

# 1 Linux基础

Linux和Windows是目前两个主流的操作系统。Linux与Windows迥然不同的风格和操作习惯往往是熟悉Windows的用户接触、接受Linux最大的障碍。在这部分，我们将向您展示Linux与Windows最主要的区别所在，以及快速上手的一些基本技能。

## 1.1 命令行

Linux是一个基于命令行的操作系统，这是与Windows在操作上最大的区别<sup>1</sup>。尽管Linux的图形界面发展已经相当完善，但命令行仍然并将一直是Linux不可或缺的重要部分。Linux的很多功能通过命令行实现起来可以方便许多，即使你刚开始不这么认为。

在进一步了解Linux的命令之前，我们需要先了解一下shell。Linux的shell是用户界面与系统内部交互的“中介”，用来接收用户的命令并将其“翻译”给系统底层。准确地说，图形界面也是一种shell。但我们一般所说的shell都特指文本终端（Terminal）。

在shell环境下所输入的Linux命令，通常包括“应用程序名称”、“选项”、“参数”。即：

```
$ ProgramName [options] [preference]
```

其中选项用来调整或指定程序的行为；参数则提供程序运行所需要的数据（通常是I/O位置）。例如：

```
$ ls -l Customs
```

这是一个显示Customs（可以是目录也可以是文件）详细信息的命令。在此例中，ls即为应用程序名称（显示目录或文件的信息。尽管它非常简单，但它的确是一个应用程序）；-l是选项（显示详细信息）；Customs则是我们指定的参数，即待操作数据；\$是shell的提示符。

同一个命令，有时候可以存在多个选项，表示同时采取多种限制条件。例如：

```
$ tar zvfz myfile.tar.gz
```

是一个解压缩文件包的命令。这里的zvfz是四个选项，分别表示“调用gzip程序”、“显示处理过的文件”、“操作对象是一个压缩包”以及“解压缩”。

有时候，我们也可以不指定任何选项，也不指定任何参数。程序会以它的默认方式运行，例如：

```
$ top
```

用来列出系统的动态进程列表，它就可以不带任何选项和参数运行。

Linux区分大小写，一般的命令（除了自定义的一些脚本）都是小写格式。也就是说，如果你误把ls输成LS终端就将报错。为了方便上手，我们为这篇论文配备了一份Linux基础命令列表。参照这张表，相信可以很快掌握Linux的基本操作。

需要了解某个命令的详细用法可以键入：

```
$ CommandName --help
```

即可列出该程序的所有可用选项及其含义。

## 1.2 普通用户和超级用户

Linux是多用户操作系统。Linux系统里的每一位用户，各有彼此不同的用户名，各拥有系统的一部分私有专区供个人工作。听起来这跟Windows差不多，Windows也允许有不同的用户，也各有各的“某某某的文档”。然而两者的本质区别在于：Linux允许多名用户同时登录到系统上，没有干扰地进行各自的工作；而Windows则不能。

在允许不同用户同时使用系统的基础上，Linux有一个特别指定的用户，其用户名为root，俗称“超级用户”（superuser）或根用户。它是“凌驾”于其他用户（普通用户）之上的，root用户拥有在系统上执行任何工作的特权。普通用户受到限制，虽然他们可以运行大部分程序，但是他们通常只能修改属于自己的私人文件，并且无法执行一些影响到系统内部的程序（如杀死某个进程）。超级用户则不然，它有权在任意时间任意地点创建、修改和删除任何文件，并运行所有程序，还包括限制普通用户的权限。

<sup>1</sup>在本质上的区别显而易见是一个开源而另一个专有。

很明显，root用户也就是通常的系统管理员（类似于Windows下Administrator但绝不一样），Linux系统中的系统管理命令只有root才能够成功运行，也只有root用户可以在除/home目录外的其它目录做任何改动<sup>2</sup>。

在Linux系统中，普通用户可以通过输入root密码成为root用户。事实上，root用户一般都另外拥有一个普通账户，只有在需要进行一些管理工作时才转换到root权限。要成为root用户，只需使用su命令。root用户的命令提示符是#：

```
$ su
Password:  *****          输入root密码
#
```

在Ubuntu系统和基于BSD UNIX的MAC OSX中，情况比较特殊。它们使用sudo命令作为系统管理的工具，不再需要切换为root用户（这对于短暂的系统管理工作非常节省时间）：

```
$ sudo mycommand
Password:  *****          输入用户密码
```

即可以普通用户的身份执行相应的root操作。其它发行版本也可以使用sudo，但需要事先运行visudo命令编辑好sudoers文件。

### 1.3 Linux的图形界面与模块化

Linux并不是人们想象的那样只有DOS般黑乎乎的界面和枯燥难懂的命令行。恰恰相反，Linux有非常棒的图形界面，并且功能齐全，完全可以和Windows的图形界面媲美。Novell最近开发的XGL更是以其炫目的透明、3D效果赢得大片掌声。

对于危言耸听的Linux纯字符界面，我们要说的是，图形界面和命令行相结合才是Linux下最常见的操作习惯。图形界面为主，命令行为辅。主要工作都是在图形界面下完成，只有在进行一些必要的系统管理工作（如安装软件、修改配置）时才需要用到命令行。因为文本终端可以让我们更方便地切换到root用户。

目前，可供Linux选择的图形环境主要有：KDE、GNOME和XFCE。

Linux的图形界面与Windows有很大不同：Linux的图形界面是一个单独的程序，称为X，图形环境如GNOME等基于X运行；而Windows的图形界面虽然也是一个程序，但它是集成到系统内核的。这两种截然不同的实现方式使得Linux可以单独使用图形界面，也可以单独使用命令行，也可以两者结合使用。而Windows在系统启动时图形界面即随之启动，直到关机。这两者各有优缺点，主要集中在稳定性和速度方面。Linux的图形界面独立于内核，因此它的崩溃与否不影响内核的运行，因而我们说Linux是相对稳定的；而Windows由于将图形界面和内核捆绑，一旦图形界面崩溃，系统也随之崩溃，即著名的蓝屏现象。另一方面，将图形界面与内核捆绑有速度优势，因为两者的程序被很好的“嵌”在了一起，这一点Linux逊色一些。但实际上，由于Windows的过分臃肿，它的运行速度往往不及Linux。

由于Linux的模块化，初次接触Linux会有“天昏地暗”的感觉，因为很多在Windows下点击几下鼠标的事在Linux下有时会复杂得不可思议。这是Linux为人诟病的地方，却也是熟悉Linux的用户引以为荣的地方<sup>3</sup>。这种情况主要体现在应用程序安装、字体配置等方面。比如音乐播放器，Linux中的音乐播放器需要安装单独的解码器才可以工作，而Windows只需要几次点击。Windows的字体几乎不需要任何设置，而Linux却要单独安装中文字体并且进行相应设置才行。

造成这种情况的原因很多，总结下来主要有两点：一是Linux的应用程序由世界各地的志愿者开发，模块化设计是必须的，最终公开的版本是整合起来的结果；二是版权问题，由于Linux本身的开源性质，它不可能包含那些需要巨额许可费用的库、字体等，而Windows显然是可以通过购买来完成这些工作。

### 1.4 Linux下的软件

在Linux系统中，软件不再是由.exe程序提供，而是封装包。封装包，顾名思义就是包含有程序源代码或者预编译源代码的压缩文件。目前主流的软件包有三种：Tarball、.rpm文件和.deb文件。

#### Source Tarball

即源码包。这类包通常被封装为.tar.gz、.tar.bz2或.tgz之类的格式。说它们是封装包其实不够准确，因为它们只是软件的作者们将软件的源代码打包压缩而成。在tarball内包含源代码树，makefile项目管理配置文件等等。安装这类软件的基本步骤如下：

<sup>2</sup>关于目录权限等问题，请参看系统管理部分之“文件系统”和“文件属性与权限”。

<sup>3</sup>因为这使得他们有更多选择的余地，有更多拼装的乐趣。Linux的高度可配置性亦来源于此。

- 用tar、unzip、unrar等工具解开压缩包。
- 进入解开的目录，阅读README/INSTALL文档。
- 依照说明文档调用GNU开发工具设定编译选项（./configure）。
- `sudo make && make install`执行安装。

这类软件包的优点是版本很新，而缺点是安装后不容易管理。

## RedHat Package Manager

RPM是Red Hat公司开发的包系统，也是目前最广为使用的包系统<sup>4</sup>。RPM包的扩展名为.rpm，这类软件包包含有已经编译好的软件、软件包信息以及该软件所需的其他相关文件。另外还有一种常见的RPM包文件是.src.rpm，它们封装的是没有预编译的源代码，可以通过安装这类包得到源代码再自行构建相应平台的.rpm文件。

安装RPM包使用rpm命令，常用选项：

- i 安装软件包
- v 显示版本信息
- h 显示进度条
- r 删除软件包
- qa 列出系统中所有软件包

RPM包的来源很多，除了系统本身的光盘外，各发行版的下载中心包含了很多光盘上没有的软件包和相应的升级包。还有下列网站：

```
http://freshmeat.net/
http://freshrpms.net/
http://rpmfind.net/
http://sourceforge.net/
http://rpm.pbone.net/
```

也可以提供软件包查找和搜索服务。

## Debian Advanced Package Tools

对应于Red Hat的rpm包，发行版Debian也有属于自己的封装格式：.deb。和rpm封装一样，deb包也是将预编译好的软件和包文件信息封装而成。处理deb包的底层命令为dpkg。

APT（Advanced Package Tools）是处理deb包的上层工具，它可以自动下载deb格式的软件包并调用低一级的dpkg工具进行安装和配置。目前APT在Debian以及基于Debian（如Ubuntu）的发行版中提供，在基于RPM的系统也可以额外安装APT作为第三方软件包管理器。

使用APT安装软件，常用的命令有：

```
$ sudo apt-get update           //更新软件列表
$ sudo apt-get upgrade          //安装有更新的软件包
$ sudo apt-get dist-upgrade      //版本间升级
$ sudo apt-get install packagename //安装指定的软件包
$ sudo apt-get remove packagename //删除指定的软件包
```

有时候遇到单独的deb包的安装（没有网络连接，或者要装的包源中没有的时候），需要手工调用dpkg命令：

```
$ sudo dpkg -i packagename.deb //安装软件包
$ sudo dpkg -r packagename.deb //删除软件包
$ sudo dpkg -l | grep packagename //查询已安装的某个软件包的信息
```

## 其他

除了上述的三种软件包格式外，还有一些软件（以非GPL软件为主）有它们自己的安装方式。有些是提供iso镜像供挂载后运行安装脚本；有些是bin的二进制文件直接运行；还有的甚至是直接解压缩即可。在安装这类软件时，只需认真阅读它所附带的说明文档，或者是其网站上的安装说明即可。

<sup>4</sup>Fedora、SuSE和Mandriva等著名发行版都是基于RPM包管理系统。

## 1.5 vi编辑器

vi是UNIX系统上经典的屏幕编辑器，Linux中包含一个vi的增强版本，称为vim。为了方便，我们一般直接称呼vim为vi。之所以辟出一部分单独讲述vi，是因为它实在是太重要了。vi基于文本终端，因此可以在图形界面没有运行的情况下支持文本编辑。

虽然我们有gedit这样方便的图形编辑器，但是为了应对一些突发情况如系统无法启动、图形界面崩溃等等，vi允许我们在文本模式下修复受损的配置文件来恢复系统。这类情况下，vi是必不可少的。更重要的是，vi本身的确是一个极其强大的编辑器，尽管初学者都会觉得它繁琐、枯燥，但它最终都会获得用户的青睐。双手不用离开键盘就可以快速编辑复杂的文本，的确是一件令人兴奋的事情！因为vi实在太强大，详细讲解它完全可以写一本书。我们在这里只简单地介绍它的基本用法，即新建、编辑和保存文件。

```
$ vi filename
```

即可打开vi编辑器。如果指定的文件不存在，则自动新建一个文件。vi编辑器有两种基本工作模式：命令模式和编辑模式。

### 命令模式

一旦打开文件，vi就进入命令模式。在命令模式中，常见的操作有：

- 激活编辑模式
- 发送编辑命令
- 移动光标至文件的不同位置
- 保存或退出文件

常用的命令有：

```
/      搜索一个匹配字符串，如/hello
:w      保存文件
:wq     保存并退出文件
:q!    不保存修改，强行退出文件
:q     在未作修改的前提下，退出
dd     删除光标所在的行
u      撤销最后一次修改
```

### 编辑模式

在编辑模式中，用户可以在文件中输入新的文本，使用方向键移动光标。我们常用下面几种命令激活编辑模式：

```
i   在光标之前插入
I   在行首插入
o   在当前行下面新增一行
O   在当前行上面新增一行
```

只有在命令模式下键入命令才有效。如果当前是编辑模式，那么输入的命令只会被当作相应的字符处理。按Esc键可以退出编辑模式并回到命令模式。

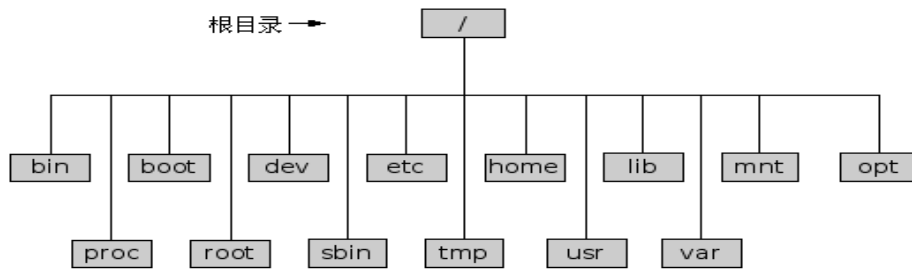
## 2 系统管理

### 2.1 文件系统

人类对计算机的操作，实际上就是对文件的操作；对文件的操作，实际上就是对磁盘的操作。因此，这一部分我们讨论Linux的文件系统，作为Linux系统管理的开篇。

几乎所有操作系统中都有文件的概念，文件是系统中用来存储信息的基本结构。在计算机系统中存有大量的文件，如何有效地组织和管理它们，并为用户提供一个方便操作它们的接口，这些都是文件系统的工作。

在Linux中文件和存储设备的使用与Windows中不同。虽然同样有文件和分层目录结构，但对Linux还是需要建立一种全新的思维模式。下面的图给出了典型的Linux目录结构：



在Linux中没有驱动器字母，不再有C盘、D盘之分。在Linux中，只有一个文件结构。它以root (/)开始，所有本地文件系统、本地设备、以及远程文件系统都表示为这个结构中的子目录。

当Linux第一次引导时，内核根据/etc/fstab文件中的信息构建文件结构。Windows为硬盘驱动器分区和其他存储设备分配驱动器字母；而Linux则是在root文件结构中为它们分配目录。

### fstab文件

Filesystem table（文件系统表）是系统中用来记录安装哪些文件系统的配置文件，定义在/etc/fstab：

```
$ cat /etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/hda5 / ext3 defaults, errors=remount-ro 0 1
/dev/hda1 /boot ext3 defaults 0 2
/dev/hda7 /home ext3 defaults 0 2
/dev/hda6 none swap sw 0 0
/dev/hdc /media/cdrom0 udf,iso9660 user,noauto 0 0
```

在这个文件中，第一列指出的是要装载的设备。第二列是装载点，也就是设备在文件系统中的位置。第三列指出了文件系统的类型。第四列是处理文件系统的选项。最后两列是文件系统的标志位。第一个数字是1 或者0，指定系统是否应该用dump（系统备份的一个选项）进行复制。第二个数字是0、1 或者2，指定了在引导时检查文件系统的次序。0 表示完全不检查。1 表示要最先检查，root (/) 文件系统需要指定为1。其他文件系统应该是2。

除了配置为noauto的那些文件系统外，其余所列的所有文件系统都在引导时被安装。在本例中，根分区 (/)、家目录 (/home) 和引导分区 (/boot) 在系统启动时安装。/proc文件系统（与设备无关）自动挂载。CD-ROM驱动器不自动安装，而是定义在fstab文件中，系统检测到有光盘放入时，将读取fstab文件自动挂载光盘。

举例我们要在Windows，Linux双启动的计算机上，让Linux启动时自动挂载一个Windows分区/dev/hda5（一般就是D盘）到/mnt/WinE。则可以在/etc/fstab里添加一行：

```
/dev/hda5 /mnt/WinE vfat defaults, umask=0022 0 2
```

即可将文件系统为FAT32的Windows分区挂载到文件系统的目录树上，其中的umask=0022指定了挂载的分区普通用户可写。

### Linux目录结构

Linux文件系统采用树形结构，文件按类型放置在不同目录下，而不是采用注册表管理。主要的目录用途如下：

/	根目录，包含下面的目录。
/bin	存放可执行二进制文件。
/boot	内核镜像以及启动所需的文件等。
/dev	存放设备文件。
/etc	存放系统的配置文件。
/home	用户文件的主目录，用户数据存放在各自主目录中。
/lib	各类库文件、头文件等。
/mnt	存放临时挂载的文件系统。
/opt	保存大型单独安装的软件，如数据库软件。
/proc	从内存中映射出来的虚拟信息文件系统。
/root	超级用户的主目录。
/sbin	存放系统管理程序，只有root才能执行
/tmp	临时文件目录。
/usr	包含了一般不需要修改的应用程序，命令程序文件、程序库、手册和其它文档。
/var	包含系统产生的经常变化的文件，如日志文件，软件库更新等。

## proc文件系统

proc（process information pseudo-filesystem）即虚拟进程信息文件系统。它是一种内核和内核模块用来向进程发送信息的机制，这个伪文件系统是直接从内存中映射出来的，可以让你和内核内部数据结构进行交互，获取有关进程的有用信息，甚至可以在运行中改变内核的运行参数。

proc通常被挂载在/proc，/proc目录下的文件可以用于访问有关内核的状态、计算机属性、正在运行的进程的状态等信息。下面是一些重要文件：

/proc/cpuinfo	CPU的信息（型号、类别、缓存大小等）。
/proc/meminfo	物理内存、交换分区等的信息。
/proc/filesystems	支持的文件系统。
/proc/mounts	已挂载的文件系统列表。
/proc/devices	可用的设备。
/proc/modules	已加载的模块。
/proc/version	内核版本。
/proc/cmdline	系统启动时内核命令行参数。

/proc中的文件远不止这些，其它还有一些重要系统模块的运行状态、当前系统的进程信息等等。想要进一步了解的话可以对其中的文件逐一查看。举一个简单的例子，如：

```
$ cat /proc/acpi/thermal_zone/THRM/temperature
```

查看当前CPU的温度。

## df [options] [disk devices | files | directories]

df（disk free）程序可显示特定分区的总容量、已用空间、挂载点。如果指定某文件或目录，df就显示该文件或目录所属的文件系统信息；若不指定任何参数，df则直接报告所有已挂载的文件系统：

```
$ df / /home /boot
Filesystem 1K-blocks    Used  Available  Use%  Mounted on
/dev/hda5   10555012  3955276   6063572   40%   /
/dev/hda7   26589196  8089488  17149048   33%  /home
/dev/hda1   120411    13791    100196   13%   /boot
```

### 常用选项

- k -m 以千字节（默认值）或兆字节为容量单位。
- h 自动选择合适的容量单位。-h选项使用的单位基数是1024；
- H 而-H使用的单位基数是1000。
- l 只显示本地文件系统，不显示NFS、SMB之类的网络文件系统。
- t type 显示特定类型的文件系统。
- x type 不显示特定类型的文件系统。

## **mount [options] device | directory**

mount命令使某硬件存储设备（通常是硬盘分区，如/dev/sda1）挂载到文件系统树的某个安装点上，也就是某个现有的目录下（例如/mnt/sda1），让用户可通过该安装点访问该存储设备：

```
$ sudo mkdir /mnt/sda1
$ sudo mount -t vfat -o iocharset=utf8 /dev/sda1 /mnt/sda1
```

mount命令有数不清的选项和用法，我们常用的是-t和-o选项。-t用来指明要挂载的分区文件系统类型；-o为额外添加的选项，如字符编码等。在上述命令中，mount命令指定要挂载的分区为Microsoft扩展的FAT文件系统，I/O字符编码为utf8。

大多数Linux发行版支持自动挂载，即光盘、U盘插入后，系统可以自动识别、分配相应的设备名称。mount命令用得越来越少，只有在部分移动存储设备不幸无法被系统自动识别，或者是挂载Windows分区的时候才需要用到。

## **umount [options] [device | directory]**

与mount对应，存在一个umount命令用来卸载存储设备。umount不能卸载使用中的设备，比如当前目录所在的分区、该分区上文件正在被使用等等。对于插入式存储媒介（光盘、U盘），在未卸载之前不应该擅自抽离存储媒介，否则有可能造成该设备上文件系统的损坏。

假定我们挂载了/dev/sda1到某个目录下，卸载它只需要知道设备名称即可，至于挂载点在哪不需要考虑：

```
$ sudo umount /dev/sda1
```

使用-a选项可一次卸载所有已挂载的文件系统：

```
$ sudo umount -a
```

**注意：**该命令会在正常关机时自动运行，在使用计算机时不得执行，因为它将使目前已挂载的（包括系统根分区在内）所有分区全部卸载。

## **2.2 文件属性与权限**

前面说过，Linux的文件属性与权限和Windows下有很大区别。一个Linux系统中可能具有多个可供不同用户同时登录的账号。为了维护私密与安全性，每位用户都只能访问系统中的部分文件。

对于任意文件和目录，用户可以分为三种身份：owner, group, other。文件的操作权限分为三种：read, write, executable。目录的操作权限与文件类似，只是含义有所不同：“可读”表示可以看到目录中包含的文件；“可写”表示可以新增或删除该目录下的文件；“可执行”则表示可以用cd命令进入该目录。

用户身份和其对应的权限合称为访问模式。使用ls的-l选项可以查看文件或目录的访问模式，在输出的信息中，最左侧的10个字符就是该文件或目录的访问模式。首字符代表文件的类型，而其余的9个字符以三个为一组，分别代表“拥有者”、“用户组”、“其他人”的文件操作权限。用r、w和x分别表示“可读”、“可写”和“可执行”。例如：

```
$ ls -ld My\ music
drwxr-xr-x 19 leeyee users 4096 2006-07-25 20:37 My music
```

即表示My music这个目录的拥有者为leeyee，所属用户组为users。从10位访问模式可以知道，这个目录的拥有者leeyee拥有完整的权限，而users用户组的成员和其他用户都可以读取、进入该目录。但是都没有写权限，不能在这个目录下增删文件。

## **stat [options] files**

stat命令能显示文件或文件系统的重要属性。文件信息的形式如下：

```
$ stat AddCamera.c
File: 'AddCamera.c'
Size: 1424          Blocks: 8          IO Block: 4096   regular file
Device: 307h/775d  Inode: 1811611   Links: 1
```

```
Access: (0644/-rw-r--r--) Uid: ( 1000/ leeyee) Gid: ( 1000/ leeyee)
Access: 2006-07-19 21:48:24.000000000 +0800
Modify: 2006-07-19 16:46:17.000000000 +0800
Change: 2006-07-19 16:46:17.000000000 +0800
```

这些信息包括文件名称、字节数、所占用的块数、I/O单位、文件类型、该文件所在的设备类型与编号、Inode编号、硬链接的数量、访问模式、拥有者的UID与名称、所属组的GID与名称、最后一次访问时间、内容修改时间、状态改变时间等等。

### **chown [options] user spec files**

用chown (change owner) 命令可以设定文件或目录的所有权。使用-R选项可以指定当前目录以及这个目录下所有递归的文件和目录，例如：

```
$ sudo chown -R leeyee /usr/local
```

即将/usr/local以及该目录下所有文件的所有权从root交付给用户leeyee。

### **chmod [options] permissions files**

chmod (change mode) 命令用来设定文件与目录的访问模式。*permissions*参数有三种形式：

- --reference=*file*, 引用另一个文件的访问模式（可以理解为同步）。
- 四位数的八进制数字，以位形式指出文件的绝对权限：最左侧的数字表示特殊属性，一般不用指出，往右三位用数字4、2、1分别对应“可读”、“可写”和“可运行”。它们加起来和作为一个用户身份的执行权限。如：

```
$ chmod -R 755 My\ music
```

中的755即表示对于My music文件夹及其所有文件：拥有者拥有读、写和运行的完整权限（7 = 4 + 2 + 1）；同一用户组的用户和其他人拥有读和运行权限（5 = 4 + 1）。

- 一个或多个以逗号相隔的字符串，这些字符串描述绝对权限或相对权限。如：

```
$ chmod u+x,g-r,o-w ~/bin/mp3togg
```

修改~/bin/mp3togg的权限：拥有者增加执行权限；同组用户减去读权限；其他减去写权限。在执行前我们的mp3togg文件的信息是：

```
$ ls -l mp3togg
-rwxr-xrwx 1 leeyee leeyee 196 2006-07-10 19:30 mp3togg
```

而执行权限修改后：

```
$ ls -l mp3togg
-rwx--xr-x 1 leeyee leeyee 196 2006-07-10 19:30 mp3togg
```

同chown一样，使用-R可以递归地改变子目录的访问模式。

## **2.3 进程管理**

进程 (process) 是Linux操作系统的工作单位，我们所运行的每一个程序都是由一个或多个进程组成的。每个进程都有自己的标记，称为Process ID或PID。进程受操作系统内核的控制，由内核负责为各个进程分配内存和CPU的使用，因此进程的控制直接关系到系统运行的效率和稳定。Linux提供了几个可以观察并操纵进程的工具。



### **ps** [*options*]

ps用来显示“当前”系统所运行的进程的列表。查看当前所有进程：

```
$ ps aux
```

指定用户的所有进程（多用户登录的情况下）：

```
$ ps -U username
```

使用ps命令得到的进程信息十分丰富，甚至到了泛滥的程度。因此，我们要想从进程列表中筛选出有用信息，需要用到管道和grep命令。所谓管道，是类UNIX系统中所特有的输出方式，是将一个命令的输出用作另一个命令的输入的“通道”，用“|”表示。grep是Linux下最常用的搜索工具，它常被用来找出文件中符合特定模式的文本行。因此，我们在查看进程时，可以输入：

```
$ ps aux | grep xmms
```

快速找出xmms程序相关的进程信息。

### **top** [*options*]

top可用来监视最活跃的进程，它将给出一个动态的进程列表，大约每隔一秒会自动更新一次信息。

```
top - 11:01:19 up 1:11, 4 users, load average: 2.44, 2.07, 1.36
Tasks:  89 total,   2 running,  86 sleeping,   0 stopped,   1 zombie
Cpu(s): 96.3% us,  1.7% sy,  0.0% ni,  0.0% id,  2.0% wa,  0.0% hi,  0.0% si
Mem:    514328k total,  510180k used,   4148k free,  12560k buffers
Swap:   1204832k total,  17132k used,  1187700k free,   293104k cached
```

top是一个交互程序，在它运行的过程中，可以用几个单键指令来改变它的行为。如设定更新间隔（s）、隐藏闲置进程（i）等。要结束top运行，按q（quit）。

### **kill** [*options*] [*process ids*]

kill的作用是传送信号（signal）给进程。信号有很多种，不同的信号进程会有不同的反应。默认的kill信号是终止信号，使用-l或者-L参数可以列出所有可用的信号，包括挂起（suspend）、中断（interrupt）、销毁（crash）等。普通用户只可以控制自己开启的进程，只有root用户才拥有控制他人或系统进程的权限。

使用kill杀死进程时，需要知道进程的PID，可以用我们上面讲到的ps命令加管道搜索得到PID，例如：

```
$ ps aux | grep gaim
leeyee  4861  0.1  2.8 42368 14524 ?        S  11:57   0:05 gaim
$ kill 4861
```

即可找出名为gaim的进程，它的PID为4861。再使用kill命令就可以终止gaim进程了。不过，并非所有程序都可以这么轻松地杀死，有些顽固分子可能会“不理睬”这些信号。这种情况下，可以使用-9参数强行杀死进程：

```
$ kill -9 process_ID
```

这个方法属于所向披靡型，无论进程在执行什么操作它几乎都可以终止。这个参数适合于僵死的进程，或者程序没有反应的时候强行关闭它（非常类似于Windows下的Ctrl+Alt+Del）。这样强制结束的程序有可能会留下一些没有还原给系统的资源，从而在系统内残留一些临时文件，可以不必理会。

## **2.4 主机信息管理**

### **uname** [*options*]

uname命令可显示关于计算机的基本信息：

```
$ uname -a
Linux myUbuntu.seu 2.6.15-23-686 #1 SMP PREEMPT
Tue May 23 14:03:07 UTC 2006 i686 GNU/Linux
```

用-a参数可以显示的信息包括主机名称、Linux内核版本、CPU数目、内核编译日期、硬件平台以及操作系统名称等。uname命令输出的信息没有固定标准，在不同的硬件平台、不同的内核版本中会有不同格式的信息。

#### 常用选项

- a 所有信息
- s 只显示内核名称（默认）
- n 只显示主机名称
- r 只显示内核版本

### ifconfig interface

ifconfig是用于观察和设定系统网络接口参数的工具，使用ifconfig命令可以方便地控制系统的网络接口，免去了打开图形配置工具的麻烦。在终端下运行ifconfig命令可以看到系统上所有打开的网络接口的配置信息。

这里面通常包含了网卡的MAC地址、IP地址、局域网广播地址、网络掩码、接收与发送数据量统计等信息。

设定特定网络接口的IP参数（以eth0为例）

```
$ sudo ifconfig eth0 192.168.0.100 netmask 255.255.255.0
```

关闭特定网络接口

```
$ sudo ifconfig eth0 down
```

重新启动特定网络接口

```
$ sudo ifconfig eth0 up
```

更多细节请参考手册页。

### uptime

uptime可以给出系统从开机到目前为止持续运行的时间。

```
$ uptime
20:54:41 up 3:35, 3 users, load average: 0.36, 0.53, 0.58
```

从左至右，各字段的意义分别为：当前时间（20:54:41）、系统已运行时间（3:35）、当前登录的人数（3）、一分钟内负载均值（0.36）、五分钟内负载均值（0.53）、十五分钟负载均值（0.58）。负载均值表征的是单位时间内活动的进程的平均数量。