
testfile1:This a Linux testfile! #testfile1 test
testfile_2:This is a linux testfile! #testfile_2 test
testfile_2:Linux test #testfile_2 test
```

3.2 以递归的方式查找符合条件的文件

查找指定目录/etc/acpi 及其子目录（如果存在子目录的话）下所有文件中包含字符串"update"的文件，并打印出该字符串所在行的内容

```
#grep -r update /etc/acpi
//
/etc/acpi/events/thinkpad-cmos:action=/usr/sbin/thinkpad-keys --update
```

3.3反向查找

查找文件名中包含 test 的文件中不包含test 的行

```
#grep -v test *test*
//
testfile1:helLinux!
testfile1:Linis a free Unix-type operating system.
testfile1:Lin
testfile_1:HELLO LINUX!
testfile_1:LINUX IS A FREE UNIX-TYPE OPERATING SYSTEM.
testfile_1:THIS IS A LINUX TESTFILE!
testfile_2:HELLO LINUX!
testfile_2:Linux is a free unix-type operating system.
```

gzip-压缩文件

- 1、功能说明
- 2、命令格式和参数
 - 2.1、命令格式
 - 2.2、命令参数
- 3、常见用法示例
 - 3.1、压缩文件
 - 3.2、解压缩列出详细信息
 - 3.3、显示压缩文件信息

1、功能说明

Linux gzip命令用于压缩文件。

gzip是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。

2、命令格式和参数

2.1、命令格式

```
gzip [-acdfhlLnNqrtvV][-S <>][-<>][--best/fast][...] gzip [-acdfhlLnNqrtvV][-S <>][-<>][--best/fast][...]
```

2.2、命令参数

简写参数	完整参数	参数说明
-c	--stdout或--to-stdout	把压缩后的文件输出到标准输出设备，不去更动原始文件。
-a	--ascii	使用ASCII文字模式。
-d	--decompress或--uncompress	解开压缩文件。
-f	--force	强行压缩文件。不理睬文件名称或硬连接是否存在以及该文件是否为符号连接。
-h	--help	在线帮助。
-l	--list	列出压缩文件的相关信息。
-L	--license	显示版本与版权信息。
-n	--no-name	压缩文件时，不保存原来的文件名称及时间戳记。
-N	--name	压缩文件时，保存原来的文件名称及时间戳记。
-q	--quiet	不显示警告信息。
-r	--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	--suffix	更改压缩字尾字符串。
-t	--test	测试压缩文件是否正确无误。
-v	--verbose	显示指令执行过程。
-V	--version	显示版本信息。
-<压缩效率>		压缩效率是一个介于1—9的数值，预设值为"6"，指定愈大的数值，压缩效率就会愈高。
	--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。
--	--------	--------------------

3、常见用法示例

3.1、压缩文件

压缩文件

```
[root@runoob.com a]# ls //
a.c b.h d.cpp
[root@runoob.com a]# gzip * //
[root@runoob.com a]# ls //
a.c.gz    b.h.gz    d.cpp.gz
```

3.2、解压缩列出详细信息

列出详细信息

```
[root@runoob.com a]# gzip -dv * //
a.c.gz:      0.0% -- replaced with a.c
b.h.gz:      0.0% -- replaced with b.h
d.cpp.gz:    0.0% -- replaced with d.cpp
[root@runoob.com a]#
```

3.3、显示压缩文件信息

显示压缩文件信息

```
[root@runoob.com a]# gzip -l *
  compressed      uncompressed ratio uncompressed_name
      24           0   0.0% a.c
      24           0   0.0% b.h
      26           0   0.0% d.cpp
```

ifconfig-显示或设置网络设备

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1显示网络设备信息
 - 3.2启动关闭指定网卡
 - 3.3修改MAC地址
 - 3.4配置IP地址

1、功能说明

ifconfig命令用于显示或设置网络设备。
ifconfig可设置网络设备的状态，或是显示目前的设置。

2、命令格式与参数

```
ifconfig [[down up -allmulti -arp -promisc][add<>][del<>][<hw<><>][io_addr<I/O>]  
[irq<IRQ>][media<>][mem_start<>][metric<>][mtu<>][netmask<>]  
[tunnel<>][-broadcast<>][-pointopoint<>][IP]
```

参数	说明
add<地址>	设置网络设备IPv6的IP地址
del<地址>	删除网络设备IPv6的IP地址
down	关闭指定的网络设备
<hw<网络设备类型><硬件地址>	设置网络设备的类型与硬件地址
io_addr<I/O地址>	设置网络设备的I/O地址
irq<IRQ地址>	设置网络设备的IRQ
media<网络媒介类型>	设置网络设备的媒介类型
mem_start<内存地址>	设置网络设备在主内存所占用的起始地址
metric<数目>	指定在计算数据包的转送次数时，所要加上的数目
mtu<字节>	设置网络设备的MTU
netmask<子网掩码>	设置网络设备的子网掩码
tunnel<地址>	建立IPv4与IPv6之间的隧道通信地址
up	建立IPv4与IPv6之间的隧道通信地址
-broadcast<地址>	将要送往指定地址的数据包当成广播数据包来处理
-pointopoint<地址>	与指定地址的网络设备建立直接连线，此模式具有保密功能
-promisc	关闭或启动指定网络设备的promiscuous模式
[IP地址]	指定网络设备的IP地址
[网络设备]	指定网络设备的名称

3、常见用法实例

3.1显示网络设备信息

显示网络设备信息

```
# ifconfig
eth0   Link encap:Ethernet HWaddr 00:50:56:0A:0B:0C
        inet addr:192.168.0.3 Bcast:192.168.0.255 Mask:255.255.255.0
        inet6 addr: fe80::250:56ff:fe0a:b0c/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:172220 errors:0 dropped:0 overruns:0 frame:0
        TX packets:132379 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:87101880 (83.0 MiB) TX bytes:41576123 (39.6 MiB)
        Interrupt:185 Base address:0x2024
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:2022 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2022 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:2459063 (2.3 MiB) TX bytes:2459063 (2.3 MiB)
```

3.2启动关闭指定网卡

启动关闭指定网卡

```
# ifconfig eth0 down
# ifconfig eth0 up
```

3.3修改MAC地址

修改MAC地址

```
# ifconfig eth0 down //
# ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE //MAC
# ifconfig eth0 up //
# ifconfig eth1 hw ether 00:1D:1C:1D:1E //MAC
# ifconfig eth1 up //
```

3.4配置IP地址

配置IP地址

```
# ifconfig eth0 192.168.1.56 //eth0IP
# ifconfig eth0 192.168.1.56 netmask 255.255.255.0 // eth0IP,
# ifconfig eth0 192.168.1.56 netmask 255.255.255.0 broadcast 192.168.1.255 // eth0IP,,
```

ifdown-禁用指定的网络接口

- [1、功能说明](#)
- [2、命令格式与参数](#)
- [3、实例](#)

1、功能说明

用于禁用指定的网络接口

2、命令格式与参数

命令格式:

ifdown (参数)

参数:

网络接口: 要禁用的网络接口

3、实例

```
ifdown eth0 #eth0
```

locate-查找文档或目录

- 1、功能说明
- 2、命令格式与参数
 - 命令格式:
 - 参数说明:
- 3、常见用法示例
 - 3.1、查找passwd文件:
 - 3.2、搜索 etc 目录下所有以 sh 开头的文件:
 - 3.3、忽略大小写搜索当前用户目录下所有以 r 开头的文件:

1、功能说明

Linux locate命令用于查找符合条件的文档，它会在保存文档和目录名称的数据库内，查找合乎范本样式条件的文档或目录。

一般情况我们只需要输入 **locate your_file_name** 即可查找指定文件。

2、命令格式与参数

命令格式:

```
locate [-d ][-help][-version][范本样式...]
```

参数说明:

简写参数	完整参数	参数说明
-b	--basename	-- 仅匹配路径名的基本名称
-c	--count	-- 只输出找到的数量
-d	--database	-- 使用 DBPATH 指定的数据库，而不是默认数据库 /var/lib/mlocate/mlocate.db
-e	--existing	-- 仅打印当前现有文件的条目
-l		--启动安全模式。在安全模式下，使用者不会看到权限无法看到的档案。这会始速度减慢，因为 locate 必须至实际的档案系统中取得档案的 权限资料。
-0	--null	-- 在输出上带有NUL的单独条目
-S	--statistics	-- 不搜索条目，打印有关每个数据库的统计信息
-q		-- 安静模式，不会显示任何错误讯息。
-p	--nofollow	-- 检查文件存在时不要遵循尾随的符号链接
-l	--limit,-n LIMIT	-- 将输出（或计数）限制为LIMIT个条目
-n		-- 至多显示 n个输出。
-m	--mmap	-- 被忽略，为了向后兼容
-r	--regexp	-- 使用基本正则表达式
	--regex	-- 使用扩展正则表达式
-q	--quiet	-- 安静模式，不会显示任何错误讯息
-s	--stdio	-- 被忽略，为了向后兼容
-o		-- 指定资料库的名称

-h	--help	-- 显示帮助
-i	--ignore-case	-- 忽略大小写
-V	-version	-- 显示版本信息

3、常见用法示例

3.1、查找passwd文件：

```

shell

[root@itemint ~]# locate passwd
#/etc/passwd
#/etc/passwd-
#/etc/pam.d/passwd
#/etc/security/opasswd
#/usr/bin/gpasswd
#/usr/bin/grub2-mkpasswd-pbkdf2
#/usr/bin/lppasswd
#/usr/bin/passwd
#/usr/bin/vncpasswd

```

3.2、搜索 etc 目录下所有以 sh 开头的文件：

```

shell

[root@itemint ~]# locate /etc/sh
#/etc/shadow
#/etc/shadow-
#/etc/shells
#/var/lib/docker/overlay/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4/root/etc/shadow
#/var/lib/docker/overlay/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4/root/etc/shadow-#/var
/lib/docker/overlay/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4/root/etc/shells

```

3.3、忽略大小写搜索当前用户目录下所有以 r 开头的文件：

```

shell

[root@itemint ~]# locate -i ~/r
#/var/lib/docker/overlay/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4
#/root/run/systemd/seats/var/lib/docker/overlay/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4
#/root/run/systemd/sessions/var/lib/docker/overlay
/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4
#/root/run/systemd/shutdown/var/lib/docker/overlay
/010d201919f76307764f6f8c2137218526fb36aaee6063df08eea701439655d4
#/root/run/systemd/users

```

ls-显示目录结构

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1、列出根目录(\)下的所有目录：
 - 3.2、列出目前工作目录下所有名称是 s 开头的文件，越新的排越后面：
 - 3.3、将 /bin 目录以下所有目录及文件详细资料列出：
 - 3.4、列出目前工作目录下所有文件及目录；目录于名称后加 "/"，可执行档于名称后加 "*"：

1、功能说明

Linux系统是由各种文件和目录构成的，就像Windows系统下文件和文件夹的关系，一级一级的向下包含扩展。使用Linux中的ls命令可以使我们方便的了解Linux系统下的目录结构。

2、命令格式与参数

Shell
ls [-alrtAFR] [name...]

参数	说明
ls	显示当前目录下文件，按照文件名排序
ls -a	显示所有项目，不忽略隐藏文件
ls -s	显示项目时附带大小
ls -l	显示项目的详细信息
ls -t	显示项目并按修改时间排序
ls -r	显示项目并按照选定排序方式逆序排列
ls -ltrah	显示所有文件易读详细信息，并按照时间倒序排列

3、常见用法实例

3.1、列出根目录(\)下的所有目录：

Shell
ls / bin dev lib media net root srv upload www boot etc lib64 misc opt sbin sys usr home lost+found mnt proc selinux tmp var

3.2、列出目前工作目录下所有名称是 s 开头的文件，越新的排越后面：

Shell
ls -ltr s*

3.3、将 /bin 目录以下所有目录及文件详细资料列出：

Shell

```
# ls -lR /bin
```

3.4、列出目前工作目录下所有文件及目录；目录于名称后加 "/", 可执行档于名称后加 "*"：

Shell

```
# ls -AF
```

mkdir-创建目录

- 1. 功能说明
 - 2. 命令格式与参数
 - 3. 实例
 - 3.1 创建目录
 - 3.2 在不存在的目录下建立目录
-

1. 功能说明

Linux mkdir（英文全拼：make directory）命令用于创建目录。

2. 命令格式与参数

1	<code>mkdir [-p] dirName</code>
---	---------------------------------

<code>-p</code>	确保目录名称存在，不存在的就建一个
-----------------	-------------------

3. 实例

3.1 创建目录

在工作目录下，建立一个名为 runoob 的子目录：

1	<code>\$ mkdir runoob</code>
---	------------------------------

3.2 在不存在的目录下建立目录

在工作目录下的 runoob2 目录中，建立一个名为 test 的子目录。

若 runoob2 目录原本不存在，则建立一个。（注：本例若不加 -p 参数，且原本 runoob2 目录不存在，则产生错误。）

1	<code>\$ mkdir -p runoob2/test</code>
---	---------------------------------------

mv-移动文件或文件夹

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例

1、功能说明

mv命令用来为文件或目录改名、或将文件或目录移入其它位置。

2、命令格式与参数

```
mv [options] source dest
```

参数	说明
-b	当目标文件或目录存在时，在执行覆盖前，会为其创建一个备份
-i	如果指定移动的源目录或文件与目标的目录或文件同名，则会先询问是否覆盖旧文件，输入 y 表示直接覆盖，输入 n 表示取消该操作
-f	如果指定移动的源目录或文件与目标的目录或文件同名，不会询问，直接覆盖旧文件
-n	不要覆盖任何已存在的文件或目录
-u	当源文件比目标文件新或者目标文件不存在时，才执行移动操作

3、常见用法实例

```
$ mv -b 1.txt ./1
$ cd 1
$ ls
1.txt  1.txt~
```

```
$ mv -i 1.txt ./1
mv './1/1.txt' y
```

```
mv source_file() dest_file()source_filedest_file
```

```
mv source_file() dest_directory()
source
```

```
mv source_directory() dest_directory()
dest_directorysource_directorydest_directorysource_directorydest_directory
```


netstat-打印网络系统的状态信息

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 列出所有端口 (包括监听和未监听的)
 - 3.2 列出所有处于监听状态的 Sockets
 - 3.3 显示每个协议的统计信息
 - 3.4 在netstat输出中显示 PID 和进程名称
 - 3.5 在netstat输出中不显示主机，端口和用户名(host, port or user)
 - 3.6 持续输出netstat信息
 - 3.7 显示系统不支持的地址族(Address Families)
 - 3.8 显示核心路由信息
 - 3.9 找出程序运行的端口
 - 3.10 显示网络接口列表
 - 3.11 IP和TCP分析

1、功能说明

netstat命令用来打印Linux中网络系统的状态信息，可让你得知整个Linux系统的网络情况。

2、命令格式与参数

netstat(option)

简写	全称	说明
-a	--all	显示所有连线中的Socket
-A	--	列出该网络类型连线中的相关地址
-c	--continuous	持续列出网络状态
-C	--cache	显示路由器配置的快取信息
-e	--extend	显示网络其他相关信息
-F	--fib	显示FIB
-g	--groups	显示多重广播功能群组组员名单
-h	--help	在线帮助
-i	--interfaces	显示网络界面信息表单
-l	--listening	显示监控中的服务器的Socket
-M	--masquerade	显示伪装的网络连线
-n	--numeric	直接使用ip地址，而不通过域名服务器
-N	--netlink	显示网络硬件外围设备的符号连接名称
-o	--timers	显示计时器
-p	--programs	显示正在使用Socket的程序识别码和程序名称
-r	--route	显示Routing Table
-s	--statistic	显示网络工作信息统计表
-t	--tcp	显示TCP传输协议的连线状况
-u	--udp	显示UDP传输协议的连线状况
-v	--verbose	显示指令执行过程

-V	--version	显示版本信息
-w	--raw	显示RAW传输协议的连线状况
-x	--unix	此参数的效果和指定"-A unix"参数相同
-ip	--inet	此参数的效果和指定"-A inet"参数相同

3、常见用法实例

3.1 列出所有端口 (包括监听和未监听的)

Shell
<pre>netstat -a # netstat -at #tcp netstat -au #udp</pre>

3.2 列出所有处于监听状态的 Sockets

Shell
<pre>netstat -l # netstat -lt # tcp netstat -lu # udp netstat -lx # UNIX</pre>

3.3 显示每个协议的统计信息

Shell
<pre>netstat -s # netstat -st #TCP netstat -su #UDP</pre>

3.4 在netstat输出中显示 PID 和进程名称

Shell
<pre>netstat -pt netstat -p #"PID/"netstatdebugging</pre>

3.5 在netstat输出中不显示主机，端口和用户名(host, port or user)

Shell

```
#netstat -n

netstat -an
#

netsat -a --numeric-ports
netsat -a --numeric-hosts
netsat -a --numeric-users
```

3.6 持续输出netstat信息

Shell

```
netstat -c #
```

3.7 显示系统不支持的地址族(Address Families)

Shell

```
netstat --verbose
#

netstat: no support for `AF IPX' on this system.
netstat: no support for `AF AX25' on this system.
netstat: no support for `AF X25' on this system.
netstat: no support for `AF NETROM' on this system.
```

3.8 显示核心路由信息

Shell

```
netstat -r
#netstat -rn
```

3.9 找出程序运行的端口

Shell

```
# root

netstat -ap | grep ssh
#

netstat -an | grep ':80'
```

3.10 显示网络接口列表

Shell

```
netstat -i  
#ifconfignetstat -ie
```

3.11 IP和TCP分析

Shell

```
#IP  
  
netstat -ntu | grep :80 | awk '{print $5}' | cut -d: -f1 | awk '++ip[$1] END {for(i in ip) print ip[i],"\t",  
i}' | sort -nr  
#TCP  
  
netstat -nt | grep -e 127.0.0.1 -e 0.0.0.0 -e ::: -v | awk '/^tcp/ {++state[$NF]} END {for(i in state) print i,"  
\t",state[i}]'  
#phpcgi  
  
netstat -anpo | grep "php-cgi" | wc -l
```

ping-检测主机网络

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1检测是否与主机连通
 - 3.2指定接收包的次数
 - 3.3多参数使用

1、功能说明

ping命令用于检测主机。
执行ping指令会使用ICMP传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。

2、命令格式与参数

```
ping [-dfnqrRv]
      [-c<>]
      [-i<>]
      [-I<>]
      [-l<>]
      [-p<>]
      [-s<>]
      [-t<>][IP]
```

参数	说明
-d	使用Socket的SO_DEBUG功能
-c<完成次数>	设置完成要求回应的次数
-f	极限检测
-i<间隔秒数>	指定收发信息的间隔时间
-I<网络界面>	使用指定的网络接口送出数据包
-l<前置载入>	设置在送出要求信息之前，先行发出的数据包
-n	只输出数值
-p<范本样式>	设置填满数据包的范本样式
-q	不显示指令执行过程，开头和结尾的相关信息除外
-r	忽略普通的Routing Table，直接将数据包送到远端主机上
-R	记录路由过程
-s<数据包大小>	设置数据包的大小
-t<存活数值>	设置存活数值TTL的大小
-v	详细显示指令的执行过程

3、常见用法实例

3.1检测是否与主机连通

检测是否与主机连通

```
# ping www.w3cschool.cc //ping
PING aries.m.alikunlun.com (114.80.174.110) 56(84) bytes of data.
64 bytes from 114.80.174.110: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 114.80.174.110: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 114.80.174.110: icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from 114.80.174.110: icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from 114.80.174.110: icmp_seq=5 ttl=64 time=0.028 ms
64 bytes from 114.80.174.110: icmp_seq=6 ttl=64 time=0.028 ms
64 bytes from 114.80.174.110: icmp_seq=7 ttl=64 time=0.034 ms
64 bytes from 114.80.174.110: icmp_seq=8 ttl=64 time=0.034 ms
64 bytes from 114.80.174.110: icmp_seq=9 ttl=64 time=0.036 ms
64 bytes from 114.80.174.110: icmp_seq=10 ttl=64 time=0.041 ms

--- aries.m.alikunlun.com ping statistics ---
10 packets transmitted, 30 received, 0% packet loss, time 29246ms
rtt min/avg/max/mdev = 0.021/0.035/0.078/0.011 ms

//Ctrl+C
```

3.2指定接收包的次数

指定接收包的次数

```
# ping -c 2 www.w3cschool.cc
PING aries.m.alikunlun.com (114.80.174.120) 56(84) bytes of data.
64 bytes from 114.80.174.120: icmp_seq=1 ttl=54 time=6.18 ms
64 bytes from 114.80.174.120: icmp_seq=2 ttl=54 time=15.4 ms

--- aries.m.alikunlun.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1016ms
rtt min/avg/max/mdev = 6.185/10.824/15.464/4.640 ms

//
```

3.3多参数使用

修改MAC地址

```
# ping -i 3 -s 1024 -t 255 g.cn //ping
PING g.cn (203.208.37.104) 1024(1052) bytes of data.
1032 bytes from bg-in-f104.1e100.net (203.208.37.104): icmp_seq=0 ttl=243 time=62.5 ms
1032 bytes from bg-in-f104.1e100.net (203.208.37.104): icmp_seq=1 ttl=243 time=63.9 ms
1032 bytes from bg-in-f104.1e100.net (203.208.37.104): icmp_seq=2 ttl=243 time=61.9 ms

--- g.cn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 6001ms
rtt min/avg/max/mdev = 61.959/62.843/63.984/0.894 ms, pipe 2
[root@linux ~]#

//-i 3 3 -s -t TTL 255
```

pwd-显示工作目录

- [1. 功能说明](#)
- [2. 命令格式与参数](#)
- [3.实例](#)

1. 功能说明

Linux **pwd**（英文全拼：print work directory）命令用于显示工作目录。

执行 pwd 指令可立刻得知您目前所在的工作目录的绝对路径名称。

2. 命令格式与参数

shell

```
pwd [--help][--version]
```

参数	参数解释
--help	在线帮助
--version	显示版本信息

3.实例

```
$ pwd  
  
/root/test #
```

rmmdir-删除空目录

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1删除空目录
 - 3.2删除包含仅包含一个空目录的目录

1、功能说明

Linux rmmdir（英文全拼：remove directory）命令删除空的目录

2、命令格式与参数

```
rmmdir [-p]
dirName
```

参数	说明
-p	是当子目录被删除后使它也成为空目录的话，则顺便一并删除。

3、常见用法实例

3.1删除空目录

将工作目录下，名为 AAA 的子目录删除

```
rmmdir AAA
```

3.2删除包含仅包含一个空目录的目录

在工作目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为空目录，则 BBB 亦予删除

```
rmmdir -p BBB/
Test
```


rm-删除文件或目录

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1基本用法
 - 3.2递归删除
 - 3.3强制删除

1、功能说明

rm 命令可任意删除一个目录中的一个或多个文件或目录，也可以将某个目录及其下属的所有文件及其子目录均删掉。

2、命令格式与参数

Shell
<code>rm []</code>

参数	说明
<code>-f</code>	强制删除文件或目录，忽略不存在是文件，不给出提示
<code>-i</code>	删除已有文件或目录之前先询问用户
<code>-r</code> 或 <code>-R</code>	将参数中列出的全部目录和子目录均递归地删除
<code>-v</code>	显示指令的详细执行过程
<code>--help</code>	显示此帮助信息并退出

3、常见用法实例

3.1基本用法

rm命令如果任何选项都不加，则默认执行“**rm -i 文件名**”，即在删除文件之间会询问用户

```
#touch file1
#rm file1
rm: "file1"y
```

3.2递归删除

经过询问地删除test目录以及其目录下的子目录study

```
#rm -r test
rm: "test"? y
rm: "test/study"? y
rm: "test"? y
```

3.3强制删除

强制删除project目录和该目录下的子目录并展示删除详情

```
#rm -rfv /project  
: "/project/docs/demands"  
: "/project/docs"  
: "/project"
```

route-路由设置

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例

1、功能说明

route命令用来显示并设置Linux内核中的网络路由表，route命令设置的路由主要是静态路由。要实现两个不同的子网之间的通信，需要一台连接两个网络的路由器，或者同时位于两个网络的网关来实现。

在Linux系统中设置路由通常是为了解决以下问题：该Linux系统在一个局域网中，局域网中有一个网关，能够让机器访问Internet，那么就需要将这台机器的ip地址设置为Linux机器的默认路由。要注意的是，直接在命令行下执行route命令来添加路由，不会永久保存，当网卡重启或者机器重启之后，该路由就失效了；可以在/etc/rc.local中添加route命令来保证该路由设置永久有效。

2、命令格式与参数

语法：

```
route( ) ( )
```

选项：

-A：设置地址类型

-C：打印将Linux核心的路由缓存

-v：详细信息模式

-n：不执行DNS反向查找，直接显示数字形式的IP地址

-c：netstat格式显示路由表

-net：到一个网络的路由表

-host：到一个主机的路由表

参数：

Add：增加指定的路由记录

Dcl：删除指定的路由记录

Target：目的网络或目的主机

gw：设置默认网关

mss：设置TCP的最大区块长度（MSS），单位MB

window：指定通过路由表的TCP连接的TCP窗口大小

dev：路由记录所表示的网络接口。

3、常见用法实例

```
[root@www ~]# route [-n]

-n    IP

[root@www ~]# route
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref  Use  Iface
192.168.0.0    *               255.255.255.0   U      0      0    0   eth0
127.0.0.0      *               255.0.0.0       U      0      0    0   lo
default        192.168.0.254  0.0.0.0         UG     0      0    0   eth0

[root@www ~]# route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref  Use  Iface
192.168.0.0    0.0.0.0         255.255.255.0   U      0      0    0   eth0
127.0.0.0      0.0.0.0         255.0.0.0       U      0      0    0   lo
0.0.0.0        192.168.0.254  0.0.0.0         UG     0      0    0   eth0
```

上面输出的数据共有八个字段，你需要注意的有几个地方：

- **Destination**：其实就是 Network 的意思；
- **Gateway**：就是该接口的 Gateway 那个 IP 啦！若为 0.0.0.0 表示不需要额外的 IP；
- **Genmask**：就是 Netmask 啦！与 Destination 组合成为一部主机或网域；
- **Flags**：共有多个旗标可以来表示该网域或主机代表的意义：
- **U**：代表该路由可用；
- **G**：代表该网域需要经由 Gateway 来帮忙转递；
- **H**：代表该行路由为一部主机，而非一整个网域；
- **Iface**：就是 Interface (接口) 的意思。

在上面的例子当中,第一栏为『目的地的网域』，例如 `192.168.0.0` 就是一个网域，最后一栏显示的是『要去到这个目的地要使用哪一个网络接口！』例如 eth0 就是网络卡的装置代号啦。如果我们要传送的封包在路由规则里面的 `192.168.0.0/255.255.255.0` 或者 `127.0.0.0/255.0.0.0` 里面时，因为第二栏 Gateway 为 `*`，所以就会直接以后面的网络接口来传出去，而不透过 Gateway！万一我们要传送的封包目的地 IP 不在路由规则里面，那么就会将封包传送到『default』所在的那个路由规则去，也就是 `192.168.0.254` 那个 Gateway 喔！所以，几乎每一部主机都会有一个 `default gateway` 来帮他们负责所有非网域内的封包转递！

scp命令-复制文件和目录

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1、从本地复制到远程
 - 3.2、从远程复制到本地
 - 3.3、说明

1、功能说明

scp 命令用于 Linux 之间复制文件和目录。

scp 是 secure copy 的缩写,scp 是 linux 系统下基于 ssh 登陆进行安全的远程文件拷贝命令。

scp 是加密的, [rsh](#) 是不加密的, scp 是 rcp 的加强版。

2、命令格式与参数

Shell

```
scp [-l246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
[-l limit] [-o ssh_option] [-P port] [-S program]
[[user@]host1:]file1 [...] [[user@]host2:]file2
```

参数	说明
-1	强制scp命令使用协议ssh1
-2	强制scp命令使用协议ssh2
-4	强制scp命令只使用IPv4寻址
-6	强制scp命令只使用IPv6寻址
-B	使用批处理模式（传输过程中不询问传输口令或短语）
-C	允许压缩。（将-C标志传递给ssh，从而打开压缩功能）
-p	保留原文件的修改时间，访问时间和访问权限。
-q	不显示传输进度条。
-r	递归复制整个目录。
-v	详细方式显示输出。scp和ssh(1)会显示出整个过程的调试信息。这些信息用于调试连接，验证和配置问题。
-c cipher	以cipher将数据传输进行加密，这个选项将直接传递给ssh。
-F ssh_config	指定一个替代的ssh配置文件，此参数直接传递给ssh。
-i identity_file	从指定文件中读取传输时使用的密钥文件，此参数直接传递给ssh。
-l limit	限定用户所能使用的带宽，以Kbit/s为单位。
-P port	注意是大写的P,port是指定数据传输用到的端口号
-S program	指定加密传输时所使用的程序。此程序必须能够理解ssh(1)的选项。

3、常见用法实例

3.1、从本地复制到远程

命令格式：

Shell

```
scp local_file remote_username@remote_ip:remote_folder  
  
scp local_file remote_username@remote_ip:remote_file  
  
scp local_file remote_ip:remote_folder  
  
scp local_file remote_ip:remote_file
```

- 第1,2个指定了用户名，命令执行后需要再输入密码，第1个仅指定了远程的目录，文件名字不变，第2个指定了文件名；
- 第3,4个没有指定用户名，命令执行后需要输入用户名和密码，第3个仅指定了远程的目录，文件名字不变，第4个指定了文件名；

应用实例：

Shell

```
scp /home/space/music/1.mp3 root@www.runoob.com:/home/root/others/music  
scp /home/space/music/1.mp3 root@www.runoob.com:/home/root/others/music/001.mp3  
scp /home/space/music/1.mp3 www.runoob.com:/home/root/others/music  
scp /home/space/music/1.mp3 www.runoob.com:/home/root/others/music/001.mp3
```

复制目录命令格式：

Shell

```
scp -r local_folder remote_username@remote_ip:remote_folder  
  
scp -r local_folder remote_ip:remote_folder
```

- 第1个指定了用户名，命令执行后需要再输入密码；
- 第2个没有指定用户名，命令执行后需要输入用户名和密码；

应用实例：

Shell

```
scp -r /home/space/music/ root@www.runoob.com:/home/root/others/  
scp -r /home/space/music/ www.runoob.com:/home/root/others/
```

上面命令将本地 music 目录复制到远程 others 目录下。

3.2、从远程复制到本地

从远程复制到本地，只要将从本地复制到远程的命令的后2个参数调换顺序即可，如下实例

应用实例：

Shell

```
scp root@www.runoob.com:/home/root/others/music /home/space/music/1.mp3  
scp -r www.runoob.com:/home/root/others/ /home/space/music/
```

3.3、说明

1.如果远程服务器防火墙有为scp命令设置了指定的端口，我们需要使用 -P 参数来设置命令的端口号，命令格式如下：

Shell

```
#scp 4588  
scp -P 4588 remote@www.runoob.com:/usr/local/sin.sh /home/administrator
```

2.使用scp命令要确保使用的用户具有可读取远程服务器相应文件的权限，否则scp命令是无法起作用的。

shutdown-关闭/重启计算机

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 立即关机
 - 3.2 指定 10 分钟后关机
 - 3.3 重新启动计算机
 - 3.3 制定20:13关机

1、功能说明

Linux系统下的shutdown命令用于安全的关闭/重启计算机，它不仅可以方便的实现定时关机，还可以由用户决定关机时的相关参数。在执行shutdown命令时，系统会给每个终端（用户）发送一条屏显，提示关机操作。定时关机只需要一个简单的参数，既可以是倒计时，也可以是确切的时间。

2、命令格式与参数

Shell

```
# shutdown [-t seconds] [-rkhncfF] time [message]
```

参数	说明
-t seconds	设定在几秒钟之后进行关机程序
-k	并不会真的关机，只是将警告讯息传递给所有使用者
-r	关机后重新开机
-h	关机后停机
-n	不采用正常程序来关机，用强迫的方式杀掉所有执行中的程序后自行关机
-c	取消目前已经进行中的关机动作
-f	关机时，不做 fck 动作(检查 Linux 档系统)
-F	关机时，强迫进行 fck 动作

3、常见用法实例

3.1 立即关机

Shell

```
# shutdown -h now
```

3.2 指定 10 分钟后关机

Shell

```
# shutdown -h 10
```


3.3 重新启动计算机

Shell

```
# shutdown -r now
```

3.3 制定20:13关机

Shell

```
# shutdown -h 20:13
```

touch-创建文件或修改文件时间

- 1、功能说明
- 2、命令格式和参数
 - 2.1、命令格式:
 - 2.2、命令参数:
- 3、常见用法实例
 - 3.1创建不存在的文件
 - 3.2更新log.log的时间和log2012.log时间戳相同
 - 3.3设定文件的时间戳

1、功能说明

Linux的touch命令最主要是在创建新的空文件，创建一个或者多个文件，且用户必须拥有该目录下的写权限。

touch命令可用来更新文件的访问时间和修改时间，也可以指定参数来更新时间。

2、命令格式和参数

2.1、命令格式:

touch [option] [file]

2.2、命令参数:

简写参数	完整参数	参数说明
-a	--time=atime或--time=access或--time=use	只更改存取时间。
-c	--no-create	不建立任何文档。
-d		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	--time=mtime或--time=modify	只更改变动时间。
-r		把指定文档或目录的日期时间，统统设成和参考文档或目录的日期时间相同。
-t		使用指定的日期时间，而非现在的时间。

3、常见用法实例

3.1创建不存在的文件

创建不存在的文件

```
touch log2012.log log2013.log

[root@localhost test]# touch log2012.log log2013.log
[root@localhost test]# ls
-rw-r--r-- 1 root root    0 10-28 16:01 log2012.log
-rw-r--r-- 1 root root    0 10-28 16:01 log2013.log
```

3.2更新log.log的时间和log2012.log时间戳相同

更新log.log的时间和log2012.log时间戳相同

```
touch -r log.log log2012.log

[root@localhost test]# ls
-rw-r--r-- 1 root root    0 10-28 16:01 log2012.log
-rw-r--r-- 1 root root    0 10-28 16:01 log2013.log
-rw-r--r-- 1 root root    0 10-28 14:48 log.log
[root@localhost test]# touch -r log.log log2012.log
[root@localhost test]# ls
-rw-r--r-- 1 root root    0 10-28 14:48 log2012.log
-rw-r--r-- 1 root root    0 10-28 16:01 log2013.log
-rw-r--r-- 1 root root    0 10-28 14:48 log.log
```

3.3设定文件的时间戳

设定文件的时间戳

```
touch -t 201211142234.50 log.log

[root@localhost test]# ls
-rw-r--r-- 1 root root    0 10-28 14:48 log2012.log
-rw-r--r-- 1 root root    0 10-28 16:01 log2013.log
-rw-r--r-- 1 root root    0 10-28 14:48 log.log
[root@localhost test]# touch -t 201211142234.50 log.log
[root@localhost test]# ls
-rw-r--r-- 1 root root    0 10-28 14:48 log2012.log
-rw-r--r-- 1 root root    0 10-28 16:01 log2013.log
-rw-r--r-- 1 root root    0 2012-11-14 log.log
```

traceroute-显示数据包到主机间的路径

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1显示到达目的地的数据包路由

1、功能说明

traceroute命令用于显示数据包到主机间的路径。

traceroute指令让你追踪网络数据包的路由途径，预设数据包大小是40Bytes，用户可另行设置。

2、命令格式与参数

```
traceroute
[-dFlnrvx]
[-f<>][-g<
>...][-i<>]
[-m<>][-p<
>][-s<>][-t
<>][-w<>][
IP][[
```

参数	说明
-d	使用Socket层级的排错功能
-f<存活数值>	设置第一个检测数据包的存活数值TTL的大小。
-F	设置勿离断位
-g<网关>	设置来源路由网关，最多可设置8个
-i<网络界面>	使用指定的网络界面送出数据包
-I	使用ICMP回应取代UDP资料信息
-m<存活数值>	设置检测数据包的最大存活数值TTL的大小
-n	直接使用IP地址而非主机名称
-p<通信端口>	设置UDP传输协议的通信端口
-r	忽略普通的Routing Table，直接将数据包送到远端主机上
-s<来源地址>	设置本地主机送出数据包的IP地址
-t<服务类型>	设置检测数据包的TOS数值
-v	详细显示指令的执行过程
-w<超时秒数>	设置等待远端主机回报的时间
-x	开启或关闭数据包的正确性检验

3、常见用法实例

3.1显示到达目的地的数据包路由

显示到达目的地的数据包路由

```
# traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 66.249.89.99
traceroute to www.l.google.com (66.249.89.99), 30 hops max, 38 byte packets
 1 192.168.0.1 (192.168.0.1) 0.653 ms 0.846 ms 0.200 ms
 2 118.250.4.1 (118.250.4.1) 36.610 ms 58.438 ms 55.146 ms
 3 222.247.28.177 (222.247.28.177) 54.809 ms 39.879 ms 19.186 ms
 4 61.187.255.253 (61.187.255.253) 18.033 ms 49.699 ms 72.147 ms
 5 61.137.2.177 (61.137.2.177) 32.912 ms 72.947 ms 41.809 ms
 6 202.97.46.5 (202.97.46.5) 60.436 ms 25.527 ms 40.023 ms
 7 202.97.35.69 (202.97.35.69) 40.049 ms 66.091 ms 44.358 ms
 8 202.97.35.110 (202.97.35.110) 42.140 ms 70.913 ms 41.144 ms
 9 202.97.35.14 (202.97.35.14) 116.929 ms 57.081 ms 60.336 ms
10 202.97.60.34 (202.97.60.34) 54.871 ms 69.302 ms 64.353 ms
11 * * *
12 209.85.255.80 (209.85.255.80) 95.954 ms 79.844 ms 76.052 ms
    MPLS Label=385825 CoS=5 TTL=1 S=0
13 209.85.249.195 (209.85.249.195) 118.687 ms 120.905 ms 113.936 ms
14 72.14.236.126 (72.14.236.126) 115.843 ms 137.109 ms 186.491 ms
15 nrt04s01-in-f99.1e100.net (66.249.89.99) 168.024 ms 140.551 ms 161.127 ms
```

useradd-建立用户账号

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 添加一般用户
 - 3.2 为添加的用户指定相应的用户组
 - 3.3 创建一个系统用户
 - 3.4 为新添加的用户指定home目录
 - 3.5 建立用户且制定ID

1、功能说明

useradd 用来建立用户帐号。帐号建好之后，再用 **passwd** 设定帐号的密码。而可用 **userdel** 删除帐号。使用 **useradd** 指令所建立的帐号，实际上是保存在 **/etc/passwd** 文本文件中。

2、命令格式与参数

```
useradd [-mMnr][-c <>][-d <>][-e <>][-f <>][-g <>][-G <>][-s <shell>][-u <uid>][[]]
```

简写	说明
-c<备注>	加上备注文字。备注文字会保存在passwd的备注栏位中。
-d<登入目录>	指定用户登入时的起始目录。
-D	变更预设值。
-e<有效期限>	指定帐号的有效期限。
-f<缓冲天数>	指定在密码过期后多少天即关闭该帐号。
-g<群组>	指定用户所属的群组。
-G<群组>	指定用户所属的附加群组。
-m	自动建立用户的登入目录。
-M	不要自动建立用户的登入目录。
-n	取消建立以用户名作为名的群组。
-r	建立系统帐号。
-s<shell>	指定用户登入后所使用的shell。
-u<uid>	指定用户ID。

3、常见用法实例

3.1 添加一般用户

Shell

useradd tt

3.2 为添加的用户指定相应的用户组

Shell
<pre>useradd -g root tt</pre>

3.3 创建一个系统用户

Shell
<pre>useradd -r tt</pre>

3.4 为新添加的用户指定home目录

Shell
<pre>useradd -d /home/myd tt</pre>

3.5 建立用户且制定ID

Shell
<pre>useradd caojh -u 544</pre>

which-显示命令的绝对路径

- 1、功能说明
- 2、命令格式与参数
- 3、实例

1、功能说明

Linux **which**命令用于查找文件。

which命令的作用是，在PATH变量指定的路径中，搜索某个系统命令的位置，并且返回第一个搜索结果。也就是说，使用which命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令。

2、命令格式与参数

```
which [--] [filename]
```

参数	解释
-a	显示查找到的所有的文件的路径信息。默认情况下只显示第一行。
--read-functions	从标准输入读取shell函数定义，将查找到的函数送到标准输出设备中显示
--skip-tilde	忽略环境变量PATH中以波浪线开头的目录
--skip-dot	忽略环境变量PATH中以点开头的隐藏目录

3、实例

```
$ which cp
alias cp='cp -i'
        /usr/bin/cp  #
$ which find
/usr/bin/find  #
```


kill-删除执行中的程序或工作

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 杀死进程
 - 3.2 强制杀死进程
 - 3.3 显示信号
 - 3.4 杀死指定用户所有进程

1、功能说明

kill 可将指定的信息送至程序。预设的信息为 SIGTERM(15)，可将指定程序终止。若仍无法终止该程序，可使用 SIGKILL(9) 信息尝试强制删除程序。程序或工作的编号可利用 ps 指令或 jobs 指令查看。

2、命令格式与参数

shell
\$ kill [-s <>][]kill [-l <>]

参数简写	解释
-l <信息编号>	若不加<信息编号>选项，则 -l 参数会列出全部的信息名称
-s <信息名称或编号>	指定要送出的信息
[程序]	[程序]可以是程序的PID或是PGID，也可以是工作编号

3、常见用法实例

3.1 杀死进程

shell
\$ kill 12345

3.2 强制杀死进程

shell
\$ kill -KILL 123456

3.3 显示信号

shell

```
$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT     19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

3.4 杀死指定用户所有进程

shell

```
$kill -9 $(ps -ef | grep hnlinux) // hnlinux
$kill -u hnlinux //
```

cmp-比较两个文件是否存在差异

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1比较两个文件是否相同
 - 3.2显示文件中所有不同
 - 3.3比较与自定义输入内容的差异

1、功能说明

cmp命令用于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。若发现有所差异，预设会标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。

2、命令格式与参数

Shell

```
cmp [-clsv][-i <>][--help][[]]
```

参数	说明
-c或--print-chars	除了标明差异处的十进制字码之外，一并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	标示出所有不一样的地方
-s或--quiet或--silent	不显示错误信息
-v或--version	显示版本信息
--help	在线帮助

3、常见用法实例

3.1比较两个文件是否相同

比较file1.txt与file2.txt文件的差异，如果文件相同，则不显示消息，如果文件不同，则显示第一个不同的位置

```
#cmp file1.txt file2.txt

//3
file1 file2 differ: byte 3, line 2
```

3.2显示文件中所有不同

```
#cmp -l file1.txt file2.txt
```

3.3 比较与自定义输入内容的差异

file1.txtCtrl+DEOF

```
#cmp file1.txt -
```

read-从标准输入读取数值

- 1、功能说明
- 2、命令格式与参数
- 3、常见用法实例
 - 3.1 简单读取
 - 3.2 允许在 read 命令行中直接指定一个提示

1、功能说明

read 内部命令被用来从标准输入读取单行数据。这个命令可以用来读取键盘输入，当使用重定向的时候，可以读取文件中的一行数据。

2、命令格式与参数

shell

```
$ read [--] [filename]
```

参数	解释
-a	后跟一个变量，该变量会被认为是个数组，然后给其赋值，默认是以空格为分割符
-d	后面跟一个标志符，其实只有其后的第一个字符有用，作为结束的标志
-p	后面跟提示信息，即在输入前打印提示信息
-c	在输入的时候可以使用命令补全功能
-n	后跟一个数字，定义输入文本的长度
-r	屏蔽\，如果没有该选项，则\作为一个转义字符，有的话\就是个正常的字符了
-s	安静模式，在输入字符时不再屏幕上显示，例如login时输入密码
-t	后面跟秒数，定义输入字符的等待时间

3、常见用法实例

3.1 简单读取

shell

```
#!/bin/bash

#
echo ": "
#
read website
echo " $website"
exit 0 #
```

测试结果为：

shell

```
:  
www.runoob.com  
www.runoob.com
```

3.2 允许在 read 命令行中直接指定一个提示

shell

```
#!/bin/bash  
  
read -p ":" website  
echo " $website"  
exit 0
```

测试结果为：

shell

```
:www.runoob.com  
www.runoob.com
```