

## ➤ Linux基本命令操作

**inspur** 浪潮

# Linux系统认识

## ►根目录下各个目录的意义

目录名称	目录功能
<code>/bin</code>	二进制可执行命令
<code>/dev</code>	设备特殊文件
<code>/etc</code>	系统管理和配置文件
<code>/etc/rc.d</code>	启动的配置文件和脚本
<code>/home</code>	用户主目录的基点，比如用户user的主目录就是/home/user，可以用~user表示
<code>/lib</code>	标准程序设计库，又叫动态链接共享库，作用类似windows里的.dll文件

# LinuxLinux系统认识

<b>/sbin</b>	系统管理命令，这里存放的是系统管理员使用的管理程序
<b>/tmp</b>	公用的临时文件存储点
<b>/root</b>	系统管理员的主目录
<b>/mnt</b>	系统提供这个目录是让用户临时挂载其他的文件系统。
<b>/lost+found</b>	这个目录平时是空的，系统非正常关机而留下碎片的文件（windows下是.chk文件）就在这里

# Linux基础

<b>/proc</b>	虚拟的目录，是系统内存的映射。可直接访问这个目录来获取系统信息
<b>/var</b>	某些大文件的溢出区，比方说各种服务的日志文件
<b>/usr</b>	最庞大的目录，要用到的应用程序和文件几乎都在这个目录。其中包含下面的目录.
<b>/usr/X11R6</b>	存放X window的目录
<b>/usr/bin</b>	众多的应用程序
<b>/usr/sbin</b>	超级用户的一些管理程序

# LinuxLinux系统认识

<b>/usr/doc</b>	linux文档
<b>/usr/include</b>	下开发和编译应用程序所需要的头文件
<b>/usr/lib</b>	常用的动态链接库和软件包的配置文件
<b>/usr/man</b>	帮助文档
<b>/usr/src</b>	源代码，linux内核的源代码就放在/usr/src/linux里
<b>/usr/local/bin</b>	本地增加的命令
<b>/usr/local/lib</b>	本地增加的库

# Linux常用命令

## ► 常用命令

### 1、cd命令

语法：`cd [目录]`

功能说明：`cd`指令改变当前目录到指定目录。

执行范例：

```
[root@localhost home]# pwd
/home
[root@localhost home]# cd
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd -
/home
[root@localhost home]# _
```

# Linux常用命令

## 2、date命令

语法：date [选择性参数] [必要参数]

功能说明：根据指定的格式显示当前的时间或者设置系统时间

执行范例：

### 1) 查看当前服务器日期

```
[root@localhost home]# date
Wed Oct 10 00:00:03 CST 2007
[root@localhost home]# _
```

### 2) 设置服务器的日期

```
[root@localhost home]# date -s 2008/02/29
Fri Feb 29 00:00:00 CST 2008
[root@localhost home]# _
```

### 3) 设置服务器的时间

```
[root@localhost home]# date -s 09:34:00
Fri Feb 29 09:34:00 CST 2008
[root@localhost home]# _
```

# Linux常用命令

## 3、ls命令

语法：ls [参数] [文件]

功能说明：列出目录或者文件名称

执行范例：

### 1) 列当前目录下的文件名

```
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost ~]# _
```

### 2) 列出所有目录的详细信息

```
[root@localhost ~]# ls -l
total 64
-rw-r--r-- 1 root root 1182 Jul 4 2007 anaconda-ks.cfg
-rw-r--r-- 1 root root 44308 Jul 4 2007 install.log
-rw-r--r-- 1 root root 3077 Jul 4 2007 install.log.syslog
[root@localhost ~]# _
```

# Linux常用命令

## 4、pwd命令

语法：pwd [选择性参数]

功能说明：pwd指令可以使用户获得目前的路径信息

执行范例：

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost ~]# _
```

# Linux常用命令

## 5、man命令

语法：man [必要参数] [选择性参数]

功能说明：格式化和显示在线手册

执行范例：

- 1) 列出ls命令的用法: man ls

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuSUX nor --sort.

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

  -a, --all
        do not hide entries starting with .

  -A, --almost-all
        do not list implied . and ..

  --author
        print the author of each file

: _
```

# Linux常用命令

## 6、exit命令

语法：exit

功能说明：exit指令用来退出当前的shell

执行范例：

- 1) 输入exit命令退出当前的登录状态

```
Red Hat Enterprise Linux AS release 4 (Nahant)
Kernel 2.6.9-5.EL on an i686

localhost login: _
```

注：请在服务器操作完成以后一定要退出登录状态。

# Linux常用命令

## 7、echo命令

语法：echo [必要参数] [选择性参数] [字符串]

功能说明：显示文本行

执行范例：

```
[root@localhost ~]# echo $HOME
/root
[root@localhost ~]# _
```

# Linux常用命令

## ►文件管理

### 1、cat命令

语法：cat [必要参数] [选择性参数] [文件]

功能说明：cat指令用于连接文件到标准输出

执行范例：

- 1) 显示Makefile文件的内容

```
[root@localhost power]# ls
Kconfig Makefile
[root@localhost power]# cat Makefile

ifeq ($(CONFIG_PM_DEBUG),y)
EXTRA_CFLAGS += -DDEBUG
endif

swsusp-smp-$(CONFIG_SMP) += smp.o

obj-y := main.o process.o console.o pm.o
obj-$(CONFIG_SOFTWARE_SUSPEND) += swsusp.o $(swsusp-smp-y) disk.o

obj-$(CONFIG_MAGIC_SYSRQ) += poweroff.o
[root@localhost power]# _
```

# Linux常用命令

## 2、more命令

语法：more [选择性参数] [必要参数]

功能说明：按屏幕输出指定文件的内容。More命令类似于cat,但是比cat功能强。按空格下翻一页,按“b”上翻一页

执行范例：

### 1) 显示Kconfig文件的内容

```
[root@localhost power1]# more Kconfig
config PM
    bool "Power Management support"
    ---help---
    "Power Management" means that parts of your computer are shut
    off or put into a power conserving "sleep" mode if they are not
    being used.  There are two competing standards for doing this: APM
    and ACPI.  If you want to use either one, say Y here and then also
    to the requisite support below.

    Power Management is most important for battery powered laptop
    computers; if you have a laptop, check out the Linux Laptop home
    page on the WWW at <http://www.linux-on-laptops.com/> or
    Tuxmobil - Linux on Mobile Computers at <http://www.tuxmobil.org/>
    and the Battery Powered Linux mini-HOWTO, available from
    <http://www.tldp.org/docs.html#howto>.

    Note that, even if you say N here, Linux on the x86 architecture
    will issue the hlt instruction if nothing is to be done, thereby
    sending the processor to sleep and saving power.

config PM_DEBUG
    bool "Power Management Debug Support"
    ---help---
--More--(33%)_
```

# Linux常用命令

## 3、rm命令

语法：rm [必要参数] [选择性参数] [目录或者文件]

功能说明：rm指令用来删除文件或者目录

执行范例：

### 1) 删除leofang.txt文件

```
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog  leofang.txt
[root@localhost ~]# rm leofang.txt
rm: remove regular empty file `leofang.txt'? y
[root@localhost ~]# _
```

### 2) 删除avcon目录

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon  install.log  install.log.syslog
[root@localhost ~]# rm avcon
rm: cannot remove directory `avcon': Is a directory
[root@localhost ~]# rm -r avcon
rm: remove directory `avcon'? y
[root@localhost ~]# _
```

### 3) 强制删除所有文件或者目录

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon  install.log  install.log.syslog  leofang.txt
[root@localhost ~]# rm -rf avcon leofang.txt
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost ~]# _
```

# Linux常用命令

## 4、grep命令

语法：grep [参数] 模板样式 文件名称或者目录

功能说明：搜索文件中符合条件的字符串

执行范例：

- 1) 搜索Makefile中含有Usage字符串的行

```
[root@localhost 2.6.9-5.EL-i686]# grep -n "Usage" Makefile
284:# Usage: cflags-y += $(call gcc-option, -march=winchip-c6, -march=i586)
294:# Usage: flag := $(call gcc-option-yn, -march=winchip-c6)
299:# Usage gcc-ver := $(call cc-version $(CC))
1237:# Usage: $(call if_changed_rule,foo)
1277:# Usage:
1282:# Usage:
1288:# Usage is deprecated, because make does not see this as an invocation of m
ake.
[root@localhost 2.6.9-5.EL-i686]# _
```

# Linux常用命令

## 5、cp命令

语法：cp [必要参数] [选择性参数] [源目录或者文件] [目标文件或者目录]

功能说明：复制源文件到目标文件，或者复制多个源到目标文件夹

执行范例：

### 1) 复制源文件到目标文件夹

```
[root@localhost ~]# cp avcon.txt /root/avcon
[root@localhost ~]# _
```

### 2) 复制多个源文件到目录文件夹

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon.txt      install.log      leofang.txt
avcon           avonline.txt  install.log.syslog
[root@localhost ~]# cp avcon.txt leofang.txt avonline.txt /root/avcon
cp: overwrite '/root/avcon/avcon.txt'? y
[root@localhost ~]# _
```

### 3) 复制源文件夹及源文件夹下的文件到目标文件夹

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon.txt      install.log      leofang.txt
avcon           avonline.txt  install.log.syslog
[root@localhost ~]# cp avcon /
cp: omitting directory 'avcon'
[root@localhost ~]# cp -R avcon /
[root@localhost ~]# ls /avcon
avcon.txt  avonline.txt  leofang.txt
[root@localhost ~]# _
```

# Linux常用命令

## 6、mv命令

语法：mv [选择性参数] [必要参数]

功能说明：移动或者重命名指定文件

执行范例：

### 1) 移动源文件到目标文件夹

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon.txt      install.log      leofang.txt
avcon            avonline.txt   install.log.syslog
[root@localhost ~]# mv avonline.txt /root/avcon/
mv: overwrite '/root/avcon/avonline.txt'? y
[root@localhost ~]# _
```

### 2) 复制多个源文件到目录文件夹

```
[root@localhost ~]# ls
anaconda-ks.cfg  avcon  avcon.txt  install.log  install.log.syslog  leofang.txt
[root@localhost ~]# mv -f avcon.txt leofang.txt /root/avcon/
[root@localhost ~]# _
```

### 3) 重命名指定文件

```
[root@localhost avcon]# ls
avcon.txt  avonline.txt  leofang.txt
[root@localhost avcon]# mv leofang.txt fangqikun.txt
[root@localhost avcon]# ls
avcon.txt  avonline.txt  fangqikun.txt
[root@localhost avcon]# _
```

# Linux常用命令

## 7、wc命令

语法：wc [必要参数] [选择性参数] [文件]

功能说明：wc指令用来计算指定文件的字数。

执行范例：

### 1) 显示Makefile的字节数目

```
[root@localhost net]# ls -l Makefile
-rw-r--r-- 3 root root 1274 Jan 6 2005 Makefile
[root@localhost net]# wc -c Makefile
1274 Makefile
[root@localhost net]# _
```

### 2) 显示Makefile的行数

```
[root@localhost net]# wc -l Makefile
48 Makefile
[root@localhost net]# _
```

# Linux常用命令

## 8、find命令

语法：find [目录] [选择性参数] [必要参数]

功能说明：find指令用于在文件树中查找文件，并做出相应的处理

执行范例：

- 1) 在系统中查找avonline.txt文件存在的位置

```
[root@localhost ~]# find / -name avonline.txt
/root/avcon/avonline.txt
/avcon/avonline.txt
[root@localhost ~]# _
```

# Linux常用命令

## ►目录管理

### 1、mkdir命令

语法：mkdir [选择性参数] [必要参数]

功能说明：如果必要参数指定的目录不存在，则创建它

执行范例：

#### 1) 创建一个avonline目录

```
[root@localhost ~]# ls
anaconda-ks.cfg avcon install.log install.log.syslog
[root@localhost ~]# mkdir avonline
[root@localhost ~]# ls
anaconda-ks.cfg avcon avonline install.log install.log.syslog
[root@localhost ~]# _
```

#### 2) 依层建立相应目录

```
[root@localhost ~]# ls
anaconda-ks.cfg avcon avonline install.log install.log.syslog
[root@localhost ~]# mkdir avonline/leofang/fangqikun
mkdir: cannot create directory 'avonline/leofang/fangqikun': No such file or directory
[root@localhost ~]# mkdir -p avonline/leofang/fangqikun
[root@localhost ~]# _
```

# Linux常用命令

## 2、rmdir命令

语法：rmdir [必要参数] [选择性参数] [目录]

功能说明：rmdir指令用来删除空目录，如果目录非空，将出现错误。

执行范例：

### 1) 删除目录

```
[root@localhost ~]# ls
anaconda-ks.cfg avcon avonline install.log install.log.syslog
[root@localhost ~]# rmdir avonline
rmdir: `avonline': Directory not empty
[root@localhost ~]# rmdir avonline/leofang/fangqikun/
[root@localhost ~]# _
```

# Linux常用命令

## 4、du命令

语法：du [必要参数] [选择性参数] [目录或者文件]

功能说明：显示每个文件的磁盘使用空间，对目录同样可以适用

执行范例：

### 1) 显示文件的大小

```
[root@localhost libexec]# du
14020  ./gcc/i386-redhat-linux/3.4.3
14028  ./gcc/i386-redhat-linux
14036  ./gcc
200    ./emacs/21.3/i386-redhat-linux
208    ./emacs/21.3
216    ./emacs
52     ./getconf
24     ./awk
400    ./openssh
14776  .
[root@localhost libexec]# du -h
14M    ./gcc/i386-redhat-linux/3.4.3
14M    ./gcc/i386-redhat-linux
14M    ./gcc
200K   ./emacs/21.3/i386-redhat-linux
208K   ./emacs/21.3
216K   ./emacs
52K    ./getconf
24K    ./awk
400K   ./openssh
15M    .
[root@localhost libexec]# _
```

# Linux常用命令

## 5、df命令

语法：df [必要参数] [选择性参数] [文件或者设备]

功能说明：df指令显示指定磁盘文件的可用空间。如果没有文件名被指定，则所有当前挂载的文件系统的可用空间将被显示。

执行范例：

### 1) 显示当前磁盘的使用情况

```
[root@localhost libexec]# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                    9643660      1143124    8010656   13% /
/dev/hda1            101086        9136      86731    10% /boot
none                 128020         0         128020    0% /dev/shm
[root@localhost libexec]#
[root@localhost libexec]# df -h
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                    9.2G  1.1G  7.7G   13% /
/dev/hda1            99M   9.0M   85M   10% /boot
none                 126M     0   126M    0% /dev/shm
[root@localhost libexec]# _
```

# Linux常用命令

## ▶ 权限命令

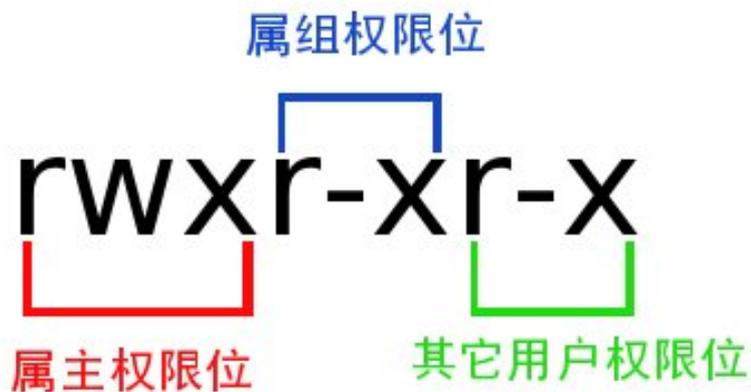
### 1、chmod命令

语法：`chmod` [必要参数] [选择性参数] [<权限范围>+/-/=<权限设置>] [文件或者目录]

功能说明：`chmod`指令依据权限设置改变给定文件的权限。

# Linux常用命令

每个文件或目录都有一组9个权限位，每三位被分为一组，他们分别是属主权限位（占三个位置）、用户组权限位（占三个位置）、其它用户权限位（占三个位置）。比如rwxr-xr-x，我们数一下就知道是不是9个位置了，正是这9个权限位来控制文件属主、用户组以及其它用户的权限。



# Linux常用命令

关于权限位；

Linux文件或目录的权限位是由9个权限位来控制，每三位为一组，它们分别是文件所有者(Ower)的读、写、执行，用户组(Group)的读、写、执行以及(Other)其它用户的读、写、执行；

文件所有者： 读r、写w、执行x

用户组： 读r、写w、执行x

其它用户： 读r、写w、执行x

如果权限位不可读、不可写、不可执行，是用-来表示。

对于普通文件的读、写、执行权限可以这样理解：

可读：意味着我们可以查看阅读；

可写：意味着，可以修改或删除

可执行：意味着如果文件就可以运行，比如二进制文件（比如命令），  
或脚本（要用脚本语言解释器来解释运行）。

# Linux常用命令

比如:

```
[root@localhost bin]# ls -l mail
-rwxr-xr-x 1 root mail 67984 Jun 16 2004 mail
[root@localhost bin]# _
```

第一个字段 -rwxr-xr-x 中的第一个字符是 -，表示 mail 是一个普通文件；mail 的权限是 rwxr-xr-x，表示 mail 文件，文件的所有者 root，拥有 rwx（可读、可写、可执行）权限，用户组 mail，拥有 r-x（可读、可写）权限，其它用户拥有 r-x（可读、可写）权限。这 9 个权限连在一起就是 rwxr-xr-x，也就是说，mail 文件，文件所有者 root 拥有可读、可写、可执行权限，用户组 mail 下的所有用户拥有可读可执行权限，其它用户拥有可读可执行权限

查看文件的属性用 ls -l 文件；查看目录的属性是 ls -d 目录。

# Linux常用命令

我们前面已经说了，文件或目录的权限位总共有9个位置，文件所有者、文件所归属的组占用三位和其它用户各占用三个位置。看个例子：

```
-rwxr-xr-x 1 root root 0 Jul 11 07:13 linuxsir007.txt
```

说明：

所有者权限位：占用三个位置，三个位置的顺序是读<sup>r</sup>、写<sup>w</sup>、执行<sup>x</sup>。如果是没有权限，则为<sup>-</sup>。在这个例子中，我们看到<sup>rwx</sup>，表示所有者在三个权位上都有权限，也就是可读可写可执行；

归属组权限位：占用三个位置，三个位置的顺序是读<sup>r</sup>、写<sup>w</sup>、执行<sup>x</sup>。如果是没有权限，则为<sup>-</sup>。在这个例子中，我们看到的是<sup>r-x</sup>，在写的位置上是<sup>-</sup>，表示没有写权限，文件所归属的组对文件拥有的是可读可执行，但没有写的权限。

# Linux常用命令

其它用户权限位：占用三个位置，三个位置的顺序是读r、写w、执行x，如果是没有权限，则为-。在这个例子中，我们看其它用户的权限位是r-x，在写的位置上是-，表示没有写权限，文件所归属的组对文件拥有的是可读可执行，但没有写的权限

chmod 的八进制语法的数字说明；

R(4)          w(2)          x(1)          -(0)

所有者的权限用数字表达：所有的那三个权限位的数字加起来的总和。比如上面的例子中所有者的权限是rwx，也就是4+2+1，应该是7；

归属组的权限用数字表达：归属组的那个权限位数字的相加的总和。比如上面的例子中的r-x，也就是4+0+1，应该是5；

其它用户的权限数字表达：其它用户权限位的数字相加的总和。比如上面例子中是 r-x，也就是4+0+1，应该是5；

# Linux常用命令

```
[root@localhost bin]# ls -l sun.txt  
-rwxr-xr-x  1 root root 0 Jul 11 07:21 sun.txt
```

注：查看sun.txt的属性，我们看到sun.txt的权限位是rwxr-xr-x，用八进制数字表示是755；

```
[root@localhost bin]# chmod 644 sun.txt
```

注：我们改变它的权限为属主可读可写、属组可读、其它用户可读，也就是rw-r--r--，用数字表示就是644；

```
[root@localhost bin]# ls -l sun.txt  
-rw-r--r--  1 root root 0 Jul 11 07:21 sun.txt
```

# Linux常用命令

**chmod** 的助记语法相对简单，对文件或目录权限的改变时，是通过比较直观的字符的形式来完成；在助记语法中，相关字母的定义；

用户或用户组定义：

u 代表属主

g 代表属组

o 代表其它用户

a 代表属主、属组和其它用户，也就是上面三个用户(或组)的所有权限定义字母：

r 代表读权限；

w 代表写权限；

x 代表执行权限；

权限增减字符：

- 代表减去相关权限；

+ 代表增加相关权限；

# Linux常用命令

示例一：

```
[root@localhost bin]# ls -l linuxsir007.txt
-rwxr-xr-x  1 root root 0 Jul 11 07:13 linuxsir007.txt
[root@localhost bin]# chmod ugo-x linuxsir007.txt
```

注：把属主、用户组及其它用户的执行权限都减掉；

```
[root@localhost bin]# ls -l linuxsir007.txt
-rw-r--r--  1 root root 0 Jul 11 07:13 linuxsir007.txt
```

示例二：

```
[root@localhost bin]# ls -l linuxsir007.txt
-rw-r--r--  1 root root 0 Jul 11 07:13 linuxsir007.txt
```

注：为文件的属主增加执行权限

```
[root@localhost bin]# chmod u+x linuxsir007.txt
```

```
[root@localhost bin]# ls -l linuxsir007.txt
-rwxr--r--  1 root root 0 Jul 11 07:13 linuxsir007.txt
[root@localhost bin]# _
```

# Linux常用命令

示例三:

```
[root@localhost bin]# ls -l linuxsir007.txt
-rwxr--r-- 1 root root 0 Jul 11 07:13 linuxsir007.txt
[root@localhost bin]# chmod u-x,go+rw linuxsir007.txt
```

注: 减去文件归属者对文件的执行权, 增加归属组和其它用户对文件的可读可写权;

```
[root@localhost bin]# ls -l linuxsir007.txt
-rw-rw-rw- 1 root root 0 Jul 11 07:13 linuxsir007.txt
[root@localhost bin]# _
```

用助记语法比较灵活, 组合起来比较方便; 比如;

**u=r+x** 为文件归属者添加读写权限;

**ug=rwx,o=r** 为文件归属者和归属组添加读、写、执行权限, 为其它用户设置读权限。

**a+x** 为文件的归属者、归属组和其它用户添加执行权限;

**g=u** 让文件的归属组和归属者的权限相同;

# Linux常用命令

## 2、chown命令

语法：chown [必要参数] [选择性参数] [用户或者群组]

功能说明：chown指令改变每一个给定文件的用户和/或群组的拥有权限

执行范例：

- 1) 把avcon.txt文件的拥有者改变为guest

```
[root@localhost avcon]# ls -l avcon.txt
-rw-r--r--  1 root root 0 Feb 29 10:04 avcon.txt
[root@localhost avcon]# chown guest avcon.txt
[root@localhost avcon]# ls -l avcon.txt
-rw-r--r--  1 guest root 0 Feb 29 10:04 avcon.txt
[root@localhost avcon]# _
```

# Linux常用命令

## 3、chgrp命令

语法：chgrp [必要参数] [选择性参数] [群组] [文件或者目录]

功能说明：chgrp指令用来改变文件或者目录所属的群组

执行范例：

- 1) 把avcon.txt文件的拥有者改变为guest

```
[root@localhost avcon]# ls -l
total 0
-rw-r--r--  1 guest root 0 Feb 29 10:04 avcon.txt
-rw-r--r--  1 root  root 0 Feb 29 10:04 avonline.txt
-rw-r--r--  1 root  root 0 Feb 29 10:04 fangqikun.txt
[root@localhost avcon]# chgrp guest avcon.txt
[root@localhost avcon]# ls -l
total 0
-rw-r--r--  1 guest guest 0 Feb 29 10:04 avcon.txt
-rw-r--r--  1 root  root  0 Feb 29 10:04 avonline.txt
-rw-r--r--  1 root  root  0 Feb 29 10:04 fangqikun.txt
[root@localhost avcon]# _
```

# Linux常用命令

## ►系统命令

### 1、mount命令

语法：mount [选择性参数] [必要参数]

功能说明：将指定设备上已存在的文件系统按照指定类型进行装载

执行范例：

#### 1) 挂载CDROM到指定位置

```
[root@localhost media]# ls -l /media/cdrom/
total 0
[root@localhost media]# mount /dev/cdrom /media/cdrom/
mount: block device /dev/cdrom is write-protected, mounting read-only
[root@localhost media]# ls /media/cdrom
autorun          README-ja.html      RELEASE-NOTES-gu.html
EULA             README-ko.html      RELEASE-NOTES-hi.html
GPL              README-pa.html      RELEASE-NOTES-it.html
images           README-pt_BR.html   RELEASE-NOTES-ja.html
isolinux         README-ta.html      RELEASE-NOTES-ko.html
README-bn.html   README-zh_CN.html   RELEASE-NOTES-pa.html
README-de.html   README-zh_TW.html   RELEASE-NOTES-pt_BR.html
README-en        RedHat              RELEASE-NOTES-ta.html
README-en.html   RELEASE-NOTES-bn.html
README-es.html   RELEASE-NOTES-de.html
README-fr.html   RELEASE-NOTES-en    RPM-GPG-KEY
README-gu.html   RELEASE-NOTES-en.html
README-hi.html   RELEASE-NOTES-es.html
README-it.html   RELEASE-NOTES-fr.html
TRANS.TBL
SRPMS
```

# Linux常用命令

## 2、umount命令

语法：umount [必要参数] [选择性参数] [文件系统]

功能说明：umount指令用来卸载挂载的文件系统

执行范例：

### 1) 卸载CDROM的挂载

```
[root@localhost media]# ls /media/cdrom/
autorun          README-ja.html      RELEASE-NOTES-gu.html
EULA             README-ko.html      RELEASE-NOTES-hi.html
GPL              README-pa.html      RELEASE-NOTES-it.html
images           README-pt_BR.html   RELEASE-NOTES-ja.html
isolinux         README-ta.html      RELEASE-NOTES-ko.html
README-bn.html   README-zh_CN.html   RELEASE-NOTES-pa.html
README-de.html   README-zh_TW.html   RELEASE-NOTES-pt_BR.html
README-en        RedHat              RELEASE-NOTES-ta.html
README-en.html   RELEASE-NOTES-bn.html RELEASE-NOTES-zh_CN.html
README-es.html   RELEASE-NOTES-de.html RELEASE-NOTES-zh_TW.html
README-fr.html   RELEASE-NOTES-en    RPM-GPG-KEY
README-gu.html   RELEASE-NOTES-en.html SRPMS
README-hi.html   RELEASE-NOTES-es.html TRANS.TBL
README-it.html   RELEASE-NOTES-fr.html
[root@localhost media]# umount /dev/cdrom
[root@localhost media]# ls /media/cdrom/
[root@localhost media]# _
```

# Linux常用命令

## 3、uname命令

语法：uname [必要参数] [选择性参数]

功能说明：uname指令用来显示系统的相关信息

执行范例：

- 1) 显示系统所有信息

```
[root@localhost media]# uname -a
Linux localhost.localdomain 2.6.9-5.EL #1 Wed Jan 5 19:22:18 EST 2005 i686 i686
i386 GNU/Linux
[root@localhost media]# _
```

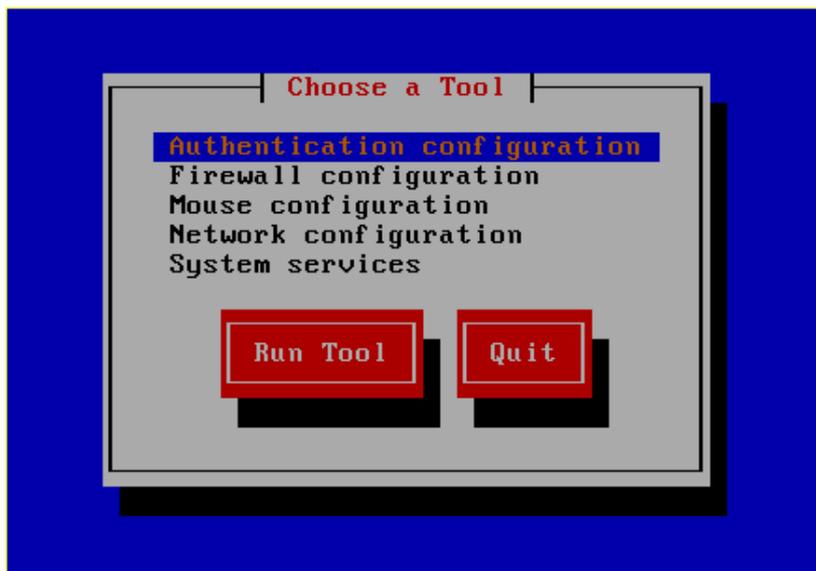
# Linux常用命令

## 4、setup命令

语法：setup

功能说明：setup指令用来设置认证、防火墙、鼠标、网络配置、打印机配置、系统服务等

执行范例：



# Linux常用命令

## 5、dmesg命令

语法：dmesg [必要参数] [选择性参数]

功能说明：dmesg指令用来检测和控制内核ring缓冲区。它帮助用户输出开机信息。

执行范例：

### 1) 显示开机信息

```
[root@localhost media]# dmesg | more
Linux version 2.6.9-5.EL (bhcompile@decompose.build.redhat.com) (gcc version 3.4
.3 20041212 (Red Hat 3.4.3-9.EL4)) #1 Wed Jan 5 19:22:18 EST 2005
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 0000000000fef0000 (usable)
 BIOS-e820: 0000000000fef0000 - 0000000000feff000 (ACPI data)
 BIOS-e820: 0000000000feff000 - 0000000000ff00000 (ACPI NVS)
 BIOS-e820: 0000000000ff00000 - 00000000010000000 (usable)
 BIOS-e820: 0000000000fec00000 - 0000000000fec10000 (reserved)
 BIOS-e820: 0000000000fee00000 - 0000000000fee01000 (reserved)
 BIOS-e820: 0000000000ffe00000 - 000000000100000000 (reserved)
0MB HIGHMEM available.
256MB LOWMEM available.
zapping low mappings.
On node 0 totalpages: 65536
 DMA zone: 4096 pages, LIFO batch:1
 Normal zone: 61440 pages, LIFO batch:15
 HighMem zone: 0 pages, LIFO batch:1
DMI present.
ACPI: RSDP (0000 PTLTD ) @ 0x000f6c20
ACPI: RSDT (0001 PTLTD RSDT 0x06040000 LTP 0x00000000) @ 0x0fefab68
--More--
```

# Linux常用命令

## 6、top命令

语法：top [必要参数] [选择性参数]

功能说明：top指令可以显示当前系统正在执行的进程的相关信息。

执行范例：

### 1) 显示系统进行信息

```
top - 11:41:16 up 2:25, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 42 total, 1 running, 41 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 0.4% sy, 0.0% ni, 99.6% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 256044k total, 254096k used, 1948k free, 56676k buffers
Swap: 524280k total, 0k used, 524280k free, 159404k cached
```

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
27253	root	16	0	2508	916	756	R	0.7	0.4	0:00.08	top
2192	root	16	0	6796	4224	1692	S	0.3	1.6	0:04.33	hald
1	root	16	0	3588	516	440	S	0.0	0.2	0:00.89	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.06	ksoftirqd/0
3	root	5	-10	0	0	0	S	0.0	0.0	0:00.33	events/0
4	root	5	-10	0	0	0	S	0.0	0.0	0:00.01	khelper
5	root	15	-10	0	0	0	S	0.0	0.0	0:00.00	kacpid
18	root	5	-10	0	0	0	S	0.0	0.0	0:00.12	kblockd/0
28	root	15	0	0	0	0	S	0.0	0.0	0:01.22	pdflush
29	root	15	0	0	0	0	S	0.0	0.0	0:00.26	pdflush
31	root	10	-10	0	0	0	S	0.0	0.0	0:00.00	aio/0
19	root	15	0	0	0	0	S	0.0	0.0	0:00.00	khubd
30	root	16	0	0	0	0	S	0.0	0.0	0:00.07	kswapd0
105	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kseriod
180	root	22	0	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_0
187	root	6	-10	0	0	0	S	0.0	0.0	0:00.00	kmirror/0
196	root	15	0	0	0	0	S	0.0	0.0	0:03.38	kjournald
1004	root	6	-10	2792	484	404	S	0.0	0.2	0:00.07	udev

# Linux常用命令

## 8、kill命令

语法：`kill [必要参数] [选择性参数] [进程号]`

功能说明：`kill`指令发送指定的信号到相应进程。

执行范例：

- 1) 终止掉<sup>27286</sup>这个进程

```
[root@localhost media]# ps -A | grep sendmail
27286 ?          00:00:00 sendmail
[root@localhost media]# kill 27286
[root@localhost media]# ps -A | grep sendmail
[root@localhost media]# _
```

# Linux常用命令

## 9、shutdown命令

语法：shutdown [参数] 时间 [警告信息]

功能说明：shutdown指令用来关闭系统，然后再通知内核是停止、重启还是关闭电源

执行范例：

- 1) 关闭系统并关闭电源(时间为分钟,如果需要为秒数则需要添加参数 -t)

```
[root@localhost media]# shutdown 10 -h
Broadcast message from root (tty1) (Fri Feb 29 11:54:06 2008):
The system is going DOWN for system halt in 10 minutes!
-
```

- 2) 重新启动系统(时间为分钟,如果需要为秒数则需要添加参数 -t)

```
[root@localhost media]# shutdown -r 10
Broadcast message from root (tty1) (Fri Feb 29 11:55:13 2008):
The system is going DOWN for reboot in 10 minutes!
-
```

# Linux常用命令

## ►用户管理

### 1、useradd命令

语法：useradd [必要参数] [选择性参数]

功能说明：useradd 指定用来建立帐号

执行范例：

#### 1) 建立用户leofang

```
[root@localhost media]# useradd leofang
[root@localhost media]# _
```

#### 2) 建立用户fangqikun并添加到avcon用户组里

```
[root@localhost media]# useradd -g avcon fangqikun
[root@localhost media]# _
```

# Linux常用命令

## 2、userdel命令

语法：userdel [必要参数] [用户]

功能说明：userdel指令用来删除用户帐号。还可以通过加入参数，删除与指定用户相关的文件

执行范例：

- 1) 删除fangqikun用户

```
[root@localhost media]# userdel fangqikun  
[root@localhost media]# _
```

# Linux常用命令

## 3、groupadd命令

语法：groupadd [选择性参数] [必要参数] 群组

功能说明：groupadd指令用来创建一个新的群组

执行范例：

- 1) 增加一个avcon的群组

```
[root@localhost media]# groupadd avcon  
[root@localhost media]# _
```

# Linux常用命令

## 4、groupdel命令

语法：groupdel [群组]

功能说明：groupdel指令用来改变系统的帐户文件，删除指定用户的入口。指定的群组必须存在。

执行范例：

- 1) 删除avcon群组

```
[root@localhost media]# groupdel avcon  
[root@localhost media]# _
```

# Linux常用命令

## 5、who命令

语法：`who`[必要参数][选择性参数][记录文件]

功能说明：`who`指令用来显示当前登录系统的用户的信息

执行范例：

- 1) 显示当前系统中的登录人员信息

```
[root@localhost media]# who
root      tty1      Feb 29 09:42
root      tty2      Feb 29 13:18
root      tty4      Feb 29 13:18
[root@localhost media]# _
```

# Linux常用命令

## 6、last命令

语法：last [必要参数]

功能说明：last指令查找/var/log/wtmp文件，并显示在此文件创建后的所有登录用户，可以指定数字或者时间来显示用户的登录情况。

执行范例：

### 1) 显示系统的人员登录信息

```
[root@localhost media]# last
root      tty4          Fri Feb 29 13:18  still logged in
root      tty2          Fri Feb 29 13:18  still logged in
root      tty1          Fri Feb 29 09:42  still logged in
root      tty1          Wed Sep  5 13:33  - 09:41 (176+20:07)
reboot    system boot   2.6.9-5.EL       Wed Sep  5 13:31  (176+23:47)
reboot    system boot   2.6.9-5.EL       Wed Sep  5 13:30  (176+23:49)
root      tty1          Wed Sep  5 13:29  - crash (00:00)
reboot    system boot   2.6.9-5.EL       Wed Sep  5 13:28  (176+23:50)
root      tty1          Wed Jul  4 10:30  - down (00:00)
reboot    system boot   2.6.9-5.EL       Wed Jul  4 10:29  (00:01)

wtmp begins Wed Jul  4 10:29:21 2007
[root@localhost media]# _
```

# Linux常用命令

## 7、passwd命令

语法：passwd [必要参数] [选择性参数] [用户名称]

功能说明：passwd指令用来设置用户帐号的密码，超级用户可以设置所有用户的密码

执行范例：

- 1) 修改用户fangqikun的密码

```
[root@localhost media]# passwd fangqikun
Changing password for user fangqikun.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost media]# _
```

# Linux常用命令

## ► 常用的网络命令

### 1、ifconfig命令

语法：ifconfig [网络设备] [必要参数] [选择性参数]

功能说明：ifconfig指令用来配置网络设备，当系统启动时，将调用此指令配置网络设备；此外，当进行调试或者系统的网络环境发生变化时，将调用此指令进行相应的网络配置。

执行范例：

# Linux常用命令

**ifconfig** 如果不接任何参数，就会输出当前网络接口的情况；

```
[root@localhost init.d]# ifconfig
eth0      Link encap:Ethernet  HWaddr AA:0C:29:1A:73:03
          inet addr:192.168.6.200  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::a80c:29ff:fe1a:7303/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1959  errors:0  dropped:0  overruns:0  frame:0
          TX packets:2272  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:162506 (158.6 KiB)  TX bytes:653307 (637.9 KiB)
          Interrupt:10  Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:780  errors:0  dropped:0  overruns:0  frame:0
          TX packets:780  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:89145 (87.0 KiB)  TX bytes:89145 (87.0 KiB)

[root@localhost init.d]# _
```

# Linux常用命令

解说：

eth0 表示第一块网卡，其中 HWaddr 表示网卡的物理地址，我们可以看到目前这个网卡的物理地址(MAC地址)是 AA:0C:29:1A:73:03 ； inet addr 用来表示网卡的IP地址，此网卡的IP地址是 192.168.6.200 ，广播地址， Bcast:192.168.6.255 ，掩码地址Mask:255.255.255.0

lo 是表示主机的回环地址，这个一般是用来测试一个网络程序，但又不想让局域网或外网的用户能够查看，只能在此台主机上运行和查看所用的网络接口。比如我们把 HTTPD服务器的指定到回环地址，在浏览器输入 127.0.0.1 就能看到你所架WEB网站了。但只是您能看得到，局域网的

# Linux常用命令

如果我们想知道主机所有网络接口的情况，请用下面的命令：

```
[root@localhost init.d]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr AA:0C:29:1A:73:03
          inet addr:192.168.6.200  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::a80c:29ff:fe1a:7303/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1959 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2272 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:162506 (158.6 KiB)  TX bytes:653307 (637.9 KiB)
          Interrupt:10 Base address:0x1400

eth1      Link encap:Ethernet  HWaddr 00:0C:29:1A:73:0D
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:985 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:173570 (169.5 KiB)  TX bytes:3366 (3.2 KiB)
          Interrupt:9 Base address:0x1480

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:780 errors:0 dropped:0 overruns:0 frame:0
```

# Linux常用命令

如果我们想查看某个端口，比如我们想查看<sup>eth0</sup>的状态，就可以用下面的方法；

```
[root@localhost init.d]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr AA:0C:29:1A:73:03
          inet addr:192.168.6.200  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::a80c:29ff:fe1a:7303/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1959 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2272 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:162506 (158.6 KiB)  TX bytes:653307 (637.9 KiB)
          Interrupt:10 Base address:0x1400

[root@localhost init.d]# _
```

# Linux常用命令

## 2、netconfig命令

语法：netconfig

功能说明：netconfig指令用来设置网络环境，它提供图形化界面，可以设置本地主机<sup>IP</sup>地址、网络掩码、网关以及域名服务器。

1) 设置网卡的DHCP模式自动获得IP

执行范例：

```
[root@localhost init.d]# netconfig -d eth0 --bootproto=dhcp
[root@localhost init.d]# _
```

2) 手动设置网卡的IP等

```
[root@localhost init.d]# netconfig -d eth0 --ip=192.168.6.200 --netmask=255.255.255.0 --gateway=192.168.6.254
[root@localhost init.d]# _
```

# Linux常用命令

## 3、ping命令

语法：ping [必要参数] [选择性参数] [主机名或者IP地址]

功能说明：ping指令向指定的主机发送ICMP包并等待回复，以确定指定的主机是否存在。

执行范例：

- 1) 探测192.168.6.101主机是否存在

```
[root@localhost ~]# ping 192.168.6.101
PING 192.168.6.101 (192.168.6.101) 56(84) bytes of data.
64 bytes from 192.168.6.101: icmp_seq=0 ttl=64 time=0.336 ms
64 bytes from 192.168.6.101: icmp_seq=1 ttl=64 time=0.001 ms
64 bytes from 192.168.6.101: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 192.168.6.101: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 192.168.6.101: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 192.168.6.101: icmp_seq=5 ttl=64 time=0.034 ms

--- 192.168.6.101 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 0.001/0.077/0.336/0.116 ms, pipe 2
[root@localhost ~]# _
```

# Linux常用命令

## 1) 向192.168.6.101主机发送5次大小为1024B的数据包

```
[root@localhost ~]# ping -c 5 -s 1024 192.168.6.101
PING 192.168.6.101 (192.168.6.101) 1024(1052) bytes of data.
1032 bytes from 192.168.6.101: icmp_seq=0 ttl=64 time=0.044 ms
1032 bytes from 192.168.6.101: icmp_seq=1 ttl=64 time=0.393 ms
1032 bytes from 192.168.6.101: icmp_seq=2 ttl=64 time=0.034 ms
1032 bytes from 192.168.6.101: icmp_seq=3 ttl=64 time=0.034 ms
1032 bytes from 192.168.6.101: icmp_seq=4 ttl=64 time=0.038 ms

--- 192.168.6.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.034/0.108/0.393/0.142 ms, pipe 2
[root@localhost ~]# _
```

# Linux常用命令

## 6、wget命令

语法：wget [必要参数] [选择性参数]

功能说明：wget指令用于从网络上下载资源

执行范例：

- 1) 从www.avms.com.cn服务器上下载avcon.exe文件

```
[root@avcon ~]# wget http://www.avms.com.cn:8080/avcon/avcon.exe
--06:27:22-- http://www.avms.com.cn:8080/avcon/avcon.exe
      => `avcon.exe'
Resolving www.avms.com.cn... 211.155.31.196
Connecting to www.avms.com.cn[211.155.31.196]:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16,091,266 [application/octet-stream]

7% [==>                               ] 1,197,178    237.57K/s    ETA 00:56
```

# Linux常用命令

## 7、lsof命令

语法：lsof [选择性参数]

功能说明：lsof可以用来查看正在运行中的进程打开了哪些文件、目录和套接字；是系统监测工具之一。

执行范例：

- 1) 查看服务器上22端口是被哪个程序所占用

```
[root@localhost ~]# lsof -i:22
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
sshd     2078 root    3u   IPv6  4745      TCP *:ssh (LISTEN)
[root@localhost ~]# _
```

# Linux常用命令

## ► 文本编辑

### 1、vi命令

语法：vi [必要参数] [选择性参数] [文件]

功能说明：vi指令是Linux默认的文字编辑器，在命令行界面下编辑文字

#### 1) 如何调用vi；

```
[root@localhost ~]# vi filename_
```

# Linux常用命令

vi 的三种命令模式；

Command（命令）模式，用于输入命令；

Insert（插入）模式，用于插入文本；

Visual（可视）模式，用于视觉化的高亮并选定正文；

2)文件的保存和退出；

Command 模式是vi或vim的默认模式，如果我们处于其它命令模式时，要通过ESC键切换过来。

当我们按ESC键后，接着再输入“:”号时，vi会在屏幕的最下方等待我们输入命令；

:w 保存；

:w filename 另存为filename；

:wq! 保存退出；

:wq! filename 注：以filename为文件名保存后退出；

:q! 不保存退出；

# Linux常用命令

## 3) 光标移动;

当我们按ESC进入Command模式后，可以用下面的一些键位来移动光标;

- j 向下移动一行;
- k 向上移动一行;
- h 向左移动一个字符;
- l 向右移动一个字符;
- ctrl+b 向上移动一屏;
- ctrl+f 向下移动一屏;
- 向上箭头 向上移动;
- 向下箭头 向下移动;
- 向左箭头 向左移动;
- 向右箭头 向右移动;

我们编辑一个文件时，对于j、k、l和h键，还能在这些动作命令的前面加上数字，比如 3j，表示向下移动3行。

# Linux常用命令

## 4) 插入模式（文本的插入）；

i 在光标之前插入；

a 在光标之后插入；

I 在光标所在行的行首插入；

A 在光标所在行的行末插入；

o 在光标所在的行的上面插入一行；

O 在光标所在的行的下面插入一行；

s 删除光标后的一个字符，然后进入插入模式；

S 删除光标所在的行，然后进入插入模式；

# Linux常用命令

## 5) 文本内容的删除操作；

x 一个字符；

#x 删除几个字符，#表示数字，比如<sup>3x</sup>；

dw 删除一个单词；

#dw 删除几个单词，#用数字表示，比如<sup>3dw</sup>表示删除三个单词；

dd 删除一行；

#dd 删除多个行，#代表数字，比如<sup>3dd</sup>表示删除光标行及光标的下两行；

d\$ 删除光标到行尾的内容；

J 清除光标所处的行与上一行之间的空格，把光标行和上一行接在一起；

# 软件安装与卸载

## 1、RPM命令安装

### 1) 用RPM安装软件包

```
[root@localhost RPMS]# rpm -ivh vsftpd-2.0.1-5.i386.rpm
warning: vsftpd-2.0.1-5.i386.rpm: U3 DSA signature: NOKEY, key ID db42a60e
Preparing...                               ##### [100%]
 1:vsftpd                                   ##### [100%]
[root@localhost RPMS]# _
```

### 2) 反安装RPM软件包

```
[root@localhost RPMS]# rpm -e vsftpd
[root@localhost RPMS]# _
```

### 3) 升级

```
[root@localhost RPMS]# rpm -Uvh vsftpd-2.0.1-5.i386.rpm
warning: vsftpd-2.0.1-5.i386.rpm: U3 DSA signature: NOKEY, key ID db42a60e
Preparing...                               ##### [100%]
package vsftpd-2.0.1-5 is already installed
[root@localhost RPMS]# _
```

用户要注意的是：rpm会自动反安装相应软件包的老版本。如果老版本软件的配置文件通新版本的不兼容，rpm会自动将其保存为另外一个文件，用户会看到下面的信息：

```
saving /etc/vsftpd.conf as /etc/vsftpd.conf.rpmsave
```

# 软件安装与卸载

## 2、源代码编译安装

- 1、源码包的打包格式；

源代码一般以 `file.tar.gz` `file.tar.bz2` 或 `file.src.rpm` 打包；`file.tar.gz` 和 `file.tar.bz2` 格式的解包

命令如下；

## 2、如何编译安装源码包；（大多数）

### 1）解开软件包查看帮助文档；

我们解开一个包后，进入解压包，一般都能发现 `README`（或 `reame`）和 `INSTALL`（或 `install`）；或 `doc`（或 `DOC`）目录；看名字就知道个差不多；

# 软件安装与卸载

比如我们下载一个比较新的<sup>fcitx</sup>的软件包，比如是<sup>fcitx-3.2-050827.tar.bz2</sup>

我们在解开这个软件包会会发现如下的文件；

```
[root@localhost fcitx]# tar jxvf fcitx-3.2-050827.tar.bz2
[root@localhost fcitx]# cd fcitx
[root@localhost fcitx]# ls
aclocal.m4  config.guess  configure  INSTALL  Makefile.in  src
AUTHORS    config.h.in   configure.in  depcomp  install-sh
missing    THANKS       autogen.sh  config.rpath  COPYING
doc        lib          mkinstalldirs  TODO    ChangeLog  config.sub
data      fcitx.spec.in  Makefile.am  README  tools
```

我们就可以看<sup>fcitx</sup>的<sup>INSTALL</sup>和<sup>doc</sup>目录的安装文档；里面都告诉我们如何安装；

# 软件安装与卸载

## 2) 编译安装软件的条件；

首先我们在Linux系统中至少得把开发工具安装上，比如gcc ;perl;python;glibc;gtk;make ;automake 等开发工具或基础包；

还要安装一些相应的开发包，一般是文件名包括<sup>dev</sup>的，比如<sup>kernel-devel</sup>；还有一些开发库，比如以<sup>lib</sup>开头的；

如果您在编译软件时，有时会提示缺少什么东西之类的，大多少的是这些开发工具和开发库等；从安装光盘中找到直接安装就可以了；有时安装光盘没有提供，请用<sup>google</sup>搜索相应的软件包进行安装。

# 软件安装与卸载

## 3) 编译安装软件的方法；

大多以 `tar.gz` 和 `tar.bz2` 打包软件，大多是通过 `./configure ;make ;make install` 来安装的；有的软件是直接 `make;make install` ；

我们可以通过 `./configure --help` 来查看配置软件的功能；大多软件是提供 `./configure` 配置软件的功能的；少数的也没有，如果没有的就不用 `./configure` ；直接 `make;make install` 就行了；

`./configure` 比较重要的一个参数是 `--prefix` ，用 `--prefix` 参数，我们可以指定软件安装目录；当我们不需要这个软件时，直接删除软件的目录就行了；

比如我们可以指定 `fcitx` 安装到 `/opt/fcitx` 目录中；

```
[root@localhost fcitx]#./configure --prefix=/opt/fcitx
```

如果我们不需要 `fcitx` 时，可以直接删除 `/opt/fcitx` 目录；

# 常见的系统设置

不同的运行级定义如下：(可以参考Red Hat Linux 里面的/etc/inittab )

- # 缺省的运行级，REDHAT 用到的级别如下：
- # 0 - 停机 ( 千万不要把initdefault 设置为0 )
- # 1 - 单用户模式
- # 2 - 多用户，但是没有 NFS
- # 3 - 完全多用户模式
- # 4 - 没有用到
- # 5 - X11
- # 6 - 重新启动 ( 千万不要把initdefault 设置为6 )

# 常见的系统设置

对各个运行级的详细解释：

- 0 为停机，机器关闭。
- 1 为单用户模式，就像<sup>Win9x</sup>下的安全模式类似。
- 2 为多用户模式，但是没有<sup>NFS</sup>支持。
- 3 为完整的多用户模式，是标准的运行级。
- 4 没有用到
- 5 就是 <sup>X11</sup>，进到 <sup>X Window</sup> 系统了。
- 6 为重启，运行 <sup>init 6</sup> 机器就会重启。

# 常见问题

## 2、root密码丢失后的处理方法

1) 我们需要在服务器本机上重新启动服务器。在这个界面时，我们按任意键

```
Press any key to enter the menu

Booting Red Hat Enterprise Linux AS (2.6.9-5.EL) in 2 seconds... |
```

2) 进入到下面这个界面后，我们按`e`键进入到编辑状态

```
GNU GRUB version 0.95 (638K lower / 260032K upper memory)

Red Hat Enterprise Linux AS (2.6.9-5.EL)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```



redhat.

# 常见问题

3) 在这个界面下,我们需要把光标移动到以`kernel`开头的那一行,并再次按`e`键,进入到编辑状态

```
GNU GRUB version 0.95 (638K lower / 260032K upper memory)

root (hd0,0)
kernel /vmlinuz-2.6.9-5.EL ro root=/dev/VolGroup00/LogVol100
initrd /initrd-2.6.9-5.EL.img
```

Use the ↑ and ↓ keys to select which entry is highlighted.  
Press 'b' to boot, 'e' to edit the selected command in the  
boot sequence, 'c' for a command-line, 'o' to open a new line  
after ('O' for before) the selected line, 'd' to remove the  
selected line, or escape to go back to the main menu.



redhat.

# 常见问题

- 4) 在出现的编辑状态中,我们在这一行的最后输入一个空格和一个数字“1”,然后回车

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.9-5.EL ro root=/dev/VolGroup00/LogVol00 1
```

- 5) 此时,应该是像下面的这个图上显示的内容一样:

```
GNU GRUB version 0.95 (638K lower / 260032K upper memory)

root (hd0,0)
kernel /vmlinuz-2.6.9-5.EL ro root=/dev/VolGroup00/LogVol00 1
initrd /initrd-2.6.9-5.EL.img

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

# 常见问题

6) 这时我们再按<sup>b</sup>键,此时系统就会自动开始引导,并且不需要你输入<sup>root</sup>密码就可以登录到系统中。

```
Telling INIT to go to single user mode.
INIT: Going single user
INIT: Sending processes the TERM signal
INIT: Sending processes the KILL signal
sh-3.00# _
```

7) 我们这个时候就可以使用<sup>passwd</sup>命令来更改<sup>root</sup>的密码了。

```
Telling INIT to go to single user mode.
INIT: Going single user
INIT: Sending processes the TERM signal
INIT: Sending processes the KILL signal
sh-3.00# passwd
Changing password for user root.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
sh-3.00# _
```

# 常见问题

## 4、SSH连接服务器时速度很慢

问：当使用SSH客户端连接服务器时，在输入IP以后要过很久才会出现输入密码的提示。

答： 请检查 `/etc/resolv.conf` 文件里的 `nameserver` 选项，请确认 `nameserver` 后的DNS IP地址是你的服务器可以访问到的，因为SSH在连接时会做反向查询，所以如果你的 `nameserver` 服务器填写的是一个无法访问的DNS IP地址时，SSH服务端会进行等待，直到超时，所以就会造成SSH等待很久才会做出回应。如果是这样，我们只要删除 `nameserver` 里不正确的IP地址，再进行连接时，速度就会很快了。

# 常见问题

## 5、查看系统硬件配置信息

查看CPU信息：`cat /proc/cpuinfo`

查看板卡信息：`cat /proc/pci`

查看PCI信息：`lspci` (相比`cat /proc/pci`更直观)

查看内存信息：`cat /proc/meminfo`

查看USB设备：`cat /proc/bus/usb/devices`

查看键盘和鼠标：`cat /proc/bus/input/devices`

# 常见问题

## 6、挂载U盘和移动硬盘

1) 插入U盘前,先用 `fdisk -l` 查看当前系统的磁盘挂载情况

```
[root@localhost ~]# fdisk -l
Disk /dev/hda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1 *          1           13        104391   83  Linux
/dev/hda2             14          1305       10377990  8e  Linux LUM
[root@localhost ~]# _
```

2) 插入U盘以后,屏幕上会出现提示信息,这时我们再输入 `fdisk -l` 就会看到多出一个磁盘,这个磁盘就是U盘的设备名,然后我们就可以使用 `mount` 命令对U盘进行挂载操作

```
[root@localhost ~]# Vendor: Generic Model: USB Flash Drive Rev: 1.03
Type: Direct-Access ANSI SCSI revision: 02
SCSI device sda: 257536 512-byte hdwr sectors (132 MB)
sda: Write Protect is off
sda: assuming drive cache: write through
Attached scsi removable disk sda at scsi2, channel 0, id 0, lun 0

[root@localhost ~]# fdisk -l
Disk /dev/hda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1 *          1           13        104391   83  Linux
/dev/hda2             14          1305       10377990  8e  Linux LUM

Disk /dev/sda: 131 MB, 131858432 bytes
16 heads, 32 sectors/track, 503 cylinders
Units = cylinders of 512 * 512 = 262144 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *          1           503       128752    b  W95 FAT32
[root@localhost ~]# _
```

# 如何学习linux

## ▶从头学习

理解一下基础的硬体知识，不用一定要全懂(没那么多时间)，但是至少要『听过、有概念』

了解一下 Linux 的基础知识，这些包含了『使用者、群组的概念』、『权限的观念』，『程序的定义』等等

必需至少学会一种以上的文书编辑器，例如最好学会通用版本的 vi！

实际操作 Linux 时，必定要学习 Shell，最好 Shell scripts 也能够了解；（Shell 是一个命令行解释器，是用户使用 linux 的桥梁，它的作用就是遵循一定的语法将输入的命令加以解释并传给系统）



# 如何学习linux

## ▶ 选择一本好的工具书

[http://linux.vbird.org/linux\\_basic/0120howtolinux/0120howtolinux\\_1.php](http://linux.vbird.org/linux_basic/0120howtolinux/0120howtolinux_1.php)

## ▶ 实践再实践

在人类记忆的曲线中，必须要『不断的重复练习』才会将一件事情记得比较熟！同样的，学习 Linux 也一样，如果你无法经常摸索的话，那么，抱歉的是，学了后面的，前面的忘光光！学了等于没学，所以，除了要实作之外，还得要常摸！才会熟悉 Linux 而且不会怕他

## ▶ 学习的方法

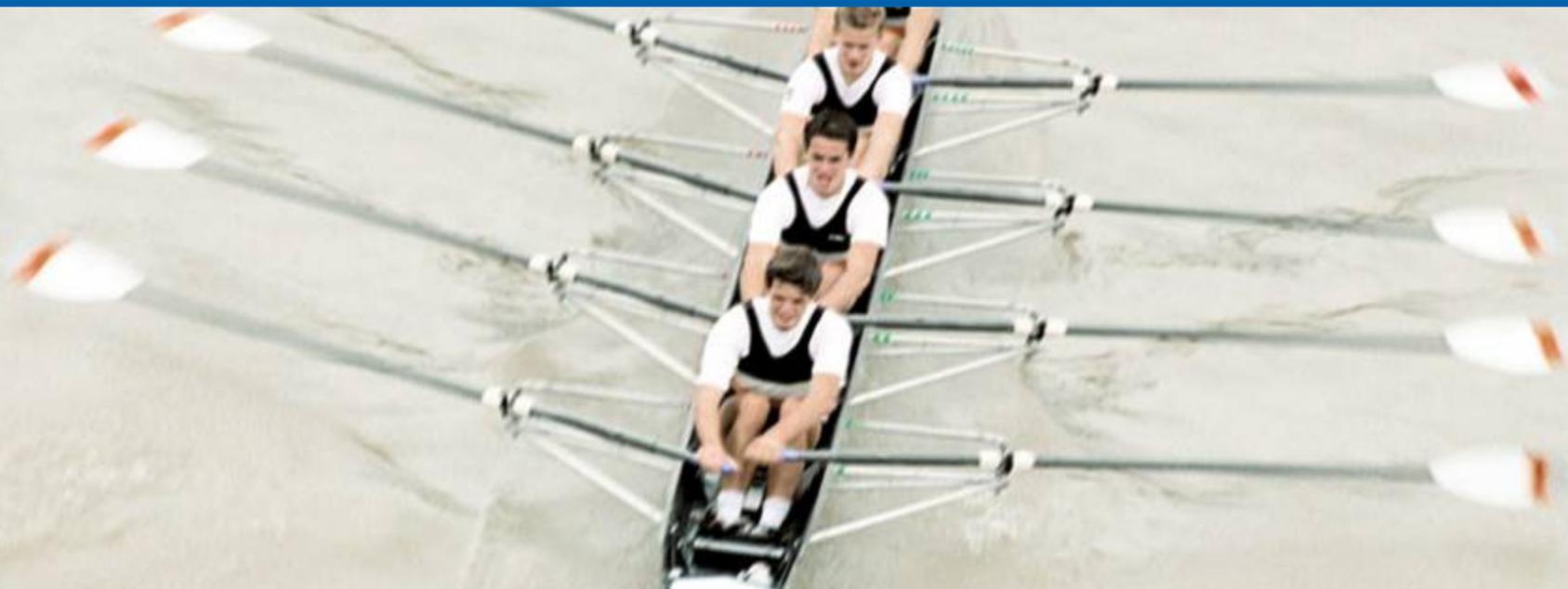
x - windows 还是 command line ?

X - windows 适合桌面版用户，一般不进行服务器的架设，而 command - line 适合主要用于服务器架设的用户，并且 x - windows 对系统资源的要求较高。不适合远程控制。

## ▶ 主机、网络资料的查询

网络资源是帮助学习和故障排查的有力助手，使用 google 对遇到故障和疑难进行搜索是查询解决方案的好办法。

谢谢大家!



**inspur** 浪潮