

# Linux常用命令

## 文件管理命令

### ls命令

ls

ls 命令是Linux下最常用的指令之一。ls命令为英文单词 list 的缩写，正如英文单词 list 的意思，其功能是列出指定目录下的内容及其相关属性信息。

默认状态下，ls命令会列出当前目录的内容。而带上参数后，我们可以用ls做更多的事情。作为最基础同时又是使用频率很高的命令，我们很有必要搞清楚ls命令的用法，那么接下来一起看看吧！

语法格式: ls [选项] [文件]

常用参数：

-a

显示所有文件及目录 (包括以“.”开头的隐藏文件)

-l

使用长格式列出文件及目录信息

-r

将文件以相反次序显示(默认依英文字母次序)

-t

根据最后的修改时间排序

-A

同 -a，但不列出“.”(当前目录)及“..”(父目录)

-S

根据文件大小排序

-R

递归列出所有子目录

参考实例

```
# 列出所有文件(包括隐藏文件):
[root@dsjprs ~]# ls -a
# 列出文件的详细信息:
[root@dsjprs ~]# ls -l
# 列出根目录(/)下的所有目录:
[root@dsjprs ~]# ls /
# 列出当前工作目录下所有名称是“s”开头的文件 :
[root@dsjprs ~]# ls -ltr s*
# 列出 /bin 目录下的所有目录及文件的详细信息 :
[root@dsjprs ~]# ls -lR /bin
# 列出当前工作目录下所有文件及目录并以文件的大小进行排序 :
[root@dsjprs ~]# ls -AS
```

## cp命令

cp命令可以理解为英文单词copy的缩写，其功能为复制文件或目录。

cp命令可以将多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，也可以同时复制多个文件到一个指定的目录中。

语法格式：cp [参数] [文件]

常用参数：

-f

若目标文件已存在，则会直接覆盖原文件

-i

若目标文件已存在，则会询问是否覆盖

-p

保留源文件或目录的所有属性

-r

递归复制文件和目录

-d

当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录

-l

对源文件建立硬连接，而非复制文件

-s

对源文件建立符号连接，而非复制文件

-b

覆盖已存在的文件目标前将目标文件备份

-v

详细显示cp命令执行的操作过程

-a

等价于“-dpr”选项

参考实例

```
# 复制目录：
[root@dsjprs ~]# cp -R dir1 dir2/
# 将文件test1改名为test2：
[root@dsjprs ~]# cp -f test1 test2
# 复制多个文件：
[root@dsjprs ~]# cp -r file1 file2 file3 dir
# 交互式地将目录 /usr/dsjprs 中的所有.c文件复制到目录 dir 中：
[root@dsjprs ~]# cp -r /usr/dsjprs/*.c dir
```

## mkdir命令

mkdir命令是“make directories”的缩写，用来创建目录。

注意：默认状态下，如果要创建的目录已经存在，则提示已存在，而不会继续创建目录。所以在创建目录时，应保证新建的目录与它所在目录下的文件没有重名。mkdir命令还可以同时创建多个目录，是不是很强大呢？

语法格式：mkdir [参数] [目录]

常用参数：

-p

递归创建多级目录

-m

建立目录的同时设置目录的权限

-z

设置安全上下文

-v

显示目录的创建过程

参考实例

在工作目录下，建立一个名为 dir 的子目录：

```
[root@dsjprs ~]# mkdir dir
# 在目录/usr/dsjprs下建立子目录dir，并且设置文件属主有读、写和执行权限，其他人无权访问
[root@dsjprs ~]# mkdir -m 700 /usr/dsjprs/dir
# 同时创建子目录dir1, dir2, dir3:
[root@dsjprs ~]# mkdir dir1 dir2 dir3
# 递归创建目录:
[root@dsjprs ~]# mkdir -p dsjprs/dir
```

## mv命令

mv命令是“move”单词的缩写，其功能大致和英文含义一样，可以移动文件或对其改名。

这是一个使用频率超高的文件管理命令，我们需要特别留意它与复制的区别：mv与cp的结果不同。mv命令好像文件“搬家”，文件名称发生改变，但个数并未增加。而cp命令是对文件进行复制操作，文件个数是有增加的。

语法格式：mv [参数]

常用参数：

-i

若存在同名文件，则向用户询问是否覆盖

-f

覆盖已有文件时，不进行任何提示

-b

当文件存在时，覆盖前为其创建一个备份

-u

当源文件比目标文件新，或者目标文件不存在时，才执行移动此操作

参考实例

```
# 将文件file_1重命名为file_2:
[root@dsjprs ~]# mv file_1 file_2
# 将文件file移动到目录dir中：
[root@dsjprs ~]# mv file /dir
# 将目录dir1移动目录dir2中（前提是目录dir2已存在，若不存在则改名):
[root@dsjprs ~]# mv /dir1 /dir2
# 将目录dir1下的文件移动到当前目录下:
[root@dsjprs ~]# mv /dir1/* .
```

## pwd命令

pwd命令是“print working directory”中每个单词的首字母缩写，其功能正如所示单词一样，为打印工作目录，即显示当前工作目录的绝对路径。

在实际工作中，我们经常会不同目录之间进行切换，为了防止“迷路”，我们可以使用pwd命令快速查看当前我们所在的目录路径。

语法格式：pwd [参数]

常用参数：

-L

显示逻辑路径

参考实例

查看当前工作目录路径：

```
[root@dsjprs ~]# pwd
/home/dsjprs
```

# 文档编辑命令

## cat命令

Linux系统中有很多个用于查看文件内容的命令，每个命令又都有自己的特点，比如这个cat命令就是用于查看内容较少的纯文本文件的。cat这个命令也很好记，因为cat在英语中是“猫”的意思，小猫咪是不是给您一种娇小、可爱的感觉呢？

注意：当文件内容较大时，文本内容会在屏幕上快速闪动（滚屏），用户往往看不清所显示的具体内容。因此对于较长文件内容可以按Ctrl+S键，停止滚屏；以及Ctrl+Q键可以恢复滚屏；而按Ctrl+C（中断）键则可以终止该命令的执行。或者对于大文件，干脆用more命令吧！

语法格式：cat [参数] [文件]

常用参数：

-n

显示行数（空行也编号）

-s

显示行数（多个空行算一个编号）

-b

显示行数（空行不编号）

-E

每行结束处显示\$符号

-T

将TAB字符显示为^I符号

-v

使用^和M-引用，除了LFD和TAB之外

-e

等价于“-vE”组合

-t

等价于“-vT”组合

-A

等价于 -vET组合

--help

显示帮助信息

--version

显示版本信息

参考实例

查看文件的内容：

```
[root@dsjprs ~]# cat filename.txt
# 查看文件的内容，并显示行数编号：
[root@dsjprs ~]# cat -n filename.txt
# 查看文件的内容，并添加行数编号后输出到另外一个文件中：
[root@dsjprs ~]# cat -n dsjprs.log > dsjprsbe.log
# 清空文件的内容：
[root@dsjprs ~]# cat /dev/null > /root/filename.txt
# 持续写入文件内容，碰到EOF符后结束并保存：
[root@dsjprs ~]# cat > filename.txt <<EOF

Hello, World
Linux!
EOF
# 将软盘设备制作成镜像文件：
[root@dsjprs ~]# cat /dev/fd0 > fdisk.iso
```

## echo命令

echo命令用于在终端设备上输出字符串或变量提取后的值，这是在Linux系统中最常用的几个命令之一，但操作却非常简单。

人们一般使用在变量前加上\$符号的方式提取出变量的值，例如：\$PATH，然后再用echo命令予以输出。或者直接使用echo命令输出一段字符串到屏幕上，起到给用户提示的作用。

语法格式：echo [参数] [字符串]

常用参数：

- n 不输出结尾的换行符
- e "\a" 发出警告音
- e "\b" 删除前面的一个字符
- e "\c" 结尾不加换行符
- e "\f" 换行，光标仍停留在原来的坐标位置
- e "\n" 换行，光标移至行首
- e "\r" 光标移至行首，但不换行
- E 禁止反斜杠转移，与-e参数功能相反

```
-version
# 查看版本信息
--help
# 查看帮助信息
# 参考实例
# 输出一段字符串：
[root@dsjprs ~]# echo "dsjprs.com"
dsjprs.com
# 输出变量提取后的值：
[root@dsjprs ~]# echo $PATH
/
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
# 对内容进行转义，不让$符号的提取变量值功能生效：
[root@dsjprs ~]# echo \$PATH
$PATH
# 结合输出重定向符，将字符串信息导入文件中：
[root@dsjprs ~]# echo "It is a test" > dsjprs
# 使用反引号执行命令，并输出其结果到终端：
[root@dsjprs ~]# echo `date`
# 输出带有换行符的内容：
[root@dsjprs ~]# echo -e "a\nb\nc"
a
a
b
b
c
c
# 输出信息中删除某个字符，注意看数字3消失了：
[root@dsjprs ~]# echo -e "123\b456"
12456
```

## rm命令

rm是常用的命令，该命令的功能为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均删除。对于链接文件，只是删除了链接，原有文件均保持不变。rm也是一个危险的命令，使用的时候要特别当心，尤其对于新手，否则整个系统就会毁在这个命令（比如在/（根目录）下执行rm \* -rf）。所以，我们在执行rm之前最好先确认一下在哪个目录，到底要删除什么东西，操作时保持高度清醒的头脑。

语法格式：rm [参数] [文件]

常用参数：

- f

忽略不存在的文件，不会出现警告信息

-i

删除前会询问用户是否操作

-r/R

递归删除

-v

显示指令的详细执行过程

参考实例

删除前逐一询问确认：

```
[root@dsjprs ~]# rm -i test.txt.bz2
r
rm: remove regular file `test.txt.bz2'?
# 直接删除，不会有任何提示：
[root@dsjprs ~]# rm -f test.txt.bz2
# 递归删除目录及目录下所有文件：
[root@dsjprs ~]# mkdir /data/log
[
[root@dsjprs ~]# rm -rf /data/log
# 删除当前目录下所有文件：
[root@dsjprs ~]# rm -rf *
# 清空系统中所有的文件（谨慎）：
[root@dsjprs ~]# rm -rf /*
```

## tail命令

tail用于显示文件尾部的内容，默认在屏幕上显示指定文件的末尾10行。如果给定的文件不止一个，则在显示的每个文件前面加一个文件名标题。如果没有指定文件或者文件名为“-”，则读取标准输入。

语法格式：tail [参数]

常用参数：

--retry

即是在tail命令启动时，文件不可访问或者文件稍后变得不可访问，都始终尝试打开文件。使用此选项时需要与选项“——follow=name”连用

-c或——bytes=

输出文件尾部的N（N为整数）个字节内容

-f<name/descriptor>

--follow：显示文件最新追加的内容

-F

与选项“-follow=name”和“--retry”连用时功能相同

-n或——line=

输出文件的尾部N（N位数字）行内容

--pid=<进程号>

与“-f”选项连用，当指定的进程号的进程终止后，自动退出tail命令

--help

显示指令的帮助信息

--version

显示指令的版本信息

参考实例

显示文件file的最后10行：

```
[root@dsjprs ~] tail file
# 显示文件file的内容，从第20行至文件末尾：
[root@dsjprs ~] tail +20 file
# 显示文件file的最后10个字符：
[root@dsjprs ~] tail -c 10 file
# 一直变化的文件总是显示后10行：
[root@dsjprs ~] tail -f 10 file
# 显示帮助信息：
[root@dsjprs ~] tail --help
```

## rmmdir命令

rmmdir命令作用是删除空的目录，英文全称：“remove directory”。

注意：rmmdir命令只能删除空目录。当要删除非空目录时，就要使用带有“-R”选项的rm命令。

rmmdir命令的“-p”参数可以递归删除指定的多级目录，但是要求每个目录也必须是空目录。

语法格式：rmmdir [参数] [目录名称]

常用参数：

-p

用递归的方式删除指定的目录路径中的所有父级目录，非空则报错

-- -- ignore-fail-on-non-empty

忽略由于删除非空目录时导致命令出错而产生的错误信息

-v

显示命令的详细执行过程

-- -- help

显示命令的帮助信息

-- -- version

显示命令的版本信息

参考实例

删除空目录：

```
[root@dsjprs ~]# rmmdir dir
# 递归删除指定的目录树：
[root@dsjprs ~]# rmmdir -p dir/dir_1/dir_2
# 显示指令详细执行过程：
[root@dsjprs ~]# rmmdir -v dir
rmmdir: 正在删除目录 'dir'
[root@dsjprs ~]# rmmdir -p -v dir/dir_1/dir_2
rmmdir: 正在删除目录 'dir/dir_1/dir_2'
rmmdir: 正在删除目录 'dir/dir_1'
rmmdir: 正在删除目录 'dir_2'
# 显示命令的版本信息：
[root@dsjprs ~]# rmmdir --version
r
rmmdir (GNU coreutils) 8.30
C
Copyright (C) 2018 Free Software Foundation, Inc.

许可证 GPLv3+: GNU 通用公共许可证第 3 版或更新版本https://gnu.org/licenses/gpl.html。

本软件是自由软件：您可以自由修改和重新发布它。

在法律范围内没有其他保证。
```

# 系统管理命令

## rpm命令

rpm命令是Red-Hat Package Manager（RPM软件包管理器）的缩写，该命令用于管理Linux下软件包的软件。在Linux操作系统下，几乎所有的软件均可以通过RPM进行安装、卸载及管理等操作。

概括的说，rpm命令包含了五种基本功能：安装、卸载、升级、查询和验证。

语法格式：rpm [参数] [软件包]

常用参数：

-a

查询所有的软件包

-b或-t

设置包装套件的完成阶段，并指定套件档的文件名称；

-c

只列出组态配置文件，本参数需配合“-l”参数使用

-d

只列出文本文件，本参数需配合“-l”参数使用

-e或--erase

卸载软件包

-f

查询文件或命令属于哪个软件包

-h或--hash

安装软件包时列出标记

-i

显示软件包的相关信息

--install

安装软件包

-l

显示软件包的文件列表

-p

查询指定的rpm软件包

-q

查询软件包

-R

显示软件包的依赖关系

-s

显示文件状态，本参数需配合“-l”参数使用

-U或--upgrade

升级软件包

-v

显示命令执行过程

-vv

详细显示指令执行过程

参考实例

直接安装软件包：



```
[root@dsjprs ~]# rpm -ivh packge.rpm
# 忽略报错，强制安装：
[root@dsjprs ~]# rpm --force -ivh package.rpm
# 列出所有安装过的包：
[root@dsjprs ~]# rpm -qa
# 查询rpm包中的文件安装的位置：
[root@dsjprs ~]# rpm -ql ls
# 卸载rpm包：
[root@dsjprs ~]# rpm -e package.rpm
# 升级软件包：
[root@dsjprs ~]# rpm -U file.rpm
```

## find命令

find命令可以根据给定的路径和表达式查找的文件或目录。find参数选项很多，并且支持正则，功能强大。和管道结合使用可以实现复杂的功能，是系统管理者和普通用户必须掌握的命令。

find如不加任何参数，表示查找当前路径下的所有文件和目录，如果服务器负载比较高尽量不要在高峰期使用find命令，find命令模糊搜索还是比较消耗系统资源的。

语法格式：find [参数] [路径] [查找和搜索范围]

常用参数：

-name

按名称查找

-size

按大小查找

-user

按属性查找

-type

按类型查找

-iname

忽略大小写

参考实例

使用-name参数查看/etc目录下面所有的.conf结尾的配置文件：

```
[root@dsjprs ~]# find /etc -name "*.conf"
# 使用-size参数查看/etc目录下面大于1M的文件：
[root@dsjprs ~]# find /etc -size +1M
# 查找当前用户主目录下的所有文件：
[root@dsjprs ~]# find $HOME -print
# 列出当前目录及子目录下所有文件和文件夹：
[root@dsjprs ~]# find .
# 在/home目录下查找以.txt结尾的文件名：
[root@dsjprs ~]# find /home -name "*.txt"
# 在/var/log目录下忽略大小写查找以.log结尾的文件名：
[root@dsjprs ~]# find /var/log -iname "*.log"
# 搜索超过七天内被访问过的所有文件：
[root@dsjprs ~]# find . -type f -atime +7
# 搜索访问时间超过10分钟的所有文件：
[root@dsjprs ~]# find . -type f -amin +10
# 找出/home下不是以.txt结尾的文件：
[root@dsjprs ~]# find /home ! -name "*.txt"
```

## startx命令

startx命令用来启动X-Window，它负责调用X-Window系统的初始化程序xinit。以完成X-Window运行所必要的初始化工作，并启动X-Window系统。

语法格式：startx [参数]

常用参数：

-d

指定在启动过程中传递给客户机的X服务器的显示名称

-m

当未找到启动脚本时，启动窗口管理器

-r

当未找到启动脚本时，装入资源文件

-w

强制启动

-x

使用startup脚本启动X-windows会话

参考实例

已默认方式启动X-windows系统：

```
[root@dsjprs ~]# startx
# 以16位颜色深度启动X-windows系统:
[root@dsjprs ~]# startx -- -depth 16
# 强制启动 X-windows系统:
[root@dsjprs ~]# startx -w
```

## uname命令

uname命令的英文全称即“Unix name”。

用于显示系统相关信息，比如主机名、内核版本号、硬件架构等。

如果未指定任何选项，其效果相当于执行“uname -s”命令，即显示系统内核的名字。

语法格式：uname [参数]

常用参数：

-a

显示系统所有相关信息

-m

显示计算机硬件架构

-n

显示主机名称

-r

显示内核发行版本号

-s

显示内核名称

-v

显示内核版本

-p

显示主机处理器类型

-o

显示操作系统名称

-i

显示硬件平台

参考实例

显示系统主机名、内核版本号、CPU类型等信息：

```
[root@dsjprs ~]# uname -a
Linux dsjprs 3.10.0-123.el7.x86_64 #1 SMP Mon May 5 11:16:57 EDT 2014 x86_64 x86_64 x86_64
GNU/Linux
# 仅显示系统主机名:
[root@dsjprs ~]# uname -n
dsjprs
# 显示当前系统的内核版本 :
[root@dsjprs ~]# uname -r
3.10.0-123.el7.x86_64
# 显示当前系统的硬件架构:
[root@dsjprs ~]# uname -i
x86_64
```

## vmstat命令

vmstat命令的含义为显示虚拟内存状态（“Virtual Memory Statistics”），但是它可以报告关于进程、内存、I/O等系统整体运行状态。

语法格式：vmstat [参数]

常用参数：

-a

显示活动内页

-f

显示启动后创建的进程总数

-m

显示slab信息

-n

头信息仅显示一次

-s

以表格方式显示事件计数器和内存状态

-d

报告磁盘状态

-p

显示指定的硬盘分区状态

-S

输出信息的单位

参考实例

显示活动内页：

```
[root@dsjprs ~]# vmstat -a
显示启动后创建的进程总数:
[root@dsjprs ~]# vmstat -f
显示slab信息:
[root@dsjprs ~]# vmstat -m
头信息仅显示一次:
[root@dsjprs ~]# vmstat -n
以表格方式显示事件计数器和内存状态:
[root@dsjprs ~]# vmstat -s
显示指定的硬盘分区状态:
[root@dsjprs ~]# vmstat -p /dev/sda1
指定状态信息刷新的时间间隔为1秒:
[root@dsjprs ~]# vmstat 1
```

# 磁盘管理命令

## df命令

df命令的英文全称即“Disk Free”，顾名思义功能是为了显示系统上可使用的磁盘空间。默认显示单位为KB，建议使用“df -h”的参数组合，根据磁盘容量自动变换合适的单位，更利于阅读。

日常普遍用该命令可以查看磁盘被占用了多少空间、还剩多少空间等信息。

语法格式：df [参数] [指定文件]

常用参数：

-a

显示所有系统文件

-B <块大小>

指定显示时的块大小

-h

以容易阅读的方式显示

-H

以1000字节为换算单位来显示

-i

显示索引字节信息

-k

指定块大小为1KB

-l

只显示本地文件系统

-t <文件系统类型>

只显示指定类型的文件系统

-T

输出时显示文件系统类型

--sync

在取得磁盘使用信息前，先执行sync命令

参考实例

显示磁盘分区使用情况：

```
[root@dsjprs ~]# df
```

文件系统	1K-块	已用	可用	已用%	挂载点
devtmpfs	1980612	0	1980612	0%	/dev
tmpfs	1994756	0	1994756	0%	/dev/shm
tmpfs	1994756	1040	1993716	1%	/run
tmpfs	1994756	0	1994756	0%	/sys/fs/cgroup
/dev/mapper/fedora_linuxhell-root	15718400	2040836	13677564	13%	/
tmpfs	1994756	4	1994752	1%	/tmp
/dev/sda1	999320	128264	802244	14%	/boot
tmpfs	398948	0	398948	0%	/run/user/0

# 以容易阅读的方式显示磁盘分区使用情况：

```
[root@dsjprs ~]# df -h
```

文件系统	容量	已用	可用	已用%	挂载点
devtmpfs	1.9G	0	1.9G	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	2.0G	1.1M	2.0G	1%	/run
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/mapper/fedora_linuxhell-root	15G	2.0G	14G	13%	/
tmpfs	2.0G	4.0K	2.0G	1%	/tmp
/dev/sda1	976M	126M	784M	14%	/boot
tmpfs	390M	0	390M	0%	/run/user/0

# 显示指定文件所在分区的磁盘使用情况：

```
[root@dsjprs ~]# df /etc/dhcp
```

文件系统	1K-块	已用	可用	已用%	挂载点
/dev/mapper/fedora_linuxcool-root	15718400	2040836	13677564	13%	/

```
# 显示文件类型为ext4的磁盘使用情况:
```

```
[root@dsjprs ~]# df -t ext4
```

文件系统	1K-块	已用	可用	已用%	挂载点
/dev/sda1	999320	128264	802244	14%	/boot

## fdisk命令

fdisk命令的英文全称是“Partition table manipulator for Linux”，即作为磁盘的分区工具。进行硬盘分区从实质上说就是对硬盘的一种格式化，用一个形象的比喻，分区就好比在一张白纸上画一个大方框，而格式化好比在方框里打上格子。

语法格式：fdisk [参数]

常用参数：

-b

指定每个分区的大小

-l

列出指定的外围设备的分区表状况

-s

将指定的分区大小输出到标准输出上，单位为区块

-u

搭配“-l”参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址

-v

显示版本信息

参考实例

查看所有分区情况：

```
[root@dsjprs ~]# fdisk -l
```

```
# 选择分区磁盘:
```

```
[root@dsjprs ~]# fdisk /dev/sdb
```

```
# 在当前磁盘上建立扩展分区:
```

```
[root@dsjprs ~]# fdisk /ext
```

```
# 不检查磁盘表面加快分区操作:
```

```
[root@dsjprs ~]# fdisk /actok
```

```
# 重建主引导记录:
```

```
[root@dsjprs ~]# fdisk /cmbr
```

## lsblk命令

lsblk命令的英文是“list block”，即用于列出所有可用块设备的信息，而且还能显示他们之间的依赖关系，但是它不会列出RAM盘的信息。

lsblk命令包含在util-linux-ng包中，现在该包改名为util-linux。

语法格式：lsblk [参数]

常用参数：

-a

显示所有设备

-b

以bytes方式显示设备大小

-d

不显示 slaves 或 holders

-D

print discard capabilities

-e

排除设备

-f

显示文件系统信息

-h

显示帮助信息

-i

use ascii characters only

-m

显示权限信息

-l

使用列表格式显示

-n

不显示标题

-o

输出列

-P

使用key="value"格式显示

-r

使用原始格式显示

-t

显示拓扑结构信息

参考实例

lsblk命令默认情况下将以树状列出所有块设备：

```
[root@dsjprs ~]# lsblk
lsblk NAME      MAJ:MIN  RM  SIZE RO type mountpoint
sda      8:0      0 232.9G  0 disk
├─sda1    8:1      0  46.6G  0 part /
├─sda2    8:2      0    1K  0 part
├─sda5    8:5      0   190M  0 part /boot
├─sda6    8:6      0    3.7G  0 part [SWAP]
├─sda7    8:7      0   93.1G  0 part /data
└─sda8    8:8      0   89.2G  0 part /personal
sr0     11:0      1  1024M  0 rom
```

默认选项不会列出所有空设备：

```
[root@dsjprs ~]# lsblk -a
# 也可以用于列出一个特定设备的拥有关系，同时也可以列出组和模式：
[root@dsjprs ~]# lsblk -m
# 要获取SCSI设备的列表，你只能使用-S选项，该选项是用来以颠倒的顺序打印依赖的：
[root@dsjprs ~]# lsblk -S
#例如，你也许想要以列表格式列出设备，而不是默认的树状格式。可以将两个不同的选项组合，以获得期望的输出：
[root@dsjprs ~]# lsblk -nl
```

## hdparm命令

hdparm命令用于检测，显示与设定IDE或SCSI硬盘的参数。

语法格式：hdparm [参数]

常用参数：

-a

设定读取文件时，预先存入块区的分区数

-f

将内存缓冲区的数据写入硬盘，并清空缓冲区

-g

显示硬盘的磁轨，磁头，磁区等参数

-I

直接读取硬盘所提供的硬件规格信息

-X

设定硬盘的传输模式

参考实例

显示硬盘的相关设置：

```
[root@dsjprs ~]# hdparm /dev/sda
/
/dev/sda:
I
IO_support = 0 (default 16-bit)
r
readonly = 0 (off)
r
readahead = 256 (on)
g
```

geometry = 19929 [柱面数] /255 [磁头数] /63 [扇区数] , sectors = 320173056 [总扇区数] , start = 0 [起始扇区数]

显示硬盘的柱面、磁头、扇区数：

```
[root@dsjprs ~]# hdparm -g /dev/sda
# 评估硬盘的读取效率：
[root@dsjprs ~]# hdparm -t /dev/sda
# 直接读取硬盘所提供的硬件规格信息：
[root@dsjprs ~]# hdparm -X /dev/sda
# 使IDE硬盘进入睡眠模式：
[root@dsjprs ~]# hdparm -Y /dev/sda
```

## vgextend

vgextend命令用于动态扩展LVM卷组，它通过向卷组中添加物理卷来增加卷组的容量。LVM卷组中的物理卷可以在使用vgcreate命令创建卷组时添加，也可以使用vgextend命令动态的添加。

语法格式：vgextend [参数]

常用参数：

-d

调试模式

-t

仅测试

参考实例

将物理卷/dev/sdb1加入卷组vglinuxprobe:

```
[root@dsjprs ~]# vgextend vglinuxprobe /dev/sdb1
```

## 文件传输命名

### tftp命令

tftp命令用于传输文件。ftp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。

语法格式：tftp [参数]

常用参数：

connect

连接到远程tftp服务器

mode

文件传输模式

put

上传文件

get

下载文件

quit

退出

verbose

显示详细的处理信息

trace

显示包路径

status

显示当前状态信息

binary

二进制传输模式

ascii

ascii 传送模式

rexmt

设置包传输的超时时间

timeout

设置重传的超时时间

help

帮助信息

?

帮助信息

参考实例

连接远程服务器"218.28.188.288":

```
[root@dsjprs ~]# tftp 218.28.188.288
远程下载file文件:
tftp> get file
g
getting from 218.28.188.288 to /dir
R
Recived 168236 bytes in 1.5 seconds[112157 bit/s]
退出tftp:
tftp> quit
```

## curl命令

curl命令是一个利用URL规则在shell终端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。

语法格式：curl [参数] [网址]

常用参数：

-O

把输出写到该文件中，保留远程文件的文件名

-u

通过服务端配置的用户名和密码授权访问

参考实例

将下载的数据写入到文件，必须使用文件的绝对地址：



```
[root@dsjprs ~]# curl https://www.dsjprs.com/abc.txt --silent -O
# 访问需要授权的页面时，可通过-u选项提供用户名和密码进行授权：
[root@dsjprs ~]# curl -u root https://www.dsjprsbe.com/
Enter host password for user 'root':
```

## fsck命令

fsck命令的英文全称是“filesystem check”，即检查文件系统的意，常用于检查并修复Linux文件系统的一些错误信息，操作文件系统需要先备份重要数据，以防丢失。

Linux fsck命令用于检查并修复Linux文件系统，可以同时检查一个或多个 Linux 文件系统；若系统掉电或磁盘发生问题，可利用fsck命令对文件系统进行检查。

语法格式：fsck [参数] [文件系统]

常用参数：

-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

-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

-t

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

参考实例

修复坏的分區文件系统：

```
[root@dsjprs ~]# fsck -t ext3 -r /usr/local
fsck from util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
fsck.ext3: Is a directory while trying to open /usr/local
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
e2fsck -b 8193
/usr/local: status 8, rss 1232, real 0.020288, user 0.002022, sys 0.005354
# 显示fsck系统安装的版本号：
[root@dsjprs ~]# fsck --version
fsck from util-linux 2.23.2
```

## ftpwho命令

ftpwho命令用于显示当前所有以FTP登入的用户会话信息。

执行该命令可得知当前用FTP登入系统的用户有哪些人，以及他们正在进行的操作。

语法格式：ftpwho [参数]

常用参数：

-h

显示帮助信息

-v

详细模式，输出更多信息

参考实例

查询当前正在登录FTP 服务器的用户：

```
[root@dsjprs ~]# ftpwho
# 在详细模式下，查询当前正在登录FTP 服务器的用户：
[root@dsjprs ~]# ftpwho -v
# 显示帮助信息：
[root@dsjprs ~]# ftpwho -h
```

## lprm命令

lprm命令的英文全称是“Remove jobs from the print queue”，意为用于删除打印队列中的打印任务。尚未完成的打印机工作会被放在打印机队列之中，这个命令可用来将未送到打印机的工作取消。

语法格式：lprm [参数] [任务编号]

常用参数：

-E

与打印服务器连接时强制使用加密

-P

指定接受打印任务的目标打印机

-U

指定可选的用户名

参考实例

将打印机hpprint中的第102号任务移除：

```
[root@dsjprs ~]# lprm -Phpprint 102
# 将第101号任务由预设打印机中移除：
[root@dsjprs ~]# lprm 101
```

## 网络通信命名

### ssh命令

ssh命令是openssh套件中的客户端连接工具，可以给予ssh加密协议实现安全的远程登录服务器，实现对服务器的远程管理。

语法格式：ssh [参数] [远程主机]

常用参数：

-1

强制使用ssh协议版本1

-2

强制使用ssh协议版本2

-4

强制使用IPv4地址

-6

强制使用IPv6地址

-A

开启认证代理连接转发功能

-a

关闭认证代理连接转发功能

-b<IP地址>

使用本机指定的地址作为对位连接的源IP地址

-C

请求压缩所有数据

-F<配置文件>

指定ssh指令的配置文件，默认的配置文件的“/etc/ssh/ssh\_config”

-f

后台执行ssh指令

-g

允许远程主机连接本机的转发端口

-i<身份文件>

指定身份文件（即私钥文件）

-l<登录名>

指定连接远程服务器的登录用户名

-N

不执行远程指令

-o<选项>

指定配置选项

-p<端口>

指定远程服务器上的端口

-q

静默模式，所有的警告和诊断信息被禁止输出

-X

开启X11转发功能

-x

关闭X11转发功能

-y

开启信任X11转发功能

参考实例

登录远程服务器：

```
[root@dsjprs ~]# ssh 202.102.240.88
# 用test用户连接远程服务器：
[root@dsjprs ~]# ssh -l test 202.102.220.88
# 查看分区列表：
[root@dsjprs ~]# ssh 202.102.220.88 /sbin/fdisk -l
# 强制使用ssh协议版本1：
[root@dsjprs ~]# ssh -1
# 开启认证代理连接转发功能：
[root@dsjprs ~]# ssh -A
```

## netstat命令

netstat 命令用于显示各种网络相关信息，如网络连接，路由表，接口状态 (Interface Statistics)，masquerade 连接，多播成员 (Multicast Memberships) 等等。

从整体上看，netstat的输出结果可以分为两个部分：一个是Active Internet connections，称为有源TCP连接，其中”Recv-Q”和”Send-Q”指%0A的是接收队列和发送队列。这些数字一般都应该是0。如果不是则表示软件包正在队列中堆积。这种情况只能在非常少的情况见到；另一个是Active UNIX domain sockets，称为有源Unix域套接口(和网络套接字一样，但是只能用于本机通信，性能可以提高一倍)。

语法格式：netstat [参数]

常用参数：

-a

显示所有连线中的Socket

-p

显示正在使用Socket的程序识别码和程序名称

-u

显示UDP传输协议的连线状况

-i

显示网络界面信息表单

-n

直接使用IP地址，不通过域名服务器

参考实例

显示详细的网络状况：

```
[root@dsjprs ~]# netstat -a
# 显示当前户籍UDP连接状况:
[root@dsjprs ~]# netstat -nu
# 显示UDP端口号的使用情况:
[root@dsjprs ~]# netstat -apu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
udp      0      0 0.0.0.0:bootpc 0.0.0.0:* 4000/dhclient
udp      0      0 localhost:323 0.0.0.0:* 3725/chronyd
udp6     0      0 localhost:323 [::]:* 3725/chronyd

显示网卡列表:
[root@dsjprs ~]# netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0 181864 0 0 0 141278 0 0 0 BMRU
lo 16436 0 3362 0 0 0 3362 0 0 0 LRU

# 显示组播组的关系:
[root@dsjprs ~]# netstat -g
IPv6/IPv4 Group Memberships Interface
RefCnt Group

-----

lo 1 ALL-SYSTEMS.MCAST.NET
eth0 1 ALL-SYSTEMS.MCAST.NET lo 1 ff02::1
eth0 1 ff02::1:ff0a:b0c eth0 1 ff02::1
```

## ping命令

ping命令主要用来测试主机之间网络的连通性，也可以用于。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。

不过值得注意的是：Linux系统下的ping命令与Windows系统下的ping命令稍有不同。Windows下运行ping命令一般会发出4个请求就结束运行该命令；而Linux下不会自动终止，此时需要我们按CTR+C终止或者使用-c参数为ping命令指定发送的请求数目。

语法格式：ping [参数] [目标主机]

常用参数：

-d

使用Socket的SO\_DEBUG功能

-c

指定发送报文的次数

-i

指定收发信息的间隔时间

-I

使用指定的网络接口送出数据包

-l

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

设置填满数据包的范本样式

-q

不显示指令执行过程

-R

记录路由过程

-s

设置数据包的大小

-t

设置存活数值TTL的大小

-v

详细显示指令的执行过程

参考实例

检测与linuxcool网站的连通性：

```
[root@dsjprs ~]# ping www.dsjprs.com
# 连续ping4次:
[root@dsjprs ~]# ping -c 4 www.dsjprs.com
# 设置次数为4, 时间间隔为3秒:
[root@dsjprs ~]# ping -c 4 -i 3 www.dsjprs.com
# 利用ping命令获取指定网站的IP地址:
[root@dsjprs ~]# ping -c 1 dsjprs.com | grep from | cut -d " " -f 4
220.181.57.216
```

## ifconfig命令

ifconfig命令的英文全称是“network interfaces configuring”，即用于配置和显示Linux内核中网络接口的网络参数。用ifconfig命令配置的网卡信息，在网卡重启后机器重启后，配置就不存在。要想将上述的配置信息永远的存的电脑里，那就要修改网卡的配置文件了。

语法格式：ifconfig [参数]

常用参数：

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

up

启动指定的网络设备

IP地址

指定网络设备的IP地址

参考实例

显示网络设备信息：

```
[root@dsjprs ~]# 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)
# 启动关闭指定网卡:
[root@dsjprs ~]# ifconfig eth0 down
[root@dsjprs ~]# ifconfig eth0 up
# 为网卡配置和删除IPv6地址:
[root@dsjprs ~]# ifconfig eth0 add 33ffe:3240:800:1005::2/64
[root@dsjprs ~]# ifconfig eth0 del 33ffe:3240:800:1005::2/64
# 用ifconfig修改MAC地址:
[root@dsjprs ~]# ifconfig eth0 down
[root@dsjprs ~]# ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE
[root@dsjprs ~]# ifconfig eth0 up
[root@dsjprs ~]# ifconfig eth1 hw ether 00:1D:1C:1D:1E
[root@dsjprs ~]# ifconfig eth1 up
# 配置IP地址:
[root@dsjprs ~]# ifconfig eth0 192.168.1.56
[root@dsjprs ~]# ifconfig eth0 192.168.1.56 netmask 255.255.255.0
[root@dsjprs ~]# ifconfig eth0 192.168.1.56 netmask 255.255.255.0 broadcast 192.168.1.255
```

## dhclient

dhclient命令的作用是：使用动态主机配置协议动态的配置网络接口的网络参数，也支持BOOTP协议。

语法格式：dhclient [参数] [网络接口]

常用参数：

-p

指定dhcp客户端监听的端口号（默认端口号86）

-d

总是以前台方式运行程序

-q

安静模式，不打印任何错误的提示信息

-r

释放ip地址

-n

不配置任何接口

-x

停止正在运行的DHCP客户端，而不释放当前租约，杀死现有的dhclient

-s

在获取ip地址之前指定DHCP服务器

-w

即使没有找到广播接口，也继续运行

参考实例

在指定网络接口上发出DHCP请求：

```
[root@dsjprs ~]# dhclient eth0
# 释放IP地址:
[root@dsjprs ~]# dhclient -r
K
Killed old client process
# 从指定的服务器获取ip地址:
[root@dsjprs ~]# dhclient -s 192.168.60.240
# 停止运行dhclient:
[root@dsjprs ~]# dhclient -x
K
Killed old client process
```

## 设备管理命令

### mount命令

mount命令用于加载文件系统到指定的加载点。此命令的最常用于挂载cdrom，使我们可以访问cdrom中的数据，因为你将光盘插入cdrom中，Linux并不会自动挂载，必须使用Linux mount命令来手动完成挂载。

语法格式：mount [参数]

常用参数：

-t

指定挂载类型

-l

显示已加载的文件系统列表

-h

显示帮助信息并退出

-V

显示程序版本

-n

加载没有写入文件“/etc/mtab”中的文件系统

-r

将文件系统加载为只读模式

-a

加载文件“/etc/fstab”中描述的所有文件系统

参考实例

查看版本：

```
[root@dsjprs ~]# mount -V
# 启动所有挂载:
[root@dsjprs ~]# mount -a
# 挂载 /dev/cdrom 到 /mnt:
[root@dsjprs ~]# mount /dev/cdrom /mnt
# 挂载nfs格式文件系统:
[root@dsjprs ~]# mount -t nfs /123 /mnt
# 挂载第一块盘的第一个分区到/etc目录 :
[root@dsjprs ~]# mount -t ext4 -o loop,default /dev/sda1 /etc
```

## MAKEDEV命令

MAKEDEV是一个脚本程序,用于在 /dev 目录下建立设备,通过这些设备文件可以访问位于内核的驱动程序。  
MAKEDEV 脚本创建静态的设备节点,通常位于/dev目录下。

语法格式: MAKEDEV [参数]

常用参数:

-v

显示出执行的每一个动作

-n

并不做真正的更新,只是显示一下它的执行效果

-d

删除设备文件

参考实例

显示出执行的每一个动作:

```
[root@dsjprs ~]# ./MAKEDEV -v update
# 删除设备:
[root@dsjprs ~]# ./MAKEDEV -d device
```

## setleds命令

setleds即是英文词组“set leds”的合并,翻译为中文就是设置LED灯。setleds命令用来设定键盘上方三个 LED 灯的状态。在 Linux 中,每一个虚拟主控台都有独立的设定。

这是一个十分神奇的命令,竟然可以通过命令来控制键盘的灯的状态。那么下面我一起来学习一下这个命令吧。

语法格式: setleds [参数]

常用参数:

-F

设定虚拟主控台的状态

-D

改变虚拟主控台的状态和预设的状态

-L

直接改变 LED 显示的状态

+num/-num

将数字键打开或关闭

+caps/-caps

把大小写键打开或关闭

+scroll /-scroll

把选项键打开或关闭

参考实例

控制键盘灯num灯亮和灯灭:

```
[root@dsjprs ~]# setleds +num
[root@dsjprs ~]# setleds -num
# 控制键盘的大小写键打开或关闭, 键盘指示灯亮与灭:
[root@dsjprs ~]# setleds +caps
[root@dsjprs ~]# setleds -caps
# 控制键盘的选项键打开或关闭, 键盘指示灯亮与灭:
[root@dsjprs ~]# setleds +scroll
# 对三灯的亮与灭的情况进行组合, 分别设置为数字灯亮, 大小写灯灭, 选项键scroll灯灭:
[root@dsjprs ~]# setleds +num -caps -scroll
```



## lspci命令

lspci命令用于显示当前主机的所有PCI总线信息，以及所有已连接的PCI设备信息。现在主流设备如网卡储存等都采用PCI总线

语法格式：lspci [参数]

常用参数：

-n

以数字方式显示PCI厂商和设备代码

-t

以树状结构显示PCI设备的层次关系

-b

以总线为中心的视图

-s

仅显示指定总线插槽的设备和功能块信息

-i

指定PCI编号列表文件，不使用默认文件

-m

以机器可读方式显示PCI设备信息

参考实例

显示当前主机的所有PCI总线信息：

```
[root@dsjprs ~]# lspci
# 以树状结构显示PCI设备的层次关系：
[root@dsjprs ~]# lspci -t
```

## sensors命令

sensors命令用于检测服务器内部降温系统是否健康，可以监控主板，CPU的工作电压，风扇转速、温度等数据。

语法格式：sensors

参考实例

检测cpu工作电压，温度等：

```
[root@dsjprs ~]# sensors
c
coretemp-isa-0000
C
Core 0:      +48.0°C  (high = +87.0°C, crit = +97.0°C)
C
Core 1:      +46.0°C  (high = +87.0°C, crit = +97.0°C)
C
Core 2:      +47.0°C  (high = +87.0°C, crit = +97.0°C)
C
Core 3:      +46.0°C  (high = +87.0°C, crit = +97.0°C)
```

## 备份命令

### zipinfo命令

zipinfo命令的全称为“zip information”，该命令用于列出压缩文件信息。执行zipinfo指令可得知zip压缩文件的详细信息。

语法格式：zipinfo [参数]

常用参数：

-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

参考实例

显示压缩文件信息：

```
[root@dsjprs ~]# zipinfo file.zip
A
Archive: file.zip  486 bytes  3 files
-
-rw-r--r--  2.3 unx      0 bx stor 24-May-10 18:54 a.c
-
-rw-r--r--  2.3 unx      0 bx stor 24-May-10 18:54 b.c
-
-rw-r--r--  2.3 unx      0 bx stor 24-May-10 18:54 c.c
3
3 files, 0 bytes uncompressed, 0 bytes compressed: 0.0%
# 显示压缩文件中每个文件的信息：
[root@dsjprs ~]# zipinfo -v file.zip
# 只显示压缩包大小、文件数目：
[root@dsjprs ~]# zipinfo -h file.zip
A
Archive:  file.zip
Z
Zip file size: 907 bytes, number of entries: 3
# 生成一个基本的、长格式的列表(而不是冗长的)，包括标题和总计行：
[root@dsjprs ~]# zipinfo -l file
# 查看存档中最近修改的文件：
[root@dsjprs ~]# zipinfo -T file | sort -nr -k 7 | sed 15q
```

## zip命令

zip程序将一个或多个压缩文件与有关文件的信息(名称、路径、日期、上次修改的时间、保护和检查信息以验证文件完整性)一起放入一个压缩存档中。可以使用一个命令将整个目录结构打包到zip存档中。

对于文本文件来说, 压缩比为2:1和3:1是常见的。zip只有一种压缩方法(通缩), 并且可以在不压缩的情况下存储文件。(如果添加了bzip 2支持, zip也可以使用bzip 2压缩, 但这些条目需要一个合理的现代解压缩来解压缩。当选择bzip 2压缩时, 它将通货紧缩替换为默认方法。)zip会自动为每个要压缩的文件选择更好的两个文件(通缩或存储, 如果选择bzip2, 则选择bzip2或Store)。

语法格式: zip [参数] [文件]

常用参数:

-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

-z

替压缩文件加上注释

-v

显示指令执行过程或显示版本信息

-n<字尾字符串>

不压缩具有特定字尾字符串的文件

参考实例

将 /home/html/ 这个目录下所有文件和文件夹打包为当前目录下的 html.zip:

```
[root@dsjprs ~]# zip -q -r html.zip /home/html
# 压缩文件 cp.zip 中删除文件 a.c:
[root@dsjprs ~]# zip -dv cp.zip a.c
# 把/home目录下面的mydata目录压缩为mydata.zip:
[root@dsjprs ~]# zip -r mydata.zip mydata
# 把/home目录下面的abc文件夹和123.txt压缩成为abc123.zip:
[root@dsjprs ~]# zip -r abc123.zip abc 123.txt
# 将 logs目录打包成 log.zip:
[root@dsjprs ~]# zip -r log.zip ./logs
```

## gzip命令

gzip命令的英文是“GNUzip”, 是常用来压缩文件的工具, gzip是个使用广泛的压缩程序, 文件经它压缩过后, 其名称后面会多处“.gz”扩展名。

gzip是在Linux系统中经常使用的一个对文件进行压缩和解压缩的命令, 既方便又好用。gzip不仅可以用来压缩大的、较少使用的文件以节省磁盘空间, 还可以和tar命令一起构成Linux操作系统中比较流行的压缩文件格式。据统计, gzip命令对文本文件有60%~70%的压缩率。减少文件大小有两个明显的好处, 一是可以减少存储空间, 二是通过网络传输文件时, 可以减少传输的时间。

语法格式: gzip [参数]

常用参数:

-a

使用ASCII文字模式

-d

解开压缩文件

-f

强行压缩文件

-l

列出压缩文件的相关信息

-c

把压缩后的文件输出到标准输出设备, 不去更动原始文件

-r

递归处理, 将指定目录下的所有文件及子目录一并处理

-q

不显示警告信息

参考实例

把rancher-v2.2.0目录下的每个文件压缩成.gz文件：

```
[root@dsjprs ~]# gzip *  
# 把上例中每个压缩的文件解压，并列出详细的信息：  
[root@dsjprs ~]# gzip -dv *  
# 递归地解压目录：  
[root@dsjprs ~]# gzip -dr rancher.gz
```

## unzip命令

unzip命令是用于.zip格式文件的解压缩工具，unzip命令将列出、测试或从zip格式存档中提取文件，这些文件通常位于MS-DOS系统上。

默认行为（就是没有选项）是从指定的ZIP存档中提取所有的文件到当前目录（及其下面的子目录）。一个配套程序zip（1L）创建ZIP存档；这两个程序都与PKWARE的PKZIP和PKUNZIP为MS-DOS创建的存档文件兼容，但许多情况下，程序选项或默认行为是不同的。

语法格式：unzip [参数] [文件]

常用参数：

-l

显示压缩文件内所包含的文件

-v

执行时显示详细的信息

-c

将解压缩的结果显示到屏幕上，并对字符做适当的转换

-n

解压缩时不要覆盖原有的文件

-j

不处理压缩文件中原有的目录路径

参考实例

把/home目录下面的mydata.zip解压到mydatabak目录里面：

```
[root@dsjprs ~]# unzip mydata.zip -d mydatabak  
# 把/home目录下面的wwwroot.zip直接解压到/home目录里面：  
[root@dsjprs ~]# unzip wwwroot.zip  
# 把/home目录下面的abc12.zip、abc23.zip、abc34.zip同时解压到/home目录里面：  
[root@dsjprs ~]# unzip abc\*.zip  
# 查看把/home目录下面的wwwroot.zip里面的内容：  
[root@dsjprs ~]# unzip -v wwwroot.zip  
# 验证/home目录下面的wwwroot.zip是否完整：  
[root@dsjprs ~]# unzip -t wwwroot.zip
```

## unarj命令

unarj命令用于解压缩.arj文件。

语法格式：unarj [参数] [arj压缩文件]

常用参数：

-c

解压缩.arj文件

-l

显示压缩文件内所包含的文件

-t

检查压缩文件是否正确

-x

解压缩时保留原有的路径

参考实例

解压缩.arj文件:

```
[root@dsjprs ~]# unarj -e test.arj
# 显示压缩文件内所包含的文件:
[root@dsjprs ~]# unarj -l test.arj
# 检查压缩文件是否正确:
[root@dsjprs ~]# unarj -t test.arj
# 解压缩时保留原有的路径:
[root@dsjprs ~]# unarj -x test.arj
# 把文件解压到当前路径:
[root@linuxcool ~]# unarj -ex test.arj
```

## 其他命令

### hash命令

hash命令负责显示与清除命令运行时系统优先查询的哈希表 (hash table) 。

当执行hash命令不指定参数或标志时，hash命令向标准输出报告路径名列表的内容。此报告含有先前hash命令调用找到的当前shell环境中命令的路径名。而且还包含通过正常命令搜索进程调用并找到的那些命令。

语法格式: hash [参数] [目录]

常用参数:

-d

在哈希表中清除记录

-l

显示哈希表中的命令

-p<指令>

将具有完整路径的命令加入到哈希表中

-r

清除哈希表中的记录

-t

显示哈希表中命令的完整路径

参考实例

显示哈希表中的命令:

```
[root@dsjprs ~]# hash -l
# 删除哈希表中的命令:
[root@dsjprs ~]# hash -r
# 向哈希表中添加命令:
[root@dsjprs ~]# hash -p /usr/sbin/adduser myadduser
# 在哈希表中清除记录:
[root@dsjprs ~]# hash -d
# 显示哈希表中命令的完整路径:
[root@dsjprs ~]# hash -t
```

## wait命令

wait命令用来等待指令的指令，直到其执行完毕后返回终端。该指令常用于shell脚本编程中，待指定的指令执行完成后，才会继续执行后面的任务。该指令等待作业时，在作业标识号前必须添加备份号“%”。

语法格式：wait [参数]

常用参数：

22 或%1

进程号 或 作业号

参考实例

等待作业号为1的作业完成后再返回：

```
[root@dsjprs ~]# wait %1
[
[root@dsjprs ~]# find / -name password
```

## bc命令

bc的英文全拼为“ Binary Calculator ”，是一种支持任意精度的交互执行的计算器语言。bash内置了对整数四则运算的支持，但是并不支持浮点运算，而bc命令可以很方便的进行浮点运算，当然整数运算也不再话下。

语法格式：bc [选项]

常用参数：

-i

强制进入交互式模式

-l

定义使用的标准数学库

-w

定义使用的标准数学库

-q

打印正常的GNU bc环境信息

参考实例

算术操作高级运算bc命令它可以执行浮点运算和一些高级函数：

```
[root@dsjprs ~]# echo "1.212*3" | bc
3.636
# 设定小数精度（数值范围）：
[root@dsjprs ~]# echo "scale=2;3/8" | bc
.37
# 计算平方和平方根：
[root@dsjprs ~]# echo "10^10" | bc
10000000000
[root@dsjprs ~]# echo "sqrt(100)" | bc
10
```

## rmmod命令

rmmod即“remove module”的简称,rmmod命令用于删除模块。执行rmmod命令，可删除不需要的模块。Linux操作系统的核心具有模块化的特性，因此在编译核心时，不需要吧全部功能都放入核心，可以将这些功能编译成一个个单独的模块，待有需要时再分别载入它们。

语法格式：rmmod [参数] [模块名称]

常用参数：

-a

删除所有目前不需要的模块

-s

把信息输出至syslog常驻服务，而非终端机界面

-v

显示指令执行的详细信息

-f

强制移除模块，使用此选项比较危险

-w

等待着，知道模块能够被除时再移除模块

参考实例

卸载模块pppoe并显示执行信息：

```
[root@dsjprs ~]# rmmod -v pppoe
# 卸载bridge模块：
[root@dsjprs ~]# rmmod bridge
# 卸载bridge模块并将错误信息写入syslog：
[root@dsjprs ~]# rmmod -s bridge
# 孤立正在使用的bridge模块，知道不被使用：
[root@dsjprs ~]# rmmod -w bridge
# 删除正在使用的bridge模块（-f可以删除正在使用的模块，非常危险，慎用）：
[root@dsjprs ~]# rmmod -f bridge
```

## history命令

history命令用于显示用户以前执行过的历史命令，并且能对历史命令进行追加和删除等操作。

如果你经常使用Linux命令，那么使用history命令可以有效地提升你的效率。

语法格式: history [参数] [目录]

常用参数：

-a

将当前shell会话的历史命令追加到命令历史文件中,命令历史文件是保存历史命令的配置文件

-c

清空当前历史命令列表

-d

删除历史命令列表中指定序号的命令

-n

从命令历史文件中读取本次Shell会话开始时没有读取的历史命令

-r

读取命令历史文件到当前的Shell历史命令内存缓冲区

-s

将指定的命令作为单独的条目加入命令历史内存缓冲区。在执行添加之前先删除命令历史内存缓冲区中最后一条命令

-w

把当前的shell历史命令内存缓冲区的内容写入命令历史文件

参考实例

显示最近的10条命令：

```
[root@dsjprs ~]# history 10
# 将本次登录的命令写入历史文件中：
[root@dsjprs ~]# history -w
# 将命令历史文件中的内容读入到目前shell的history记忆中：
[root@dsjprs ~]# history -r
# 将当前Shell会话的历史命令追加到命令历史文件中：
[root@dsjprs ~]# history -a
# 清空当前历史命令列表：
[root@dsjprs ~]# history -c
```