**1.1打开终端**

在以后的学习中，我们离不开一个界面，那就是终端(Terminal)，首先介绍一下如何打开终端。在左上角的应用程序中，找到附件-终端，单击即可以打开。  
还有一个快捷键，如果是在Ubuntu 10.04以上的系统，默认配置了一个快捷键

(Ctrl + Alt + T)就可以打开终端。  
在一个终端界面下，还可以打开选项卡，在终端空白处单击鼠标右键，点击新建标签页可以打开一个新的终端。  
当然这也有快捷键(Ctrl + Shift + T)。

## 1.2 man

查看关键词的manpage(manual page) 查看’ls’的manpage(manual page)可以输入

$ man lsdisplay the help man-page of ls

就可以查看ls的manpage

## 1.3 --help

查看关键词的快捷帮助而不是manpage,可以使用–help  
查看’ls’的快速帮助，输入

$ ls --helpshow the help information of ls

快速帮助信息就会打印到终端上

# 1.4 info

另外一个帮助命令，可以显示出更完整的最新的信息。 查看’ls’的info信息，可以输入

$ info lsdisplay the info of ls

可以显示ls的info信息。

# 1.5 ls

## ls

查看指定目录下的内容

$lslistmain.c Makefile

### 1.5.1 ls -l

查看当前目录内容的详细信息。

$ ls -llist -long-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 main.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 Makefile

### 1.5.2 ls -a

查看当前目录中的所有文件(包括隐藏文件)

$ ls -alist -all. .. .data.txt main.c Makefile

# 1.6 mkdir

## 1.6 mkdir

### 1.6.1 创建单个目录

在当前目录下创建一个名为akaedu的目录

$mkdir akaedumake a directory named akaedu$lsmain.c Makefile akaedu

### 1.6.2 创建多层目录

在当前目录下创建目录a, 再在a目录下创建b

$mkdir -p a/bmake a directory named akaedu$lsmain.c Makefile akaedu a$ls ab

# 1.8 cd

## 1.8 cd (change directory)

### 1.8.1 改变活动目录

假设当前目录下有一个a目录，要切换到a目录下可以输入

$lsmain.c Makefile akaedu a$pwd/home/ymqqqqdx/$cd a$pwd/home/ymqqqqdx/a

### 1.8.2 绝对路径

假设要切换到用户主目录下的a目录，可以使用绝对路径

$pwd/$cd /home/ymqqqqdx/a$pwd/home/ymqqqqdx/a

### 1.8.3 相对路径

同1.2.3.2,假设当前就在用户主目录下，可以使用相对路径

$pwd/home/ymqqqqdx$cd a$pwd/home/ymqqqqdx/a

### 1.8.4 主目录

每一个用户都有一个对应的主目录，其绝对路径为**/home/usrname**。  
要切换到主目录时，只需输入**cd**，即可回到用户主目录。  
另外**~**代表当前用户的主目录，**～/a**即代表主目录下的a目录

$pwd/home/ymqqqqdx/a/b/c/d$cd$pwd/home/ymqqqqdx/$cd ~/a/home/ymqqqqdx/a

### 1.8.5 ..(上层目录)和.(当前目录)

..代表上层目录 .代表当前目录，若要运行当前目录下的a.out程序，即可输入./a.out运行程序。

$pwd/home/ymqqqqdx/a/b/c/d$cd ..$pwd/home/ymqqqqdx/a/b/c$./a.outhello world!

# 1.9 rmdir

### 1.9 rm -r

**rmdir**用来删除空目录。

$pwd/home/ymqqqqdx/$lsa main.c Makefile$rmdir a$lsmain.c Makefile

# 1.10 touch

## 1.10 touch

创建一个空文件,**touch foo.c**，这样就在当前目录创建了foo.c文件。  
touch的另外一个作用时更新当前文件的修改时间，若已经存在foo.c文件，输入**touch foo.c**后，用**ls -l**可以看到文件修改时间已更新为当前时间。

$lsmain.c Makefile$touch foo.cmain.c Makefile foo.c$ls -l-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 01:04 foo.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 main.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 Makefile$touch main.c$ls -l-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 01:04 foo.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 01:06 main.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 Makefile

# 1.11 cat/more/less

## 1.11.1 cat

**cat filename**   
cat用来查看文本文件中的内容。

$lsmain.c Makefile$cat main.cthe content of main.c

若cat后不加任何文件名，则cat进程就会一起运行，不能自行退出，这时按下**Ctrl + Z**即可结束该进程。

$ cat^C$

## 1.11.2 more

**more filename**   
more与cat的作用差不多，只是当文件内容多于一屏时，按下空格可以滚到下一屏继续阅读，但是不可以回退。

$ more main.ccontent of main.c  
...  
press space to continue

## 1.11.3 less

**less filename**  
less的功能比more更强大，当文件内容多于一屏时，按j可以向下滚动，按k可以向上滚动。

$ less main.ccontent of main.c   
...  
press j to down   
press k to up 

## 1.12 rm

### 1.12.1 删除文件

删除1.c文件，**rm 1.c**

$lsmain.c Makefile 1.c$rm 1.c$lsmain.c Makefile

### 1.12.2 强制删除目录rm -rf

删除a目录，**rm -r a**,若目录非空，会提示是否删除，若不需要提示，输入**rm -rf a**就可以直接删除a目录不需要确认。

$lsmain.c Makefile a$rm -rf a$lsmain.c Makefile

# 1.13 cp

## 1.13 cp

### 1.13.1 复制单个文件

**cp filename newfilename.c**，复制filename为newfilename。

$lsmain.c Makefile$cp main.c new.c$lsmain.c Makefile new.c$cp main.c ~/new.c$ls ~/new.c ...

### 1.13.2 复制文件夹

**cp -r a/ b/**，复制a目录为b目录。

$lsmain.c Makefile a$cp -r a b$lsmain.c Makefile a b

## 1.15 ln

### 1.15.1 创建硬链接

**ln file1 file2**,file2为file1的硬链接

$lsmain.c Makefile$ln main.c 1.c$lsmain.c Makefile 1.c$vim main.c$vim 1.c$rm main.c$vim 1.c

### 1.15.2 创建符号链接

**ln -s file1 file2**,file2为file1的符号链接

$lsmain.c Makefile$ln -s main.c 1.c$lslrwxrwxrwx 1 ymqqqqdx ymqqqqdx 6 2011-08-27 01:34 1.c -> main.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 01:04 foo.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 115 2011-08-27 01:15 main.c  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 0 2011-08-27 00:31 Makefile$vim main.c$vim 1.c$rm main.c$vim 1.c

### 1.15.3 硬链接与符号链接的区别

  linux系统下提供ln指令来进行文件链接。文件链接主要分为硬链接和软链接。

   硬链接：由于linux下的文件是通过索引节点（Inode）来识别文件，硬链接可以认为是一个指针，指向文件索引节点的指针，系统并不为它重新分配inode。每添加一个一个硬链接，文件的链接数就加1。

    可以用:ln命令来建立硬链接。语法:

**[php:collapse]** [+ expand sourceview plaincopy](http://blog.csdn.net/hairetz/article/details/4168296)

  用法: 第一种:为”existingfile”创建硬链接,文件名为”newfile”。第二种:在”directory”目录中,为 ”existingfile-list”中包含的所有文件创建一个同名的硬链接。常用可选[options] –f 无论”newfile”存在与否,都创建链接。-n 如果”newfile”已存在,就不创建链接。

  例子：

**[php:collapse]** [+ expand sourceview plaincopy](http://blog.csdn.net/hairetz/article/details/4168296)

注意在创建链接前,file1 显示的链接数目为1,创建链接后(1)file1和file1hard的链接数目都变为2;(2) file1和file1hard在inode号是一样的(3) file1和file1hard显示的文件大小也是一样。可见进行了ln命令的操作结果：file1和file1hard是同一个文件的两个名字，它们具有同样的索引节点号和文件属性，建立文件file1的硬链接，就是为file1的文件索引节点在当前目录上建立一个新指针。如下图,你可以删除其中任何一个,如rm file2 ,每次只会删除一个指针，链接数同时减一,只有将所有指向文件内容的指针,也即链接数减为0时，内核才会把文件内容从磁盘上删除。

 不足：

尽管硬链接节省空间，也是Linux系统整合文件系统的传统方式，但是存在一下不足之处：（1）不可以在不同文件系统的文件间建立链接（2）只有超级用户才可以为目录创建硬链接。

软链接（符号链接）：

软链接克服了硬链接的不足，没有任何文件系统的限制，任何用户可以创建指向目录的符号链接。因而现在更为广泛使用，它具有更大的灵活性，甚至可以跨越不同机器、不同网络对文件进行链接。

      　如果给ln命令加上- s选项，则建立软链接。如果[链接名]已经存在但不是目录，将不做链接。[链接名]可以是任何一个文件名（可包含路径），也可以是一个目录，并且允许它与“目标”不在同一个文件系统中。如果[链接名]是一个已经存在的目录，系统将在该目录下建立一个或多个与“目标”同名的文件，此新建的文件实际上是指向原“目标”的符号链接文件。

   例子：

**[php]** [view plaincopy](http://blog.csdn.net/hairetz/article/details/4168296)

1. $ ls -il
2. 13058 -rwx - - - - - - 1 longcheng longcheng 48 8月 5 16:38 file1
3. 13059 -rwx - - - - - - 2 longcheng longcheng 57 8月 5 16:40 file2
4. 13059 -rwx - - - - - - 2 longcheng longcheng 57 8月 5 16:40 file2hard
5. $ln –s file1 file1soft
6. $ls -il
7. 13058 -rwx - - - - - - 1 longcheng longcheng 48 8月 5 16:38 file1
8. 13059 -rwx - - - - - - 2 longcheng longcheng 57 8月 5 16:40 file2
9. 13059 -rwx - - - - - - 2 longcheng longcheng 57 8月 5 16:40 file2hard
10. 13061 lrwxrwxrwx 1 longcheng longcheng 5 8月 5 16:58 file1soft->file1

软链接与硬链接，区别不仅仅是在概念上，在实现上也是不同的。区别：硬链接原文件＆链接文件公用一个inode号，说明他们是同一个文件，而软链接原文件＆链接文件拥有不同的inode号，表明他们是两个不同的文件；在文件属性上软链接明确写出了是链接文件，而硬链接没有写出来，因为在本质上硬链接文件和原文件是完全平等关系；链接数目是不一样的，软链接的链接数目不会增加；文件大小是不一样的，硬链接文件显示的大小是跟原文件是一样的，这用强调，因为是等同的嘛，而这里软链接显示的大小与原文件就不同了，file1大小是48B，而file1soft是5B，这里面的5实际上就是“file1”的大小。

总之，建立软链接就是建立了一个新文件。当访问链接文件时，系统就会发现他是个链接文件，它读取链接文件找到真正要访问的文件。

缺点：

因为链接文件包含有原文件的路径信息，所以当原文件从一个目录下移到其他目录中，再访问链接文件，系统就找不到了，而硬链接就没有这个缺陷，你想怎么移就怎么移；还有它要系统分配额外的空间用于建立新的索引节点和保存原文件的路径。

# 1.16 tar

## 1.16 tar

为文件和目录创建档案(压缩包)

### 1.16.1 打包

将a目录用gzip打包，tar zcvf a.tar.gz a  
将a目录用bzip2打包，tar jcvf a.tar.bz2 a

$lsmain.c Makefile$tar zcvf foo.tar.gz main.c$lsmain.c Makefile foo.tar.gz$tar jcvf foo.tar.bz2 main.c$lsmain.c Makefile foo.tar.gz foo.tar.bz2

### 1.16.2 解包

解压gzip包，tar zxvf a.tar.gz  
解压bzip2包，tar jxvf a.tar.bz2

$lsfoo.tar.gz foo.tar.bz2$tar zxvf foo.tar.gz$lsmain.c foo.tar.gz$rm main.c$tar jxvf foo.tar.bz2$lsmain.c foo.tar.gz foo.tar.bz2

**1.17.1文件的权限**

在Linux系统中使用的Ext文件系统中，支持文件的权限，用ls -l查看当前目录的文件权限。

$ls -l-rwxr-xr-x 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c

由图中我们可以看出这两个文件的权限是不一样的:  
最前边的一位，可以看出backup/的前边有一个d，这代表它是一个目录，a.out和chess.c为一个-，这说是它们是普通文件。  
后边有三位代表文件所有者的权限，分别为可读，可写，可执行，这说明a.out为一个可执行文件，chess.c为一个不可执行文件。  
随后的三位表示同组用户所拥有的权限，最后的三位表示其它用户所拥有的权限。

**1.17.2文字设定法**

可以使用chmod who +|-|= 文件名来改变权限。  
who可以是下述字母中的任一个或者它们的组合：

 u 表示文件或目录的所有者

 g 表示同组用户

 o 表示其它用户

 a 表示所有用户mode的常用权限为以下几个：

 r 可读

 w 可写

 x 可执行

$ls -l-rwxr-xr-x 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c$chmod -x a.out$ls -l-rw-r--r-- 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c

**1.17.3数字设定法**

在chmod后边可以加上数字，000即代表不可读，不可写，不可执行  
011则代表不可读，可写，可执行。用二进制进行转换011为3，所以用chmod 333 a.out就可以将a.out的变为–rx-rx-rx权限。

$ls -l-rwxr-xr-x 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c$chmod 333 a.out$ls -l--wx-wx-wx 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c

# 1.18 chown

### 1.18 chown

更改某个文件或目录的所有者，这个命令是很常用的，例如root用户把自己的一个文件拷贝给用户akaedu，为了能让用户akaedu存取这个文件，root用户应该把这个文件的所有者设为akaedu，否则，akaedu无法存取这个文件。  
例如把一个文件改为akaedu用户所有：

$ls -l-rwxr-xr-x 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 ymqqqqdx ymqqqqdx 3681 2011-09-02 17:49 chess.c$chown akaedu chess.c$ls -l-rwxr-xr-x 1 ymqqqqdx ymqqqqdx 7438 2011-09-02 15:08 a.out  
drwxr-xr-x 2 ymqqqqdx ymqqqqdx 4096 2011-09-03 09:33 backup/  
-rw-r--r-- 1 akaedu ymqqqqdx 3681 2011-09-02 17:49 chess.c

# 1.19 find、grep

### 1.19.1 find

在目录中搜索文件，使用find命令

$find . -name 'chess.\\*'./chess.c

### 1.19.2 grep

在指定文件中搜索特定内容，并将含有这些内容的行输出到标准输出。若不指定文件名，则从标准输入读取。

$grep 'printf' ./ -R./chess.c:30: printf("\n ");  
./chess.c:32: printf(" %d", i + 1);  
./chess.c:33: printf("\n");  
./chess.c:35: printf(" -");  
./chess.c:36: printf("\n");  
./chess.c:39: printf("%d ", i + 1);  
./chess.c:42: printf(" %d", chessboard[i][j]);  
./chess.c:44: printf("\n");  
./chess.c:115: printf("current: p1: %d, p2: %d.\n", p1, p2);  
./chess.c:126: printf("Welcome!\n");  
./chess.c:127: printf("Input link this, 3 3\n");  
./chess.c:131: printf("Player %d: ", who);  
./chess.c:140: printf("Board full!\n");  
./chess.c:143: printf("Current step: %d\n", step);  
./chess.c:146: printf("out of range or place   
occupied!\n"

# 1.20 mount/umount

### 1.20.1 挂载U盘

新版本的Linux发行版大多支持自动挂载U盘分区，默认的挂载位置为/media/卷名，但是如果发生问题不能自动挂载，所以要介绍U盘的手动挂载。

$ls /dev/sd\\*/dev/sda /dev/sda2 /dev/sda5 /dev/sda7 /dev/sda9 /dev/sda1 /dev/sda3 /dev/sda6 /dev/sda8

插入U盘后

$ls /dev/sd\\*/dev/sda /dev/sda2 /dev/sda5 /dev/sda7 /dev/sda9 /dev/sda1 /dev/sda3 /dev/sda6 /dev/sda8 /dev/sdb /dev/sdb1

我们发现这里多了一个sdb 和一个sdb1，这就是我们的U盘。

$sudo mount /dev/sdb1 /mnt$cd /mnt$lsfile1 file2 .. .

这样就将U盘挂载到了/mnt目录下，对/mnt目录的操作就可以读写U盘上的文件。

# 1.21 df/du

### 1.21.1 df

df用来检查文件系统的磁盘空间占用情况，可以利用该命令来获取硬盘被占用了多少空间，日前还剩下多少空间等信息。

$df文件系统 1K-块 已用 可用 已用% 挂载点  
/dev/sda8 15480816 3693868 11000568 26% /  
none 1024060 332 1023728 1% /dev   
none 1028300 1460 1026840 1% /dev/shm   
none 1028300 104 1028196 1% /var/run   
none 1028300 0 1028300 0% /var/lock   
none 1028300 0 1028300 0% /lib/init/rw   
/dev/sda9 15472768 11540932 3145860 79% /home   
/dev/sda1 52428796 15280160 37148636 30%   
/media/7CE619AAE6196624   
/dev/sda2 62915580 2578292 60337288 5% /media/New Volume   
/dev/sda5 101448700 55005980 46442720 55% /media/Entertainment   
/dev/sda6 62918652 519376 62399276 1% /media/New Volume 

### 1.21.2 du

du用来统计目录或文件所占用磁盘空间的大小，du意为disk usage。  
使用du -s可显示当前目录所占用的空间总大小。

$du -s1697668 .

这说明当前目录占用空间大小为1697668K。

# 1.22 ps

## 1.22 ps

监视后台进程的工作情况

$psPID TTY TIME CMD  
2575 pts/2 00:00:00 bash  
4190 pts/2 00:00:00 ps

### 1.22.1 ps -a

显示终端上的所有进程，包括其他用户的进程。

$ps -aPID TTY TIME CMD  
1753 pts/0 00:00:00 tig  
3640 pts/1 00:00:13 vi  
4207 pts/2 00:00:00 ps

### 1.22.2 ps -u

查看进程所有者及其他一些详细信息。

$ps -uUSER PID %CPU %MEM VSZ RSS TTY STAT  
START TIME COMMAND  
ymqqqqdx 1563 0.0 0.1 6488 3664 pts/0 Ss 00:01 0:00 bash  
ymqqqqdx 1753 0.0 0.0 4092 1492 pts/0 S+ 00:03 0:00 tig  
ymqqqqdx 1801 0.0 0.1 6492 3668 pts/1 Ss 00:04 0:00 bash  
ymqqqqdx 2575 0.0 0.1 6492 3656 pts/2 Ss 00:30 0:00 bash  
ymqqqqdx 3640 1.0 0.3 13164 6340 pts/1 S+ 01:19 0:14 vi 1.11.md  
ymqqqqdx 4226 0.0 0.0 2716 1072 pts/2 R+ 01:42 0:00 ps -u

### 1.22.3 ps -x

显示没有控制终端的进程

$ps -x1356 ? Sl 0:00 /usr/bin/gnome-keyring-daemon --daemonize --login   
1374 ? Ssl 0:00 gnome-session   
1405 ? Sl 0:30 /usr/bin/ibus-daemon --xim   
1409 ? Ss 0:00 /usr/bin/ssh-agent /usr/bin/dbus-launch --exit-with-s   
1412 ? S 0:00 /usr/bin/dbus-launch --exit-with-session gnome-sessio   
1413 ? Ss 0:01 /bin/dbus-daemon --fork --print-pid 5 --print-address   
1416 ? S 0:00 /usr/lib/libgconf2-4/gconfd-2   
1417 ? S 0:00 /usr/lib/ibus/ibus-gconf   
1419 ? S 0:11 python /usr/share/ibus/ui/gtk/main.py   
1421 ? S 0:00 /usr/lib/ibus/ibus-x11 --kill-daemon   
1423 ? S 0:21 python /usr/share/ibus-table/engine/main.py --ibus   
1435 ? Ss 0:02 /usr/lib/gnome-settings-daemon/gnome-settings-daemon   
1438 ? S 0:00 /usr/lib/gvfs/gvfsd   
1445 ? Sl 0:02 gnome-power-manager   
1446 ? SL 0:00 nm-applet --sm-disable   
1447 ? S 0:00 /usr/lib/policykit-1-gnome/polkit-gnome-authenticatio   
1449 ? S 0:00 bluetooth-applet   
1451 ? S 1:16 /usr/bin/compiz 

# 1.23 kill

## 1.23 kill

给一个进程发信号，或终止一个进程的运行。

### 1.23.1 用PID来终止一个进程

**cat**创建一个进程  
按下**Ctrl + Z**来挂起进程，输入**ps -aux**查看当前进程，假设cat进程的PID为1760  
输入**kill -SIGKILL 1760**即可结束cat进程。

$ps -aux...  
ymqqqqdx 1729 0.9 0.1 6492 3672 pts/1 Ss 13:09 0:00 bash  
ymqqqqdx 1760 0.0 0.0 3464 600 pts/1 T 13:10 0:00 cat  
ymqqqqdx 1761 0.0 0.0 2716 1076 pts/1 R+ 13:10 0:00 ps -aux  
$kill -SIGKILL 1760$ps -aux...  
ymqqqqdx 1729 0.9 0.1 6492 3672 pts/1 Ss 13:09 0:00 bash  
ymqqqqdx 1761 0.0 0.0 2716 1076 pts/1 R+ 13:10 0:00 ps -aux

### 1.23.2 用进程名来终止一个进程

同上来创建一个进程，查看到cat进程存在，这是输入  
**killall -9 cat**即可结束cat进程.

$ps -aux...  
ymqqqqdx 1729 0.9 0.1 6492 3672 pts/1 Ss 13:09 0:00 bash  
ymqqqqdx 1760 0.0 0.0 3464 600 pts/1 T 13:10 0:00 cat  
ymqqqqdx 1761 0.0 0.0 2716 1076 pts/1 R+ 13:10 0:00 ps -aux  
$killall -9 cat$ps -aux...  
ymqqqqdx 1729 0.9 0.1 6492 3672 pts/1 Ss 13:09 0:00 bash  
ymqqqqdx 1761 0.0 0.0 2716 1076 pts/1 R+ 13:10 0:00 ps -aux

## 1.25 ifconfig

查看或设置网络

### 1.25.1 查看网络设置

直接输入**ifconfig**即可查看网络的IP,网关和子网掩码。

$ifconfigeth0   
Link encap:以太网   
硬件地址 00:26:9e:27:21:1e   
inet 地址:192.168.0.4   
广播:192.168.0.255   
掩码:255.255.255.0   
inet6 地址: fe80::226:9eff:fe27:211e/64   
Scope:Link UP BROADCAST RUNNING MULTICAST   
MTU:1500 跃点数:1 接收数据包:5456 错误:0 丢弃:0 过载:0 帧数:0   
发送数据包:1380 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:1139542 (1.1 MB) 发送字节:244972 (244.9 KB)   
中断:17

### 1.25.2 设置网卡参数

**ifconfig eth0 192.168.0.1 netmask 255.255.255.0**  
设置IP地址和子网掩码。

$ifconfig eth0 192.168.0.1 netmask 255.255.255.0

[更多ifconfig语法。](http://baike.baidu.com/view/1001503.htm)

# 1.26 ping

## 1.26 ping

利用它可以检查网络是否能够连通，用好它可以很好地帮助我们分析判定网络故障. 检查192.168.0.1是否可以连通，输入**ping 192.168.0.1**,按下Ctrl + C停止。

$ping 192.168.0.1PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.  
64 bytes from 192.168.0.1: icmpseq=1 ttl=64 time=0.417 ms   
64 bytes from 192.168.0.1: icmpseq=2 ttl=64 time=0.399 ms   
64 bytes from 192.168.0.1: icmpseq=3 ttl=64 time=0.410 ms   
64 bytes from 192.168.0.1: icmpseq=4 ttl=64 time=0.388 ms   
64 bytes from 192.168.0.1: icmpseq=5 ttl=64 time=0.403 ms   
^C   
--- 192.168.0.1 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 3999ms   
rtt min/avg/max/mdev = 0.388/0.403/0.417/0.020 ms

不加任何参数时会一直不停地进行Ping操作。使用-c参数可以指定次数：

$ping 192.168.0.1 -c 4PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.   
64 bytes from 192.168.0.1: icmpseq=1 ttl=64 time=0.417 ms   
64 bytes from 192.168.0.1: icmpseq=2 ttl=64 time=0.399 ms   
64 bytes from 192.168.0.1: icmpseq=3 ttl=64 time=0.410 ms   
64 bytes from 192.168.0.1: icmpseq=4 ttl=64 time=0.388 ms   
^C   
--- 192.168.0.1 ping statistics ---   
4 packets transmitted, 4 received, 0% packet loss, time 3999ms   
rtt min/avg/max/mdev = 0.388/0.403/0.417/0.020 ms

### 1.26.1 检查当前网络是否连通外网

**ping www.baidu.com**  
若可以ping通则说明主机与外网连接正常且DNS服务正常。

$ping www.baidu.com -c 4PING www.baidu.com (61.135.169.125) 56(84) bytes of data.   
64 bytes from www.baidu.com: icmpseq=1 ttl=64 time=0.417 ms   
64 bytes from www.baidu.com: icmpseq=2 ttl=64 time=0.399 ms   
64 bytes from www.baidu.com: icmpseq=3 ttl=64 time=0.410 ms   
64 bytes from www.baidu.com: icmpseq=4 ttl=64 time=0.388 ms   
^C   
--- 61.135.169.125 ping statistics ---   
4 packets transmitted, 4 received, 0% packet loss, time 3999ms   
rtt min/avg/max/mdev = 0.388/0.403/0.417/0.020 ms

# 1.27 netstat

## 1.27 netstat

netstat是一个监控TCP/IP网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息.Netstat用于显示与IP、TCP、UDP和ICMP协议相关的统计数据，一般用于检验本机各端口的网络连接情况。  
netstat -a可以显示所有的socket。  
netstat | grep “5789” 查看5789端口的的连接。

$netstat | grep "5789"unix 3 [ ] 流 已连接 5789 /var/run/acpid.socket

[更多关于netstat。](http://baike.baidu.com/view/28008.htm)

# 1.28 环境变量

## 1.28 环境变量

### 1.28.1 环境变量的概念

在我们的当前目录下有一个可执行文件，它的作用是打印出hello world!，我们执行一下：

$lshello.c hello$hellohello: 找不到命令$./hellohello world!

想一下，为什么我们执行ls的时候就可以执行，而要执行hello的时候就必须要加上./呢？  
我们先用which命令看一下ls这个文件在哪里存放：

$which ls/bin/ls

在我们没有告诉系统完整路径时，系统会在一个名为PATH的的环境变量所存的路径下去找这个文件，我们用env打印当前的环境变量看一下：

$env...  
DESKTOPSESSION=gnome  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games  
QTIMMODULE=xim  
PWD=/home/ymqqqqdx   
XMODIFIERS=@im=ibus  
...

我们找到PATH项可以看到/bin在这个环境变量中，所以我们输入ls后，系统会到这个目录下去寻找。

### 1.28.1 更改环境变量

如果我们要把hello所在的目录也添加到PATH里面，是不是直接输入hello就可以直接运行我们这个程序了呢？  
下面我们来试一下:

pwd/home/ymqqqqdxexport PATH=/home/ymqqqqdxhellohello world!lsls: 找不到命令

这是怎么了？怎么连ls都找不到了呢？  
我们上面把PATH设置为了/home/ymqqqqdx/，而这个目录下没有ls，所以系统就报错了。不要急，我们重新启动一下终端就好了。  
正确的做法是这样的：

$export PATH=$PATH:/home/ymqqqqdxhellohello world!lshello.c hello

### 1.28.3 修改终端启动脚本

由上面的例子我们可以看出，我们做的修改在关闭终端后没有保存，怎么才能将我们做的修改保存起来呢？  
修改~/.bashrc：

$gedit ~/.bashrc

此时我们用gedit打开了这个文件，到这个文件的最下方，添加我们上面输入的export PATH=$PATH:/home/ymqqqqdx，然后保存退出。  
这时我们重启一下终端，终端就会按着这个脚本顺序执行，把我们的路径添加到了环境变量之中。

# 1.29 ssh/scp

## 1.29.1 ssh

大多数人以前没有接触过Linux系统，都是从Windows系统过来的，Windows下的一些实用功能例如共享文件，这样的功能在Linux系统也是可以实现的，而且更便捷。  
先在要登陆的机器上安装ssh-server服务：

$ sudo apt-get install openssh-server

查看该机器的IP：

$ ifconfig

假设我们查询到这台机器的IP为192.168.0.247，并且知道这台机器的用户名和密码，在自己的机器上进行如下操作：

$ ssh xwp@192.168.0.247The authenticity of host '192.168.0.247 (192.168.0.247)' can't be established.   
RSA key fingerprint is 6a:0d:44:dc:41:5e:8b:f7:a2:9f:9c:2d:47:92:4b:c8. Are you sure you want to continue connecting (yes/no)?yesxwp@192.168.0.247's password:xwp@xwp-desktop:~$

这样我们就用自己的机器连接上了另一台机器，我们可以用命令行对这台机器进行操作。

## 1.29.2 scp

下面我们介绍如何复制另一台机器上的文件，比如我们要复制192.168.0.247:/home/xwp/mind 目录到本地，如下操作：

$ scp -r xwp@192.168.0.247:/home/xwp/mind ./The authenticity of host '192.168.0.247 (192.168.0.247)' can't be established.  
RSA key fingerprint is 6a:0d:44:dc:41:5e:8b:f7:a2:9f:9c:2d:47:92:4b:c8.  
Are you sure you want to continue connecting (yes/no)? yes   
Failed to add the host to the list of known hosts   
(/home/ymqqqqdx/.ssh/known\\_hosts).  
xwp@192.168.0.247's password:   
linux.mm 100% 2070 2.0KB/s 00:00   
gfh.mm 100% 1792 1.8KB/s 00:00$ ls. .. mind

下面介绍一个如何将本地的文件复制到远端，这个一定要掌握，因为在以后的学习中，提交代码采用这种方式。例如：本地的目录为yourname，要提交这个目录，教师机的IP地址为192.168.0.247，代码提交到教师机的/home/xwp/code/下：

$ scp -r yourname xwp@192.168.0.247:/home/xwp/code/the authenticity of host '192.168.0.247 (192.168.0.247)' can't be established.   
RSA key fingerprint is 6a:0d:44:dc:41:5e:8b:f7:a2:9f:9c:2d:47:92:4b:c8.   
Are you sure you want to continue connecting (yes/no)? yes   
Failed to add the host to the list of known hosts   
(/home/ymqqqqdx/.ssh/knownhosts).  
xwp@192.168.0.247's password:   
main.c 100% 34KB 33.9KB/s 00:00   
add.c 100% 2417 2.4KB/s 00:00 