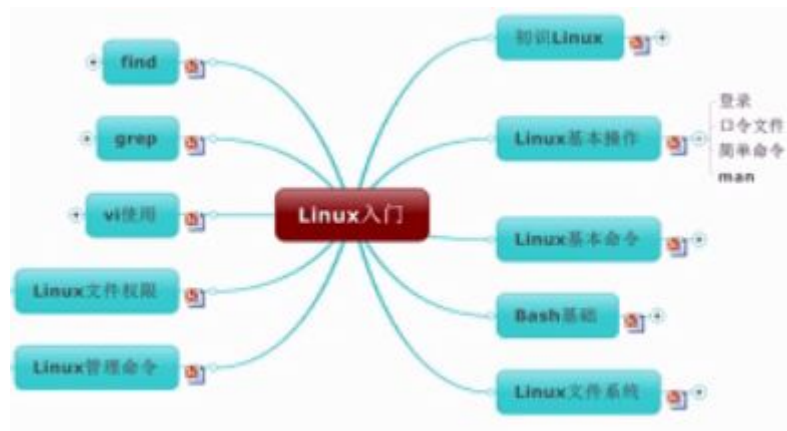


## 02 Linux 基本操作



### 1. Linux 系统的登录:

文本登陆方式和图像化界面的登录两种。首先看一下文本方式登录:

#### 【1】输入账号和密码

教程中是在 windows 下安装的 linux 系统虚拟机, 启动系统进入 linux 系统会提示输入用户名和密码, **密码是不会显示的**。回车登录成功, 并且进入了一个 shell 环境。

当输入完账号和密码之后, 系统是这样工作的:

【2】系统会找到 / etc/passwd 文件 (该文件存放账户信息, 这些用户信息包括用户名, 用户的 iD, 用户所属的组 ID, 用户的家目录以及用户所使用的一些 shell 程序) 和 etc/shadow (影子文件存方式的是与密码相关的信息, 该文文件存放的用户的密码以及密码的过期时间等等) 文件进行对照。当输入的用户名和密码验证通过之后, 系统会启动一个程序,

【3】也就是外壳程序。也就是一个 shell 程序。

【4】并且获得一个虚终端, 这样就可以与系统交互。

【5】并且进入到用户的家目录。

### 2. 口令文件的格式:

【1】username:用户名

【2】password:密码 (密码这项已经转移到了影子文件里面, 因此这一项不代表真正的密码)。

【3】uid:用户的标识号

【4】gid:组标识号 (用户所属的组)

【5】comment:备注信息 (可选的, 包含用户的其他一些注释信息)

【6】home; 家目录

【7】shell;shell 程序 (等我们登录成功之后, 要启动的 shell 程序。)

比如说我们可以看一下 root 用户口令文件的格式是这样的:

```
root:x: 0 : 0 : root:/root:/bin/bash
```

[1 用户名(root)]

[2 密码(x)]

[3 uid(0)]

[4 gid(0)]

[5 描述信息，备注信息（root），]这是可选的，可以为空

[6 家目录（root）]

[7 最后一项用户登录要启动的 shell 程序 bash（:/bin/bash）]linux 默认的 shell 程序是 bash。这些信息是保存在口令文件当中，也就是 `etc/passwd` 文件，可以用命令查看该文件的信息：通过命令：`cat /etc/passwd` 可以查看文件的内容。

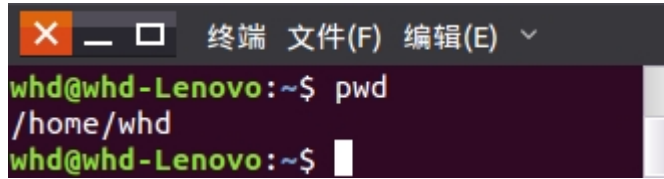
```
whd@whd-Lenovo:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
refox 网络浏览器 james:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:109::/home/syslog:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
uidd:x:106:111::/run/uidd:/bin/false
avahi:x:107:114:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
whoopsie:x:108:115::/nonexistent:/bin/false
avahi-autoipd:x:109:118:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:111:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:112:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:114:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:116:124:RealtimeKit,,,:/proc:/bin/false
saned:x:117:125::/var/lib/saned:/bin/false
usbmux:x:118:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
lightdm:x:119:126:Light Display Manager:/var/lib/lightdm:/bin/false
clickpkg:x:120:129::/nonexistent:/bin/false
whd:x:1000:1000:whd,,,:/home/whd:/bin/bash
```

如果显示的内容太长无法找到前面的行，可以通过 `cat /etc/passwd more` 来分页查看内容。我们可以发现第一行就是 root 用户的信息。也就是前面所讲的内容。虽然它叫做密码文件但是密码实际上不存放在这个文件当中，而是存放在 `etc/shadow`（影子文件中）。同样的我们可以看一下影子文件的内容：



```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:114:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:116:124:RealtimeKit,,,:/proc:/bin/false
saned:x:117:125::/var/lib/saned:/bin/false
usbmux:x:118:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
lightdm:x:119:126:Light Display Manager:/var/lib/lightdm:/bin/false
clickpkg:x:120:129::/nonexistent:/bin/false
whd:x:1000:1000:whd,,,:/home/whd:/bin/bash
whd@whd-Lenovo:~$ cat /etc/shadow
cat: /etc/shadow: 权限不够
whd@whd-Lenovo:~$ sudo cat /etc/shadow
[sudo] whd 的密码:
root:!:16993:0:99999:7:::
daemon*:16729:0:99999:7:::
bin*:16729:0:99999:7:::
sys*:16729:0:99999:7:::
sync*:16729:0:99999:7:::
games*:16729:0:99999:7:::
man*:16729:0:99999:7:::
lp*:16729:0:99999:7:::
mail*:16729:0:99999:7:::
news*:16729:0:99999:7:::
uucp*:16729:0:99999:7:::
proxy*:16729:0:99999:7:::
www-data*:16729:0:99999:7:::
backup*:16729:0:99999:7:::
list*:16729:0:99999:7:::
irc*:16729:0:99999:7:::
gnats*:16729:0:99999:7:::
nobody*:16729:0:99999:7:::
systemd-timesync*:16729:0:99999:7:::
systemd-network*:16729:0:99999:7:::
systemd-resolve*:16729:0:99999:7:::
systemd-bus-proxy*:16729:0:99999:7:::
syslog*:16729:0:99999:7:::
messagebus*:16729:0:99999:7:::
uuidd*:16729:0:99999:7:::
avahi*:16729:0:99999:7:::
whoopsie*:16729:0:99999:7:::
avahi-autoipd*:16729:0:99999:7:::
dnsmasq*:16729:0:99999:7:::
colord*:16729:0:99999:7:::
speech-dispatcher:!:16729:0:99999:7:::
hplip*:16729:0:99999:7:::
kernoops*:16729:0:99999:7:::
pulse*:16729:0:99999:7:::
rtkit*:16729:0:99999:7:::
saned*:16729:0:99999:7:::
usbmux*:16729:0:99999:7:::
lightdm*:16729:0:99999:7:::
clickpkg*:16731:0:99999:7:::
whd:$6$1225mnfh$.LODPGOTVZYGjV.c3JAQD2dBWs9PqUSqze53aJSv03G6r1AEcQanQ4C9H7Y3L1Ek
cxwoy5kwU8W9vjkpmNciM0:16993:0:99999:7:::
whd@whd-Lenovo:~$
```

可以看一下密码是加密过的密码，它的密码是加密过的，而用户输入的密码是明文，系统会对输入的密码加密，并且将加密过的密码和影子文件中的密码比较，验证通过的话，才允许它登录，并且启动一个 shell 程序。启动的这个 shell 文件就是口里文件当中所配置的默认的 shell。然后这个 shell 会获得一个控制终端，最后，然后将用户的目录更改到用户所对应的家目录，这个家目录也是存放在密码文件当中的。我们可以通过 pwd 命令查看当前路径（也就是 root 目录对应的家目录）：



```
whd@whd-Lenovo:~$ pwd
/home/whd
whd@whd-Lenovo:~$
```

这就是登录的过程。

### 3. Linux 登录时的 shell 环境文件:

Linux 登录完成之后, linux 还会执行一些配置文件, 进行一些环境的初始化, 这些文件大家稍微做一些了解, 比如说:

【1】 / etc/bashrc

包含系统定义的命令别名和 bash 的环境变量的定义。

【2】 /etc/profile

包含系统的环境定义并指定启动时必须运行的文件。

【3】 / etc/inputrc

包含系统的键盘设定, 以及针对不同终端程序的键位配置信息。

【4】 \$HOME/.bashrc

包含为用户定义的命令别名和 bash 的环境变量的定义。

【5】 \$HOME/.bash\_profile

(\$HOME 表示家目录) 包含为用户定义的环境变量, 并制定用户登录需要启动的程序。

【6】 ¥\$HOME/.inputrc

包含用户的键盘设定, 以及针对用户终端的键位配置信息。

其中前 3 个是针对全局的配置 (针对每一个用户登录时都要执行的配置文件, 是全局的) s h i, 后三个是针对不同用户的配置 (是局部的), 注意后面都有点号。这些东西了解就可以了。实际上登录是不止需要配置这些文件, 目前只需要指导这些就可以了。

### 4. 多终端登录:

实际上我们可以以通过图形界面登录, 如果我们在文本界面 (也就是命令行下), 可以通过 **startx** 命令转换到图形界面 (教程中的 redhat linux 默认的是进入文本界面)。

Ubuntu 系统不要轻易的使用该命令, 因为会把周围的菜单栏和一些使用工具隐藏掉, 只剩下桌面和几个图标, 而且重启之后无法正常进入系统。参考: [运行 startx 后 Ubuntu 密码正确进不去的问题](#)

如果想要从图像化界面直接转换到文本界面, 长按 **ctl+alt+backspace**.但是在 ubuntu 系统测试没有效果。Ubuntu 下两种界面的切换可以看一下博文

<http://blog.csdn.net/fightingforcv/article/details/52337155>

如果想进入系统后直接进入图像化文件可以配置一个文件, 在终端输入 **cat /etc/inittab** 命令:

```
#
# Default runlevel. The runlevels used by RMS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
14:3: initdefault:

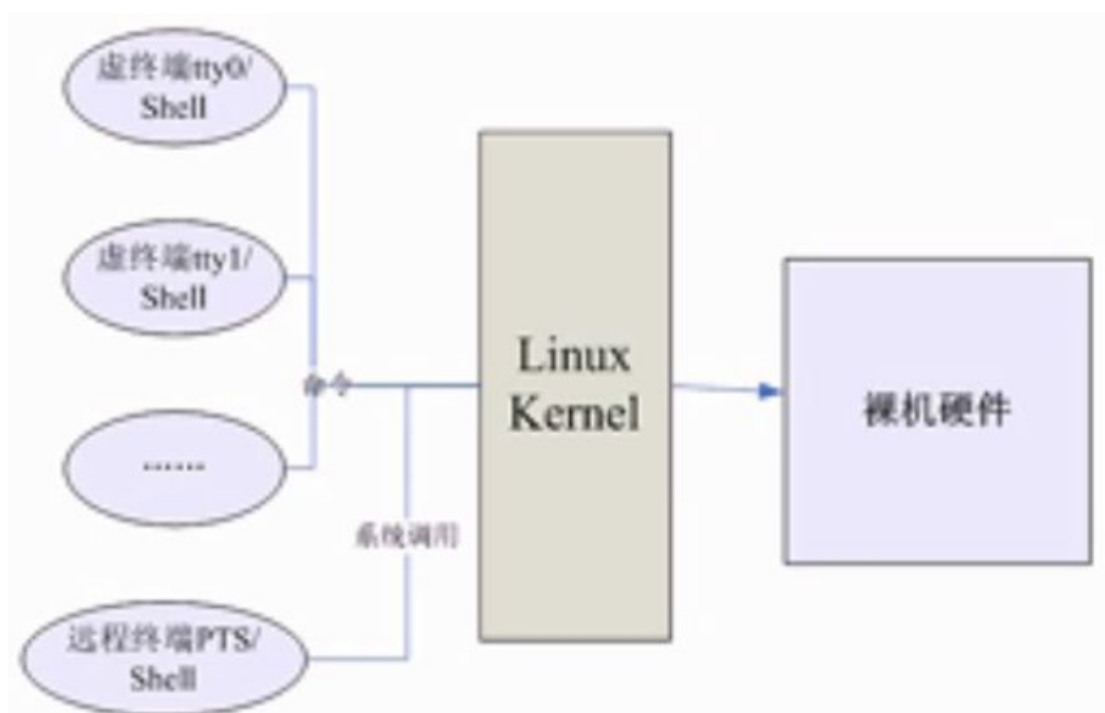
# System initialization.
x1::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```

当前默认的方式是 3: initdefault, 是文本的方式, 如果想以图像化的方式进入则系统, 需要修改为 5. 这里只了解。退去用 q 命令。同时这也是对 redhat 来说的, ubuntu 没有找到这样的文件。

接下来我们讲一下另外一种终端登录方式: **多终端登录**

主要有那些终端呢? 主要有虚终端, 先看一下用虚终端的方式登录, 实际上我们刚才使用的就是虚终端。Linux 有 6 个虚终端, 我们可以用 alt + Fn(n=1~6)进行切换。Ubuntu 测试可用。参见 <http://blog.csdn.net/fightingforev/article/details/52337155>, 虚终端中“虚”的含义就是共享的含义, 共享一个屏幕, 在美国字典里“虚”有共享的含义。通过快捷键可以切换不同的终端, 输入账号和密码就可以进入不同的终端。各虚终端共享键盘和显示器, 通过 alt + Fn(n=1~6)进行切换。各与虚终端相对应的是物理终端。物理终端是有什么构成的呢?



物理终端是由屏幕(或者说显示器)和键盘构成的。这些物理终端通过串口连接到主机上面, 比如说有一台 linux 主机, linux 主机接出来一个多用户接口卡, 这个卡和主机之间的连接是串口, 多用户接口卡上面有很多的口, 这些口用来连接终端, 终端就是键盘+显示器。这个终端我们就称为物理终端。物理终端和我们前面说到的虚终端是有区别的, 刚才说的虚终端是在 linux 主

机上面的终端,主机上面的键盘和显示器被多个虚终端共享,

第三种终端我们称为(也叫仿真终端)**远程终端**,远程终端是通过软件进行模拟的,这里我们可以用一个演示软件 **secureCRT**,这个软件实际上也是一个终端,我们称为仿真终端.可以用该软件登录到 **root** 用户.该终端的特征是?他和物理终端不同的地方是,它的链接线是通过,仿真终端是一个软件,这个软件安装在一台机器上面,这是一台客户机,这个机器和主机之间的链接是通过网络来进行的.也就是说仿真终端和主机之间的链接不受地域范围的影响,只要网络你能够通就可以,可以由中国的一台客户机访问美国的一台服务器,这就是仿真终端,它仅仅是客户机上面的一个仿真的软件,那么这个软件和服务器之间的通信符合一个协议,这个协议是 **SSH 协议**.(可以想到 **git/github 主机和从机的关系**)也就是说仿真终端和主机形成了一个系统,这个系统一般称为 C/S(C=customer S=service,客户服务模型)架构的系统,那么我们把仿真终端就做 ssh 的客户端,因为他们符合 ssh 协议,linux 主机上面必定提供了一个 ssh 服务的软件,这个软件实际上叫做 **sshd** 软件,有这样的一个守护进程,仿真终端才能够链接上去,并且它的端口号是 22(sshd(22)).这里大家做一个了解.这既是仿真终端,刚才我们一共讲到了三种终端,一种是虚终端,虚终端共享屏幕和键盘,另外一个物理终端,每一个物理终端都有独立的屏幕和键盘,它是通过串口和主机相连.第三种是仿真终端,它实际上是一个客户端的一个软件,通过 ssh 协议链接到了服务器端,服务器端安装了一个 ssh 协议的一个软件,叫做 **sshd**.那么他们是通过网络链接的.

#### 5. 简单的命令:

命令名称	用途	命令举例
date	显示日期	date
cal	显示日历	cal 2007
who	查看谁在使用系统	who
who am i	查看我是谁	who am i
clear	清除屏幕	clear
bc	计算器	bc

教程是在仿真终端上做演示,以下我们在主机上演示:



```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whd@whd-Lenovo:~$ date
2016年 08月 28日 星期日 14:23:51 CST
whd@whd-Lenovo:~$ cal
      八月 2016
日 一 二 三 四 五 六
   1  2  3  4  5  6
  7  8  9 10 11 12 13
 14 15 16 17 18 19 20
 21 22 23 24 25 26 27
 28 29 30 31

whd@whd-Lenovo:~$ who
whd      tty1          2016-08-27 23:40
guest-gQW6G9 :0          2016-08-27 19:38 (:0)
guest-Yp84rG :1          2016-08-27 19:41 (:1)
whd      tty3          2016-08-27 19:46
whd      :2            2016-08-27 19:47 (:2)
whd      pts/36        2016-08-27 23:34 (:2)
whd      pts/41        2016-08-28 07:51 (:2)
whd      pts/50        2016-08-28 09:16 (:2)
whd      pts/51        2016-08-28 09:24 (:2)
whd      pts/52        2016-08-28 09:32 (:2)
whd      pts/53        2016-08-28 09:35 (:2)
whd      pts/54        2016-08-28 09:41 (:2)
whd      pts/55        2016-08-28 09:43 (:2)
whd      pts/56        2016-08-28 09:46 (:2)
whd      pts/57        2016-08-28 09:50 (:2)
whd      pts/58        2016-08-28 09:54 (:2)
whd      pts/59        2016-08-28 09:59 (:2)
whd      pts/60        2016-08-28 10:04 (:2)
whd@whd-Lenovo:~$
```

Cal(calendar 日历)命令默认显示的是当月的日历 ,如果要显示某年的日历可以 cal 2008

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whd@whd-Lenovo:~$ cal 2008
2008
  一月          二月          三月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  5          1  2          1
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    2  3  4  5  6  7  8
13 14 15 16 17 18 19    10 11 12 13 14 15 16    9 10 11 12 13 14 15
20 21 22 23 24 25 26    17 18 19 20 21 22 23    16 17 18 19 20 21 22
27 28 29 30 31          24 25 26 27 28 29    23 24 25 26 27 28 29
                                30 31

  四月          五月          六月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  5          1  2  3          1  2  3  4  5  6  7
  6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19    11 12 13 14 15 16 17    15 16 17 18 19 20 21
20 21 22 23 24 25 26    18 19 20 21 22 23 24    22 23 24 25 26 27 28
27 28 29 30          25 26 27 28 29 30 31    29 30

  七月          八月          九月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  5          1  2          1  2  3  4  5  6
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    7  8  9 10 11 12 13
13 14 15 16 17 18 19    10 11 12 13 14 15 16    14 15 16 17 18 19 20
20 21 22 23 24 25 26    17 18 19 20 21 22 23    21 22 23 24 25 26 27
27 28 29 30 31          24 25 26 27 28 29 30    28 29 30
                        31

  十月          十一月          十二月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4          1          1  2  3  4  5  6
  5  6  7  8  9 10 11    2  3  4  5  6  7  8    7  8  9 10 11 12 13
12 13 14 15 16 17 18    9 10 11 12 13 14 15    14 15 16 17 18 19 20
19 20 21 22 23 24 25    16 17 18 19 20 21 22    21 22 23 24 25 26 27
26 27 28 29 30 31      23 24 25 26 27 28 29    28 29 30 31
                        30
```

Who 命令谁在使用这个系统,教程中的截图:可以看到有两个虚终端,tty1,tty2,用仿真终端还给出 IP 地址.为什么是这个 IP 呢?这就设计虚拟机的网络配置问题.当前虚拟机的配置使用的 NAT,它使用的是 VMnet 这块网卡进入到虚拟机的,我们可以看一下这块网卡的配置信息,可以看到它的 ip 地址就是 192.168.234.1.这里我们就简单的理解成,这就是客户端的 ip 地址.

```
[root@linuxserver root]# who
root      tty1      Oct  5 22:02
study     tty2      Oct  5 22:15
root      pts/0     Oct  5 22:22 (192.168.234.1)
[root@linuxserver root]#
```

服务器端的 ip 地址是多少呢,可以用命令 ifconfig,来查看一下 ip 地址.



```
whd@whd-Lenovo:~$ ifconfig
enp3s0    Link encap:以太网  硬件地址 00:23:24:3d:f4:dc
          inet 地址:192.168.1.100 广播:192.168.1.255 掩码:255.255.255.0
          inet6 地址: fe80::223:24ff:fe3d:f4dc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:1147542 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:561742 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:1246824895 (1.2 GB)  发送字节:84153836 (84.1 MB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1 掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:69162 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:69162 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:11039247 (11.0 MB)  发送字节:11039247 (11.0 MB)

whd@whd-Lenovo:~$
```

这就是当前主机的 ip 地址.

还可以查看我自己是谁:who am i

```
whd@whd-Lenovo:~$ who am i
whd      pts/60      2016-08-28 10:04 (:2)
whd@whd-Lenovo:~$
```

Bc 是计算器命令

```
whd@whd-Lenovo:~$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5*6
30

```

Ctrl+D 是结束符,bc 中表示计算结束().

Ctrl+C 表示中断前台进程.

比如说我们刚才的 bc,然后键入 Ctrl +C:

```
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5*6
30
whd@whd-Lenovo:~$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
^C
(interrupt) use quit to exit.

```

当前中断不了,是因为这个终端信号被 bc 程序捕捉到了,按住 ctrl+D 把它退出来.

```
whd@whd-Lenovo:~$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
^C
(interrupt) use quit to exit.
whd@whd-Lenovo:~$
```

这里只做简单的介绍.中断不了的原因是因为 bc 这个程序捕捉了这个中断.并且捕捉这样都额信号,打印了(interrupt)use quit to exit 这样的信息.

## 6. Man 命令:

[1]Linux 的命令有很多的参数,我们不可能都记住,我们可以通过查看联机的手册获取帮助,访问 linux 手册的命令就是 man 命令.使用的方法就是 **man + 命令**,比如说我们查看 date 的用法.**man +date.**

```

X  —  终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
DATE(1)                                User Commands                                DATE(1)

NAME
    date - print or set the system date and time

SYNOPSIS
    date [OPTION]... [+FORMAT]
    date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
    Display the current time in the given FORMAT, or set the system date.

    Mandatory arguments to long options are mandatory for short options
    too.

    -d, --date=STRING
        display time described by STRING, not 'now'

    -f, --file=DATEFILE
        like --date once for each line of DATEFILE

    -I[TIMESPEC], --iso-8601[=TIMESPEC]
        output date/time in ISO 8601 format. TIMESPEC='date' for date
        only (the default), 'hours', 'minutes', 'seconds', or 'ns' for
        date and time to the indicated precision.

    -r, --reference=FILE
        display the last modification time of FILE

    -R, --rfc-2822
        output date and time in RFC 2822 format. Example: Mon, 07 Aug
        2006 12:34:56 -0600

    --rfc-3339=TIMESPEC
        output date and time in RFC 3339 format. TIMESPEC='date', 'sec-
        onds', or 'ns' for date and time to the indicated precision.
        Date and time components are separated by a single space:
        2006-08-07 12:34:56-06:00

    -s, --set=STRING
        set time described by STRING

    -u, --utc, --universal
        print or set Coordinated Universal Time (UTC)

    --help display this help and exit

Manual page date(1) line 1 (press h for help or q to quit)

```

Date(1) 1 代表 date 的帮助在第一章.

```

X  —  终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whd@whd-Lenovo:~$ date +%A
星期日
whd@whd-Lenovo:~$

```

[2]man(2.2)

Man 手册是按照一定的章节来划分的:

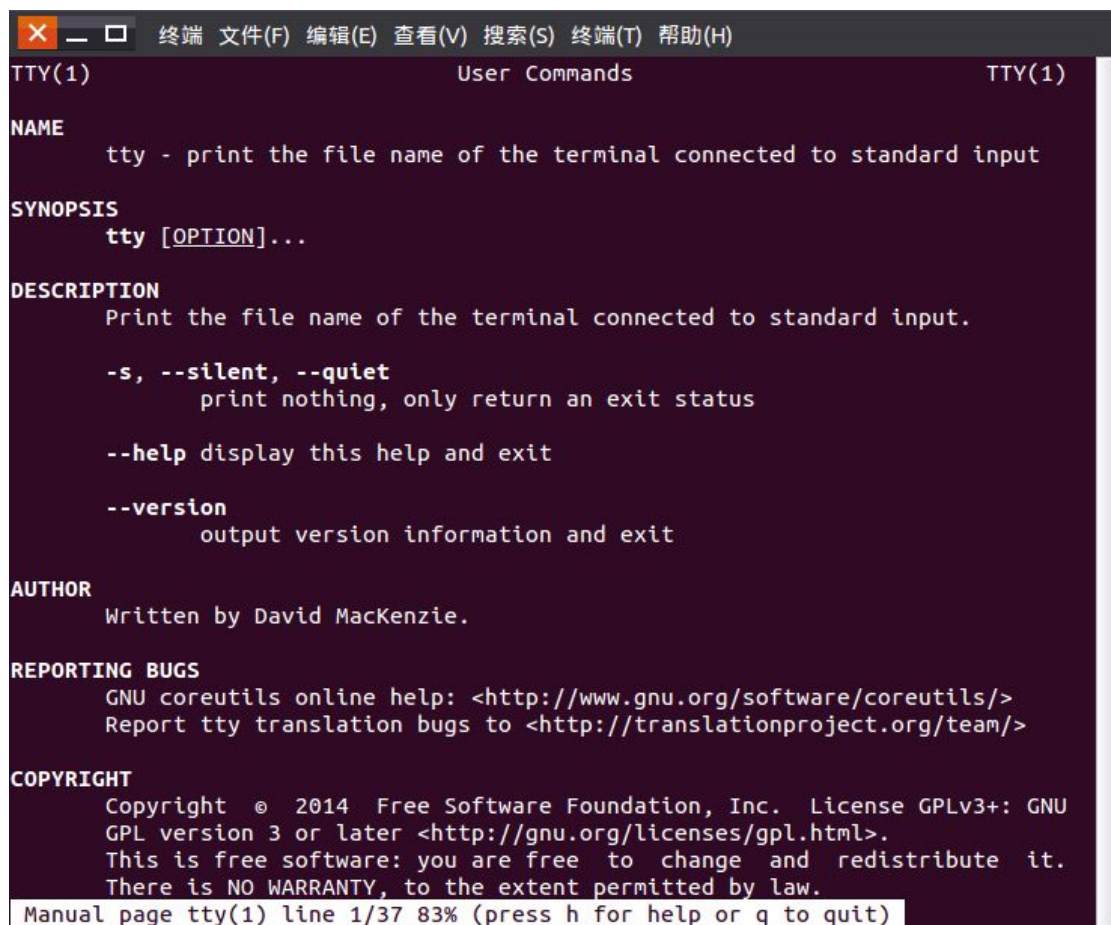


- 命令
- 系统调用
- 库函数
- 特殊文件
- 配置文件
- 游戏
- 惯例与协议
- 系统管理命令
- 内核

命令是第一章放的是和命令相关的帮助.

库函数包含比如说 c 语言的一些库函数的使用等,以及一些第三方的库的手册.

特殊文件;一些特殊的设备文件的一些帮助.什么是特殊文件后面还会介绍,比如说,man tty  
Tty 就是终端就是特殊文件



```

X  _  终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
TTY(1)                                User Commands                                TTY(1)

NAME
    tty - print the file name of the terminal connected to standard input

SYNOPSIS
    tty [OPTION]...

DESCRIPTION
    Print the file name of the terminal connected to standard input.

    -s, --silent, --quiet
        print nothing, only return an exit status

    --help display this help and exit

    --version
        output version information and exit

AUTHOR
    Written by David MacKenzie.

REPORTING BUGS
    GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
    Report tty translation bugs to <http://translationproject.org/team/>

COPYRIGHT
    Copyright © 2014 Free Software Foundation, Inc. License GPLv3+: GNU
    GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.

Manual page tty(1) line 1/37 83% (press h for help or q to quit)
```

这样显示的还是第一张的内容,就是 tty 命令的介绍,如果想看第四章关于 tty 的介绍:man 4 tty

```
✕ — □ 终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
TTY(4)                                Linux Programmer's Manual                                TTY(4)

NAME
    tty - controlling terminal

DESCRIPTION
    The file /dev/tty is a character file with major number 5 and minor
    number 0, usually of mode 0666 and owner.group root.tty. It is a syn-
    onym for the controlling terminal of a process, if any.

    In addition to the ioctl(2) requests supported by the device that tty
    refers to, the ioctl(2) request TIOCNOTTY is supported.

    TIOCNOTTY
        Detach the calling process from its controlling terminal.

        If the process is the session leader, then SIGHUP and SIGCONT signals
        are sent to the foreground process group and all processes in the cur-
        rent session lose their controlling tty.

        This ioctl(2) call works only on file descriptors connected to
        /dev/tty. It is used by daemon processes when they are invoked by a
        user at a terminal. The process attempts to open /dev/tty. If the
        open succeeds, it detaches itself from the terminal by using TIOCNOTTY,
        while if the open fails, it is obviously not attached to a terminal and
        does not need to detach itself.

FILES
    /dev/tty

SEE ALSO
    Manual page tty(4) line 1 (press h for help or q to quit)
```

如果想看手册中所有的 `tty` 的帮助,可以:`man -a tty`

```
✕ — □ 终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
星期日
whd@whd-Lenovo:~$ man tty
whd@whd-Lenovo:~$ man 4 tty
whd@whd-Lenovo:~$ man -a tty
```

配置文件:比如说我们之前说道的口令文件

Man 5 passwd

```

X  —  终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
PASSWD(5)                                文件格式和转化                                PASSWD(5)

名称
passwd - 密码文件

描述
/etc/passwd 为每个用户账户包含一行，包含使用冒号 (":")
分隔的七个字段，分别是：

· 登录名
· 可选的加密后的密码
· 数字用户 ID
· 数字组 ID
· 用户名和注释字段
· 用户主目录
· 可选的用户命令解释器

如果
加密的密码字段可以为空，此时使用指定的登录名登录时不会要求认证。然而，如
password 为空，一些读取 /etc/passwd 文件的程序可能会不允许 任何
访问。如果 password 字段是一个小写的 "x"，那么加密的密码实际上存储于
shadow(5) 中；在 /etc/shadow 文件中 必须
有对应的行，否则用户账户就会无效。如果 password
字段是其他任何字符串，将会被视为加密过的密码，如 crypt(3) 中的说明。

此注释字段被多个系统工具使用，例如 finger(1)。

主目录字段提供了初始工作目录的名称。login 程序根据此信息设置 $HOME
环境变量。

命令解释器字段提供了用户命令语言解释器的名称，或者开始时执行的程序的名称
。login
程序使用此信息设置 $SHELL 环境变量的值。如果字段为空，默认值为
/bin/sh。

Manual page passwd(5) line 1 (press h for help or q to quit)
```