**显示目录和文件的命令**

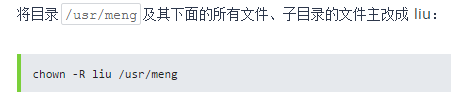
Ls：用于查看所有文件夹的命令,

显示当前目录下非影藏文件与目录:  
 [root@localhost ~]# ls  
显示当前目录下包括影藏文件在内的所有文件列表:  
 [root@localhost ~]# ls -a  
列出文件和文件夹的详细信息:  
 [root@localhost /]# ls -l  
  
Du：显示目录或文件大小

## 显示目录或者文件所占空间： [root@localhost test]# du 只显示当前目录下面的子目录的目录大小和当前目录的总的大小，最下面的1288为当前目录的总大小 显示指定文件所占空间： [root@localhost test]# du log2012.log

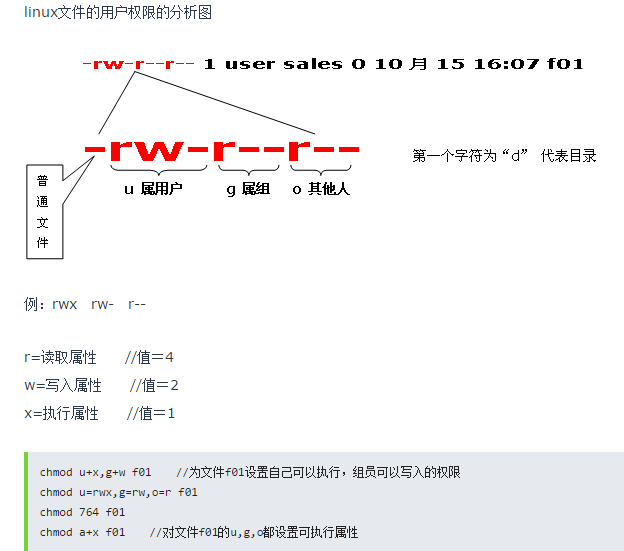
## **df命令**

## 用于显示磁盘分区上的可使用的磁盘空间。默认显示单位为KB。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。 **修改目录，文件权限和属主及数组命令**

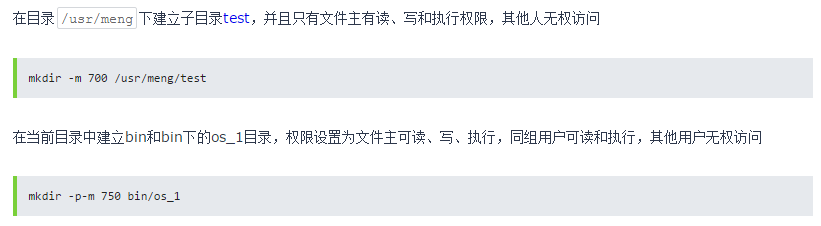
Chown：用于改变文件拥有属性的命令只有文件主和超级用户才可以便用该命令  
  


## -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 **创建和删除目录的命令**

Chmod：用于改变指定目录或文件的权限命令



   Mkdir：用于创建目录



-m<目标属性>或--mode<目标属性>建立目录的同时设置目录的权限； -p或--parents 若所要建立目录的上层目录目前尚未建立，则会一并建立上层目录；  
  
Rmdir：用于删除空的目录

rmdir -p bin/os\_1  
  
-p或--parents：删除指定目录后，若该目录的上层目录已变成空目录，则将其一并删除；

   Rm -f：用于删除不为空的目录

删除当前目录下除隐含文件外的所有文件和子目录

# rm -r \*

**创建和删除，重命名，复制文件的命令**

Touch：创建一个新的文件

touch ex2

在当前目录下建立一个空文件ex2，然后，利用ls -l命令可以发现文件ex2的大小为0，表示它是空文件。

 Vi:创建一个新的文件

 Rm：删除文件或目录

 Mv：重命名或移动文件的命令

将文件ex3改名为new1

mv ex3 new1

将目录/usr/men中的所有文件移到当前目录（用.表示）中：

mv /usr/men/\* .

 Cp：复制命令

如果把一个文件复制到一个目标文件中，而目标文件已经存在，那么，该目标文件的内容将被破坏。此命令中所有参数既可以是绝对路径名，也可以是相对路径名。通常会用到点.或点点..的形式。例如，下面的命令将指定文件复制到当前目录下： cp ../mary/homework/assign .

所有目标文件指定的目录必须是己经存在的，cp命令不能创建目录。如果没有文件复制的权限，则系统会显示出错信息。 将文件file复制到目录/usr/men/tmp下，并改名为file1

cp file /usr/men/tmp/file1

将目录/usr/men下的所有文件及其子目录复制到目录/usr/zh中

cp -r /usr/men /usr/zh  
  
Scp：用于将本地的文件或目录复制到远程服务器

从远处复制到本地

scp -r root@10.10.10.10:/opt/soft/mongodb /opt/soft/

从10.10.10.10机器上的/opt/soft/中下载mongodb目录到本地的/opt/soft/目录来。  
  
Wget：用于下载ftp或http服务器文件到本地。

wget命令用来从指定的URL下载文件。wget非常稳定，它在带宽很窄的情况下和不稳定网络中有很强的适应性，如果是由于网络的原因下载失败，wget会不断的尝试，直到整个文件下载完毕。如果是服务器打断下载过程，它会再次联到服务器上从停止的地方继续下载。这对从那些限定了链接时间的服务器上下载大文件非常有用。  
使用wget下载单个文件

wget http://www.linuxde.net/testfile.zip

## 以下的例子是从网络下载一个文件并保存在当前目录，在下载的过程中会显示进度条，包含（下载完成百分比，已经下载的字节，当前下载速度，剩余下载时间） **显示文件内容的命令**

 Cat：用于显示指定文件的全部内容

   More：用分页的形式显示指定文件的内容

   Less：用分页的形式显示指定文件的内容，区别是more和less翻页使用的操作键不同。

   Head：用于显示文件的前n行内容。

   Tail：用于显示文件的后n行内容。

   Tail -f：用于自动刷新的显示文件后n行数据内容。

**查找命令**

Find：查找指定目录或文件的命令。

在/home目录下查找以.txt结尾的文件名

find /home -name "\*.txt"

同上，但忽略大小写 find /home -iname "\*.txt"

当前目录及子目录下查找所有以.txt和.pdf结尾的文件

find . \( -name "\*.txt" -o -name "\*.pdf" \) 或

find . -name "\*.txt" -o -name "\*.pdf"

匹配文件路径或者文件 find /usr/ -path "\*local\*"

基于正则表达式匹配文件路径 find . -regex ".\*\(\.txt\|\.pdf\)$"  
  
Whereis：查找指定的文件源,源代码文件,和二进制文件和手册等

和find相比，whereis查找的速度非常快，这是因为linux系统会将 系统内的所有文件都记录在一个数据库文件中  
[root@localhost ~]# whereis svn  
  
svn安装找出了很多相关文件 只将二进制文件查找出来

[root@localhost ~]# whereis -b svn svn: /usr/bin/svn /usr/local/svn

[root@localhost ~]# whereis -m svn svn: /usr/share/man/man1/svn.1.gz

[root@localhost ~]# whereis -s svn svn:

说明：whereis -m svn查出说明文档路径，whereis -s svn找source源文件

  Which：用于查询命令或别名的位置。

查找文件、显示命令路径：

[root@localhost ~]# which pwd /bin/pwd

[root@localhost ~]# which adduser /usr/sbin/adduser

说明：which是根据使用者所配置的 PATH 变量内的目录去搜寻可运行档的！所以，不同的 PATH 配置内容所找到的命令当然不一样的！

  Locate：快速查找系统[数据库](http://www.51testing.com/html/60/javascript:;" \t "http://www.51testing.com/html/60/_self)中指定的内容。

查找字符串：要查找的文件名中含有的字符串。

实例

搜索etc目录下所有以sh开头的文件： locate /etc/sh

搜索用户主目录下，所有以m开头的文件： locate ~/m

搜索用户主目录下，所有以m开头的文件，并且忽略大小写： locate -i ~/m  
  
 Grep：在指定的文件或标准输出，标准输入内，查找满足条件的内容。使用正则表达式搜索文本，并把匹配的行打印出来  
grep match\_pattern file\_name

grep "match\_pattern" file\_name  
统计文件或者文本中包含匹配字符串的行数 -c 选项：

grep -c "text" file\_name

输出包含匹配字符串的行数 -n 选项：

grep "text" -n file\_name 或 cat file\_name | grep "text" -n

## #多个文件 grep "text" -n file\_1 file\_2 **关机和重启计算机的命令**

 Shutdown：-r 关机后立即重启

             -k 并不真正的关机，而只是发出警告信息给所有用户

             -h 关机后不重新启动

  Poweroff：用于关机和关闭电源

  Init：改变系统运行级别

        0级用于关闭系统

        1 级用于单一使用者模式

        2级用来进行多用户使用模式（但不带网络功能）

        3级用来进行多用户使用模式（带网络全功能）

        4级用来进行用户自定义使用模式

        5级表示进入x  windows时的模式

        6级用来重启系统

  Reboot： 用于计算机重启

  Halt：用于关闭计算机系统

## **压缩和打包命令**

 Tar：用于多个文件或目录进行打包，但不压缩，同时也用命令进行解包

 Gzip：用于文件进行压缩和解压缩命令，文件扩展名为.gz结尾。

Gunzip：用于对gzip压缩文档进行解压缩。

Zip/unzip:压缩解压缩.zip文件

## **用户操作命令**

Su：切换用户命令

   Sudo：以系统管理员的身份执行命令

   Passwd：用于修改用户的密码

**文件连接命令**

Ln：为源文件创建一个连接，并不将源文件复制一份，即占用的空间很小。

        可以分为软件连接和硬链接。

        软连接：也称为符号连接，即为文件或目录创建一个快捷方式。

硬链接：给一个文件取多于一个名字，放在不同目录中，方便用户使用。

Ln命令参数如下：

   -f：在创建连接时，先将与目的对象同名的文件或目录删除。

   -d：允许系统管理者硬链接自己的目录。

   -i：在删除与目的对象同名文件或目录时先询问用户。

   -n：在创建软连接时，将目的对象视为一般的文件。

   -s：创建软连接，即符号连接。

   -v：在连接之前显示文件或目录名。

   -b：将在连接时会被覆盖或删除的文件进行备份。

**帮助命令-----man**

## [其他](http://www.51testing.com/html/60/javascript.:;" \t "http://www.51testing.com/html/60/_self)**命令**

Who：显示系统中有那些用户在使用。

        -ami  显示当前用户

        -u：显示使用者的动作/工作

        -s：使用简短的格式来显示

        -v：显示程序版本

   Free：查看当前系统的内存使用情况

   Uptime：显示系统运行了多长时间

   Ps：显示瞬间进程的动态

   Pstree：以树状方式显示系统中所有的进程

   Date：显示或设定系统的日期与时间。

   Last：显示每月登陆系统的用户信息

   Kill： 杀死一些特定的进程

   Logout：退出系统

   Useradd/userdel:添加用户/删除用户

   Clear：清屏

   Passwd：设置用户密码

**vi编辑器**

   首先用vi命令打开一个文件

末行模式命令：

   :n,m w path/filename 保存指定范围文档（ n表开始行，m表结束行）

   :q!    对文件做过修改后，强制退出

   :q     没有对文件做过修改退出

   Wq或x  保存退出

**dd   删除光标所在行**

**： set number 显示行号**

**：n 跳转到n行**

**：s  替换字符串 :s/test/test2/g  /g全局替换 /也可以用%代替**

**/ 查找字符串**

Linux查看进程和杀死进程

用ps查看进程

-e 显示所有进程,环境变量  
-f 全格式

$ ps -ef

smx       1822     1  0 11:38 ?        00:00:49 gnome-terminal  
smx       1823  1822  0 11:38 ?        00:00:00 gnome-pty-helper  
smx       1824  1822  0 11:38 pts/0    00:00:02 bash  
smx       1827     1  4 11:38 ?        00:26:28 /usr/lib/firefox-3.6.18/firefox-bin  
smx       1857  1822  0 11:38 pts/1    00:00:00 bash  
smx       1880  1619  0 11:38 ?        00:00:00 update-notifier  
……  
smx      11946  1824  0 21:41 pts/0    00:00:00 ps -ef

或者：

$ ps -aux（  -aux 显示所有状态）

smx       1822  0.1  0.8  58484 18152 ?        Sl   11:38   0:49 gnome-terminal  
smx       1823  0.0  0.0   1988   712 ?        S    11:38   0:00 gnome-pty-helper  
smx       1824  0.0  0.1   6820  3776 pts/0    Ss   11:38   0:02 bash  
smx       1827  4.3  5.8 398196 119568 ?       Sl   11:38  26:13 /usr/lib/firefox-3.6.18/firefox-bin  
smx       1857  0.0  0.1   6688  3644 pts/1    Ss   11:38   0:00 bash  
smx       1880  0.0  0.6  41536 12620 ?        S    11:38   0:00 update-notifier  
……  
smx      11953  0.0  0.0   2716  1064 pts/0    R+   21:42   0:00 ps -aux

此时如果我想杀了火狐的进程就在终端输入：

$ kill -s 9 1827

其中-s 9 制定了传递给进程的信号是９，即强制、尽快终止进程

1827则是上面ps查到的火狐的PID。

简单吧，但有个问题，进程少了则无所谓，进程多了，就会觉得痛苦了，无论是ps -ef 还是ps -aux，每次都要在一大串进程信息里面查找到要杀的进程，看的眼都花了。

***进阶篇：***

**改进１**：

把ps的查询结果通过管道给grep查找包含特定字符串的进程。管道符“|”用来隔开两个命令，管道符左边命令的输出会作为管道符右边命令的输入。

$ ps -ef | grep firefox  
smx       1827     1  4 11:38 ?        00:27:33 /usr/lib/firefox-3.6.18/firefox-bin  
smx      12029  1824  0 21:54 pts/0    00:00:00 grep --color=auto firefox

这次就清爽了。然后就是

$kill -s 9 1827

还是嫌打字多？

**改进２——使用pgrep**：

一看到pgrep首先会想到什么？没错，grep！pgrep的p表明了这个命令是专门用于进程查询的grep。

$ pgrep firefox  
1827

看到了什么？没错火狐的PID，接下来又要打字了：

$kill -s 9 1827