Linux基础

# Vmware

<http://www.softown.cn/post/13.html>

VMware网络连接模式—桥接、NAT以及仅主机模式的详细介绍和区别。

# 初识linux

## 初始化安装

Ubuntu16.04 LST 64位安装步骤如下所示：

<http://jingyan.baidu.com/article/c275f6ba07e269e33d756714.html>

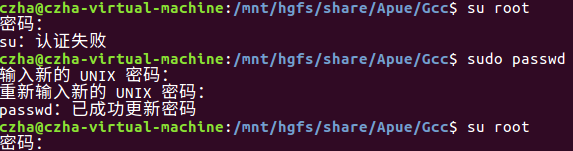
<https://docs.vmware.com/cn/VMware-Workstation-Pro/12.0/com.vmware.ws.using.doc/GUID-08BB9465-D40A-4E16-9E15-8C016CC8166F.html>

包括：系统安装、Vmware Tools安装、联网、与windows文件夹共享

## 设置root密码

安装完Ubuntu 16.04后默认是没有主动设置root密码的，无法使用root权限。

终端输入：



## 屏幕显示

在初始化安装过程中，安装完vmware tools后，只需要设置“查看-自动调整大小-自动适应窗口”即可完成屏幕自适应显示。

## 控制台切换

Linux是一个多用户操作系统，它可以同时接受多个用户登录。Linux还允许一个用户进行多次登录，这是因为Linux和UNIX一样，提供了虚拟控制台的访问方式，允许用户在同一时间从不同的虚拟控制台进行多次登录。虚拟控制台的选择可以通过按下Ctrl+Alt+一个功能键来实现，通常使用F1-F7，例如，用户登录后，按一下Ctrl+Alt+F2键，用户又可以看到"login:"提示符，说明用户看到了第二个虚拟控制台。然后只需按Ctrl+Alt+F1键，就可以回到第一个虚拟控制台。 用户可以在某一虚拟控制台上进行的工作尚未结束时，切换到另一虚拟控制台开始另一项工作。

# 第六章 Linux的文件权限与目录配置

## 1 基础概念

文件所有者：文件所有者对某些文件具有特定的操作权限，别人无法访问；

用户组概念：两个团体都有各自的成员，团体A与团体B竞争，无法访问各自内容，但团体内部成员可以互相访问。

以下3个文件时Linux系统里账号、密码、用户组信息的集中地，不能随便删除。；

/etc/passwd 记录所有系统上的账号与一般身份用户，包括root的相关信息

/etc/shadow 记录个人的密码

/etc/group 记录所有的组名

## 2 Linux文件属性

**访问权限**

系统中的每个文件和目录都有访问许可权限，用它来确定**谁**可以通过何种方式对文件和目录进行访问。

谁：

有三种不同类型的用户可对文件或目录进行访问：文件所有者，与所有者同组的用户、其他用户。所有者一般是文件的创建者。

何种方式：

访问权限分为**只读，只写和可执行**三种。

查看文件：

root@localhost ~]# ls -al

总用量 72

dr-xr-x---. 6 root root 4096 10月 12 09:39 .

dr-xr-xr-x. 18 root root 238 9月 6 21:18 ..

-rw-------. 1 root root 1775 2月 15 2017 anaconda-ks.cfg

-rw-------. 1 root root 15029 10月 11 22:39 .bash\_history

-rw-r--r--. 1 root root 18 12月 29 2013 .bash\_logout

-rw-r--r--. 1 root root 176 12月 29 2013 .bash\_profile

-rw-r--r--. 1 root root 176 12月 29 2013 .bashrc

drwx------. 4 root root 31 2月 15 2017 .cache

drwx------. 6 root root 58 9月 14 21:05 .config

-rw-r--r--. 1 root root 100 12月 29 2013 .cshrc

drwx------. 3 root root 25 2月 15 2017 .dbus

-rw-r--r--. 1 root root 1806 2月 15 2017 initial-setup-ks.cfg

-rw-------. 1 root root 54 10月 11 16:37 .lesshst

drwxr-xr-x. 3 root root 19 9月 11 17:58 .local

-rw-r--r--. 1 root root 129 12月 29 2013 .tcshrc

-rw-------. 1 root root 9943 10月 11 17:49 .viminfo

-rw-------. 1 root root 132 10月 12 09:39 .xauthkNJaFf

-rw-------. 1 root root 132 9月 12 14:21 .xauthU4kGtI

[root@localhost ~]#

[ 1 ] [ 2 ] [ 3 ] [ 4 ] [ 5 ] [ 6 ] [ 7 ]

权限 连接 所有者 用户组 文件容量 修改日期 文件名

（即文件数）

### 第一列

第一列共有 10 个字符；

第一个字符指定了文件类型：目录、文件或链接文件等：

d 表示目录

- 表示普通文件

l 表示连接文件（linkfile），类似于快捷方式

b 表示设备文件里面的可供存储的接口设备

c 表示设备文件里面的串行端口设备，例如键盘、鼠标（一次性读取设备）。

s 套接字，数据接口文件

p FIFO文件，pipe

接下来的字符中，每3个一组，按照“rwx”的顺序，某个权限没有，则该位置为“-”。

组1： 表示文件所有者的权限

组2： 表示与文件所有者同组的用户的组权限

组3： 表示系统中其他用户的权限

### 第二列

第二列表示有多少文件名连接到此节点（i-node）

每个文件都会将它的权限与属性记录到文件系统的i-node中，不过我们使用的目录树却是使用文件名来记录，因此每个文件名就会连接到一个i-node。

这个属性记录的就是有多少不同的文件名连接到相同的一个i-node号码。

### 第三列

第三列表示这个文件（或目录）的“所有者账号”。

### 第四列

第四列表示这个文件的所属用户组。

### 第五列

第五列为这个文件的容量大小，默认单位为B。

### 第六列

第六列为这个文件的创建日期或者是最近的修改日期。

### 第七列

第七列为该该文件名。

比较特殊的是：如果文件名之前多一个“ . ”，则代表这个文件为“隐藏文件”，例如.bash\_history。

## 3 文件权限与用户组的重要性

**（1）文件权限的重要性**

主要体现在“数据安全性”上面。比如/etc/shadow这个账号管理文件，由于其记录了系统中所有账号的数据，当然不能让所有人读取（否则密码会被窃取），只有root才能读取。

**（2）用户组的重要性**

团队开发软件或数据共享功能

如果你有一个软件开发团队，在你的团队中，你希望每个人都可以使用某一些目录下的文件，而非你的团队的其他人则不予以开发，怎么做到呢？

举例：testgroup的团队共有三个人，分别是test1，test2，test3，那么我们就可以将团队所需的文件权限限定为[-rwxrwx---]来提供给testgroup的工作团队使用。

**（3）未将权限设置妥当的危害**

举例：如果你的目录权限没有设置好的话，可能造成其他人都可以在你的系统上面乱来。

例如，本来只有root才能执行的开关机、ADSL拨号、新增或删除用户等的命令，若被你改成任何人都可以执行的话，那么如果用户不小心给你重新启动、重新拨号。那么你的系统就会莫名奇妙的挂掉了。而且万一你的用户密码被其他不明人士取得的话，只要他登录你的系统就可以轻而易举地执行一些root的工作。

因此，在**你修改你的Linux文件与目录的属性之前，一定要先搞清楚，什么数据时可变的，什么是不可变的**。

## 4 修改文件属性与权限

上面知道了文件权限对于一个系统的安全性的重要性了，也知道文件的权限对于用户与用户组的相关性，那么如何修改一个文件的属性与权限呢？又有多少文件的权限我们可以修改呢？

### chgrp

改变所属用户组：chgrp （change group）

语法：chgrp -选项 组名 文件（或目录）名

将文件或目录的组修改为指定的组。

前提：要被改变的组名必须要在/etc/group文件内存在才行，否则显示错误。

举例：

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root root 269 10月 2 23:53 sh01.sh

[root@localhost ShellFile]# chgrp czha sh01.sh

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root czha 269 10月 2 23:53 sh01.sh

[root@localhost ShellFile]# su czha

[czha@localhost ShellFile]$ ls -al sh01.sh

-rw-r--r--. 1 root czha 269 10月 2 23:53 sh01.sh

[czha@localhost ShellFile]$ vim sh01.sh

此时，如果以czha账号的身份试图去修改sh01.sh文件的内容，系统会提示该文件只读，不允许修改。

[root@localhost ShellFile]# chgrp nihao sh01.sh

chgrp: 无效的组："nihao"

由于，nihao这个组名不在/etc/group文件中，因此修改失败。

### chown

改变文件所有者：chown （change owner）

语法：chgrp -选项 账号名称 文件（或目录）名

将文件或目录的所有者改为指定的用户账号。

前提：要被改变的用户必须是已存在于系统中的账号，也就是在/etc/passwd这个文件中有记录的用户名称才能改变。

实例：

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root czha 269 10月 2 23:53 sh01.sh

[root@localhost ShellFile]# chown czha sh01.sh //将sh01.sh的所有者改为czha这个账号

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 czha czha 269 10月 2 23:53 sh01.sh

[root@localhost ShellFile]# su czha

[czha@localhost ShellFile]$ vim sh01.sh //此时，发现czha用户下，该文件也可以进行修改

[czha@localhost ShellFile]$ chown root:root sh01.sh

chown: 正在更改"sh01.sh" 的所有者: 不允许的操作

[czha@localhost ShellFile]$ su root

密码：

//将sh01.sh的所有者与用户组改回为root，注意是切换为root用户的权限下才能执行成功

[root@localhost ShellFile]# chown **root:root** sh01.sh

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root root 269 10月 12 15:33 sh01.sh

chown也可以单纯的修改用户组。

例如：

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root root 269 10月 12 15:33 sh01.sh

[root@localhost ShellFile]# chown **.**czha sh01.sh //“.”点号+组名

[root@localhost ShellFile]# ls -al sh01.sh

-rw-r--r--. 1 root czha 269 10月 12 15:33 sh01.sh

**使用场景**

假设你要讲.bashrc这个文件复制成为.bashrc\_test文件名，并且是要给bin这个用户：

cp .bashrc .bashrc\_test

由于复制行为（cp）会复制执行者的属性和权限，两个文件的属性一样，bashrc\_test还是属于root所有，即使你将文件给予bin这个用户了，那他仍然无法修改的，所以你必须要修改这个文件的所有者与用户组。

### chmod

文件权限的改变使用的是chmod这个命令，但是权限的设置方法有两种，分别可以使用数字或者是符号来进行权限的更改。

详细见书本。

* **数字类型改变文件权限**

r： 4

w： 2

x： 1

[owner group others]

例如：当权限为[-rwxrwx---]，分数则是：

owner = rwx = 4+2+1 = 7

group = rwx = 4+2+1 = 7

others = --- = 0+0+0 = 0

即：chmod 770 filename/dirname

* **符号类型改变文件权限**

从前面可知，共9个权限，分别是user、group、others 3种身份。那么我们可以通过u，g，o来代表三种身份的权限。此外a代表all，也即全部的身份。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chmod | u  g  o  a | +（加入）  -（除去）  =（设置） | r  w  x | 文件或目录 |

举例：设置一个文件的权限成为“-rwxr-xr-x”

chmod u=rwx,go=rx .bashrc （注意：u=rwx,go=rx是连在一起的，中间并没有任何空格）

举例：如果我不知道原先文件的属性，而我只想增加.bashrc这个文件的每个人均可写入的权限，则：

chmod a+w .bashrc

举例：去掉全部人的可执行权限呢？

chmod a-x .bashrc

目录与文件的权限意义

[root@localhost scripts]# ls -al

总用量 10480

drwxr-xr-x. 3 root root 4096 10月 2 19:54 .

……

drwxr-xr-x. 2 root root 4096 10月 10 16:03 ShellFile

[root@localhost scripts]# chmod a-x ShellFile/ //此时czha用户无法进入ShellFile目录作为工作目录

[root@localhost scripts]# su czha

[czha@localhost scripts]$ cd ShellFile/

bash: cd: ShellFile/: 权限不够

[czha@localhost scripts]$ chmod a+x ShellFile/ //czha用户可以进入了，因为others加了可执行权限x

chmod: 更改"ShellFile/" 的权限: 不允许的操作

[czha@localhost scripts]$ su root

密码：

[root@localhost scripts]# chmod a+x ShellFile

[root@localhost scripts]# su czha

[czha@localhost scripts]$ cd ShellFile

目录的x代表的是用户能否进入该目录成为工作目录的用途。

目录的w权限就与该目录下面的文件名变动有关。

## 5 Linux文件种类与扩展名

详细见书本

## 6 Linux目录配置

### Linux目录配置标准

为什么每套Linux distribution版本的配置文件、执行文件、每个目录内放置的内容其实都差不多？

为了避免造成混乱，有了一套标准，Filesystem Hierarchy Standard（FHS）标准。

其主要目的是：希望让用户可以了解到已安装软件通常放置于哪个目录下，所以希望独立的软件开发商、操作系统制造者以及想要维护系统的用户，都能够遵循FHS的标准。

事实上，FHS是根据过去的经验一直在持续改版的，FHS依据文件系统使用的频繁与否与是否允许用户随意改动，而将目录定义成为四种交互作用的形态，如下：

|  |  |  |
| --- | --- | --- |
|  | **可分享的（shareable）** | **不可分享的（unshareable）** |
| 不变的（static） | /usr（软件放置处） | /etc（配置文件） |
| /opt（第三方软件） | /boot（开机与内核文件） |
| 可变的（variable） | /var/mail（用户邮件信箱） | /var/run（程序相关） |
| /var/spool/news（新闻组） | /var/lock（程序相关） |

上面是一些代表性的目录，我们要了解的是，什么是那四个类型：

可分享的：可以分享给其他系统挂载使用的目录，所以包括执行文件与用户的邮件等数据，是能够分享给网络上其他主机挂载使用的目录。

不可分享的：自己机器上面运行的设备文件或者是与程序有关的socket文件等，由于仅与自身机器有关，所以当然就不适合分享给其他主机了。

不变的：有些数据是不会经常变动的，跟随着distribution而不变动。例如函数库、文件说明文件、系统管理员所管理的主机服务配置文件等。

可变的：经常改变的数据，例如登录文件、新闻组等。

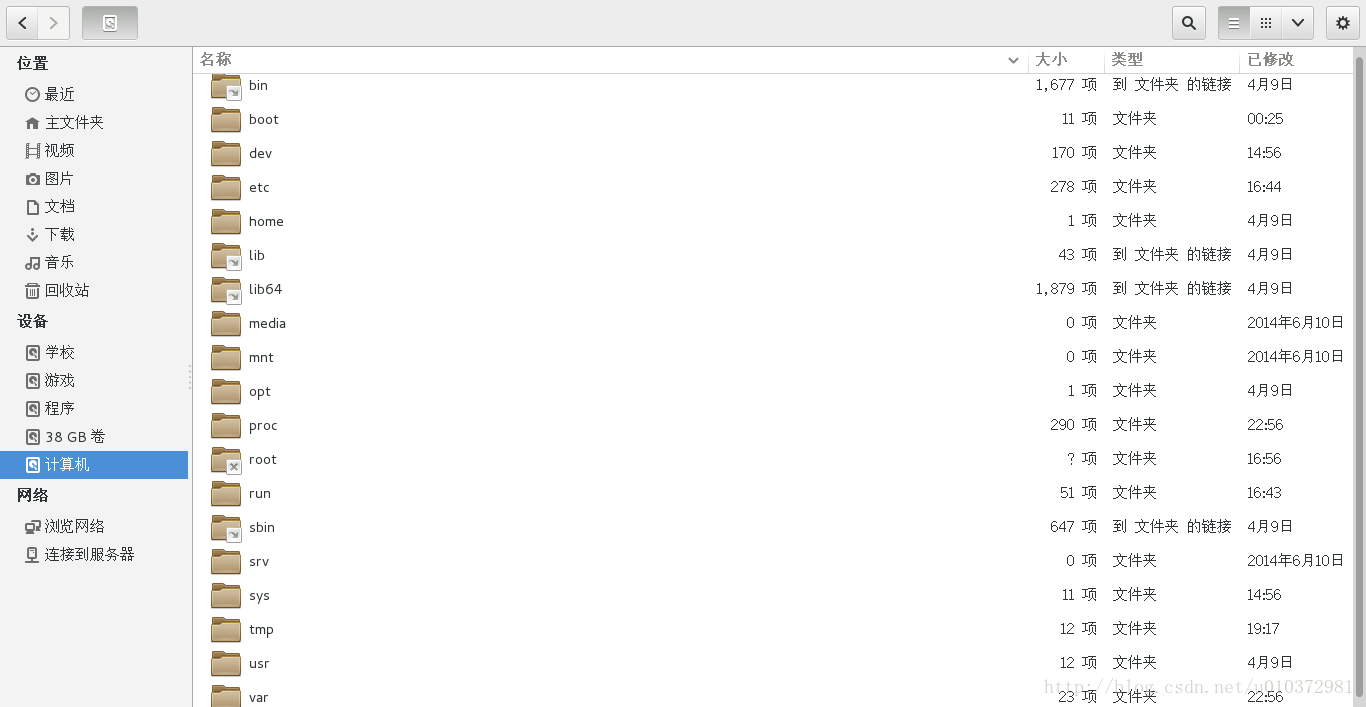
### 目录树

**摘自网络：**

<http://blog.csdn.net/u010372981/article/details/44983929>

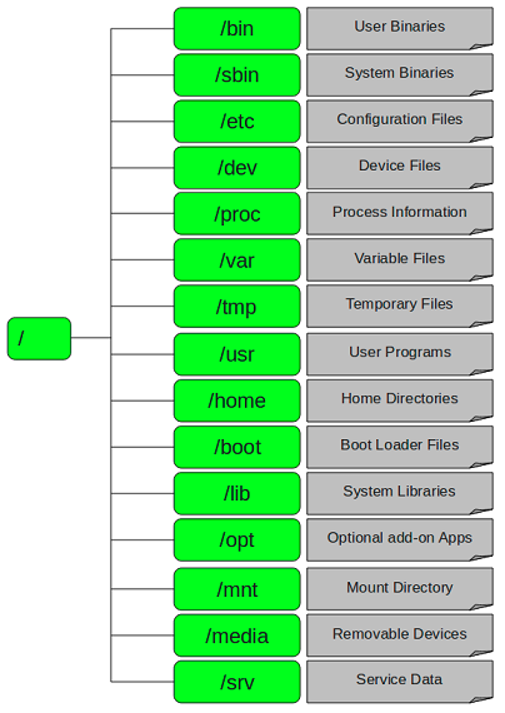
<http://www.cnblogs.com/JCSU/articles/2770249.html>

我最近安装了Centos7.0的版本，觉得相比于6系列有了很大的不同，但是这两个版本在根目录“/”下的文件夹结构还是大体上相同的，我的centos7文件结构截图如下：



首先，我要说明我在安装centos系统的过程中，勾选了GNOME的图形界面，功能当中勾选了办公工具和开发工具，办公工具就是类似于微软的office，现在在微软office2007以后的版本已经和linux的办公软件实现兼容，而如果装机的时候，勾选开发工具这个选项的话，系统就自动配置安装好了java环境，git环境等等。想要知道你想安装的开发工具是否安装在了系统当中，可以在终端中码java -version  git --version等等来查看安装的开发环境的信息。

开机之后，根目录下的文件夹目录就如上图所示。下面要说一说centos6和centos7中都有的目录，及其目录下的文件。



**1、/- 根**

每一个文件和目录从根目录开始。

只有root用户具有该目录下的写权限。请注意，/root是root用户的主目录，这与/.不一样

**2、/bin中 - 用户二进制文件**

包含二进制可执行文件。

在单用户模式下，你需要使用的常见Linux命令都位于此目录下。系统的**所有用户**使用的命令都设在这里。

例如：ps、ls、ping、grep、cp

**3、/sbin目录 - 系统二进制文件**

就像/bin，/sbin同样也包含二进制可执行文件。

但是，在这个目录下的linux**命令**通常**由系统管理员使用**，对系统进行维护。例如：iptables、reboot、fdisk、ifconfig、swapon命令

**4、/etc - 配置文件**

包含所有程序所需的配置文件。

也包含了用于启动/停止单个程序的启动和关闭shell脚本。例如：/etc/resolv.conf、/etc/logrotate.conf

**5、/dev - 设备文件**

包含设备文件。

这些包括终端设备、USB或连接到系统的任何设备。例如：/dev/tty1、/dev/usbmon0

**6、/proc - 进程信息**

包含系统进程的相关信息。

这是一个虚拟的文件系统，包含有关正在运行的进程的信息。例如：/proc/{pid}目录中包含的与特定pid相关的信息。

这是一个虚拟的文件系统，系统资源以文本信息形式存在。例如：/proc/uptime

**7、/var - 变量文件**

var代表变量文件。

这个目录下可以找到内容可能增长的文件。

这包括 - 系统日志文件（/var/log）;包和数据库文件（/var/lib）;电子邮件（/var/mail）;打印队列（/var/spool）;锁文件（/var/lock）;多次重新启动需要的临时文件（/var/tmp）;

**8、/tmp - 临时文件**

包含系统和用户创建的临时文件。

当系统重新启动时，这个目录下的文件都将被删除。

**9、/usr - 用户程序**

包含二进制文件、库文件、文档和二级程序的源代码。

/usr/bin中包含用户程序的二进制文件。如果你在/bin中找不到用户二进制文件，到/usr/bin目录看看。例如：at、awk、cc、less、scp。

/usr/sbin中包含系统管理员的二进制文件。如果你在/sbin中找不到系统二进制文件，到/usr/sbin目录看看。例如：atd、cron、sshd、useradd、userdel。

/usr/lib中包含了/usr/bin和/usr/sbin用到的库。

/usr/local中包含了从源安装的用户程序。例如，当你从源安装Apache，它会在/usr/local/apache2中。

**10、/home - HOME目录**

所有用户用home目录来存储他们的个人档案。

例如：/home/john、/home/nikita

**11、/boot - 引导加载程序文件**

包含引导加载程序相关的文件。

内核的initrd、vmlinux、grub文件位于/boot下。

例如：initrd.img-2.6.32-24-generic、vmlinuz-2.6.32-24-generic

**12、/lib - 系统库**

包含支持位于/bin和/sbin下的二进制文件的库文件.

库文件名为 ld\*或lib\*.so.\*

例如：ld-2.11.1.so，libncurses.so.5.7

**13、/opt - 可选的附加应用程序**

opt代表可选的。

包含从个别厂商的附加应用程序。

**附加应用程序**应该安装在/opt/或者/opt/的子目录下。

**14、/mnt - 挂载目录**

临时安装目录，系统管理员可以挂载文件系统。

**15、/media - 可移动媒体设备**

用于挂载可移动设备的临时目录。

举例来说，挂载CD-ROM的/media/cdrom，挂载软盘驱动器的/media/floppy;

**16、/srv - 服务数据**

srv代表服务。

包含服务器特定服务相关的数据。

例如，/srv/cvs包含cvs相关的数据。

### 绝对路径与相对路径

# 第七章 Linux文件与目录管理

## 目录与路径

注意：所有目录下面都会存在的两个目录，分别是“.”与“..”，分别代表此层与上层目录的意思。

例题：请问在Linux下面，根目录下有没有上层目录（..）存在？

答：若使用“ls -al /”去查询，可以看到根目录下确实存在“.”与“..”两个目录，再仔细查阅，可以发现这两个目录的属性与权限完全一致，这代表根目录的上一层（..）与根目录自己（.）是同一个目录。

### 目录的相关操作

几个常见的处理目录的命令：

cd： 切换目录（change directory）

pwd： 显示当前目录（print working directory）

mkdir： 新建一个新的目录

rmdir： 删除一个空的目录

### 关于执行文件路径的变量：PATH

为什么我可以在任何地方执行/bin/ls这个命令呢？并且不会说找不到该命令呢？这是因为环境变量PATH的帮助。

当我们执行一个命令的时候，举例来说“ls”好了，系统会依照PATH的设置去每个PATH定义的目录下查询文件名为ls的可执行文件，如果在PATH定义的目录中含有多个文件名为ls的可执行文件，那么先查询到的同名命令先被执行。

* 不同身份用户默认的PATH不同，默认能够随意执行的命令也不同（如root与vbird）；
* PATH是可以修改的，所以一般用户还是可以通过修改PATH来执行某些位于/sbin或/usr/sbin下的命令来查询；
* 使用绝对路径或相对路径直接指定某个命令的文件名来执行，会比查询PATH来得正确；
* 命令应该要放置在正确的目录下，执行才会方便；
* 本目录（.）最好不要放到PATH当中。

## 文件与目录管理

查看文件与目录：ls

复制、删除与移动：cp，rm，mv

**复制文件**

* 比如我们在进行备份的时候，某些需要特别注意的特殊权限文件，例如密码文件（/etc/shadow）以及一些配置文件，就不能直接以cp来复制，而必须要加上-a或是-p等可以完全复制文件权限的参数才行。
* 另外，如果你想要复制文件给其他用户，也必须要注意到文件的权限（包括读、写、执行以及文件所有者等），否则，其他人还是无法对你给予的文件进行修订的操作。

总之，由于cp有种种的文件属性与权限的特性，所以，在复制时，你必须要清楚了解到： （1）是否需要完整保留来源文件的信息？

（2）源文件是否为软连接文件（symbolic link file）？

（3）源文件是否为特殊的文件，例如FIFO、socket等？

（4）源文件是否问目录？

**basename与dirname**

[czha@localhost scripts]$ basename /server/scripts/m2.sh

m2.sh

[czha@localhost scripts]$ dirname /server/scripts/m2.sh

/server/scripts

## 文件内容查阅

cat：由第一行开始显示文件内容

tac：从最后一行开始显示，可以看出tac是cat的倒写形式。

nl：显示的时候，顺便输出行号。

more：一页一页地显示文件内容。

less：与more类似，但是more更好的是，它可以往前翻页！

head：只看头几行。

tail：只看尾巴几行。

od：以二进制的方式读取文件内容。

**文件时间**

每个文件在linux下面都会记录许多的时间参数，其实有三个主要的变动时间。那么这三个时间的意义是什么呢？

* **modification time（mtime）**

当该文件的“内容数据”更改时，就会更新这个时间。内容数据指的是文件的内容，而不是文件的属性和权限。

* **status time（ctime）**

当该文件的“状态”（status）改变时，就会更新这个时间。举例来说，像是权限与属性被更改了，都会更新这个时间。

* **access time（atime）**

当“该文件的内容被取用”时，就会更新这个读取时间（access）。举例来说，我们使用cat去读取/etc/man.config，就会更新该文件的atime了。

**非存文本文档：od**

## 文件与目录的默认权限与隐藏权限

除了基本 r, w, x 权限外，在Linux传统的 Ext2/Ext3/Ext4 文件系统下，我们还可以设定其他的系统隐藏属性，这部份可使用 chattr 来设定，而以 lsattr 来查看，最重要的属性就是可以设定其不可修改的特性！让连文件的拥有者都不能进行修改！ 这个属性可是相当重要的，尤其是在安全机制上面(security)！比较可惜的是，在 CentOS 7.x 当中利用 xfs 作为预设文件系统， 但是 xfs 就没有支持所有的 chattr 的参数了！仅有部份参数还有支持而已！

### umask

**当你新建一个新的文件或目录时，它的默认权限会是什么？**

umask就是指定“目前用户在新建文件或目录时候的权限默认值”，指定方式如下：

### chattr

### lsattr

### 查看文件类型：file

如果你想知道某个文件的基本数据类型，例如是ASCII或者是data文件，或者是binary，且其中没有使用到动态库函数（share library）等等的信息，就可以利用file这个命令来查看。

file xxx

## 命令与文件的查询

**脚本文件名的查询**

* which（查找“执行文件”）

which ifconfig

which which

which cd

**文件名的查找**

先whereis，locate，再find

whereis，locate是利用数据库来查找数据，所以相当快速，没有实际查询硬盘，比较节省时间；

find是很强大的命令，由于是直接查找硬盘，花费时间较多。

**更新数据库**：updatedb

# 4 服务

FTP（File Transfer Protocol）文件传输协议

TFTP（Trivial File Transfer Protocol）简单文件传输协议

Telnet

SSH

Linux守护进程——xinetd <http://www.cnblogs.com/zhouhbing/p/5106675.html>

文件传输协议(FTP)实际上就是传输文件的协议，它可以应用在任意两台主机之间，但是FTP不仅仅是一个协议，它同时也是一个程序。**作为协议**，FTP是被应用程序所使用的；而**作为程序**，用户需要通过手动方式来使用FTP并完成文件的传送。FTP允许执行对目录和文件的访问，并且可以完成特定类型的目录操作，例如将文件重新定位到不同的目录中。显然，FTP是与Telnet合作一同来完成对FTP服务器的登录操作，并在这之后再开始提供文件传送服务的。

然而，通过FTP访问主机这只是第一步。随后，用户必须通过一个由系统管理员为保护系统资源而设置的安全登录认证，这个认证需要输入正确的口令和用户名。但是，也可以通过使用用户名“anonymous” 来尝试登录，当然，通过这种方式完成登录后，所能访问的内容将会受到某些限制。即使FTP可以被用户以应用程序的方式来使用，FTP的功能也只限于列表和目录操作、文件内容输人，以及在主机间进行文件拷贝。它不能远程执行程序文件。

简单文件传输协议(TFTP)是FTP的简化版本，只有在你确切地知道想要得到的义件名及它的准确位置时，才可有选择地使用TFTP。TFTP是一个非常易用的、快捷的程序！TFTP并不提供像FTP那样的强大功能。TFTP不提供目录浏览的功能，它只能完成文件的发送和接收操作。这个紧凑的小协议在传送的数据单元上也是节省的，它发送比FTP更小的数据块，同时它也没有FTP所需要的传送确认，因而它是不可靠的。正是由于这个内在的安全风险，事实上只有很少的站点支持TFTP服务。

FTP 是完整、 面向会话、常规用途文件传输协议。而 TFTP 用作 bones bare - 特殊目的文件传输协议。

交互使用 FTP。 TFTP 允许仅单向传输的文件。

FTP 提供身份验证。而TFTP 不。

FTP 使用已知 TCP 端口号： 20 的数据和 21 用于连接对话框。 TFTP 用于 UDP 端口号 69 其文件传输活动。

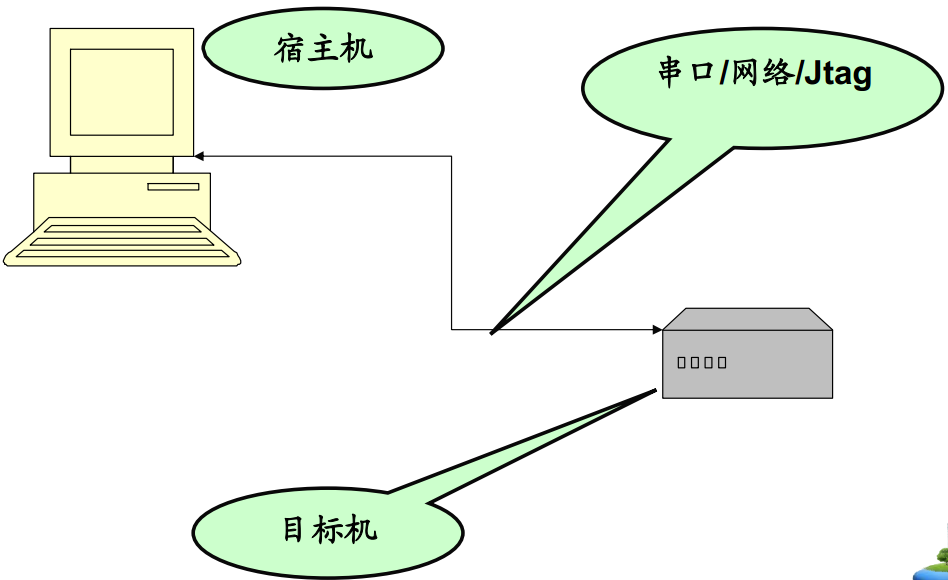
因为 TFTP 不支持验证 WindowsNT ，所以FTP 服务器服务不支持 TFTP。

FTP 依赖于 TCP，是面向连接并提供可靠的控件。 TFTP 依赖 UDP， 需要减少开销， 几乎不提供控件。

## Tftp服务器

为什么需要TFTP服务器？

在嵌入式开发过程中有宿主机和目标机的角色之分：宿主机是执行编译、链接嵌入式软件的计算机；目标机是运行嵌入式软件的硬件平台。



结论：

TFTP服务器作为工作于**宿主机**上的软件，主要提供对目标机的主要映像文件的下载工作。

## NFS介绍

网络文件系统（NFS，Network FileSystem）是一种将远程主机上的分区（目录）经网络挂载到本地的一种机制，通过对网络文件系统的支持，用户可以在本地系统上像操作本地分区一样来对远程主机的共享分区（目录）进行操作。（类似于windows共享目录）

## Samba

**博客**：<http://www.linuxidc.com/Linux/2016-06/132609.htm>

### 网络设置

<http://blog.csdn.net/gxw255613/article/details/48489743>

在Vmware的编辑->虚拟网络编辑器中将网络模式设置为桥接（具体作用见Vmware章节）;

在Centos中将ip地址设置为静态IP，与windows设置为同一个网段：

（1）临时设置

在命令行输入：ifconfig ens33(网卡名) ip地址 netmask 子网掩码，只能临时指定网卡的IP地址，但重启后会失效。

（2）永久设置

vim /etc/sysconfig/network-scripts/ifcfg-ens33，修改IPADDR=192.168.0.154 和 GATEWAY=192.168.0.1，

修改完毕后，输入service network restart，该命令会重启network服务，使配置文件生效。状态必须为OK，否则有问题。

此时，虚拟机和Windows主机均能相互ping通，OK。

**查看路由器IP的方法**：<https://jingyan.baidu.com/article/fb48e8be475c196e622e14b1.html>

### 安装SAMBA

（1）安装前检查是否已安装Samba，#rpm –qa | grep samba

（2）#yum installsamba , yum会自动查找软件的关联性

（3）#vi /etc/samba/smb.conf:配置文件已键值对的形式存在

### 配置SAMBA

目录：cd /etc/samba/

备份：cp smb.conf smb.conf.bak

设置：vim smb.conf

参考博客：

<http://blog.csdn.net/xiaoxinna1/article/details/48708665>

<http://www.linuxidc.com/Linux/2016-06/132609.htm>

在/home/cha/目录下新建一个文件夹:mkdir smbroot，设置其属性为可读可写可执行：chmod -R 777 /home/czha/smbroot

新建工作组，WORKGROUP：groupadd WROKGROUP；

设置smbroot目录的工作组为WORKGROUP：chgrp WORKGROUP /home/czha/smbroot

配置/etc/samba/ smb.conf文件：cp smb.conf smb.conf.bak，

[global]

workgroup = WORKGROUP

server string = Samaba Server %v

netbios name = centos

security = user

map to guest = bad user

dns proxy = no

[smbroot]

path=/home/czha/smbroot

available = yes

browseable = yes

writable = yes

public = yes

read only = no

重启samba服务

systemctl restart smb //一般执行该指令即可

systemctl reload smb

systemctl status smb

关闭防火墙和SELinux

关闭防火墙

systemctl stop fitewalld

systemctl disable firewalld

关闭SELinux

vi /etc/selinux/config 修改 SELINUX=disabled

设置开机启动samba

chkconfig smb on

修改完后reboot重启

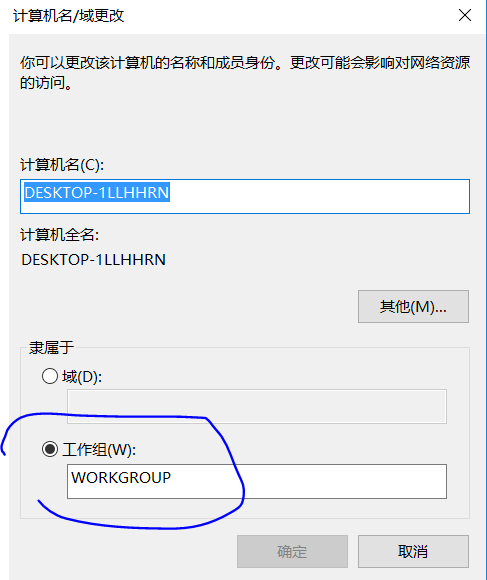
### 配置windows

* **修改windosw工作组名**

怎么查看和修改windows的工作组名呢？右键此电脑（计算机）->属性->工作组，即可查看：



怎么修改呢？进入高级系统设置->计算机名->更改，或者直接在上图中点击更改设置选项：



打开 此电脑->右键-> 添加一个网络位置->自定义网络位置 -> [\\192.168.0.154\smbroot](file:///\\192.168.0.154\smbroot)即可。

## 日志

Linux的系统日志文件：/var/log/messages，可以查看系统异常

**Q：ubuntu16.04没有/var/log/messages？**

编辑/etc/rsyslog.d/50-default.conf

其中有这么一段

\*.=info;\*.=notice;\*.=warn;\

auth，authpriv.none;\

cron，daemon.none;\

mail，news.none -/var/log/messages

这是本来就有，却被注释了。现在解注释就行。

保存后重启服务：sudo restart rsyslog

czha@ubuntu:/etc/rsyslog.d$ sudo vim 50-default.conf

修改完后：

czha@ubuntu:/etc/rsyslog.d$ sudo restart rsyslog

restart: 无法连接到 Upstart: Failed to connect to socket /com/ubuntu/upstart: 拒绝连接

解决办法：

czha@ubuntu:/etc/rsyslog.d$ sudo dpkg-divert --local --rename --add /sbin/initctl

czha@ubuntu:/etc/rsyslog.d$ sudo ln -s /bin/true /sbin/initctl

czha@ubuntu:/etc/rsyslog.d$ sudo restart rsyslog

## 时间同步服务器NTP

在使用CentOS系统的时候，我们可能会遇到时间不准的问题，需要调整时区并调整时间，那我们如何解决这个我问题呢，下面就来教大家一个CentOS系统时间同步的方法，希望大家可以解决自己所存在的疑问。

CentOS系统时间同步的步骤如下：

CentOS系统使用NTP来从一个时间服务器同步

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

ntpdate us.pool.ntp.org

解析：

* 第一句是把当前时区调整为上海就是+8区，想改其他时区也可以去看看/usr/share/zoneinfo目录;
* 然后第二句是利用ntpdate同步标准时间.

没有安装ntpdate的可以yum一下: yum install -y ntpdate

加入定时计划任务，每隔10分钟同步一下时钟

crontab -e

0-59/10 \* \* \* \* /usr/sbin/ntpdate us.pool.ntp.org | logger -t NTP

这样，我们就可以来解决在CentOS系统中时间不准确的问题了。

# 软件安装

**参考博文**：

<http://www.aboutyun.com/thread-9226-1-1.html>

我们在使用cloudera和openstack的时候，经常会看到yum、apt-get、rpm，那么它们是什么？下面介绍一下。

rpm是由红帽公司开发的软件包管理方式，使用rpm我们可以方便的进行软件的安装、查询、卸载、升级等工作。但是rpm软件包之间的依赖性问题往往会很繁琐，尤其是软件由多个rpm包组成时。

Yum（全称为 Yellow dog Updater， Modified）是一个在Fedora和RedHat以及SUSE中的Shell前端软件包管理器。基於RPM包管理，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载、安装。

rpm 是linux的一种软件包名称，以.rmp结尾，安装的时候语法为：rpm -ivh，rpm包的安装有一个很大的缺点就是文件的关联性太大，有时候装一个软件要安装很多其他的软件包，很麻烦，所以为此RedHat小红帽开发了yum安装方法，他可以彻底解决这个关联性（依赖）的问题，很方便，只要配置两个文件即可安装，安装方法是：yum -y install ，yum并不是一中包，而是安装包的软件

简单点说， rpm 只能安装已经下载到本地机器上的rpm包；yum能在线下载并安装rpm包，能更新系统，且还能自动处理包与包之间的依赖问题，这个是rpm 工具所不具备的。

## 简述

在Linux上有两种常见的软件管理器，分别是RPM与Debian的dpkg。CentOS主要以RPM为主，但也需要知道dpkg。

1. RedHat 系列：Redhat、Centos、Fedora 等

常见的安装包格式 rpm 包，安装rpm包的命令是“rpm -参数”

包管理工具 yum

支持 tar 包

1. Debian 系列：Debian、Ubuntu 等

常见的安装包格式 deb 包，安装 deb 包的命令是“dpkg -参数”

包管理工具 apt-get

支持 tar 包

## RPM

|  |  |  |  |
| --- | --- | --- | --- |
| distribution代表 | 软件管理机制 | 使用命令 | 在线升级机制（命令） |
| Red Hat/Fedora | RPM | rpm，rmpbuild | YUM（yum） |
| Debian/Ubuntu | DPKG | dkpg | APT（apt-get） |

**RPM与SRPM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件格式 | 文件名格式 | 直接安装与否 | 内含程序类型 | 可否修改参数并编译 |
| RPM | xxx.rpm | 可 | 已编译 | 不可 |
| SRPM | xxx.src.rpm | 不可 | 未编译的源代码 | 可 |

**rpm软件包的格式**

软件名称-软件版本信息-发布版本的次数.适合的硬件平台.扩展名

eg：rp-pppoe-3.1-5.i386.rpm

除了后面适合的硬件平台外，主要是以“-”来隔开各个部分。

* **软件名称**：每一个软件的名称，例如：rp-pppoe.++++656 ；
* **版本信息**：主版本与次版本，例如：上面的主版本为3，在主版本的架构下改动部分源代码内容而释放出一个新的版本，就是次版本；
* **发布版本的次数**：通常就是编译的次数，那么为何需要重复编译呢？这是由于在同一版的软件中，可能由于有些bug或者是安全上的顾虑，所以必须要进行小幅度的patch或重设一些编译参数，设置完成后重新编译并打包成RPM文件，因此就有不同的打包数出现了；

**RPM默认安装路径**

一般来说，RMP类型的文件在安装的时候，会先去读取文件内记载的设置参数内容，然后将该数据用来比较Linux系统的环境，以找出是否有属性以来的软件尚未安装的问题。例如Openssh这个联网软件需要通过Openssl这个加密软件的帮忙，所以得先安装openssl才能装openssh。那你的环境如果没有openssl，你就无法安装openssh。

若环境检查合格了，那么RPM文件就开始被安装到你的Linux系统上。安装完毕后，该软件相关的信息就会被写入/var/lib/rpm/目录下的数据库文件中了。上面这个目录内的数据很重要。因为未来如果我们有任何软件升级的需求，版本之间的比较就是来自于这个数据库，而如果你想要查询系统已经安装的软件，也是从这里查询的。同时，目前的 RPM 也提供数字证书信息，这些数字证书也是在这个目录内记录的呢！所以说，这个目录得要注意不要被删除了。

那么软件内的文件到底是放置到哪里去啊？当然与文件系统有关对吧！我们在第六章的目录配置谈过每个目录的意义，这里再次的强调：

|  |  |
| --- | --- |
| etc | 一些配置档放置的目录，例如 /etc/crontab |
| /usr/bin | 一些可运行文件 |
| /usr/lib | 一些程序使用的动态函式库 |
| /usr/share/doc | 一些基本的软件使用手册与说明档 |
| /usr/share/man | 一些 man page 文件 |

**RPM安装（install）**

安装软件是root的工作，需要root身份才能操作rpm这个命令。

命令：

rpm -i 需要安装的包文件名

举例如下：

rpm -i example.rpm 安装 example.rpm 包；

rpm -iv example.rpm 安装 example.rpm 包并在安装过程中显示正在安装的文件信息；

rpm -ivh example.rpm 安装 example.rpm 包并在安装过程中显示正在安装的文件信息及安装进度；

rpm -ivh <http://website.name/path/pkgname.rpm> 直接在网络上面的某个文件安装，以网址来安装

**RPM 查询操作（quary）**

RPM在查询的时候，其实查询的地方是/var/lib/rpm/这个目录下的数据库文件。另外，RPM也可以查询未安装的RPM文件内的信息。

命令：rpm -q …

* **查询已安装软件的信息：**

-q：仅查询，后面接的软件是否有安装；

-qa：列出所有的已经安装在本机Linux系统上面的所有软件名称；

-qi：列出该软件的详细信息（information），包含开发商、版本与说明等；

-ql：列出该软件所有的文件与目录所在完整文件文件名（list）；

-qc：列出该软件的所有设置文件（找出在/etc/下面的文件名而已）；

-qd：列出该软件的所有帮助文件（找出与man有关的文件名而已）；

-qR：列出与该软件有关的依赖软件所含的文件（Required的意思）；

-qf：由后面接的文件名称找出该文件属于哪一个已安装的软件

* **查询某个RPM文件内含有的信息**：

-qp[icdlR]：注意 -qp后面接的所有参数同上面的说明一致，但用途仅在于找出某个RPM文件内的信息，而非已安装的软件信息！注意！

在查询的部分，所有的参数之前都需要加上-q才是所谓的查询。查询主要分为两个部分，一个是查已安装到系统上面的软件信息，这部分的信息都是由/var/lib/rpm/所提供；另一个则是查某个rpm文件内容，等于是由RPM文件内找出一些要写入数据库内的信息，这部分就得要使用-qp（p是package的意思）。

举例如下：

rpm -qa | grep tomcat4 查看 tomcat4 是否被安装；

rpm -qip example.rpm 查看 example.rpm 安装包的信息；

rpm -qif /bin/df 查看/bin/df 文件所在安装包的信息；

rpm -qlf /bin/df 查看/bin/df 文件所在安装包中的各个文件分别被安装到哪个目录下；

详细见书籍P693

**RPM 卸载操作**

命令：

rpm -e 需要卸载的安装包

在卸载之前，通常需要使用rpm -q …命令查出需要卸载的安装包名称。

举例如下：

rpm -e tomcat4 卸载 tomcat4 软件包

**RPM 升级操作**

命令：

rpm -U 需要升级的包

举例如下：

rpm -Uvh example.rpm 升级 example.rpm 软件包

**RPM 验证操作**

命令：

rpm -V 需要验证的包

举例如下：

rpm -Vf /etc/tomcat4/tomcat4.conf

输出信息类似如下：

S.5....T c /etc/tomcat4/tomcat4.conf

其中，S 表示文件大小修改过，T 表示文件日期修改过。限于篇幅，更多的验证信息请您参考rpm 帮助文件：man rpm

**RPM 的其他附加命令**

--force 强制操作 如强制安装删除等；

--requires 显示该包的依赖关系；

--nodeps 忽略依赖关系并继续操作；

## YUM

### 查询、安装、升级、删除功能

* **查询功能**：yum [list|info|search|provides|whatprovides] 参数

如果想要查询利用 yum 来查询原版 distribution 所提供的软件，戒已知某软件的名称，想知道该软件的功能， 可以利用 yum 相关的参数为：

[root@www ~]# yum [option] [查询工作项目] [相关参数]

选项与参数：

[option]：主要的选项，包括有：

-y ：当 yum 要等待用户输入时，这个选项可以自动提供 yes 的响应；

--installroot=/some/path ：将该软件安装在 /some/path 而不适用默认路径

[查询工作项目] [相关参数]：这方面的参数有：

search ：搜寻某个软件名称或者是描述 (description) 的重要关键字；

list ：列出目前 yum 所管理的所有的软件名称与版本，有点类似 rpm -qa；

info ：同上，不过有点类似 rpm -qai 的执行结果；

provides：从档案去搜寻软件！类似 rpm -qf 的功能

yum search xxx-soft

yum info xxx-soft

yum list

yum list xx\*

yum list update

yum provides xxx-file

举例：利用yum的功能，找出以pam为开头的软件名称有哪些，而其中尚未安装的又有哪些。

答：yum list pam\*

* **安装升级功能：** **yum [install|update] 软件**

yum install xxx

yum update xxx

* **删除功能：yum [remove] 软件**

yum remove xxx

### yum的设置文件

/etc/yum.repos.d/ CentOS-Base.repo

### yum群组功能

yum grouplist

## 回收站Trash

<http://www.cnblogs.com/jkmiao/p/4418568.html>

## CentOS 7系统中安装Eclipse

<http://blog.csdn.net/gongxifacai_believe/article/details/53106179?locationNum=3&fps=1>

# 工具

## 快捷键

Ctrl + Tab键 窗口切换

## 网络设置

<http://www.linuxidc.com/Linux/2011-03/33873.htm>

## man手册

Linux提供了丰富的帮助手册，当你需要查看某个命令的参数时不必到处上网查找，只要man一下即可。

Linux 的man手册共有以下几个章节：

1、Standard commands （标准命令）

2、System calls （系统调用）

3、Library functions （库函数）

4、Special devices （设备说明）

5、File formats （文件格式）

6、Games and toys （游戏和娱乐）

7、Miscellaneous （杂项）

8、Administrative Commands （管理员命令

在ubuntu的shell中执行man man其中部分：

描述

man 是系统的手册分页程序。指定给 man 的 页

选项通常是程序、工具或函数名。程序将显示每一个找到的相关

手册页。如果指定了 章节，man 将只在手册的指定 章节

搜索。默认将按预定的顺序查找所有可用的 章节 (默认是“1 n l 8 3 2 3posix

3pm 3perl 5 4 9 6 7”，除非被 /etc/manpath.config 中的 SECTION

指令覆盖)，并只显示找到的第一个 页，即使多个 章节 中都有这个 页面。

下表显示了手册的 章节 号及其包含的手册页类型。

1

可执行程序或 shell 命令 （如ls 、cd等）

2

系统调用(内核提供的函数)

3

库调用(程序库中的函数) （C标准函数）

4

特殊文件(通常位于 /dev) （设备说明）

5

文件格式和规范，如

/etc/passwd

6

游戏

7

杂项(包括宏包和规范，如

man(7)，groff(7))

8

系统管理命令(通常只针对

root 用户)

9

内核例程 [非标准

一个手册 页面 包含若干个小节。

小节名称通常包括 NAME， 概述(SYNOPSIS)， 配置(CONFIGURATION)，

描述(DESCRIPTION)， 选项(OPTIONS)， 退出状态(EXIT STATUS)，

返回值(RETURN VALUE)， 错误(ERRORS)， 环境(ENVIRONMENT)， 文件(FILES)，

版本(VERSIONS)， 符合标准(CONFORMING TO)， 注(NOTES)， 缺陷(BUGS)，

示例(EXAMPLE)， 作者(AUTHORS)， 和 亦见(SEE ALSO).

以下规范适用于 概述(SYNOPSIS) 小节，也可作为其他小节的指南。

示例

man ls

显示 项目 (程序) ls 对应的手册页

## Vim

* 命令行模式：最初进入的一般模式，该模式下可以移动光标进行浏览，整行删除，但无法编辑文字。
* 插入模式：只有在该模式下，用户才能进行文字的编辑输入，用户可以使用[ESC]键回到命令行模式。
* 底行模式：该模式下，光标位于屏幕底行，用户可以进行文件保存或退出操作，也可以设置编辑环境，如寻找字符串、列出行号

**命令行模式功能键：**

• yy: 复制当前光标所在行

• [n]yy：n为数字，复制当前光标开始的n行

• p: 粘贴复制的内容到光标所在行

• dd：删除当前光标所在行

• [n]dd：删除当前光标所在行开始的n行

• /name：查找光标之后的名为“name”的字符串

• G：光标移动到文件尾（注意是大写）

• U: 取消前一个动作（注意是大写）

**底行模式功能键：**

• :w 保存

• :q 退出vi(系统会提示保存修改)

• :q! 强行退出（对修改不做保存）

• :wq 保存后退出

• :w [filename] 另存为filename的文件

• :set nu 显示行号

• :set nonu 取消行号

**假死：vim按了Ctrl + s后假死的解决办法**

使用vim时，如果你不小心按了 **Ctrl + s**后，你会发现不能输入任何东西了，像死掉了一般，其实vim并没有死掉，这时vim只是停止向终端输出而已，要想退出这种状态，只需按**Ctrl + q** 即可恢复正常。

Vim快速移动光标至行首和行尾

1、 需要按行快速移动光标时，可以使用键盘上的编辑键Home，快速将光标移动至当前行的行首。除此之外，也可以在命令模式中使用快捷键"^"（即Shift+6）或0（数字0）。

2、 如果要快速移动光标至当前行的行尾，可以使用编辑键End。也可以在命令模式中使用快捷键"$"（Shift+4）。与快捷键"^"和0不同，快捷键"$"前可以加上数字表示移动的行数。例如使用"1$"表示当前行的行尾，"2$"表示当前行的下一行的行尾。

a 移动到末尾字符的后面，并进入编辑模式

## gedit

Q: gedit显示行号?

编辑->首选项->显示行号。

Linux特殊符号大全

<http://blog.chinaunix.net/uid-16946891-id-5088144.html>

## Wireshark (ethereal)

sudo apt-get install wireshare

解决“couldn't run /usr/bin/dumpcap in child process permission denied”的问题

sudo chgrp czha /usr/bin/dumpcap