

---

# Linux 基础

## 一、Linux 简介

### 1. Linux 概述

Linux 是基于 Unix 的开源免费的操作系统，由于系统的稳定性和安全性几乎成为程序代码运行的最佳系统环境。Linux 是由 Linus Torvalds（林纳斯·托瓦兹）起初开发的，由于源代码的开放性，现在已经衍生出了千上百种不同的 Linux 系统。

Linux 系统的应用非常广泛，不仅可以长时间的运行我们编写的程序代码，还可以安装在各种计算机硬件设备中，比如手机、平板电脑、路由器等。尤其在这里提及一下，我们熟知是 Android 程序最底层就是运行在 linux 系统上的。



图 1 林纳斯·托瓦兹



图 2 linux 的 logo

### 2. Linux 的分类

(1) Linux 根据市场需求不同，基本分为两个方向：

**图形化界面版：**注重用户体验，类似 window 操作系统，但目前成熟度不够

**服务器版：**没有好看的界面，是以在控制台窗口中输入命令操作系统的，类似于 DOS，是我们假设服务器的最佳选择。

## (2) Linux 根据原生程度，又分为两种：

**内核版本：**在 Linus 领导下的内核小组开发维护的系统内核的版本号

**发行版本：**一些组织或公司在内核版基础上进行二次开发而重新发行的版本

## (3) Linux 发行版本不同，又可以分为 n 多种：



## 二、Linux 的安装

### 1. 虚拟机的安装

因为 Linux 也是一个系统，本质上跟我们电脑的 Window 没有区别，所以我们要学习 Linux 就首先将我们电脑的 Window 系统换成 Linux 系统，或者在我们电脑上安装双系统，听上去是不是很可怕。其实我们可以在我们电脑上安装一个软件，这个软件可以模拟一台或多台虚拟的电脑机器，这就是虚拟机

虚拟机常用的有两种：

VMware（威睿）公司的虚拟机软件，功能强大，收费产品；

VitruaBox 虚拟机软件 Oracle 公司的虚拟机软件，免费商品

由于市场上 VMware 使用率很高，所以我们安装 VMware。

## 2. CentOS 的安装

CentOS 是一个 Linux 的发行版本，是目前企业中用来做应用服务器系统的主要版本，CentOS 的安装，其实是将该系统安装到 VMware 虚拟机软件中，让 VMware 虚拟机软件模拟出一台 Linux 系统的电脑。

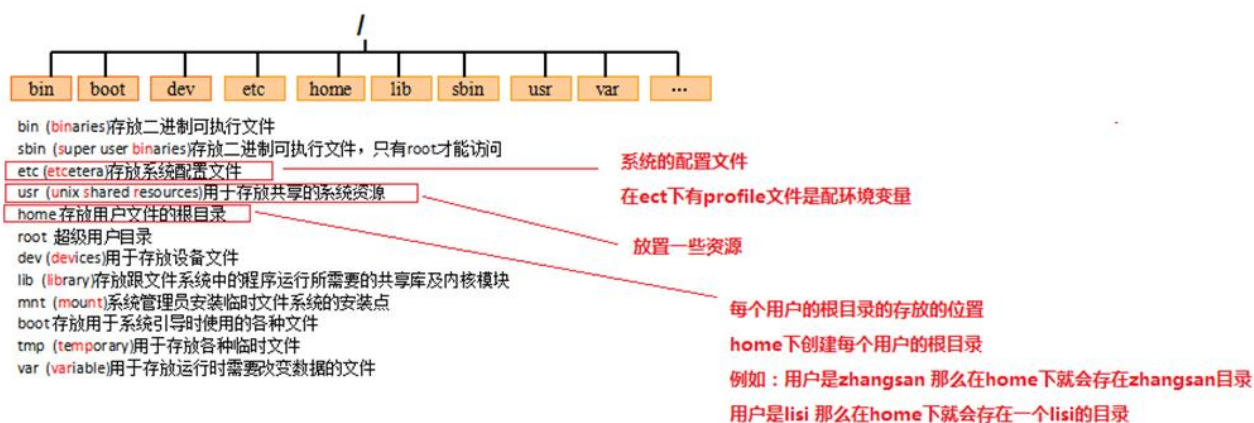
## 三、Linux 的基本命令

服务器版的 Linux 安装完毕了，但没有图形化界面，我们只能通过控制台去操作系统，我们就要使用类似 DOS 命令的 Linux 命令去操作系统，那么下面我们就进行 Linux 的命令的学习。Linux 命令成百上千，课程不能一一涉及，也没有必要一一涉及，根据我们程序员在日常工作中的需求，总结出如下几种供学习。

### 1. 目录切换命令

window 中通过图形化界面和鼠标可以任意切换需要进入的目录，但 Linux 必须通过 cd 命令切换目录。

### Linux 目录结构



`cd usr` 切换到该目录下 `usr` 目录↵  
`cd ../` 切换到上一层目录↵  
`cd /` 切换到系统根目录↵  
`cd ~` 切换到用户主目录↵  
`cd -` 切换到上一个所在目录↵

## 2. 目录的操作命令（增删改查）

### (1)增加目录操作（增）

命令：mkdir 目录名称

示例：在根目录 / 下 mkdir test , 就会在根目录 / 下产生一个 test 目录

```
[root@CentOS ~]# ls
bin  cgroup  etc  lib          media  mnt  opt  root  selinux  sys  usr
boot dev    home lost+found  misc  net  proc sbin  srv      tmp  var
[root@CentOS ~]# mkdir test
[root@CentOS ~]# ls
bin  cgroup  etc  lib          media  mnt  opt  root  selinux  sys  tmp  var
boot dev    home lost+found  misc  net  proc sbin  srv      test usr
[root@CentOS ~]# _
```

### (2)查看目录（查）

命令：ls [-al] 父目录

示例：在根目录 / 下使用 ls , 可以看到该目录下的所有的目录和文件

```
[root@CentOS ~]# ls
bin  cgroup  etc  lib          media  mnt  opt  root  selinux  sys  tmp  var
boot dev    home lost+found  misc  net  proc sbin  srv      test usr
[root@CentOS ~]# _
```

示例：在根目录 / 下使用 ls -a , 可以看到该目录下的所有文件和目录，包括隐藏的

```
[root@CentOS ~]# ls -a
.      bin  dev  lib          misc  opt  sbin  sys  usr
..     boot etc  lost+found  mnt   proc selinux test var
.autofsck 都是隐藏文件
[root@CentOS ~]# _
```

示例：在根目录 / 下使用 ls -l , 可以看到该目录下的所有目录和文件的详细信息

```
dr-xr-xr-x.  2 root root  4096 Jun 14 18:08 bin
dr-xr-xr-x.  5 root root  1024 Jun 14 17:22 boot
drwxr-xr-x.  2 root root  4096 Jul 24 2015 cgroup
drwxr-xr-x. 19 root root  3820 Jun 14 17:25 dev
drwxr-xr-x. 105 root root 12288 Jun 14 18:08 etc
drwxr-xr-x.  2 root root  4096 Sep 23 2011 home
dr-xr-xr-x. 18 root root 12288 Jun 14 18:08 lib
```

**注意：ls -l 可以缩写成 ll**

### (3)寻找目录 ( 查 )

命令：find 目录 参数

示例：查找/root 下的与 test 相关的目录(文件) `find /root -name 'test*'`

### (4)修改目录的名称 ( 改 )

命令：mv 目录名称 新目录名称

示例：test 目录下有一个 oldTest 目录，使用 mv oldTest newTest 命令修改

```
[root@CentOS test]# ls
oldTest
[root@CentOS test]# mv oldTest newTest
[root@CentOS test]# ls
newTest
[root@CentOS test]# _
```

注意：mv 的语法不仅可以对目录进行重命名而且也可以对各种文件，压缩包等进行重命名的操作

### (5)移动目录的位置---剪切 ( 改 )

命令：mv 目录名称 目录的新位置

示例：在 test 下将 newTest 目录剪切到 /usr 下面，使用 mv newTest /usr

```
[root@CentOS test]# ls
newTest
[root@CentOS test]# mv newTest /usr
[root@CentOS test]# ls
[root@CentOS test]# cd /usr
[root@CentOS usr]# ls
bin etc games include lib libexec local newTest sbin share
[root@CentOS usr]# cd -
/test
[root@CentOS test]# ls
[root@CentOS test]# _
```

原test目录中已经没有newTest了

注意：mv 语法不仅可以对目录进行剪切操作，对文件和压缩包等都可执行剪切操作

### (6)拷贝目录 ( 改 )

命令：cp -r 目录名称 目录拷贝的目标位置 -----r 代表递归拷贝

示例：将/usr下的 newTest 拷贝到根目录下的 test 中，使用 cp -r /usr/newTest /test

```
[root@CentOS test]# ls
[root@CentOS test]# cp -r /usr/newTest /test
[root@CentOS test]# ls
newTest
[root@CentOS test]# cd /usr
[root@CentOS usr]# ls
bin etc games include lib libexec local newTest sbin share
[root@CentOS usr]# _
```

## (7)删除目录（删）

命令：rm [-rf] 目录

示例：删除/usr下的 newTest，进入/usr下使用 rm -r newTest

```
[root@CentOS usr]# ls
bin etc games include lib libexec local newTest sbin share src tmp
[root@CentOS usr]# rm -r newTest/
rm: remove directory 'newTest'? y
[root@CentOS usr]# ls
bin etc games include lib libexec local sbin share src tmp
[root@CentOS usr]# _
```

示例：删除/test下的 newTest 而不需要询问强制删除，在/test下使用 rm -rf newTest

```
[root@CentOS test]# ls
newTest
[root@CentOS test]# rm -rf newTest/
[root@CentOS test]# ls
[root@CentOS test]# _
```

注意：rm 不仅可以删除目录，也可以删除其他文件或压缩包，为了增强大家的记忆，无论删除任何目录或文件，都直接使用 rm -rf 目录/文件/压缩包

## 3. 文件的操作命令（增删改查）

### (1)文件的创建（增）

命令：touch 文件名称 ----- 空文件

示例：在 test 目录下创建一个空文件 touch aaa.txt

```
[root@CentOS test]# ls
[root@CentOS test]# touch aaa.txt
[root@CentOS test]# ls
aaa.txt
[root@CentOS test]# _
```

## (2)文件的查看（查）

命令：cat/more/less/tail 文件

示例：使用 cat 查看/etc/sudo.conf 文件，只能显示最后一屏内容

```
#
# Sudo noexec:
#
# Path to a shared library containing dummy versions of the execv(),
# execve() and fexecve() library functions that just return an error.
# This is used to implement the "noexec" functionality on systems that
# support C<LD_PRELOAD> or its equivalent.
# The compiled-in value is usually sufficient and should only be changed
# if you rename or move the sudo_noexec.so file.
#
#Path noexec /usr/libexec/sudo_noexec.so

#
# Core dumps:
#
# By default, sudo disables core dumps while it is executing (they
# are re-enabled for the command that is run).
# To aid in debugging sudo problems, you may wish to enable core
# dumps by setting "disable_coredump" to false.
#
# Set to false here so as not to interfere with /proc/sys/fs/suid_dumpable
#
Set disable_coredump false
[root@CentOS etc]# _
```

示例：使用 more 查看/etc/sudo.conf 文件，可以显示百分比，回车可以向下一行，空格可以向下一页，q 可以退出查看

```
#
# Format:
#   Plugin plugin_name plugin_path plugin_options ...
#   Path askpass /path/to/askpass
#   Path noexec /path/to/sudo_noexec.so
#   Debug sudo /var/log/sudo_debug all@warn
#   Set disable_coredump true
#
# Sudo plugins:
#
# The plugin_path is relative to ${prefix}/libexec unless fully qualified.
# The plugin_name corresponds to a global symbol in the plugin
#   that contains the plugin interface structure.
# The plugin_options are optional.
#
# The sudoers plugin is used by default if no Plugin lines are present.
Plugin sudoers_policy sudoers.so
Plugin sudoers_io sudoers.so
#
# Sudo askpass:
#
# An askpass helper program may be specified to provide a graphical
# password prompt for "sudo -A" support. Sudo does not ship with its
--More--(43%)_
```

示例：使用 less 查看/etc/sudo.conf 文件，可以使用键盘上的 PgUp 和 PgDn 向上和向下翻页，q 结束查看

```
#
# Sudo noexec:
#
# Path to a shared library containing dummy versions of the execv(),
# execve() and fexecve() library functions that just return an error.
# This is used to implement the "noexec" functionality on systems that
# support C<LD_PRELOAD> or its equivalent.
# The compiled-in value is usually sufficient and should only be changed
# if you rename or move the sudo_noexec.so file.
#
#Path noexec /usr/libexec/sudo_noexec.so
#
# Core dumps:
:_
```



示例：使用 tail -10 查看/etc/sudo.conf 文件的后 10 行，Ctrl+C 结束

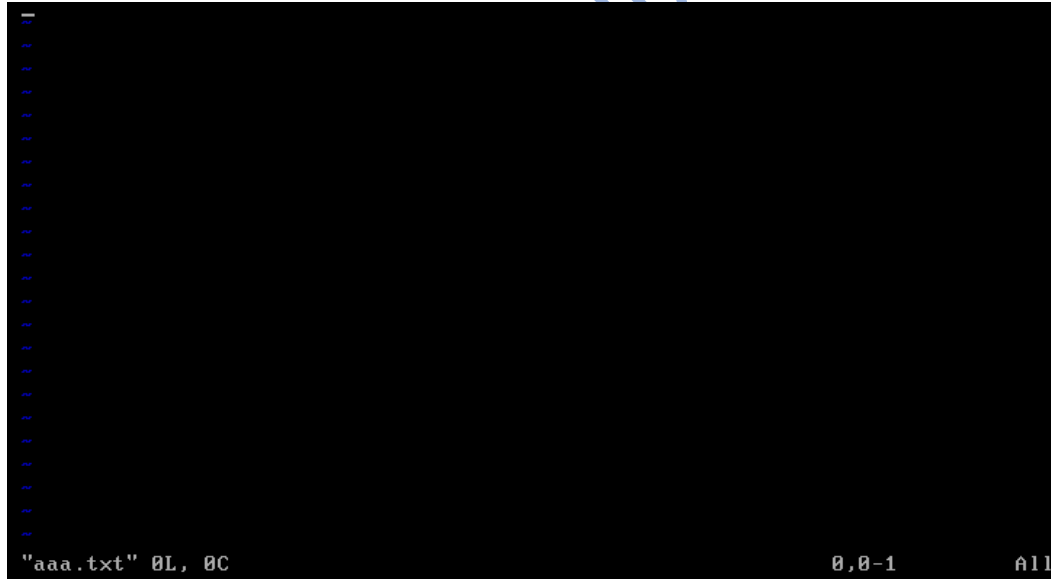
```
[root@CentOS etc]# tail -10 sudo.conf
# Core dumps:
#
# By default, sudo disables core dumps while it is executing (they
# are re-enabled for the command that is run).
# To aid in debugging sudo problems, you may wish to enable core
# dumps by setting "disable_coredump" to false.
#
# Set to false here so as not to interfere with /proc/sys/fs/suid_dumpable
#
Set disable_coredump false
[root@CentOS etc]# _
```

注意：命令 tail -f 文件 可以对某个文件进行动态监控，例如 tomcat 的日志文件，会随着程序的运行，日志会变化，可以使用 tail -f catalina-2016-11-11.log 监控文件的变化

### (3)修改文件的内容（改）

命令：vim 文件

示例：编辑/test 下的 aaa.txt 文件，使用 vim aaa.txt



但此时并不能编辑，因为此时处于命令模式，点击键盘 i/a/o 进入编辑模式，可以编辑文件

```

aaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbb
cccccccccccc_

```

编辑完成后，按下 Esc，退回命令模式

```

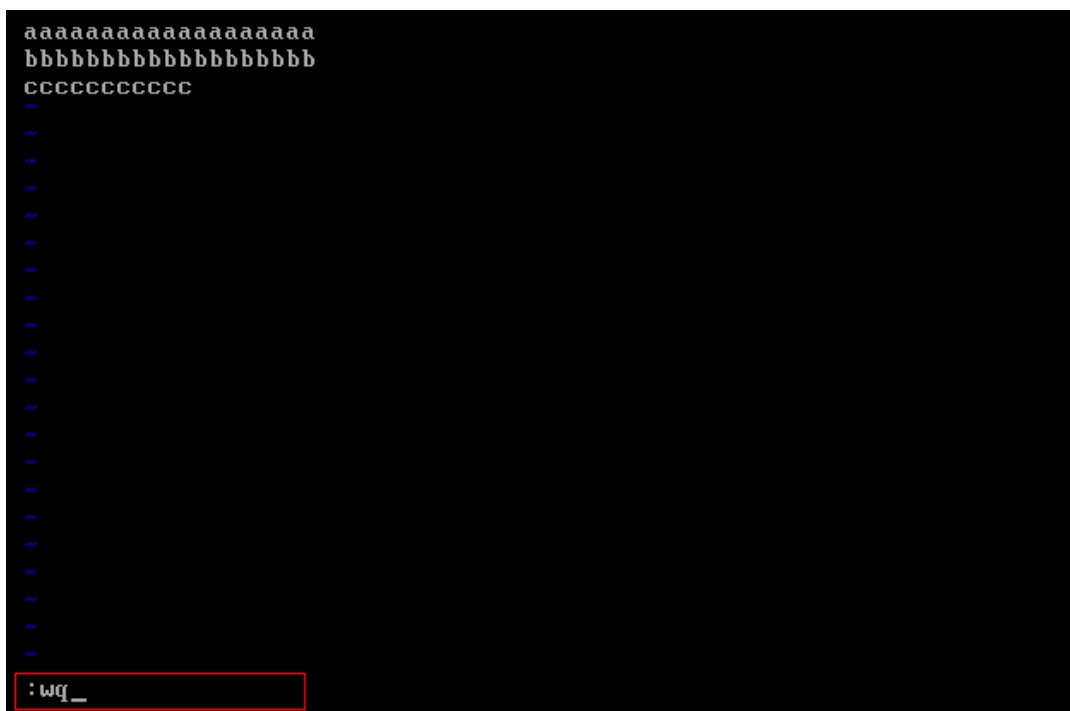
aaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbb
ccccccccccc

[... many blank lines ...]

3,11      All

```

此时文件虽然已经编辑完成，但是没有保存，需输入冒号：进入底行模式，在底行模式下输入 wq 代表写入内容并退出，即保存；输入 q!代表强制退出不保存。



### 总结：

vim 编辑器是 Linux 中的强大组件，是 vi 编辑器的加强版，vim 编辑器的命令和快捷方式有很多，但此处不一一阐述，大家也无需研究的很透彻，使用 vim 编辑修改文件的方式基本会使用就可以了。

关于 vim 使用过程：

在实际开发中，使用 vim 编辑器主要作用就是修改配置文件

**vim 文件----->进入文件----->命令模式----->按 i 进入编辑模式----->编辑文件----->按 Esc 进入底行模式----->输入:wq/q!**

### (4)删除文件（删）

同目录删除：熟记 rm -rf 文件 即可

## 4. 压缩文件的操作命令

### (1)打包并压缩文件

Linux 中的打包文件一般是以.tar 结尾的，压缩的命令一般是以.gz 结尾的。

而一般情况下打包和压缩是一起进行的，打包并压缩后的文件的后缀名一般.tar.gz。

命令：tar -zcvf 打包压缩后的文件名 要打包压缩的文件

其中：z：调用 gzip 压缩命令进行压缩

c：打包文件

v：显示运行过程

f：指定文件名

示例：打包并压缩/test 下的所有文件 压缩后的压缩包指定名称为 xxx.tar.gz

tar -zcvf xxx.tar.gz aaa.txt bbb.txt ccc.txt

或：tar -zcvf xxx.tar.gz /test/\*

```
[root@CentOS test]# ls
aaa.txt bbb.txt ccc.txt
[root@CentOS test]# tar -zcvf xxx.tar.gz aaa.txt bbb.txt ccc.txt
aaa.txt
bbb.txt
ccc.txt
[root@CentOS test]# ls
aaa.txt bbb.txt ccc.txt xxx.tar.gz
[root@CentOS test]# _
```

### (2)解压压缩包（重点）

命令：tar [-xvf] 压缩文件

其中：x：代表解压

示例：将/test 下的 xxx.tar.gz 解压到当前目录下

tar -xvf xxx.tar.gz

```
[root@CentOS test]# ls
xxx.tar.gz
[root@CentOS test]# tar -xvf xxx.tar.gz
./aaa.txt
./bbb.txt
./ccc.txt
[root@CentOS test]# ls
aaa.txt bbb.txt ccc.txt xxx.tar.gz
[root@CentOS test]# _
```

示例：将/test 下的 xxx.tar.gz 解压到根目录/usr 下

**tar -xvf xxx.tar.gz -C /usr-----C 代表指定解压的位置**

```
[root@CentOS test]# tar -xvf xxx.tar.gz -C /usr
./aaa.txt
./bbb.txt
./ccc.txt
[root@CentOS test]# cd /usr
[root@CentOS usr]# ls
aaa.txt  bin      etc      include  libexec  sbin     src
bbb.txt  ccc.txt  games    lib       local    share    tmp
[root@CentOS usr]# _
```

## 5. 其他命令

### (1)显示当前所在位置

pwd

```
[root@itheima32 local]# pwd
/usr/local
[root@itheima32 local]# _
```

### (2)搜索命令

命令：grep 要搜索的字符串 要搜索的文件

示例：搜索/usr/sudo.conf 文件中包含字符串 to 的行

```
[root@CentOS etc]# grep to sudo.conf
# Path askpass /path/to/askpass
# Path noexec /path/to/sudo_noexec.so
# The plugin_path is relative to ${prefix}/libexec unless fully qualified.
# The plugin_name corresponds to a global symbol in the plugin
# An askpass helper program may be specified to provide a graphical
# Path to a shared library containing dummy versions of the execv(),
# This is used to implement the "noexec" functionality on systems that
# To aid in debugging sudo problems, you may wish to enable core
# dumps by setting "disable_coredump" to false.
# Set to false here so as not to interfere with /proc/sys/fs/suid_dumpable
[root@CentOS etc]# _
```

示例：搜索/usr/sudo.conf 文件中包含字符串 to 的行 to 要高亮显示

```
[root@CentOS etc]# grep to sudo.conf --color
# Path askpass /path/to/askpass
# Path noexec /path/to/sudo_noexec.so
# The plugin_path is relative to ${prefix}/libexec unless fully qualified.
# The plugin_name corresponds to a global symbol in the plugin
# An askpass helper program may be specified to provide a graphical
# Path to a shared library containing dummy versions of the execv(),
# This is used to implement the "noexec" functionality on systems that
# To aid in debugging sudo problems, you may wish to enable core
# dumps by setting "disable_coredump" to false.
# Set to false here so as not to interfere with /proc/sys/fs/suid_dumpable
[root@CentOS etc]# _
```

### (3)管道命令

命令：| 将前一个命令的输出作为本次目录的输入

示例：查看当前系统中所有的进程中包括 system 字符串的进程

```
[root@CentOS etc]# ps -ef | grep system
dbus      1516      1  0 17:25 ?        00:00:00 dbus-daemon --system
root      25999  2015  0 23:30 tty1    00:00:00 grep system
[root@CentOS etc]# _
```

### (4)查看进程

命令：ps -ef

示例：查看当前系统中运行的进程

### (5)杀死进程

命令：kill -9 进程的 pid

## (6)网络通信命令

查看当前系统的网卡信息：ifconfig

```
[root@CentOS ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:00:79:73
          inet addr:192.168.193.128  Bcast:192.168.193.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe00:7973/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:765 errors:0 dropped:0 overruns:0 frame:0
          TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:85381 (83.3 KiB)  TX bytes:14864 (14.5 KiB)
          Interrupt:19 Base address:0x2000

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:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@CentOS ~]# _
```

查看与某台机器的连接情况：ping

```
[root@CentOS ~]# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=128 time=2.38 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=128 time=0.982 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=128 time=0.960 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=128 time=1.03 ms
64 bytes from 192.168.1.5: icmp_seq=5 ttl=128 time=1.62 ms
64 bytes from 192.168.1.5: icmp_seq=6 ttl=128 time=1.48 ms
64 bytes from 192.168.1.5: icmp_seq=7 ttl=128 time=0.968 ms
64 bytes from 192.168.1.5: icmp_seq=8 ttl=128 time=1.48 ms
64 bytes from 192.168.1.5: icmp_seq=9 ttl=128 time=0.980 ms
^C
--- 192.168.1.5 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8927ms
rtt min/avg/max/mdev = 0.960/1.321/2.387/0.455 ms
[root@CentOS ~]# _
```

查看当前系统的端口使用：netstat -an

```
unix 3      [ ]          STREAM     CONNECTED   13114      @/var/run/hald/dbus-zg3
BRXE7WG
unix 3      [ ]          STREAM     CONNECTED   13113
unix 3      [ ]          STREAM     CONNECTED   13033      @/var/run/hald/dbus-zg3
BRXE7WG
unix 3      [ ]          STREAM     CONNECTED   13018
unix 3      [ ]          STREAM     CONNECTED   13030      @/var/run/hald/dbus-zg3
BRXE7WG
unix 3      [ ]          STREAM     CONNECTED   12987
unix 3      [ ]          STREAM     CONNECTED   12872      @/var/run/hald/dbus-JXD
t3ohkzx
unix 3      [ ]          STREAM     CONNECTED   12871
unix 3      [ ]          STREAM     CONNECTED   12847      /var/run/dbus/system_bu
```

## 6. Linux 的权限命令

权限是 Linux 中的重要概念，每个文件/目录等都具有权限，通过 `ls -l` 命令我们可以查看某个目录下的文件或目录的权限。

示例：在随意某个目录下 `ls -l`

```
-rw-r--r--. 1 root root 128 Jun 14 23:04 aaa.txt
-rw-r--r--. 1 root root 0 Jun 14 23:03 bbb.txt
dr-xr-xr-x. 2 root root 36864 Jun 14 18:08 bin
-rw-r--r--. 1 root root 0 Jun 14 23:04 ccc.txt
drwxr-xr-x. 2 root root 4096 Sep 23 2011 etc
drwxr-xr-x. 35 root root 4096 Jun 14 17:20 include
dr-xr-xr-x. 80 root root 36864 Jun 14 18:08 lib
drwxr-xr-x. 17 root root 4096 Jun 14 18:08 libexec
drwxr-xr-x. 11 root root 4096 Jun 14 17:17 local
dr-xr-xr-x. 2 root root 12288 Jun 14 18:08 sbin
drwxr-xr-x. 125 root root 4096 Jun 14 17:21 share
drwxr-xr-x. 4 root root 4096 Jun 14 17:17 src
lrwxrwxrwx. 1 root root 10 Jun 14 17:17 tmp -> ../var/tmp
```

第一列的内容的信息解释如下：



文件的类型：

d：代表目录

-：代表文件

l：代表链接（可以认为是 window 中的快捷方式）

后面的 9 位分为 3 组，每 3 位置一组，分别代表属主的权限，与当前用户同组的 用户的权限，其他用户的权限

r：代表权限是可读，r 也可以用数字 4 表示



w：代表权限是可写，w 也可以用数字 2 表示

x：代表权限是可执行，x 也可以用数字 1 表示

属主 ( user )			属组 ( group )			其他用户		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

修改文件/目录的权限的命令：chmod

示例：修改/test 下的 aaa.txt 的权限为属主有全部权限，属主所在的组有读写权限，其他用户只有读的权限

chmod u=rwx,g=rw,o=r aaa.txt

```
[root@CentOS test]# ll
total 8
-rw-r--r--. 1 root root 2237 Jun 14 23:24 aaa.txt
-rw-r--r--. 1 root root    0 Jun 14 23:03 bbb.txt
-rw-r--r--. 1 root root    0 Jun 14 23:04 ccc.txt
-rw-r--r--. 1 root root 308 Jun 14 23:05 xxx.tar.gz
[root@CentOS test]# chmod u=rwx,g=rw,o=r aaa.txt
[root@CentOS test]# ll
total 8
-rwxrw-r--. 1 root root 2237 Jun 14 23:24 aaa.txt
-rw-r--r--. 1 root root    0 Jun 14 23:03 bbb.txt
-rw-r--r--. 1 root root    0 Jun 14 23:04 ccc.txt
-rw-r--r--. 1 root root 308 Jun 14 23:05 xxx.tar.gz
[root@CentOS test]# _
```

上述示例还可以使用数字表示：chmod 764 aaa.txt

## 四、远程连接工具的使用

实际开发中，Linux 服务器都在其他的地方，我们要通过远程的方式去连接 Linux 并操作它，Linux 远程的操作工具有很多，企业中常用的有 Putty、secureCRT、SSH Secure 等。目前自己用的是 WinSCP 和 Xshell。

关于远程工具乱码的解决：

使用远程工具进行连接时，如果 linux 有中文文件或目录，显示时会出现乱码，原因是 linux 编码是 UTF-8，而远程工具默认是当前系统本地编码即 GBK。所以解决方案是统一两者编码就 OK 了，但是该 SSH Secure 工具不能设置编码，所以通过修改 linux 的系统编码的方式进行统一编码。

在 linux 的/etc/sysconfig 目录下有一个 i18n 的文件代表 linux 的系统编码

```
[root@CentOS ~]# cd /etc/sysconfig
[root@CentOS sysconfig]# ls
atd          init          network       samba
auditd       ip6tables     networking    sandbox
authconfig   ip6tables-config network-scripts saslauthd
autofs       ip6tables.old nfs            selinux
cbq          iptables      ntpd          smartmontools
cgconfig     iptables-config ntpdate       sshd
cgroup.conf  iptables.old  prelink       sysstat
clock        irqbalance    quota_nld     sysstat.ioconf
console      kdump         raid-check    system-config-firewall
cpuspeed     kernel        readahead     system-config-firewall.old
crond        keyboard      readonly-root udev
grub         modules       rngd
i18n         netconsole    rsyslog
[root@CentOS sysconfig]# _
```

将其从 UTF-8 修改成 GBK 重现连接 linux 即可：

```
LANG="zh_CN.UTF-8"
```

```
LANG="zh_CN.GBK"
```