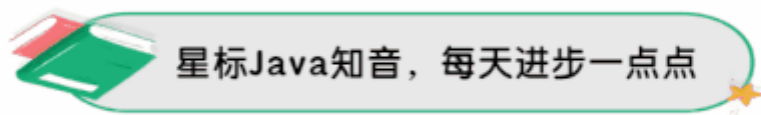


# Linux命令速查手册出炉!

Java知音 今天



来源: MarkerHub

## 查看Linux系统信息

```
arch                #显示机器的处理器架构(1)
uname -m            #显示机器的处理器架构(2)
uname -r            #显示正在使用的内核版本
dmidecode -q        #显示硬件系统部件 - (SMBIOS / DMI)
hdparm -i /dev/hda   #罗列一个磁盘的架构特性
hdparm -tT /dev/sda  #在磁盘上执行测试性读取操作
cat /proc/cpuinfo    #显示CPU info的信息
cat /proc/interrupts #显示中断
cat /proc/meminfo    #校验内存使用
cat /proc/swaps       #显示哪些swap被使用
cat /proc/version    #显示内核的版本
cat /proc/net/dev    #显示网络适配器及统计
cat /proc/mounts     #显示已加载的文件系统
lspci -tv            #罗列PCI设备
lsusb -tv            #显示USB设备
```

## date 显示系统日期

```
cal 2007                #显示2007年的日历表
date 041217002007.00    #设置日期和时间 - 月日時分年. 秒
clock -w                #将时间修改保存到 BIOS
```

## 关机 (关机、重启、登出)

```
shutdown -h now          #关闭系统(1)
init 0                   #关闭系统(2)
telinit 0                #关闭系统(3)
shutdown -h hours:minutes &    #按预定时间关闭系统
shutdown -c              #取消按预定时间关闭系统
shutdown -r now          #重启(1)
reboot                  #重启(2)
logout                  #注销
```

## 文件和目录

```
cd /home                #进入 '/ home' 目录'
cd ..                   #返回上一级目录
cd ../..               #返回上两级目录
cd                      #进入个人的主目录
cd ~user1              #进入个人的主目录
cd -                   #返回上次所在的目录
pwd                    #显示工作路径

ls                      #查看目录中的文件
ls -F                 #查看目录中的文件
ls -l                 #显示文件和目录的详细资料
ls -a                 #显示隐藏文件
ls *[0-9]*            #显示包含数字的文件名和目录名
tree                  #显示文件和目录由根目录开始的树形结构(1)
lstree                #显示文件和目录由根目录开始的树形结构(2)

mkdir dir1             #创建一个叫做 'dir1' 的目录'
mkdir dir1 dir2        #同时创建两个目录
mkdir -p /tmp/dir1/dir2    #创建一个目录树
rm -f file1            #删除一个叫做 'file1' 的文件'
rmdir dir1             #删除一个叫做 'dir1' 的目录'
rm -rf dir1            #删除一个叫做 'dir1' 的目录并同时删除其内容
rm -rf dir1 dir2       #同时删除两个目录及它们的内容
mv dir1 new_dir        #重命名/移动 一个目录

cp file1 file2         #复制一个文件
```

```

cp dir/* .          #复制一个目录下的所有文件到当前工作目录
cp -a /tmp/dir1 .   #复制一个目录到当前工作目录
cp -a dir1 dir2      #复制一个目录

ln -s file1 lnk1     #创建一个指向文件或目录的软链接
ln file1 lnk1        #创建一个指向文件或目录的物理链接

touch -t 0712250000 file1    #修改一个文件或目录的时间戳 - (YYMMDD)
file file1 outputs the mime type of the file as text
iconv -l              #列出已知的编码

iconv -f fromEncoding -t toEncoding inputFile > outputFile create

find . -maxdepth 1 -name *.jpg -print -exec convert "{}" -resize

```

## 文件搜索

```

find / -name file1      #从 '/' 开始进入根文件系统搜索文件和目录
find / -user user1      #搜索属于用户 'user1' 的文件和目录
find /home/user1 -name \*.bin    #在目录 '/ home/user1' 中搜索
find /usr/bin -type f -atime +100    #搜索在过去100天内未被使用过的
find /usr/bin -type f -mtime -10     #搜索在10天内被创建或者修改过的
find / -name \*.rpm -exec chmod 755 '{}' \;    #搜索以 '.rpm' 结尾的文件
find / -xdev -name \*.rpm    #搜索以 '.rpm' 结尾的文件，忽略光驱
locate \*.ps                #寻找以 '.ps' 结尾的文件 - 先运行 'updatedb' 命令
whereis halt                #显示一个二进制文件、源码或man的位置
which halt                  #显示一个二进制文件或可执行文件的完整路径

```

## 挂载文件系统

```

mount /dev/hda2 /mnt/hda2    #挂载一个叫做hda2的盘 - 确定目录 '/ mnt/hda2'
umount /dev/hda2             #卸载一个叫做hda2的盘 - 先从挂载点 '/ mnt/hda2'
fuser -km /mnt/hda2          #当设备繁忙时强制卸载
umount -n /mnt/hda2          #运行卸载操作而不写入 /etc/mtab 文件- 挂载点
mount /dev/fd0 /mnt/floppy    #挂载一个软盘
mount /dev/cdrom /mnt/cdrom   #挂载一个cdrom或dvdrom
mount /dev/hdc /mnt/cdrecorder #挂载一个cdrw或dvdrom

```

```
mount /dev/hdb /mnt/cdrecorder #挂载一个cdrw或dvdrom
mount -o loop file.iso /mnt/cdrom #挂载一个文件或ISO镜像文件
mount -t vfat /dev/hda5 /mnt/hda5 #挂载一个Windows FAT32文件系统
mount /dev/sdal /mnt/usdisk #挂载一个usb 捷盘或闪存设备
mount -t smbfs -o username=user,password=pass //WinClient/share /m
```

## 磁盘空间相关

```
df -h #显示已经挂载的分区列表
ls -lSr |more #以尺寸大小排列文件和目录
du -sh dir1 #估算目录 'dir1' 已经使用的磁盘空间
du -sk * | sort -rn #以容量大小为依据依次显示文件和目录的大小
rpm -q -a --qf '%10{SIZE}t%{NAME}n' | sort -k1,1n
#以大小为依据依次显示已安装的rpm包所使用的空间 (fedora, redhat类系统)
dpkg-query -W -f='${Installed-Size;10}t${Package}n' | sort -k1,1n
#以大小为依据显示已安装的deb包所使用的空间 (ubuntu, debian类系统)
```

## 用户和群组

```
groupadd group_name #创建一个新用户组
groupdel group_name #删除一个用户组
groupmod -n new_group_name old_group_name #重命名一个用户组
useradd -c "Name Surname" -g admin -d /home/user1 -s /bin/bash
useradd user1 #创建一个新用户
userdel -r user1 #删除一个用户 ( '-r' 排除主目录)
usermod -c "User FTP" -g system -d /ftp/user1 -s /bin/nologin u

passwd #修改口令
passwd user1 #修改一个用户的口令 (只允许root执行)
chage -E 2005-12-31 user1 #设置用户口令的失效期限
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 group_name #登陆进一个新的群组以改变新建文件的预设群组
```

## 文件权限

使用 `"+"` 设置权限, 使用 `"-"` 用于取消

```
ls -lh          #显示权限
ls /tmp | pr -T5 -W$COLUMNS  #将终端划分成5栏显示
chmod ugo+rwx directory1      #设置目录的所有人(u)、群组(g)以及其
chmod go-rwx directory1      #删除群组(g)与其他人(o)对目录的读写操
chown user1 file1             #改变一个文件的所有人属性
chown -R user1 directory1     #改变一个目录的所有人属性并同时改变改
chgrp group1 file1            #改变文件的群组
chown user1:group1 file1      #改变一个文件的所有人和群组属性
find / -perm -u+s             #罗列一个系统中所有使用了SUID控制
chmod u+s /bin/file1          #设置一个二进制文件的 SUID 位 -
chmod u-s /bin/file1          #禁用一个二进制文件的 SUID位
chmod g+s /home/public        #设置一个目录的SGID 位 - 类似SUID
chmod g-s /home/public        #禁用一个目录的 SGID 位
chmod o+t /home/public        #设置一个文件的 STIKY 位 - 只允许
chmod o-t /home/public        #禁用一个目录的 STIKY 位
```

## 文件的特殊属性

- 使用 `"+"` 设置权限, 使用 `"-"` 用于取消

```
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、重命名或者被
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
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
lsattr                #显示特殊的属性
```

## 打包和压缩文件

```
bunzip2 file1.bz2    #解压一个叫做 'file1.bz2' 的文件
bzip2 file1           #压缩一个叫做 'file1' 的文件
gunzip file1.gz       #解压一个叫做 'file1.gz' 的文件
gzip file1            #压缩一个叫做 'file1' 的文件
```

```

gzip -9 file1 #最大程度压缩

rar a file1.rar test_file #创建一个叫做 'file1.rar'
rar a file1.rar file1 file2 dir1 #同时压缩 'file1', 'file2'
rar x file1.rar #解压rar包
unrar x file1.rar #解压rar包

tar -cvf archive.tar file1 #创建一个非压缩的 tarball
tar -cvf archive.tar file1 file2 dir1 #创建一个包含了 'file1',
tar -tf archive.tar #显示一个包中的内容
tar -xvf archive.tar #释放一个包
tar -xvf archive.tar -C /tmp #将压缩包释放到 /tmp目录下
tar -cvfj archive.tar.bz2 dir1 #创建一个bzip2格式的压缩包
tar -jxvf archive.tar.bz2 #解压一个bzip2格式的压缩包
tar -cvfz archive.tar.gz dir1 #创建一个gzip格式的压缩包
tar -zxvf archive.tar.gz #解压一个gzip格式的压缩包

zip file1.zip file1 #创建一个zip格式的压缩包
zip -r file1.zip file1 file2 dir1 #将几个文件和目录同时压缩成
unzip file1.zip #解压一个zip格式压缩包

```

## RPM包 (Fedora, Redhat及类似系统)

```

rpm -ivh package.rpm #安装一个rpm包
rpm -ivh --nodeeps 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
rpm -q package_name --whatrequires #显示与一个rpm包存在依赖关
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
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包所挂
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、
rpm -Va                                  #检查系统中所有已安装的rpm包- 小
rpm -Vp package.rpm                      #确认一个rpm包还未安装
rpm2cpio package.rpm | cpio --extract --make-directories *bin*
rpm -ivh /usr/src/redhat/RPMS/`arch`/package.rpm      #从一个rpm源码
rpmbuild --rebuild package_name.src.rpm      #从一个rpm源码构建

```

## YUM (Fedora, RedHat及类似系统)

```

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
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

```

## DEB包 (Debian, Ubuntu及类似系统)

```

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
dpkg --contents package.deb              #显示尚未安装的一个包所提供的文件列
dpkg -S /bin/ping                         #确认所给的文件由哪个deb包挂

```

## APT软件工具 (Debian, Ubuntu及类似系统)



```

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

```

## 查看文件内容

```

cat file1                            #从第一个字节开始正向查看文件的内容
tac file1                            #从最后一行开始反向查看一个文件的内容
more file1                           #查看一个长文件的内容
less file1                           #类似于 'more' 命令，但是它允许在文件中和正向操作一样
head -2 file1                        #查看一个文件的前两行
tail -2 file1                        #查看一个文件的最后两行
tail -f /var/log/messages            #实时查看被添加到一个文件中的内容

```

## 文本处理

```

cat file1 file2 ... | command <> file1_in.txt_or_file1_out.txt go

cat file1 | command( sed, grep, awk, grep, etc...) > result.txt
#合并一个文件的详细说明文本，并将简介写入一个新文件中
cat file1 | command( sed, grep, awk, grep, etc...) >> result.txt
#合并一个文件的详细说明文本，并将简介写入一个已有的文件中

grep Aug /var/log/messages           #在文件 '/var/log/messages' 中查找以 Aug 开头的行
grep ^Aug /var/log/messages          #在文件 '/var/log/messages' 中查找以 Aug 开头的行
grep [0-9] /var/log/messages         #选择 '/var/log/messages' 文件中所含数字的行
grep Aug -R /var/log/*               #在目录 '/var/log' 及随后的目录中查找 Aug

sed 's/string1/string2/g' example.txt
#将example.txt文件中的 "string1" 替换成 "string2"
sed '/^$/d' example.txt              #从example.txt文件中删除所有空行
sed '/^#/d; /$#/d' example.txt      #从example.txt文件中删除所有注释行

```



```

echo 'esempio' | tr '[:lower:]' '[:upper:]'      #合并上下单元格内
sed -e '1d' result.txt                          #从文件example.txt 中排除第-
sed -n '/string1/p'                             #查看只包含词汇 "string1"的
sed -e 's/ *$//' example.txt                    #删除每一行最后的空白字符
sed -e 's/string1//g' example.txt
#从文档中只删除词汇 "string1" 并保留剩余全部
sed -n '1,5p;5q' example.txt                    #查看从第一行到第5行内容
sed -n '5p;5q' example.txt                      #查看第5行
sed -e 's/00*/0/g' example.txt                  #用单个零替换多个零

cat -n file1                                     #标示文件的行数
cat example.txt | awk 'NR%2==1'                 #删除example.txt文件中的所
echo a b c | awk '{print $1}'                   #查看一行第一栏
echo a b c | awk '{print $1,$3}'                #查看一行的第一和第三栏
paste file1 file2                               #合并两个文件或两栏的内容
paste -d '+' file1 file2                       #合并两个文件或两栏的内容，中间用"+"区

sort file1 file2                                #排序两个文件的内容
sort file1 file2 | uniq                         #取出两个文件的并集(重复的行只保留
sort file1 file2 | uniq -u                      #删除交集，留下其他的行
sort file1 file2 | uniq -d                      #取出两个文件的交集(只留下同时存在-

comm -1 file1 file2                             #比较两个文件的内容只删除 'file1' 所包含的
comm -2 file1 file2                             #比较两个文件的内容只删除 'file2' 所包含的
comm -3 file1 file2                             #比较两个文件的内容只删除两个文件共有的部分

```

## 字符设置和文件格式转换

```

dos2unix filedos.txt fileunix.txt              #将一个文本文件的格式从MSD
unix2dos fileunix.txt filedos.txt              #将一个文本文件的格式从UNI
recode ..HTML < page.txt > page.html          #将一个文本文件转换成html
recode -l | more                               #显示所有允

```

## 文件系统分析

```

badblocks -v /dev/hda1                         #检查磁盘hda1上的坏磁块
fsck /dev/hda1                                 #修复/检查hda1磁盘上linux文件系统的

```

```

fsck.ext2 /dev/hda1      #修复/检查hda1磁盘上ext2文件系统的完整性
e2fsck /dev/hda1        #修复/检查hda1磁盘上ext2文件系统的完整性
e2fsck -j /dev/hda1      #修复/检查hda1磁盘上ext3文件系统的完整性
fsck.ext3 /dev/hda1      #修复/检查hda1磁盘上ext3文件系统的完整性
fsck.vfat /dev/hda1      #修复/检查hda1磁盘上fat文件系统的完整性
fsck.msdos /dev/hda1     #修复/检查hda1磁盘上dos文件系统的完整性
dosfsck /dev/hda1        #修复/检查hda1磁盘上dos文件系统的完整性

```

## 初始化一个文件系统

```

mkfs /dev/hda1           #在hda1分区创建一个文件系统
mke2fs /dev/hda1         #在hda1分区创建一个linux ext2的文件系统
mke2fs -j /dev/hda1      #在hda1分区创建一个linux ext3(日志型)的文件系统
mkfs -t vfat 32 -F /dev/hda1 #创建一个 FAT32 文件系统
fdformat -n /dev/fd0      #格式化一个软盘
mkswap /dev/hda3          #创建一个swap文件系统

```

## SWAP文件系统

```

mkswap /dev/hda3          #创建一个swap文件系统
swapon /dev/hda3          #启用一个新的swap文件系统
swapon /dev/hda2 /dev/hdb3 #启用两个swap分区

```

## 备份

```

dump -0aj -f /tmp/home0.bak /home #制作一个 '/home' 目录的完整备份
dump -laj -f /tmp/home0.bak /home #制作一个 '/home' 目录的交互式备份
restore -if /tmp/home0.bak         #还原一个交互式备份

```

```

rsync -rogpav --delete /home /tmp #同步两边的目录
rsync -rogpav -e ssh --delete /home ip_address:/tmp
rsync -az -e ssh --delete ip_addr:/home/public /home/local #
rsync -az -e ssh --delete /home/local ip_addr:/home/public #

```

```

dd bs=1M if=/dev/hda | gzip | ssh user@ip_addr 'dd of=hda.gz'
#通过ssh在远程主机上执行一次备份本地磁盘的操作
dd if=/dev/sda of=/tmp/file1

```

#备份磁盘内容到一个文件

```
tar -Puf backup.tar /home/user 执行一次对 '/home/user'
```

#目录的交互式备份操作

```
( cd /tmp/local/ && tar c . ) | ssh -C user@ip_addr 'cd /hom
```

#通过ssh在远程目录中复制一个目录内容

```
( tar c /home ) | ssh -C user@ip_addr 'cd /home/backup-home &&
```

#通过ssh在远程目录中复制一个本地目录

```
tar cf - . | (cd /tmp/backup ; tar xf - )
```

#本地将一个目录复制到另一个地方, 保留原有权限及链接

```
find /home/user1 -name '*.txt' | xargs cp -av --target-directory:
```

#从一个目录查找并复制所有以 '.txt' 结尾的文件到另一个目录

```
find /var/log -name '*.log' | tar cv --files-from=- | bzip2 >
```

#查找所有以 '.log' 结尾的文件并做成一个bzip包

```
dd if=/dev/hda of=/dev/fd0 bs=512 count=1
```

#做一个将 MBR (Master Boot Record)内容复制到软盘的动作

```
dd if=/dev/fd0 of=/dev/hda bs=512 count=1
```

#从已经保存到软盘的备份中恢复MBR内容

## 光盘

```
cdrecord -v gracetime=2 dev=/dev/cdrom -eject blank=fast -force
```

#清空一个可复写的光盘内容

```
mkisofs /dev/cdrom > cd.iso #在磁盘上创建一个并
```

```
mkisofs /dev/cdrom | gzip > cd_iso.gz #在磁盘上创建一个压缩了
```

```
mkisofs -J -allow-leading-dots -R -V "Label CD" -iso-level 4 -o
```

#创建一个目录的iso镜像文件

```
cdrecord -v dev=/dev/cdrom cd.iso #刻录一个
```

```
gzip -dc cd_iso.gz | cdrecord dev=/dev/cdrom - #刻录一个压缩了的
```

```
mount -o loop cd.iso /mnt/iso #挂载-
```

```
cd-paranoia -B #从一个CD光盘转录音轨到 wav 文件
```

```
cd-paranoia -- "-3" #从一个CD光盘转录音轨到 wav 文件中
```

```
cdrecord --scanbus #扫描总线以识别scsi通道
```

```
dd if=/dev/hdc | md5sum #校验一个设备的md5sum编码, 例如一张 CD
```

## 网络 - (以太网和WIFI无线)

```
ifconfig eth0          #显示一个以太网的配置
ifup eth0              #启用一个 'eth0' 网络设备
ifdown eth0           #禁用一个 'eth0' 网络设备
ifconfig eth0 192.168.1.1 netmask 255.255.255.0          #控制IP地址
ifconfig eth0 promisc   #设置 'eth0' 成混杂模式以嗅探数据包 (s
dhclient eth0          #以dhcp模式启用 'eth0'

route -n               #查看路由表
route add -net 0/0 gw IP_Gateway          #配置默认网关
route add -net 192.168.0.0 netmask 255.255.0.0 gw 192.168.1.1
#配置静态路由到达网络'192.168.0.0/16'
route del 0/0 gw IP_gateway              #删除静态路由

hostname #查看机器名
host www.example.com                    #把一个主机名解析到一个网际地址或把一个
nslookup www.example.com              #用于查询DNS的记录，查看域名解析是否正常，
ip link show                          #查看网卡信息
mii-tool                              #用于查看、管理介质的网络接口的状态
ethtool                              #用于查询和设置网卡配置
netstat -tupl                         #用于显示TCP/UDP的状态信息
tcpdump tcp port 80                   #显示所有http协议的流量
```

END

推荐好文

[强大，10k+点赞的 SpringBoot 后台管理系统竟然出了详细教程！](#)

[为什么MySQL不推荐使用uuid或者雪花id作为主键？](#)

[为什么建议大家使用 Linux 开发？爽（外加七个感叹号）](#)

[IntelliJ IDEA 15款 神级超级牛逼插件推荐（自用，真的超级牛逼）](#)

[炫酷，SpringBoot+Echarts实现用户访问地图可视化（附源码）](#)

[记一次由Redis分布式锁造成的重大事故，避免以后踩坑！](#)

[十分钟学会使用 Elasticsearch 优雅搭建自己的搜索系统（附源码）](#)

