

# Linux 常用命令大全

Author: @Marlous (整理、编写。)

E-mail: [goonecat@foxmail.com](mailto:goonecat@foxmail.com)

Date: 2018/10/18

参考资料: [参考 1](#) [参考 2](#) [参考 3](#) 不同发行版本 Linux 命令会略有不同。

## 一 补充之查看各种信息

### 1 系统信息

```
uname -a      # 显示Linux系统信息
uname -r      # 显示内核发布版本信息
uptime        # 显示系统已经运行的时间和系统负载
hostname      # 显示系统主机名
last reboot   # 显示系统重启历史
date          # 显示当前日期和时间
cal           # 显示本月日历
w             # 显示谁当前正登录这台主机
whoami        # 显示你的登录名
```

### 2 硬件信息

```
dmesg          # 监测硬件和启动消息
cat /proc/cpuinfo  # CPU信息
cat /proc/meminfo  # 硬件内存信息
free -m        # 已使用的和可用内存, -m表示单位为M
lspci -tv      # 显示PCI设备信息
lsusb -tv      # 显示USB设备信息
lsmod          # 显示加载的模块
hdparm -l /dev/sda  # 显示sda硬盘信息
hdparm -tT /dev/sda  # 对sda硬盘进行读取速度测试
hdparm -s /dev/sda  # 测试sda硬盘上不可读的块
```

### 3 统计信息

```
top                # 显示并不断更新最耗CPU的进程
mpstat 1           # 显示CPU统计信息
vmstat 2           # 显示虚拟内存统计信息
free               # 显示物理内存使用情况
iostat 2           # 显示IO统计信息（2s采样间隔）
tcpdump -i eth1    # 捕获eth1网络接口上的所有数据包
tcpdump -i eth0 'port 80' # 监控80端口的网络流量
lsof               # 列出所有活跃进程打开的文件
lsof -u testuser   # 列出所有testuser用户打开的文件
ipcs               # 提供IPC设施信息
ipcrm              # 删除消息队列、信号量集或共享内存ID
lslk               # 列出本地锁
```

## 4 磁盘使用率

```
df -h    # 查看磁盘可用空间，和其他参数
du -ah    # 以人类可读形式显示磁盘使用情况
du -sh    # 以人类可读形式显示当前目录下磁盘使用情况
```

# 二 补充之日常操作

## 1 获取命令的系统帮助信息

获取命令的系统帮助信息

```
help  # 查看内部shell命令的帮助信息(常用)
man   # 显示在线帮助手册(常用)
info  # info格式的帮助文档
```

## 2 时间与日期

```
cal      # 显示日历信息
date     # 显示和设置系统日期和时间
hwclock  # 查看和设置硬件时钟
clockdiff # 主机之间测量时钟差
rdate    # 通过网络获取时间
sleep    # 暂停指定的时间
```

## 3 切换所在位置

```
cd ..    # 跳到上一层目录
cd /test  # 进入某目录
cd        # 回到$HOME目录
```

## 4 搜索

```
locate keywords # 预先建立数据库，默认每天更新一次，用 updatedb 命令手动更新
```

grep 选项 字符串 文件名

```
-i # 忽略大小写
-v # 排除指定字符串
-n # 显示结果所在行
```

find 位置 参数 参数值

```
-name
-perm # 权限。例 find -perm 777
-user
-group
-ctime
-type
-size
```

补充：通配符

```
* # 匹配零到多个
? # 匹配一个字符
[0-9] # 匹配一个数字范围
[abc] # 匹配列表中任意字符
[^abc] # 匹配列表外的任意字符
```

## 5 SSH 与 Telnet

```
ssh user@host
ssh -p port user@host
telnet host
```

## 6 Vi 编辑器使用

[参考资料](#) 两种模式：命令模式、插入模式、ex 模式。按 esc 键退出命令模式。

命令模式下：

```
i # 进入编辑模式
o # 当前行下插入新行
dd # 删除整行
yy # 复制当前行
n+yy # 复制 n 行
p # 粘贴
u # 撤销上一个操作
r # 替换当前字符
/ # 查找关键字
```

ex 模式下:

```
:w          # 保存当前修改
:wq         # 存盘退出
:q          # 退出
:q!         # 强制推出
:set number # 显示行号
:! command  # 执行一个系统命令并显示结果。
:sh         # 切换到命令行, ctrl+d 切回 vi
```

## 三 user

### 1 用户与群组

```
su - username      # 切换用户 (全新的环境)
su username        # 切换用户 (仅切换用户)
su                 # 切换到 root 用户 (仅切换用户)
id                 # 当前用户唯一标识信息
who                # 类似w
passwd             # 修改当前用户密码
passwd user        # 修改 user 用户的密码
chage -E 2018-10-21 user # 设置用 user 口令的失效期限
pwck               # 检查 '/etc/passwd' 的文件格式和语法修正以及存在的用户
grpck              # 检查 '/etc/passwd' 的文件格式和语法修正以及存在的群组
newgrp groupname   # 登陆进一个新的群组以改变新创建文件的预设群组
```

-----

```
users              # 查看所有用户
```

```
useradd 参数 用户名 # 添加用户
-d 家目录
-g 主组
-G 附属组           # 最多 31 个, 用逗号隔开
-s 登陆shell
-u userid
```

```
usermod 参数 用户名 # 修改用户信息
-l 新用户名         # 家目录东西不变
-u 新userid
-d 用户家目录位置
-g 用户所属组
-G 用户所属附属组
-L                  # 锁定用户使其不能登陆
-U                  # 解除锁定
```

```
userdel 用户名      # 删除用户
-r                  # 同时删除用户的家目录
```

-----

```
groups              # 查看所有组
```

```
groupadd 组名          # 添加组

groupmod -n 新组名 旧组名  # 修改组名
groupmod -g 新gid 旧gid    # 修改组 id

groupdel 组名          # 删除一个用户组
```

## 2 文件权限

```
# 用数字修改。注意：4-read, 2-write, 1-execute
chmod 755 file-or-dir-name

chown owner-user file          # 更改文件所有者
chown owner-user:owner-group file  # 更改文件所有者和所有者所在组
chown owner-user:owner-group directory

# 用 UGO 修改。注意：UGO (user、group、other, other 为除此用户、此组外的成员)
chmod u+x file-or-dir-name
chmod u-r file-or-dir-name
-r    # 读
-w    # 写
-x    # 执行

-----

chown username filename      # 修改所属用户
-R                                # 递归修改目录下所有文件所属用户

chgrp groupname filename     # 修改所属组
-R

-----

# 特殊属性
chattr +a file1      # 只允许以追加方式读写文件
chattr +c file1      # 允许这个文件能被内核自动压缩/解压
chattr +d file1      # 在进行文件系统备份时，dump程序将忽略这个文件
chattr +i file1      # 设置成不可变的文件，不能被删除、修改、重命名或者链接
chattr +s file1      # 允许一个文件被安全地删除
chattr +S file1      # 一旦应用程序对这个文件执行了写操作，使系统立刻把修改的结果写到磁盘
chattr +u file1      # 若文件被删除，系统会允许你在以后恢复这个被删除的文件
lsattr                # 显示特殊的属性
```

## 四 system

### 1 关机/重启/登陆

```
shutdown -h now          # 关闭系统
shutdown -h hours:minutes # 按预定时间关闭系统
shutdown -h +10          # 10分钟后关机
shutdown -c              # 取消按预定时间关闭系统
reboot                   # 重启
logout                   # 注销
```

## 2 进程

```
ps                        # 显示所有进程
pstree                   # 查看进程树
ps -s | grep ssh         # 查看某应用状态
service 服务名 status    # 查看某服务状态
service --status-all    # 列出所有服务
chkconfig --list | grep 'XXX' # 查看某服务配置
chkconfig --level 35 XXX on # XXX 服务在 3, 5启动级别下启动

pidof                    # 根据进程名查找正在运行的进程的进程号
pgrep                    # 按名称和其他属性查找进程
pmap pid                 # 进程内存消耗信息
top                      # 显示当前正在运行的进程
kill pid                 # 杀死某进程
killall procname         # 杀死所有名为procname的进程
pkill                    # 按名称和其他属性杀死进程
timeout                  # 在指定时间后仍在运行则杀死该进程
wait                     # 等待指定的进程

bg                        # 恢复在后台暂停工作的作业
fg                        # 将程序或命令放到前台执行
atq                       # 列出用户等待执行的作业
jobs                      # 列出活动的作业
atrm                      # 删除作业
watch                     # 定期执行一个程序
at                        # 在指定时间执行命令
sleep                     # 延迟执行
sleep 10s                 # 推迟10秒, s、m、h、d

chkconfig                 # 为系统服务更新和查询运行级别信息
fuser                     # 显示哪些进程使用指定的文件、套接字或文件系统
nohup                     # 运行指定的命令不受挂起
```

## 五 data

### 1 文件操作

```
pwd                      # 显示当前所在路径
file 文件目录名          # 查看目录类型
ln -s /path/to/filename link-name # 建立软链接
```

```
-----
ls                                # 列出所有文件和文件夹信息
-a      # 显示所有文件（包括系统文件）
-l      # 显示详细信息
-ls     # 显示目录和连接信息
-R      # 递归显示子目录结构

mkdir dirname                    # 创建目录
rmdir                                       # 删除空目录

rm filename                      # 删除文件或目录
-i      # 交互式
-r      # 递归
-f      # 强制删除

cp file1 file2                  # 拷贝
-r      # 递归
-v      # 详细信息

mv file1 file2                  # 移动。如果file2是一个目录，则移动file1到file2目录，否则重命名文件。

-----

echo                             # 显示输入的内容
touch file                       # 创建文件
more file                       # 翻页显示内容
head file                       # 显示文件开头10行内容
tail file                       # 显示文件末尾10行内容
less                             # 回卷显示文本文件的内容
cat filename                    # 显示文件内容

ln -s source target            # 软链接（符号链接）
ln source target               # 硬链接（实体链接）

-----

gpg -c file                    # 加密文件，文件以gpg为后缀
gpg file.gpg                  # 解密文件
```

## 2 文件处理

```
# 文本处理
cut 参数参数值 文件
-d      # 指定分隔符（默认为 tab），例 cut -d: /etc/xxx（用逗号分隔）
-f      # 制定输出列号
-c      # 基于字符进行切割

# 文本统计
wc 文件
-l      # 只统计和行数
-w      # 只统计单词
-c      # 只统计字节数
-m      # 只统计字符数
```

# 对文件中的数据进行排序

sort

-r # 降序  
-n # 基于数字进行排序  
-o # 标准输出  
-f # 忽略大小写  
-u # 删除重复行  
-t c # 使用c作为分隔符为列排序  
-k x # 指定以第x列来排序  
-f # 忽略大小写  
-c # 会检查文件是否已排好序, 如果乱序, 则输出第一个乱序的行的相关信息, 最后返回1  
-C # 会检查文件是否已排好序, 如果乱序, 不输出内容, 仅返回1  
-M # 会以月份来排序, 比如JAN小于FEB等等  
-b # 会忽略每一行前面的所有空白部分, 从第一个可见字符开始比较。

# 文件比较

diff 文件1 文件2

-i # 忽略大小写  
-b # 忽略空格数变化  
-u # 统一显示比较信息 (一般用于生成patch文件), 例 diff -u file newfile > final.patch

# 搜索替换

sed

参数:

-e<script> 或 --expression=<script> # 以选项中指定的script来处理输入的文本文件。  
-f<script文件> 或 --file=<script文件> # 以选项中指定的script文件来处理输入的文本文件。  
-n 或 --quiet 或 --silent # 仅显示script处理后的结果。

动作:

a # 新增, a 的后面可以接字符串, 而这些字符串会在新的一行出现(目前的下一行) ~  
c # 取代, c 的后面可以接字符串, 这些字符串可以取代 n1,n2 之间的行!  
d # 删除, 因为是删除啊, 所以 d 后面通常不接任何咚咚;  
i # 插入, i 的后面可以接字符串, 而这些字符串会在新的一行出现(目前的上一行);  
p # 打印, 亦即将某个选择的数据印出。通常 p 会与参数 sed -n 一起运行 ~  
s # 取代, 可以直接进行取代的工作哩! 通常这个 s 的动作可以搭配正规表示法! 例如 1,20s/old/new/g

例 sed -e 4a\newline testfile # 在testfile文件的第四行后添加一行, 并将结果输出到标准输出

# 检查拼写

aspell check filename

aspell list < filename

# 处理文本内容

tr -d 'XXX' < filename

tr 'a-z' 'A-Z' < filename

# 其它

uniq # 将重复行从输出文件中删除  
comm # 逐行比较两个已排序的文件  
cmp # 按字节比较两个文件  
split # 将输入文件进行分割成片, 输出固定大小的块  
tee # 将标准输入复制到每一个指定的文件  
awk # 模式扫描和处理语言, 比较复杂, 功能强大常用



## 3 压缩解压与归档

```
tar
-c 创建.tar格式的文件
-x 解开.tar格式的文件
-f 使用归档文件
-v 显示详细信息
-t 查看包内文件
-j 使用bzip2程序
-z 使用gzip程序

-p 打包时保留文件及目录的权限
-P 打包时保留文件及目录的绝对路径
-C 释放的目的地

-----
# 补充:
*.tar 用 tar -xvf 解压
*.gz 用 gzip -d 或者 gunzip 解压
*.tar.gz 和 *.tgz 用 tar -xzf 解压
*.bz2 用 bzip2 -d 或者用 bunzip2 解压
*.tar.bz2 用 tar -xjf 解压
*.Z 用 uncompress 解压
*.tar.Z 用 tar -xZf 解压
*.rar 用 unrar e 解压
*.zip 用 unzip 解压
```

## 4 文件传输

```
# scp
scp file.txt server2:/tmp          # 安全拷贝file.txt到远程主机的/tmp目录下
scp noodle@server2:/www/*.html /www/tmp # 拷贝远程主机的/www/目录下的所有HTML文件到本地的/www/tmp
目录
scp -r noodle@server2:/www /www/tmp # 递归拷贝远程主机/www目录下的所有文件和文件夹到本
地/www/tmp目录

# rsync
rsync -a /home/apps /backup/      # 源目录和目标目录同步
rsync -avz /home/apps noodle@192.168.10.1:/backup # 本地目录和远程主机目录同步，启用压缩
```

## 5 挂载文件系统

```
# 自动挂载: 配置文件 /etc/fstab
fuser -km          # 当设备繁忙时强制卸载
fuser -m           # 查看使用某文件系统的进程
lsof               # 查看正在被使用的文件

mount 要挂载的分区 挂载点          # 挂载一个叫做hda2的盘（确定目录 '/ mnt/hda2' 已经存在 ）。
```

```
-t    # 指定文件系统类型
-o    # 指定挂载选项
ro (只读)、rw (读写)
sync (直接写入硬盘, 不用缓存)
async (用缓存, 默认)
notime (每次访问文件不更新文件访问时间)
atime (每次访问文件更新文件访问时间)
remount (重新挂载文件系统)

umount /dev/hda2          # 卸载一个叫做hda2的盘 (先从挂载点 '/ mnt/hda2' 退出)。
umount -n /mnt/hda2       # 运行卸载操作而不写入 /etc/mtab 文件, 当文件为只读或当磁盘写
                           # 满时非常有用
```

## 6 磁盘

[使用 fdisk 分区参考资料](#)

```
# fdisk 和 mkfs 命令必须以管理员身份运行。
fdisk -l /dev/sda          # 查看第一块硬盘, 无设备名则是全部信息

# 一些操作, 可见帮助。 fdisk 要操作的磁盘或分区; 删除或创建分区; 保存后退出, 然后格式化。
m  # 显示菜单和帮助信息
a  # 活动分区标记/引导分区
d  # 删除分区
l  # 显示分区类型
n  # 新建分区
p  # 显示分区信息
q  # 退出不保存
t  # 设置分区号
v  # 进行分区检查
w  # 保存修改
x  # 扩展应用, 高级功能
```

## 7 软件管理

1. 补充, 下载, 编译与安装:

```
# 从链接下载
wget 链接

# 编译好的软件压缩包: 直接解压后放入 /opt 目录, 运行该软件文件夹中的可执行文件。
# 未编译好的: 解压, 进入目标软件的文件夹, 运行配置脚本, 编译, 安装。
./configure --prefix=/opt/filename
make
make install
```

2. Debian 及分支操作系统的包管理 (dpkg) :

```
apt-get install package_name    # 安装/更新一个 deb 包
```

```

apt-cdrom install package_name      # 从光盘安装/更新一个 deb 包
apt-get update                      # 升级列表中的软件包
apt-get upgrade                     # 升级所有已安装的软件
apt-get remove package_name        # 从系统删除一个deb包
apt-get check                       # 确认依赖的软件仓库正确
apt-get clean                       # 从下载的软件包中清理缓存
apt-cache search searched-package  # 返回包含所要搜索字符串的软件包名称

```

补充: 安装 deb 安装包。

```

dpkg -i package.deb                # 安装/更新一个 deb 包
dpkg -r package_name               # 从系统删除一个 deb 包
dpkg -l                             # 显示系统中所有已经安装的 deb 包
dpkg -l | grep httpd               # 显示所有名称中包含 "httpd" 字样的deb包
dpkg -s package_name               # 获得已经安装在系统中一个特殊包的信息
dpkg -L package_name               # 显示系统中已经安装的一个deb包所提供的文件列表
dpkg --contents package.deb        # 显示尚未安装的一个包所提供的文件列表
dpkg -S /bin/ping                  # 确认所给的文件由哪个deb包提供

```

### 3. RedHat 及分支操作系统的包管理 (rpm、yum) :

- rpm:

```

rpm -ivh package.rpm                # 安装一个rpm包
rpm -ivh --nodeps package.rpm       # 安装一个rpm包而忽略依赖关系警告
rpm -U package.rpm                  # 更新一个rpm包但不改变其配置文件
rpm -F package.rpm                  # 更新一个确定已经安装的rpm包
rpm -e package_name.rpm              # 删除一个rpm包
rpm -qa                             # 显示系统中所有已经安装的rpm包
rpm -qa | grep httpd                # 显示所有名称中包含 "httpd" 字样的rpm包
rpm -qi package_name                 # 获取一个已安装包的特殊信息
rpm -qg "System Environment/Daemons" # 显示一个组件的rpm包
rpm -ql package_name                 # 显示一个已经安装的rpm包提供的文件列表
rpm -qc package_name                 # 显示一个已经安装的rpm包提供的配置文件列表
rpm -q package_name --whatrequires   # 显示与一个rpm包存在依赖关系的列表
rpm -q package_name --whatprovides   # 显示一个rpm包所占的体积
rpm -q package_name --scripts        # 显示在安装/删除期间所执行的脚本1
rpm -q package_name --changelog       # 显示一个rpm包的修改历史
rpm -qf /etc/httpd/conf/httpd.conf   # 确认所给的文件由哪个rpm包所提供
rpm -qp package.rpm -l                # 显示由一个尚未安装的rpm包提供的文件列表
rpm --import /media/cdrom/RPM-GPG-KEY # 导入公钥数字证书
rpm --checksig package.rpm            # 确认一个rpm包的完整性
rpm -qa gpg-pubkey                   # 确认已安装的所有rpm包的完整性
rpm -V package_name                  # 检查文件尺寸、 许可、类型、所有者、群组、MD5检查以及最后修
改时间
rpm -Va                              # 检查系统中所有已安装的rpm包- 小心使用
rpm -Vp package.rpm                  # 确认一个rpm包还未安装
rpm2cpio package.rpm | cpio --extract --make-directories *bin* # 从一个rpm包运行可执行文件
rpm -ivh /usr/src/redhat/RPMS/`arch`/package.rpm                # 从一个rpm源码安装一个构建好的包
rpmbuild --rebuild package_name.src.rpm                          # 从一个rpm源码构建一个 rpm 包

```

- yum:

yum install package_name	# 下载并安装一个rpm包
yum localinstall package_name.rpm	# 将安装一个rpm包, 使用你自己的软件仓库为你解决所有依赖关系
yum update package_name.rpm	# 更新当前系统中所有安装的rpm包
yum update package_name	# 更新一个rpm包
yum remove package_name	# 删除一个rpm包
yum list	# 列出当前系统中安装的所有包
yum search package_name	# 在rpm仓库中搜寻软件包
yum clean packages	# 清理rpm缓存删除下载的包
yum clean headers	# 删除所有头文件
yum clean all	# 删除所有缓存的包和头文件

## 六 net

网络相关配置文件:

网卡配置文件: /etc/sysconfig/network-scripts/ifcfg-eth0 # 配置IP可能要先关掉守护进程, network-manager服务

DNS 配置文件: /etc/resolv.conf

主机名配置文件: /etc/sysconfig/network

静态主机名配置文件: /etc/hosts

ipaddr	# 查看IP地址
ifconfig eth0 up	# 网卡开启, down 为关闭
ifconfig -a	# 列出所有网络端口和IP地址
ifconfig eth0	# 列出指定以太网端口对应的IP地址和详细信息
ethtool eth0	# 查看以太网状态
nslookup	# 查看DNS, >server 为查看本机
ping host	# ping测试
ip route	# 显示路由表
traceroute	# trace追踪
mtr	# 网络质量测试
route	# 路由相关
whois domain	# 获取指定域名的信息
dig domain	# 获取指定域名的DNS信息
dig -x host	# 根据主机地址反向查找
host goole.com	# 根据域名查找DNS IP地址
wget file	# 下载文件
netstat	# 网络相关状态
-t	# 列出TCP协议端口
-u	# 列出UDP协议端口
-n	# 不用域名服务名, 用IP和端口号
-l	# 反列出在监听状态的网络服务
-a	# 列出所有网络连接