

Linux

1 Linux 介绍

1.1 Linux 是什么？

linux 是一个开源、免费的操作系统，其稳定性、安全性、处理多并发能力已经得到业界的认可，目前大多数企业级应用甚至是集群项目都部署运行在 linux 操作系统之上，很多软件公司考虑到开发成本都首选 linux，在中国软件公司得到广泛的使用。

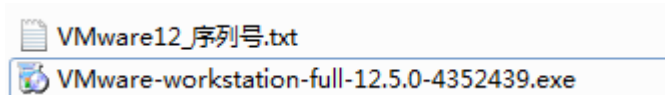
1.2 Linux 主要的发行版本

Ubuntu(乌班图)、RedHat(红帽)、CentOS、Debian[蝶变]、Fedora、SuSE、OpenSUSE、红旗 Linux(国产)

2 Linux 的安装：

2.1 下载软件

1、虚拟机软件下载：



2、CentOS 下载：

➤ 网易镜像：<http://mirrors.163.com/centos/6/isos/>

➤ 搜狐镜像：<http://mirrors.sohu.com/centos/6/isos>

2.2 安装虚拟机：

参考：“虚拟机安装手册.docx”

2.3 安装 CentOS:

参考: “CentOS 安装手册.docx”

3 Linux 的目录结构

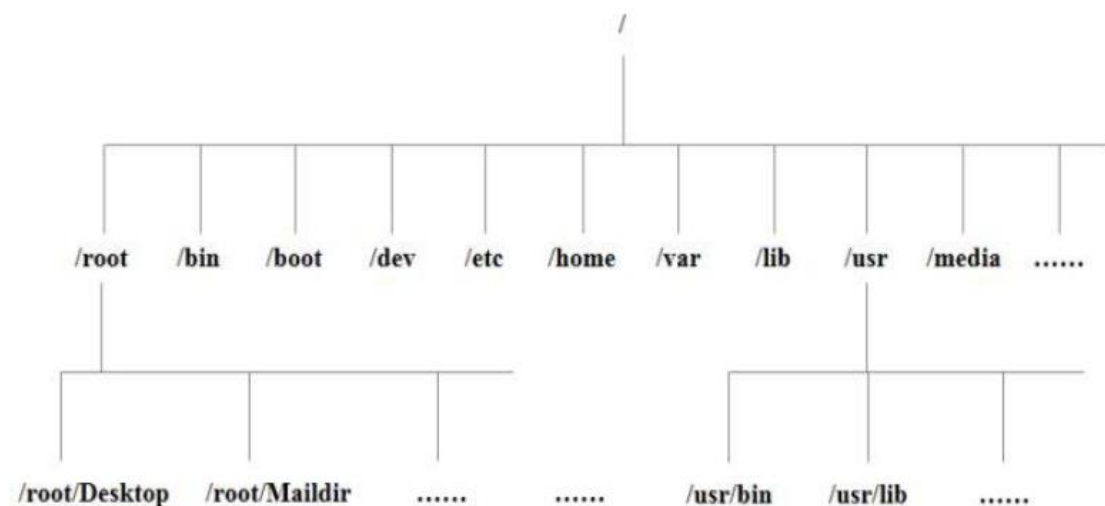
3.1 Linux 目录结构概述

Linux没有炫目的可视化操作界面，它的操作大部分都是直接执行命令，而可执行文件都是保存在相应的目录中的，所以我们对linux的操作大多数时候都是查找和执行这些可执行文件的。

linux的文件系统是采用级层式的树状目录结构，在此结构中的最上层是根目录“/”，然后在此目录下再创建其他的目录。

在Linux世界里，一切皆文件。

3.2 Linux 具体目录结构



- 1) **root** : 该目录为系统管理员目录，**root** 是具有超级权限的用户。
- 2) **bin** -> **usr/bin** : 存放系统预装的可执行程序，这里存放的可执行文件可以在系统的任何目录下执行。
- 3) **usr** 是 linux 的系统资源目录，里边存放的都是一些系统可执行文件或者系统以来的一些文件库。
- 4) **usr/local/bin** : 存放用户自己的可执行文件，同样这里存放的可执行文件可以在系统的任何目录下执行。
- 5) **lib** -> **usr/lib** : 这个目录存放着系统最基本的动态连接共享库，其作用类似于 Windows 里的 DLL 文件，几乎所有的应用程序都需要用到这些共享库。
- 6) **boot** : 这个目录存放启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件。
- 7) **dev** : **dev** 是 Device(设备)的缩写，该目录下存放的是 Linux 的外部设备，Linux 中的设备也是以文件的形式存在。
- 8) **etc** : 这个目录存放所有的系统管理所需要的配置文件。
- 9) **home** : 用户的主目录，在 Linux 中，每个用户都有一个自己的目录，一般该目录名以用户的账号命名，叫作用户的根目录；用户登录以后，默认打开自己的根目录。
- 10) **var** : 这个目录存放着在不断扩充着的东西，我们习惯将那些经常被修改的文件存放在该目录下，比如运行的各种日志文件。
- 11) **mnt** : 系统提供该目录是为了让用户临时挂载别的文件系统，我们可以将光驱挂载在 **/mnt/** 上，然后进入该目录就可以查看光驱里的内容

12) **opt**: 这是给 linux 额外安装软件所存放的目录。比如你安装一个 Oracle 数据库则就可以放到这个目录下，默认为空。

13) **tmp**: 这个目录是用来存放一些临时文件的。

4 Linux 的远程操作工具

4.1 Linux 的远程操作

在软件公司中，无论是测试服务器还是正式服务器都是部署在公司 Linux 系统上的，而 Linux 系统通常是各开发小组共享的，因此程序员需要远程登录到 centos 进行项目管理或者开发。远程登录客户端有 Xshell、Xftp 等多种工具，我们学习使用 Xshell 和 Xftp，其它的远程工具大同小异。

4.2 Xshell

1. 简介：

Xshell 是一个强大的安全终端模拟软件，它支持 SSH1, SSH2, 以及 Microsoft Windows 平台的 TELNET 协议。它通过互联网到远程系统的安全连接以及它创新性的设计和特色帮助用户在复杂的网络环境中享受他们的工作。

Xshell 是目前最好的远程登录到 Linux 操作系统的软件，流畅的速度并且完美解决了中文乱码的问题，是目前程序员首选的软件。

2. 下载：<https://www.netsarang.com>

3. 安装和使用：参考“Xshell 安装手册.docx”

4.3 Xftp

1. 简介：

是一个基于 windows 平台的功能强大的 SFTP、FTP 文件传输软件。使用了 Xftp 以后，windows 用户能安全地在 UNIX/Linux 和 Windows PC 之间传输文件。

2. 下载：<https://www.netsarang.com>

4. 安装和使用：参考“Xftp 安装手册.docx”

5 Linux 系统管理

5.1 vi 和 vim 的使用

5.1.1 vi 和 vim 简介：

vi 编辑器是 Linux 和 Unix 上最基本的文本编辑器，工作在字符模式下。由于不需要图形界面，vi 是效率很高的文本编辑器。尽管在 Linux 上也有很多图形界面的编辑器可用，但 vi 在系统和服务器管理中的功能是那些图形编辑器所无法比拟的。

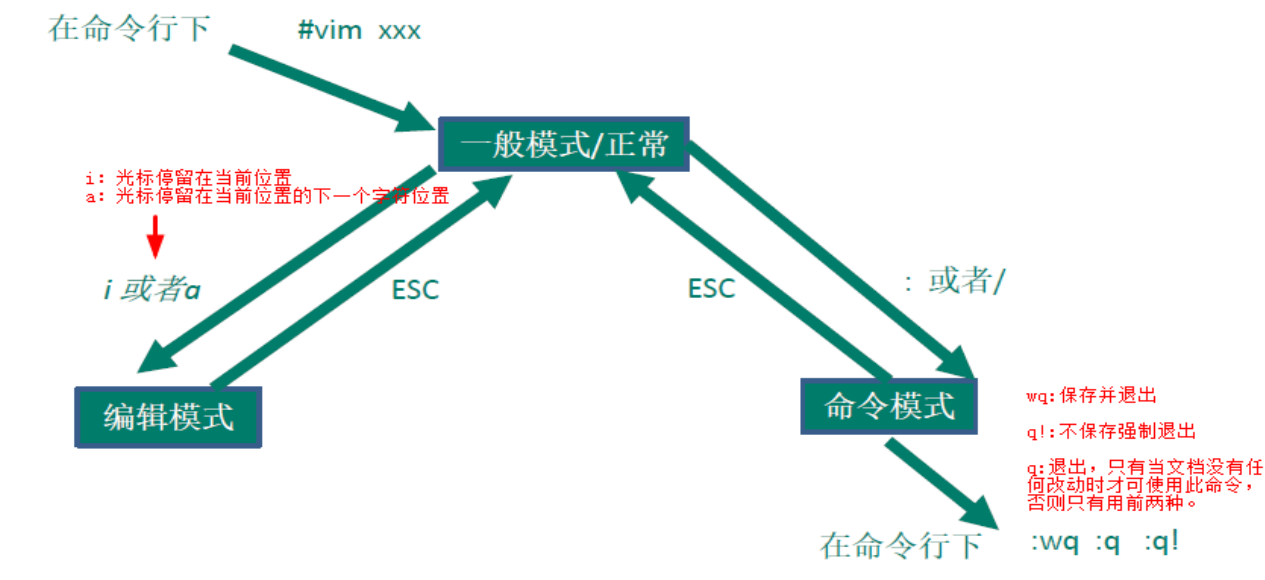
vim 是 vi 的增强版，比 vi 更容易使用。vi 的命令几乎全部都可以在 vim 上使用。

5.1.2 Vi 和 vim 三种常见模式：

一般模式：以 vi/vim 命令打开一个档案就直接进入一般模式了(这是默认的模式)。在这个模式中，你可以使用『上下左右』按键来移动光标，你可以使用『删除字符』或『删除整行』快捷键来处理档案内容，也可以使用『复制、贴上』快捷键来处理你的文件数据。

编辑模式：按下 i, l, o, O, a, A, r, R 等任何一个字母之后才会进入编辑模式，一般来说按 i 即可。

命令行模式：在这个模式当中，可以提供你相关指令，完成读取、存盘、替换、离开 vim 、显示行号等的动作则是在此模式中达成的！



5.1.3 Vi 和 vim 常用快捷键

- 一般模式拷贝当前行(yy)，拷贝当前行向下的 5 行(5yy)，并粘贴(p)。
- 一般模式删除当前行(dd)，删除当前行向下的 5 行(5dd)。
- 一般模式下，在文件中查找某个单词，[命令模式下：(/关键字)，回车查找，输入(n) 就是查找下一个]。
- 一般模式下，使用快捷键到达文档的最首行[gg]和最末行[G]。
- 一般模式下，在一个文件中输入"xxxx" ,然后又撤销这个动作(u)。
- 一般模式下，并将光标移动到 10 行 shift+g
 - 第一步：输入 10
 - 第二步：输入 shift+g
 - 第三步：回车
- 命令行模式下，设置文件的行号，取消文件的行号.[命令行下(: set nu) 和(:set nonu)]。
- 其它快捷键：



5.2 Linux 用户管理

5.2.1 Linux 用户简介:

Linux 系统是一个多用户多任务的操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。root 用户是系统默认创建的管理员账号。

5.2.2 添加用户:

`useradd [选项] 用户名`

`useradd zhangsan`: 创建一个账号叫 zhangsan，此时会创建账号、创建一个组 zhangsan 并且把 zhangsan 分到此组中、还会在 /home 下创建一个目录叫 zhangsan 作为新创建用户的根目录。

`useradd -d /home/lisi lisi`: 创建一个账号叫 lisi，并且给 lisi 指定家目录 /ls。

`passwd zhangsan`: 给 zhangsan 设置密码。

5.2.3 删除用户:

`userdel [选项] 用户名`

`userdel zhangsan`: 删除用户 zhangsan，保留 zhangsan 的主目录。

`userdel -r lisi`: 删除用户 lisi，并且把 lisi 的主目录也删除。

5.2.4 查询用户信息:

`id 用户名`

`id zhangsan`: 查看用户 zhangsan 的信息。

5.2.5 切换用户：

`su 用户名`

`su zhangsan`：切换到 `zhangsan` 用户。

注意：从高权限用户切换到低权限用户时，不需要输密码；否则，需要输密码。

另：`exit` 命令可以回到原来的用户。

5.3 Linux 组管理

5.3.1 Linux 的组简介：

Linux 的组类似于角色，系统可以对有共性的多个用户进行统一的管理。每一个用户都至少属于一个组，创建用户时如果不指定组，会默认创建一个跟用户名相同的组，并且把新创建的用户分配到组中，`root` 用户默认属于 `root` 组。

5.3.2 添加组：

`groupadd 组名`

`groupadd devgroup`：创建一个组 `devgroup`。

5.3.3 删除组：

`groupdel 组名`

`groupdel devgroup`：删除组 `devgroup`。

5.3.4 添加用户时指定组：

`useradd -g 组名 用户名`

`useradd -g devgroup zhangsan`：添加用户 `zhangsan`，并且指定 `zhangsan` 属于组 `devgroup`。

5.3.5 将用户添加到组/从组中移除：

`gpasswd -a/-d 用户名 组名`

`gpasswd -a zhangsan test`

`gpasswd -d zhangsan test`

5.4 Linux 的系统操作

`shutdown now`：立刻进行关机

`shutdown -h 1`：1 小时后会关机了

`shutdown -r now`：现在重新启动计算机

`reboot`：现在重新启动计算机

`sync`：把内存的数据同步到磁盘。

6 Linux 实操指令

6.1 帮助指令

在使用 Linux 过程中，如果遇到不熟悉的指令，一个是可以到网络上去查找，另一个是可以使用帮助指令。

6.1.1 帮助指令 man

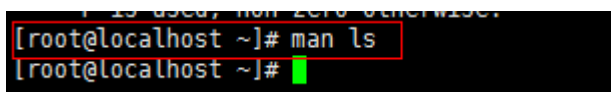
6.1.1.1 基本语法:

`man [命令或配置文件]`

功能描述：获得帮助手册上的信息

6.1.1.2 实操案例:

查看命令 `ls` 的用法:



```
[root@localhost ~]# man ls
[root@localhost ~]#
```

6.1.2 帮助指令 help

6.1.2.1 基本语法:

`help [命令]`

功能描述：获得命令内置的帮助信息

6.1.2.2 实操案例:

查看命令 `cd` 的用法:

```
[root@localhost ~]# help cd
cd: cd [-L|[-P [-e]]] [dir]
Change the shell working directory.

Change the current directory to DIR. The default DIR is the value of the
HOME shell variable.

The variable CDPATH defines the search path for the directory containing
DIR. Alternative directory names in CDPATH are separated by a colon (:).
A null directory name is the same as the current directory. If DIR begins
with a slash (/), then CDPATH is not used.

If the directory is not found, and the shell option `cdable_vars' is set,
the word is assumed to be a variable name. If that variable has a value,
its value is used for DIR.

Options:
-L      force symbolic links to be followed
-P      use the physical directory structure without following symbolic
links
-e      if the -P option is supplied, and the current working directory
cannot be determined successfully, exit with a non-zero status

The default is to follow symbolic links, as if `-L' were specified.

Exit Status:
Returns 0 if the directory is changed, and if $PWD is set successfully when
-P is used; non-zero otherwise.
```

6.2 文件目录指令

6.2.1 pwd 指令

6.2.1.1 基本语法:

pwd

用于显示当前目录的路径

6.2.1.2 实操案例:

```
[root@localhost ~]# pwd
/root
```

6.2.2 ls 指令

6.2.2.1 基本语法:

- ls [选项] [目录或是文件]
- 常用选项
- -a : 显示当前目录所有的文件和目录, 包括隐藏的。
- -l : 以列表的方式显示信息, 相当于 ll

6.2.2.2 实操案例:

列出当前目录下所有的文件和目录:


```

[root@localhost ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg
[root@localhost ~]# ls -l
total 8
-rw-----. 1 root root 1607 Nov 13 10:36 anaconda-ks.cfg
-rw-r--r--. 1 root root 1655 Nov 13 10:45 initial-setup-ks.cfg
[root@localhost ~]# ls -la
total 40
dr-xr-x---. 5 root root 221 Nov 14 20:54 .
dr-xr-xr-x. 17 root root 233 Nov 13 10:31 ..
-rw-----. 1 root root 1607 Nov 13 10:36 anaconda-ks.cfg
-rw-----. 1 root root 441 Nov 14 20:50 .bash_history
-rw-r--r--. 1 root root 18 Dec 29 2013 .bash_logout
-rw-r--r--. 1 root root 176 Dec 29 2013 .bash_profile
-rw-r--r--. 1 root root 176 Dec 29 2013 .bashrc
drwx-----. 4 root root 31 Nov 13 21:38 .cache
drwx-----. 4 root root 30 Nov 13 21:38 .config
-rw-r--r--. 1 root root 100 Dec 29 2013 .cshrc
drwx-----. 3 root root 25 Nov 13 10:40 .dbus
-rw-r--r--. 1 root root 1655 Nov 13 10:45 initial-setup-ks.cfg
-rw-r--r--. 1 root root 129 Dec 29 2013 .tcshrc
-rw-----. 1 root root 5176 Nov 14 20:54 .viminfo
[root@localhost ~]#

```

简单列表列出

长列表列出

长列表列出，包括.开头的文件

6.2.3 cd 指令

6.2.3.1 基本语法：

- cd [参数]
功能描述：切换到指定目录
- 常用参数
绝对路径(以/开头的目录)和相对路径(以目录名开头的目录，从当前目录下开始查找)
cd ~ 或者cd : 回到自己的主目录
cd .. 回到当前目录的上一级目录

6.2.3.2 实操案例：

```

[root@localhost home]# cd ~
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /home
[root@localhost home]# pwd
/home
[root@localhost home]# cd ../home/zhangsan
[root@localhost zhangsan]# pwd
/home/zhangsan
[root@localhost zhangsan]#

```

进入root目录

进入家目录，/home是绝对目录

进入zhangsan的家目录，../home/zhangsan是相对目录

6.2.4 mkdir 命令

6.2.4.1 基本语法：

- mkdir [选项] 要创建的目录
- 常用选项
-p : 创建多级目录

6.2.4.2 实操案例：

```
[root@localhost zhangsan]# mkdir test
[root@localhost zhangsan]# ls
test
[root@localhost zhangsan]# mkdir /home/zhangsan/test2
[root@localhost zhangsan]# ls
test test2
[root@localhost zhangsan]# mkdir /home/zhangsan/students/xiaozhang
mkdir: cannot create directory '/home/zhangsan/students/xiaozhang': No such file or directory
[root@localhost zhangsan]# mkdir -p /home/zhangsan/students/xiaozhang
[root@localhost zhangsan]# ls
students test test2
[root@localhost zhangsan]# cd students
[root@localhost students]# ls
xiaozhang
[root@localhost students]#
```

在当前目录下创建一个目录test

在/home/zhangsan目录下创建一个目录test2

在/home/zhangsan下创建多级目录/students/xiaozhang出错，因为mkdir只能创建单级目录。

在/home/zhangsan下创建多级目录成功，因为mkdir带上参数-p可以创建多级目录。

6.2.5 rmdir 指令

6.2.5.1 基本语法：

rmdir 目录

功能描述：删除一个空目录

6.2.5.2 实操案例：

```
[root@localhost ~]# rmdir /home/zhangsan/students/xiaozhang
[root@localhost ~]# ls /home/zhangsan/students
t1 t2
[root@localhost ~]# mkdir -p /home/zhangsan/test/t1/t2
[root@localhost ~]# ls /home/zhangsan/test/t1
t2
[root@localhost ~]# rmdir /home/zhangsan/test/t1
rmdir: failed to remove '/home/zhangsan/test/t1': Directory not empty
[root@localhost ~]# rm -rf /home/zhangsan/test/t1
[root@localhost ~]# ls /home/zhangsan/test
[root@localhost ~]#
```

删除目录/home/zhangsan/students/xiaozhang

删除目录/home/zhangsan/test/t1出错，因为rmdir只能删除一个空目录，而t1下还有其它目录，所以删除失败。

要删除一个非空目录，可以使用命令rm并且带参数-rf

6.2.6 touch 指令

6.2.6.1 基本语法：

touch 文件名称列表

功能描述：创建一个或多个空文件

6.2.6.2 实操案例：

```
[root@localhost zhangsan]# touch t1.txt
[root@localhost zhangsan]# ls
students t1.txt test test2
[root@localhost zhangsan]# touch t2.txt t3.txt t4.txt
[root@localhost zhangsan]# ls
students t1.txt t2.txt t3.txt t4.txt test test2
[root@localhost zhangsan]#
```

创建一个空文件t1.txt

创建多个空文件t2.txt, t3.txt, t4.txt

6.2.7 cp 指令

6.2.7.1 基本语法:

- `cp [选项] source dest`
- 常用选项:
- r : 递归复制整个文件夹

6.2.7.2 实操案例:

```
[root@localhost zhangsan]# ls
students test test2 t.txt
[root@localhost zhangsan]# cp t.txt test
[root@localhost zhangsan]# ls test
t.txt
[root@localhost zhangsan]# cp test2 students
cp: omitting directory 'test2'
[root@localhost zhangsan]# cp -r test2 students
[root@localhost zhangsan]# ls students
test2
[root@localhost zhangsan]# cp -r test students
[root@localhost zhangsan]# sp -r test students
bash: sp: command not found...
Similar commands are::
'ps'
'cp'
[root@localhost zhangsan]# cp -r test students
cp: overwrite 'students/test/t.txt'? y
[root@localhost zhangsan]# \cp -r test students
[root@localhost zhangsan]# ls students
test test2
[root@localhost zhangsan]# ls students/test
t.txt
[root@localhost zhangsan]#
```

准备当前目录zhangsan下三个空目录和一个文件

拷贝当前目录下的一个文件t.txt到当前目录下的test目录

拷贝目录到另一个目录出错，因为被拷贝的目录下可能有内容

拷贝当前目录下的一个目录到当前目录下的另一个目录，递归拷贝成功

拷贝文件到另一个目录时，如果有冲突的情况，每一个文件都要确认是否覆盖。

拷贝时，cp指令前带上/，在有冲突时强制覆盖，无需再确认。

6.2.8 rm 指令

6.2.8.1 基本语法:

- `rm [选项] 要删除的文件或目录`
- 常用选项:
- r : 递归删除整个文件夹
- f : 强制删除不提示

6.2.8.2 实操案例:

```
[root@localhost zhangsan]# ls
students t2.txt test test2 t.txt
[root@localhost zhangsan]# ls test
t.txt
[root@localhost zhangsan]# rm t.txt
rm: remove regular empty file 't.txt'? y
[root@localhost zhangsan]# rm -f t2.txt
[root@localhost zhangsan]# ls
students test test2
[root@localhost zhangsan]# rm test
rm: cannot remove 'test': Is a directory
[root@localhost zhangsan]# rm -r test
rm: descend into directory 'test'? n
[root@localhost zhangsan]# ls
students test test2
[root@localhost zhangsan]# rm -rf test
[root@localhost zhangsan]# ls
students test2
[root@localhost zhangsan]#
```

当前目录下准备目录和文件

删除当前目录下的文件t.txt，需要确认是否删除

删除当前目录下的文件，-f表示强制删除，无需确认

删除当前目录下的目录，必须带上-r表示递归删除

递归强制删除当前目录下的目录

6.2.9 mv 指令

6.2.9.1 基本语法:

mv oldNameFile newNameFile (功能描述: 重命名)

mv /temp/movefile /targetFolder (功能描述: 移动文件)

6.2.9.2 实操案例:

```
[root@localhost zhangsan]# ls
aaa.txt students test test2
[root@localhost zhangsan]# mv aaa.txt bbb.txt
[root@localhost zhangsan]# ls
bbb.txt students test test2
[root@localhost zhangsan]# mv test2 test3
[root@localhost zhangsan]# ls
bbb.txt students test test3
[root@localhost zhangsan]# mv bbb.txt test
[root@localhost zhangsan]# ls
students test test3
[root@localhost zhangsan]# ls test
bbb.txt
[root@localhost zhangsan]# mv test/bbb.txt test3
[root@localhost zhangsan]# ls
students test test3
[root@localhost zhangsan]# ls test3
bbb.txt
[root@localhost zhangsan]# mv test3 test
[root@localhost zhangsan]# ls
students test
[root@localhost zhangsan]# ls test
test3
[root@localhost zhangsan]# ls test/test3
bbb.txt
[root@localhost zhangsan]#
```

准备演示目录和文件

在同一目录下给文件aaa.txt重命名bbb.txt

在同一目录下给目录test2重命名test3

把当前目录下的文件bbb.txt移动到当前目录下的test目录中

把目录test下的文件bbb.txt移动到目录test3中

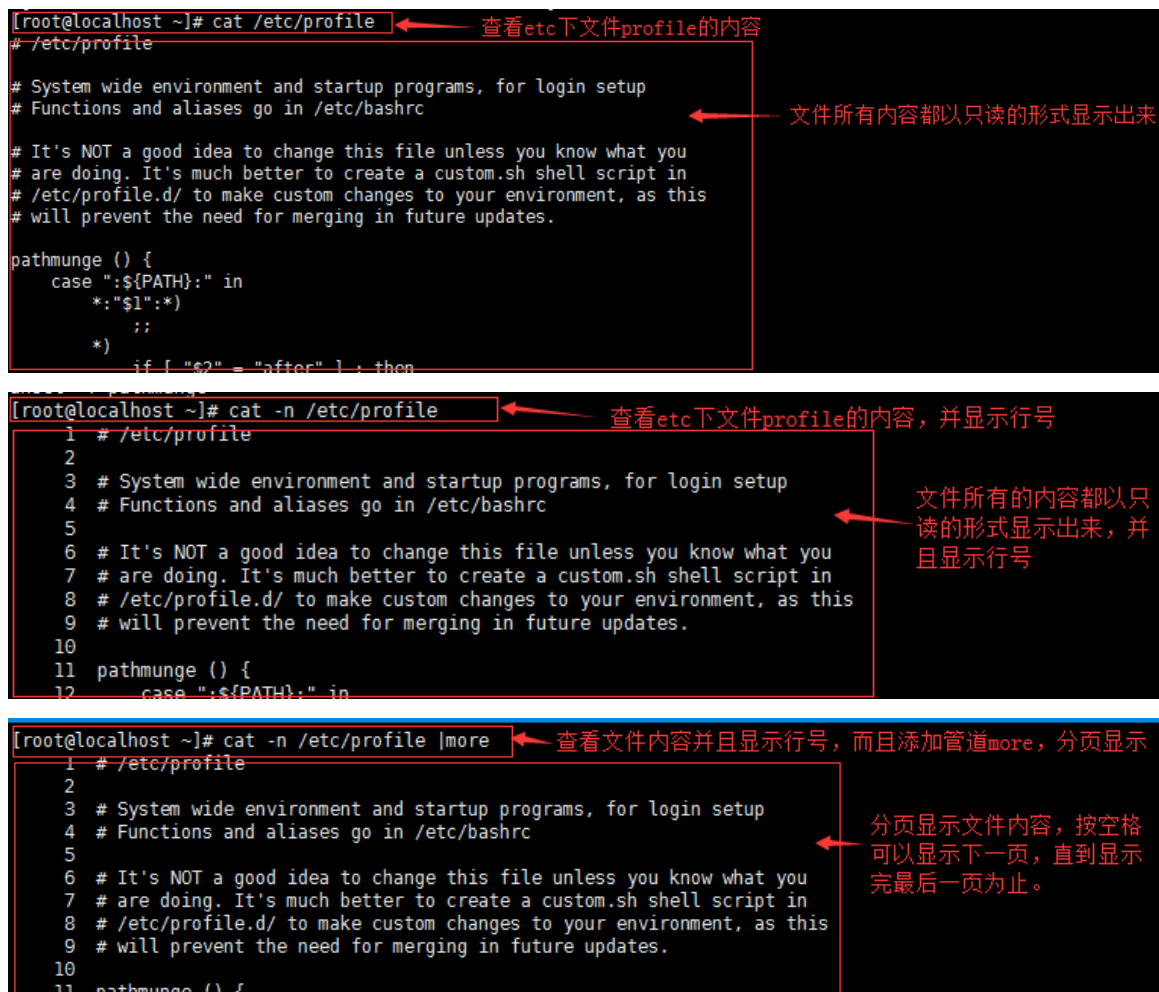
把目录test3整体移动到目录test中，连同test3中的所有内容

6.2.10 cat 指令

6.2.10.1 基本语法:

- cat [选项] 要查看的文件名
- 常用选项
 - n : 显示行号

6.2.10.2 实操案例：



```
[root@localhost ~]# cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case "${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
```

查看etc下文件profile的内容

文件所有内容都以只读的形式显示出来

```
[root@localhost ~]# cat -n /etc/profile
1 # /etc/profile
2
3 # System wide environment and startup programs, for login setup
4 # Functions and aliases go in /etc/bashrc
5
6 # It's NOT a good idea to change this file unless you know what you
7 # are doing. It's much better to create a custom.sh shell script in
8 # /etc/profile.d/ to make custom changes to your environment, as this
9 # will prevent the need for merging in future updates.
10
11 pathmunge () {
12     case "${PATH}:" in
```

查看etc下文件profile的内容，并显示行号

文件所有的内容都以只读的形式显示出来，并且显示行号

```
[root@localhost ~]# cat -n /etc/profile | more
1 # /etc/profile
2
3 # System wide environment and startup programs, for login setup
4 # Functions and aliases go in /etc/bashrc
5
6 # It's NOT a good idea to change this file unless you know what you
7 # are doing. It's much better to create a custom.sh shell script in
8 # /etc/profile.d/ to make custom changes to your environment, as this
9 # will prevent the need for merging in future updates.
10
11 pathmunge () {
```

查看文件内容并且显示行号，而且添加管道more，分页显示

分页显示文件内容，按空格可以显示下一页，直到显示完最后一页为止。

6.2.11more 指令

6.2.11.1 基本语法：

more 要查看的文件名

功能说明：more 指令是一个基于 VI 编辑器的文本过滤器，它以全屏幕的方式按页显示文本文件的内容。

6.2.11.2 快捷键:

操作	功能说明
空白键 (space)	代表向下翻一页;
Enter	代表向下翻「一行」;
q	代表立刻离开 more , 不再显示该文件内容。
Ctrl+F	向下滚动一屏
Ctrl+B	返回上一屏
=	输出当前行的行号
:f	输出文件名和当前行的行号

6.2.11.3 实操案例:

```
[root@localhost ~]# more /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
```

分页查看文件内容，使用空格可以查看下一页，使用回车可以逐行查看

以只读的形式逐页显示文件内容

6.2.12 less 指令

6.2.12.1 基本语法:

less 要查看的文件名

功能说明: less 指令用来分屏查看文件内容，它的功能与 more 指令类似，但是比 more 指令更加强大，支持各种显示终端。less 指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

6.2.12.2 快捷键:

操作	功能说明
空白键	向下翻动一页;
[pagedown]	向下翻动一页
[pageup]	向上翻动一页;
/字串	向下搜寻【字串】的功能; n: 向下查找; N: 向上查找;
?字串	向上搜寻【字串】的功能; n: 向上查找; N: 向下查找;
q	离开 less 这个程序;

6.2.12.3 实操案例:

```
[root@localhost ~]# less /etc/profile 查看/etc下文件profile的内容
[root@localhost ~]#
```

6.2.13 head 指令

6.2.13.1 基本语法:

head 文件(功能描述: 默认查看文件头10行内容)

head -n 5 文件(功能描述: 查看文件头 5 行内容, 5 可以是任意行数)

6.2.13.2 实操案例:

```
[root@localhost home]# head /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

[root@localhost home]# head -n 5 /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

[root@localhost home]#
```

查看文件profile的前10行。

查看文件profile的前5行

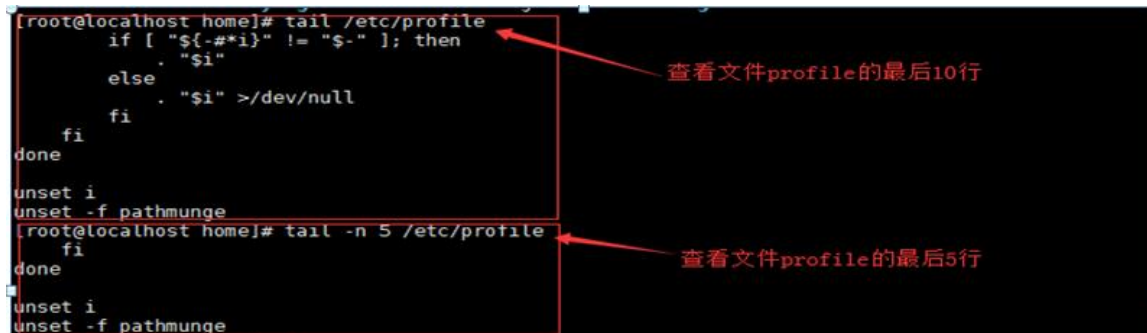
6.2.14tail 指令

6.2.14.1 基本语法:

tail 文件（功能描述：默认查看文件尾 10 行内容）

tail -n 5 文件（功能描述：查看文件尾 5 行内容，5 可以是任意行数）

6.2.14.2 实操案例:



```
[root@localhost home]# tail /etc/profile
if [ "${-#*i}" != "$-" ]; then
    . "$i"
else
    . "$i" >/dev/null
fi
done
unset i
unset -f pathmunge
[root@localhost home]# tail -n 5 /etc/profile
done
unset i
unset -f pathmunge
```

查看文件profile的最后10行

查看文件profile的最后5行

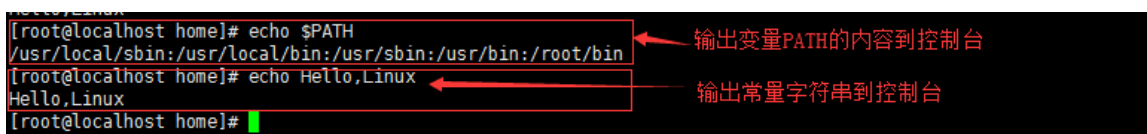
6.2.15echo 指令

6.2.15.1 基本语法:

echo [选项] [输出内容]

功能介绍：输出变量或常量内容到控制台

6.2.15.2 实操案例:



```
[root@localhost home]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@localhost home]# echo Hello, Linux
Hello, Linux
[root@localhost home]#
```

输出变量PATH的内容到控制台

输出常量字符串到控制台

6.2.16>指令

6.2.16.1 基本语法:

查看指令 > 目标文件

功能说明：将前一个查看指令中的结果覆盖写入到目标文件中，如果目标文件不存在，则新建。

6.2.16.2 实操案例:

```
[root@localhost home]# ll
total 4
drwx-----, 15 bruceyang bruceyang 4096 Nov 15 10:50 bruceyang
drwx-----, 3 lisi testgroup 78 Nov 14 15:17 lisi
drwx-----, 5 lisi devgroup 128 Nov 14 14:52 ls
drwx-----, 7 zhangsan zhangsan 156 Nov 15 13:20 zhangsan
[root@localhost home]# ll > aa.txt
[root@localhost home]# cat aa.txt
total 4
-rw-r--r--, 1 root root 0 Nov 15 18:03 aa.txt
drwx-----, 15 bruceyang bruceyang 4096 Nov 15 10:50 bruceyang
drwx-----, 3 lisi testgroup 78 Nov 14 15:17 lisi
drwx-----, 5 lisi devgroup 128 Nov 14 14:52 ls
drwx-----, 7 zhangsan zhangsan 156 Nov 15 13:20 zhangsan
[root@localhost home]# ls > aa.txt
[root@localhost home]# cat aa.txt
aa.txt
bruceyang
lisi
ls
zhangsan
[root@localhost home]# cat aa.txt > bb.txt
[root@localhost home]# cat bb.txt
aa.txt
bruceyang
lisi
ls
zhangsan
```

准备测试目录内容

将文件列表内容覆盖写到文件 aa.txt 中。

将ls文件内容覆盖写到文件 aa.txt 中。

将文件aa.txt中的内容覆盖写到文件bb.txt中。

6.2.17>>指令

6.2.17.1 基本语法:

查看指令 >> 目标文件

功能说明: 将前一个查看指令中的结果追加写入到目标文件中, 如果目标文件不存在, 则新建。

6.2.17.2 实操案例:

```
[root@localhost home]# cal
November 2018
Su Mo Tu We Th Fr Sa
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
[root@localhost home]# cal >> dd.txt
[root@localhost home]# ls
aa.txt bb.txt bruceyang dd.txt lisi ls zhangsan
[root@localhost home]# ls >> dd.txt
[root@localhost home]# cat dd.txt
November 2018
Su Mo Tu We Th Fr Sa
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

aa.txt
bb.txt
bruceyang
dd.txt
lisi
ls
zhangsan
```

把日历追加写到文件dd.txt中

把文件列表追加写到文件dd.txt中

```

[root@localhost home]# echo "Hello,Linux">>dd.txt
[root@localhost home]# cat dd.txt
November 2018
Su Mo Tu We Th Fr Sa
    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

aa.txt
bb.txt
bruceyang
dd.txt
lisi
ls
zhangsan
Hello,Linux

```

把字符串"Hello,Linux"追加写到文件dd.txt中

6.3 时间日期指令

6.3.1 date 指令

6.3.1.1 基本语法：

- 1) date (功能描述：显示当前时间)
- 2) date +%Y (功能描述：显示当前年份)
- 3) date +%m (功能描述：显示当前月份)
- 4) date +%d (功能描述：显示当前是哪一天)
- 5) date "+%Y-%m-%d %H:%M:%S" (功能描述：显示年月日时分秒)
- 6) date -s 字符串时间 (功能描述：设置日期)

6.3.1.2 实操案例：

```

[roo history
[root@localhost ~]# date
Thu Nov 15 21:45:07 CST 2018
[root@localhost ~]# date +%Y
2018
[root@localhost ~]# date +%m
11
[root@localhost ~]# date +%d
15
[root@localhost ~]# date "+%Y-%m-%d %H:%M:%S"
2018-11-15 21:46:06
[root@localhost ~]# date -s '2018-11-16 21:46:30'

```

← 显示当前时间

← 显示当前年份

← 显示当前月份

← 显示当前哪一天

← 按指定格式显示当前时间

6.3.2 cal 指令

6.3.2.1 基本语法：

cal [选项] (功能描述：不加选项，显示本月日历)

6.3.2.2 实操案例:

```
[root@localhost ~]# cal
November 2018
Su Mo Tu We Th Fr Sa
  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
```

显示当前月的日历

```
[root@localhost ~]# cal 2019
2019
January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5          1  2          3  4  5  6  7  8  9
 6  7  8  9 10 11 12    3  4  5  6  7  8  9    10 11 12 13 14 15 16
13 14 15 16 17 18 19    10 11 12 13 14 15 16    17 18 19 20 21 22 23
20 21 22 23 24 25 26    17 18 19 20 21 22 23    24 25 26 27 28 29 30
27 28 29 30 31         24 25 26 27 28         31
April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6      1  2  3  4          2  3  4  5  6  7  8
 7  8  9 10 11 12 13    5  6  7  8  9 10 11    9 10 11 12 13 14 15
14 15 16 17 18 19 20    12 13 14 15 16 17 18    16 17 18 19 20 21 22
```

显示指定年份的日历

6.4 搜索查找指令

6.4.1 find 指令

find 指令将从指定目录向下递归地遍历其各个子目录，将满足条件的文件或者目录显示在终端。

6.4.1.1 基本语法:

find [搜索范围] [选项]

选项说明:

- name:按名称查找，支持通配符。
- user:按用户名查找
- size:按文件大小查找

6.4.1.2 实操案例:

```
[root@bogon test]# find -name aa.txt
./aa.txt
[root@bogon test]# find -name '*.txt'
./t1/tt2/my.txt
./test.txt
./aa.txt
[root@bogon test]#
```

```
[root@localhost ~]# find /home -user lisi
/home/ls
/home/ls/.mozilla
/home/ls/.mozilla/extensions
/home/ls/.mozilla/plugins
/home/ls/.bash_logout
/home/ls/.bash_profile
/home/ls/.bashrc
```

查找home目录下属于lisi的所有文件

```
[root@localhost ~]# find /home -size +5M
/home/bruceyang/.mozilla/firefox/hz9efs06.default/places.sqlite
[root@localhost ~]# find /home -size -1k
/home/bruceyang/.mozilla/firefox/hz9efs06.default/.parentlock
/home/bruceyang/.mozilla/firefox/hz9efs06.default/sessionstore.js
/home/bruceyang/.mozilla/firefox/hz9efs06.default/healthreport.sqlite-wal
/home/bruceyang/.cache/gdm/session.log.old
/home/bruceyang/.cache/gdm/session.log
/home/bruceyang/.cache/abrt/applet_dirlist
/home/bruceyang/.cache/gnome-shell/update-check-3.14
/home/bruceyang/.local/share/gnome-settings-daemon/input-sources-converted
/home/bruceyang/.local/share/tracker/data/.meta.isrunning
/home/bruceyang/.local/share/.converted-launchers
/home/zhangsan/students/test/t.txt
/home/zhangsan/test/test3/hbb.txt
[root@localhost ~]# find /home -size 5M
[root@localhost ~]#
```

查找home目录下所有大于5M的文件

查找home目录下所有小于1k的文件

查找home目录下所有大小为5M的文件

6.4.2 locate 指令

6.4.2.1 基本语法:

locate 要搜索的文件名

说明: locate 指令可以快速在整个系统中定位文件路径。locate 指令利用事先建立的系统中所有文件名称及路径的 locate 数据库实现快速定位给定的文件。locate 指令无需遍历整个文件系统, 查询速度较快。为了保证查询结果的准确度, 管理员必须定期更新 locate 时刻。

6.4.2.2 实操案例:

```
[root@localhost ~]# updatedb
[root@localhost ~]# locate bb.txt
/home/bb.txt
/home/zhangsan/test/test3/bbb.txt
[root@localhost ~]#
```

更新locate数据库

查找文件名含有bb.txt的所有文件所在位置

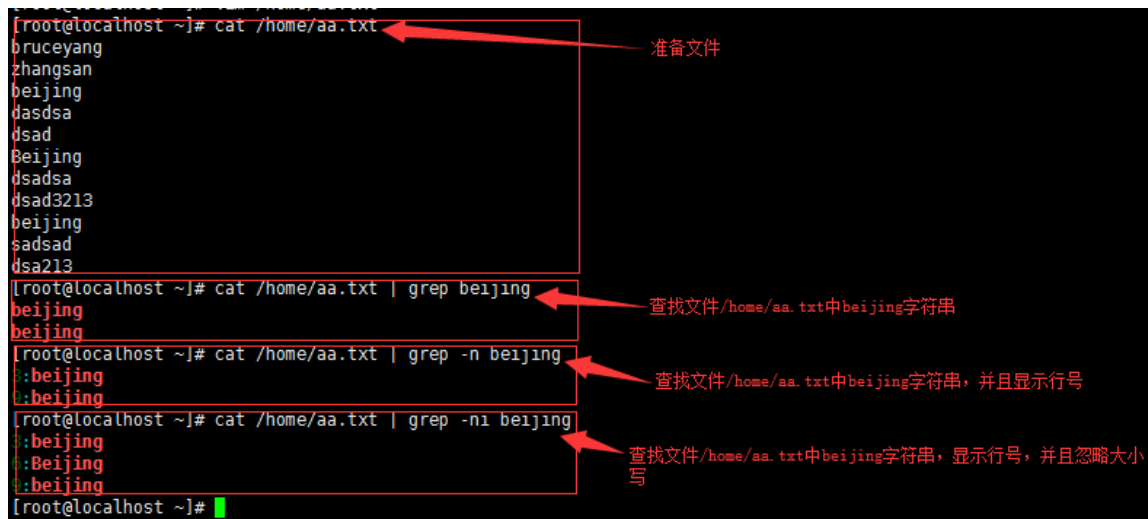
6.4.3 grep 指令

6.4.3.1 基本语法:

- grep [选项] 查找的源文件内容
- 常用选项:
 - n: 显示匹配行和行号
 - i: 忽略大小写

说明: grep 过滤查找, 管道符, “|”, 表示将前一个命令的处理结果输出传递给后面的命令处理。

6.4.3.2 实操案例:



```
[root@localhost ~]# cat /home/aa.txt  
bruceyang  
zhangsan  
beijing  
dasdsa  
dsad  
Beijing  
dsadsa  
dsad3213  
beijing  
sadsad  
dsa213  
[root@localhost ~]# cat /home/aa.txt | grep beijing  
beijing  
beijing  
[root@localhost ~]# cat /home/aa.txt | grep -n beijing  
:beijing  
:beijing  
[root@localhost ~]# cat /home/aa.txt | grep -ni beijing  
:beijing  
:Beijing  
:beijing  
[root@localhost ~]#
```

准备文件

查找文件/home/aa.txt中beijing字符串

查找文件/home/aa.txt中beijing字符串, 并且显示行号

查找文件/home/aa.txt中beijing字符串, 显示行号, 并且忽略大小写

6.5 压缩和解压缩指令

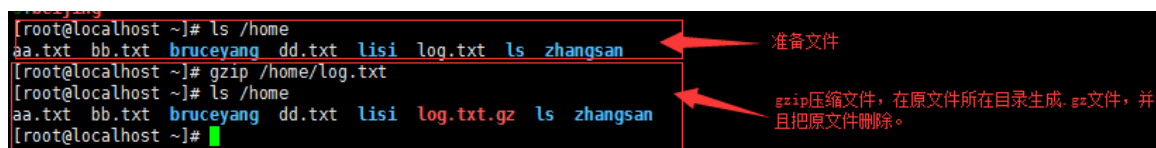
6.5.1 gzip 指令

6.5.1.1 基本语法:

gzip 文件

功能说明: 压缩文件, 将文件压缩为*.gz 文件存放在原文件所在目录, 压缩成功后会把原文件删除。用于压缩单个文件。

6.5.1.2 实操案例:



```
[root@localhost ~]# ls /home  
aa.txt bb.txt bruceyang dd.txt lisi log.txt ls zhangsan  
[root@localhost ~]# gzip /home/log.txt  
[root@localhost ~]# ls /home  
aa.txt bb.txt bruceyang dd.txt lisi log.txt.gz ls zhangsan  
[root@localhost ~]#
```

准备文件

gzip压缩文件, 在原文件所在目录生成.gz文件, 并且把原文件删除。

6.5.2 gunzip 指令

6.5.2.1 基本语法:

gunzip 文件

功能说明: 解压缩文件命令, 解压成功后存放在原压缩文件所在目录, 并且把原压缩文件删除。

6.5.2.2 实操案例:

```
[root@localhost ~]# ls /home
aa.txt bb.txt bruceyang dd.txt lisi log.txt.gz ls zhangsan
[root@localhost ~]# gunzip /home/log.txt.gz
[root@localhost ~]# ls /home
aa.txt bb.txt bruceyang dd.txt lisi log.txt ls zhangsan
[root@localhost ~]#
```

准备文件

解压文件，在原来压缩文件所在目录生成解压后的文件，并删除原压缩文件。

6.5.3 zip 指令

6.5.3.1 基本语法:

- zip [选项] XXX.zip 将要压缩的内容
功能描述: 将是指定文件或目录压缩成 XXX.zip 文件, 用于压缩所有文件结构。
- 选项说明:
 - r: 递归压缩, 即压缩目录

6.5.3.2 实操案例:

```
[root@localhost ~]# ls /home
aa.txt bb.txt bruceyang dd.txt lisi log.txt ls zhangsan
[root@localhost ~]# zip -r myhome.zip /home
zip warning: name not matched: /home/bruceyang/.mozilla/firefox/hz9efs06.default/lock
adding: home/ (stored 0%)
adding: home/bruceyang/ (stored 0%)
adding: home/bruceyang/.mozilla/ (stored 0%)
adding: home/bruceyang/.mozilla/extensions/ (stored 0%)
adding: home/bruceyang/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}/ (stored 0%)
adding: home/bruceyang/.mozilla/plugins/ (stored 0%)
adding: home/bruceyang/.mozilla/firefox/ (stored 0%)
adding: home/bruceyang/.mozilla/firefox/hz9efs06.default/ (stored 0%)
adding: home/bruceyang/.mozilla/firefox/hz9efs06.default/times.json (stored 0%)
adding: home/bruceyang/.mozilla/firefox/hz9efs06.default/.parentlock (stored 0%)
```

准备文件

递归压缩/home下所有文件

6.5.4 unzip 指令

6.5.4.1 基本语法:

- unzip [选项] XXX.zip
常用选项:
- d 目录: 指定解压后文件的存放目录

6.5.4.2 实操案例:

```
[root@localhost ~]# ls
anaconda-ks.cfg initial-setup-ks.cfg myhome.zip
[root@localhost ~]# unzip myhome.zip -d /home/lisi
Archive: myhome.zip
replace /home/lisi/home/bruceyang/.mozilla/firefox/hz9efs06.default/times.json? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
extracting: /home/lisi/home/bruceyang/.mozilla/firefox/hz9efs06.default/times.json
replace /home/lisi/home/bruceyang/.mozilla/firefox/hz9efs06.default/.parentlock? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
```

把当前目录下myhome.zip解压到/home/lisi目录下

6.5.5 tar 指令

6.5.5.1 基本语法:

- `tar [选项] XXX.tar.gz [打包的内容]`
功能描述: 打包或者解压文件
- 选项说明:
 - c: 产生.tar.gz 打包文件
 - v: 显示详细信息
 - f: 指定压缩后的文件名
 - z: 打包同时压缩
 - x: 解压.tar.gz 文件
 - C: 指定解压到哪个目录

6.5.5.2 实操案例:

```
[root@localhost home]# ls
aa.txt bb.txt bruceyang dd.txt lisi log.txt ls zhangsan
[root@localhost home]# tar -zcvf ab.tar.gz aa.txt bb.txt
aa.txt
bb.txt
[root@localhost home]# ls
aa.txt ab.tar.gz bb.txt bruceyang dd.txt lisi log.txt ls zhangsan
[root@localhost home]# tar -zcvf zhangsan.tar.gz zhangsan
zhangsan/
zhangsan/.mozilla/
zhangsan/.mozilla/extensions/
zhangsan/.mozilla/plugins/
zhangsan/.bash_logout
zhangsan/.bash_profile
zhangsan/.bashrc
zhangsan/.cache/
zhangsan/.cache/abrt/
zhangsan/.cache/abrt/lastnotification
zhangsan/.config/
zhangsan/.config/abrt/
zhangsan/.bash_history
zhangsan/students/
zhangsan/students/test2/
zhangsan/students/test/
zhangsan/students/test/t.txt
zhangsan/test/
zhangsan/test/test3/
zhangsan/test/test3/bbb.txt
[root@localhost home]# ls
aa.txt ab.tar.gz bb.txt bruceyang dd.txt lisi log.txt ls zhangsan zhangsan.tar.gz
[root@localhost home]#
```

准备文件

把当前目录下的aa.txt和bb.txt文件打包成ab.tar.gz文件

把目录zhangsan下所有的内容打包成zhangsan.tar.gz文件

```
[root@localhost home]# ls
ab.tar.gz bruceyang dd.txt lisi log.txt ls zhangsan zhangsan.tar.gz
[root@localhost home]# tar -zxvf ab.tar.gz
aa.txt
bb.txt
[root@localhost home]# ls
aa.txt ab.tar.gz bb.txt bruceyang dd.txt lisi log.txt ls zhangsan zhangsan.tar.gz
[root@localhost home]# tar -zxvf zhangsan.tar.gz -C ls
zhangsan/
zhangsan/.mozilla/
zhangsan/.mozilla/extensions/
zhangsan/.mozilla/plugins/
zhangsan/.bash_logout
zhangsan/.bash_profile
zhangsan/.bashrc
zhangsan/.cache/
zhangsan/.cache/abrt/
zhangsan/.cache/abrt/lastnotification
zhangsan/.config/
zhangsan/.config/abrt/
zhangsan/.bash_history
zhangsan/students/
zhangsan/students/test2/
zhangsan/students/test/
zhangsan/students/test/t.txt
zhangsan/test/
zhangsan/test/test3/
zhangsan/test/test3/bbb.txt
[root@localhost home]# ls ls
zhangsan
[root@localhost home]#
```

准备文件

把文件ab.tar.gz解压到当前目录

把zhangsan.tar.gz解压到当前目录下的ls目录中

6.6 组管理指令

6.6.1 Linux 组的基本介绍:

在linux中的每个用户必须属于一个组，不能独立于组外，可以改变用户所属组。
在linux中每个文件有所有者、所在的组、其它组，也可以改变文件所在组。

6.6.2 文件/目录的所有者

一般为文件的创建者,谁创建了该文件，就自然的成为该文件的所有者，默认情况下所有者所在的组也即使文件所在的组。

6.6.2.1 查看文件所有者和所在组指令

6.6.2.1.1 基本语法:

ls -ahl 文件名
(a-all,h-human,l-list)

6.6.2.1.2 实操案例:

```
[root@localhost home]# ls -ahl
total 28K
drwxr-xr-x. 6 root root 158 Nov 16 20:13 .
dr-xr-xr-x. 17 root root 233 Nov 13 10:31 ..
-rw-r--r--. 1 root root 85 Nov 16 18:34 aa.txt
-rw-r--r--. 1 root root 209 Nov 16 20:08 ab.tar.gz
-rw-r--r--. 1 root root 34 Nov 15 18:04 bb.txt
drwx----- 15 bruceyang bruceyang 4.0K Nov 15 18:41 bruceyang
-rw-r--r--. 1 root root 205 Nov 15 18:16 dd.txt
drwx----- 4 lisi testgroup 90 Nov 16 19:55 lisi
-rw-r--r--. 1 root root 145 Nov 15 18:42 log.txt
drwx----- 6 lisi devgroup 144 Nov 16 20:13 ls
drwx----- 7 zhangsan zhangsan 156 Nov 15 13:20 zhangsan
-rw-r--r--. 1 root root 899 Nov 16 20:08 zhangsan.tar.gz
```

查看文件所有者
文件所有者，一般为文件的创建者
文件所在组，一般为文件所有者所在的组，也可以改。
文件名

6.6.2.2 修改文件所有者指令

6.6.2.2.1 基本语法:

chown 新所有者 文件名
chown newowner:newgroup file 同时修改用户的所有者和所有组
-R 如果是目录则使其下所有子文件或目录递归生效
功能说明：把文件所有者修改为指定的用户。

6.6.2.2.2 实操案例:

```
[root@localhost home]# ls -ahl
total 28K
drwxr-xr-x. 6 root    root      158 Nov 16 20:13 .
dr-xr-xr-x. 17 root    root      233 Nov 13 10:31 ..
-rw-r--r--. 1 root    root      85 Nov 16 18:34 aa.txt
-rw-r--r--. 1 root    root     209 Nov 16 20:08 ab.tar.gz
-rw-r--r--. 1 root    root      34 Nov 15 18:04 bb.txt
drwx----- 15 bruceyang bruceyang 4.0K Nov 15 18:41 bruceyang
-rw-r--r--. 1 root    root     205 Nov 15 18:16 dd.txt
drwx----- 4 lisi     testgroup 90 Nov 16 19:55 lisi
-rw-r--r--. 1 root    root     145 Nov 15 18:42 log.txt
drwx----- 6 lisi     devgroup  144 Nov 16 20:13 ls
drwx----- 7 zhangsan zhangsan  156 Nov 15 13:20 zhangsan
-rw-r--r--. 1 root    root     899 Nov 16 20:08 zhangsan.tar.gz
[root@localhost home]# chown zhangsan aa.txt
[root@localhost home]# ls -ahl
total 28K
drwxr-xr-x. 6 root    root      158 Nov 16 20:13 .
dr-xr-xr-x. 17 root    root      233 Nov 13 10:31 ..
-rw-r--r--. 1 zhangsan root      85 Nov 16 18:34 aa.txt
-rw-r--r--. 1 root    root     209 Nov 16 20:08 ab.tar.gz
-rw-r--r--. 1 root    root      34 Nov 15 18:04 bb.txt
drwx----- 15 bruceyang bruceyang 4.0K Nov 15 18:41 bruceyang
-rw-r--r--. 1 root    root     205 Nov 15 18:16 dd.txt
drwx----- 4 lisi     testgroup 90 Nov 16 19:55 lisi
-rw-r--r--. 1 root    root     145 Nov 15 18:42 log.txt
drwx----- 6 lisi     devgroup  144 Nov 16 20:13 ls
drwx----- 7 zhangsan zhangsan  156 Nov 15 13:20 zhangsan
-rw-r--r--. 1 root    root     899 Nov 16 20:08 zhangsan.tar.gz
[root@localhost home]#
```

准备文件

把文件aa.txt所有者修改为zhangsan

6.6.2.3 修改文件所在组指令

6.6.2.3.1 基本语法:

chgrp 新组名 文件名

-R 如果是目录则使其下所有子文件或目录递归生效

功能描述: 把文件所在组改为指定组

6.6.2.3.2 实操案例:

```
[root@localhost home]# touch hello.txt
[root@localhost home]# ls -ahl
total 24K
drwxr-xr-x. 4 root    root      130 Nov 18 17:12 .
dr-xr-xr-x. 17 root    root      233 Nov 13 10:31 ..
-rw-r--r--. 1 zhangsan root      85 Nov 16 18:34 aa.txt
-rw-r--r--. 1 root    root     209 Nov 16 20:08 ab.tar.gz
-rw-r--r--. 1 root    root      34 Nov 15 18:04 bb.txt
drwx----- 15 bruceyang bruceyang 4.0K Nov 15 18:41 bruceyang
-rw-r--r--. 1 root    root     205 Nov 15 18:16 dd.txt
-rw-r--r--. 1 root    root       0 Nov 18 17:12 hello.txt
-rw-r--r--. 1 root    root     145 Nov 15 18:42 log.txt
drwx----- 3 zhangsan devgroup  78 Nov 18 16:54 zhangsan
[root@localhost home]# chgrp devgroup hello.txt
[root@localhost home]# ls -ahl
total 24K
drwxr-xr-x. 4 root    root      130 Nov 18 17:12 .
dr-xr-xr-x. 17 root    root      233 Nov 13 10:31 ..
-rw-r--r--. 1 zhangsan root      85 Nov 16 18:34 aa.txt
-rw-r--r--. 1 root    root     209 Nov 16 20:08 ab.tar.gz
-rw-r--r--. 1 root    root      34 Nov 15 18:04 bb.txt
drwx----- 15 bruceyang bruceyang 4.0K Nov 15 18:41 bruceyang
-rw-r--r--. 1 root    root     205 Nov 15 18:16 dd.txt
-rw-r--r--. 1 root    devgroup  0 Nov 18 17:12 hello.txt
-rw-r--r--. 1 root    root     145 Nov 15 18:42 log.txt
drwx----- 3 zhangsan devgroup  78 Nov 18 16:54 zhangsan
[root@localhost home]#
```

root用户创建文件hello.txt

默认情况下,文件的所有者就是创建者,文件所在组就是创建者所在组

修改文件hello.txt所在组为devgroup

6.7 权限管理指令

6.7.1 权限基本介绍

这里所说的权限都是文件和目录的权限。在 Linux 中，每一个文件和目录都有自己的访问权限，通过文件列表可以查看到。

```
[root@localhost zhangsan]# ls -ld /etc/passwd
lrwxrwxrwx. 1 root root 10 Nov 15 18:16 dd.txt
```

文件类型：-：普通文件，d：目录，l：软链接，c：字符设备（键盘、鼠标等），b：块文件（硬盘等）。

文件所有者拥有的权限，此处为拥有读和写的权限

文件所在组的用户所拥有的权限，此处为只拥有读的权限

其它组的用户所拥有的权限，此处为只拥有读的权限

如果是文件，表示硬链接数，都是1；如果是死目录，表示该目录下子目录的数量。

文件所有者名称

文件所在组名称

如果是文件，表示文件的大小(以字节为单位)；如果是目录，则为4096.

文件最后修改时间

文件名称

6.7.2 RWX 权限详解

rwX 作用到文件

- 1) [r]代表可读(read): 可以读取, 查看
- 2) [w]代表可写(write): 可以修改, 但是不代表可以删除该文件, 删除一个文件的前提条件是对该文件所在录有写权限, 才能删除该文件.
- 3) [x]代表可执行(execute): 可以被执行

rwX 作用到目录

- 1) [r]代表可读(read): 可以读取, ls 查看目录内容
- 2) [w]代表可写(write): 可以修改, 目录内创建+删除+重命名目录
- 3) [x]代表可执行(execute): 可以进入该目录

rwX 用数字表示

$$r=4 \text{ (即 } 2^2 \text{)}, w=2 \text{ (即 } 2^1 \text{)}, x=1 \text{ (即 } 2^0 \text{)}$$

6.7.3 权限管理指令

6.7.3.1 修改文件/目录权限的指令 **chmod**

第一种方式：通过 r、w、x 变更变更权限

- 1) `chmod u=rwx,g=rx,o=x` 文件目录名
- 2) `chmod o+w` 文件目录名
- 3) `chmod a-x` 文件目录名

说明: **u、g、o、a** 分别代表文件所有者、文件所在组用户、其它组用户、所有用户
=、+、- 分别代表设置权限、增加权限、去掉权限

第二种方式：通过数字变更权限

chmod 一组三个数字 文件目录名

说明: $r=4$ $w=2$ $x=1$ $rw+x=4+2+1=7$

6.7.3.2 实操案例：



The following terminal sessions demonstrate the step-by-step modification of file permissions for `dd.txt`:

```
[root@localhost zhangsan]# ls -l
total 4
-rw-r--r--. 1 root root 205 Nov 15 18:16 dd.txt
```

← 准备文件，注意观察此文件的权限

```
[root@localhost zhangsan]# chmod u=rwx,g=rx,o=rx dd.txt
[root@localhost zhangsan]# ls -l
total 4
-rwxr-xr-x. 1 root root 205 Nov 15 18:16 dd.txt
```

← 修改文件dd.txt的权限为：文件所有者拥有读、写和执行权限，文件所在组用户拥有读和执行权限，其它组用户拥有读和执行权限。

```
[root@localhost zhangsan]# chmod u-w,g+w dd.txt
[root@localhost zhangsan]# ls -l
total 4
-r-xrwxr-x. 1 root root 205 Nov 15 18:16 dd.txt
```

← 给文件所有者去掉写权限，给文件所在组用户增加写权限

```
[root@localhost zhangsan]# chmod a+w dd.txt
[root@localhost zhangsan]# ls -l
total 4
-rwxrwxrwx. 1 root root 205 Nov 15 18:16 dd.txt
```

← 给所有用户增加写权限

```
[root@localhost zhangsan]# chmod 751 dd.txt
[root@localhost zhangsan]# ls -l
total 4
-rwxr-x--x. 1 root root 205 Nov 15 18:16 dd.txt
```

← 修改文件dd.txt的权限为rwxr-x--x，
rwx=7
r-x=5
--x=1

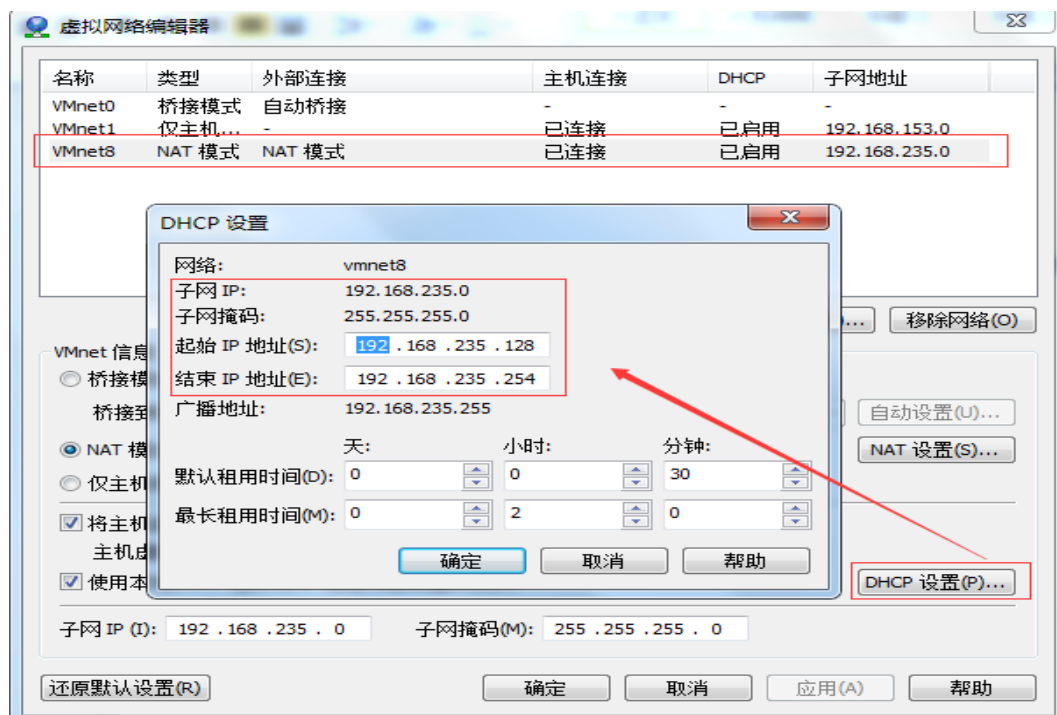
7 网络配置

7.1 查看和配置网络

7.1.1 查看虚拟网络编辑器



7.1.2 查看 IP 和网关



7.1.3 查看 windows 下虚拟网卡 VMnet8 的 IP 地址: cmd 下 ipconfig

```
以太网适配器 VMware Network Adapter VMnet8:

   连接特定的 DNS 后缀 . . . . . : 
   本地链接 IPv6 地址. . . . . : fe80::94c9:d904:35cb:d478%22
   IPv4 地址 . . . . . : 192.168.235.1
   子网掩码 . . . . . : 255.255.255.0
   默认网关. . . . . :
```

7.2 配置 Linux 网络

修改文件: /etc/sysconfig/network-scripts/ifcfg-eth33

```
TYPE="Ethernet"
BOOTPROTO="static"
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="41a34689-acc5-40c7-bd30-06de28ab1a95"
DEVICE="ens33"
ONBOOT="yes"

IPADDR=192.168.235.130
GATEWAY=192.168.235.2
DNS1=192.168.235.2
```

重启网络服务(service network restart)或者重启 linux(reboot)即可。

Linux 终端查看配置后的 IP 地址:

```
[bruceyang@localhost ~]$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.235.130 netmask 255.255.255.0 broadcast 192.168.235.255
    inet6 fe80::b2bb:e635:b8ae:fcc7 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:5f:91:e5 txqueuelen 1000 (Ethernet)
    RX packets 2355 bytes 170577 (166.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 730 bytes 94463 (92.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Linux 终端 ping 外网:

```
[bruceyang@localhost ~]$ ping www.baidu.com
PING www.a.shifen.com (119.75.217.26) 56(84) bytes of data.
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=1 ttl=128 time=12.4 ms
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=2 ttl=128 time=17.0 ms
```

Linux 终端 ping windows:

```
[bruceyang@localhost ~]$ ping 192.168.31.87
PING 192.168.31.87 (192.168.31.87) 56(84) bytes of data.
64 bytes from 192.168.31.87: icmp_seq=1 ttl=128 time=1.06 ms
64 bytes from 192.168.31.87: icmp_seq=2 ttl=128 time=0.964 ms
```

windows ping Linux:



```
C:\Users\Administrator>ping 192.168.235.130

正在 Ping 192.168.235.130 具有 32 字节的数据:
来自 192.168.235.130 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.235.130 的回复: 字节=32 时间<1ms TTL=64
```

都是通的。至此，Linux 静态 IP 设置完成。

8 进程管理

8.1 进程介绍

在Linux中，每个执行的程序（代码）都称为一个进程。每一个进程都分配一个ID号。每一个进程，都会对应一个父进程。

进程有两种运行方式：前台和后台。前台方式是用户可以在前台操作的，后台方式是实际在运行，但用户在前台看不见。

一般系统的服务都是以后台进程的方式存在，而且都会常驻在系统中。直到关机才结束。

8.2 查看系统运行的进程

8.2.1 基本语法:

ps 参数

参数说明:

ps -a:显示当前终端下的所有进程信息

ps -u:以用户的格式显示进程信息

ps -x:显示后台进程运行的参数

ps -e:显示所有进程信息

ps -f:以全格式显示进程信息

结果说明:

•USER: 用户名

•PID: 进程号

•%CPU: 进程占用CPU的百分比

•%MEM: 进程占用物理内存的百分比

•VSZ: 进程占用的虚拟内存大小 (单位: KB)

•RSS: 进程占用的物理内存大小 (单位: KB)

•TT: 终端名称,缩写.

•STAT: 进程状态, 其中S-睡眠, s-表示该进程是会话的先导进程, N-表示进程拥有比普通优先级更低的优先级, R-正在运行, D-短期等待, Z-僵死进程, T-被跟踪或者被停止等等

•STARTED: 进程的启动时间

•TIME: CPU时间, 即进程使用CPU的总时间

•COMMAND: 启动进程所用的命令和参数, 如果过长会被截断显示

•PPID 父进程的ID

•C CPU使用的资源百分比

•PRI指进程的执行优先权(Priority的简写), 其值越小越早被执行

•SZ 使用掉的内存大小

8.2.2 实操案例:

```
[root@localhost ~]# ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.3	128092	6716	?	Ss	15:00	0:09	/usr/lib/systemd/systemd --switched-root --system --deserialize 21
root	2	0.0	0.0	0	0	?	S	15:00	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	15:00	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	15:00	0:00	[kworker/0:0H]
root	7	0.0	0.0	0	0	?	S	15:00	0:01	[migration/0]
root	8	0.0	0.0	0	0	?	S	15:00	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	15:00	0:03	[rcu_sched]
root	10	0.0	0.0	0	0	?	S	15:00	0:01	[watchdog/0]
root	11	0.0	0.0	0	0	?	S	15:00	0:00	[watchdog/1]
root	12	0.0	0.0	0	0	?	S	15:00	0:01	[migration/1]
root	13	0.0	0.0	0	0	?	S	15:00	0:00	[ksoftirqd/1]
root	15	0.0	0.0	0	0	?	S<	15:00	0:00	[kworker/1:0H]
root	16	0.0	0.0	0	0	?	S	15:00	0:01	[watchdog/2]
root	18	0.0	0.0	0	0	?	S	15:00	0:00	[migration/2]
root	18	0.0	0.0	0	0	?	S	15:00	0:00	[ksoftirqd/2]
root	20	0.0	0.0	0	0	?	S<	15:00	0:00	[kworker/2:0H]
root	21	0.0	0.0	0	0	?	S	15:00	0:03	[watchdog/3]
root	22	0.0	0.0	0	0	?	S	15:00	0:00	[migration/3]
root	23	0.0	0.0	0	0	?	S	15:00	0:00	[ksoftirqd/3]
root	25	0.0	0.0	0	0	?	S<	15:00	0:00	[kworker/3:0H]
root	27	0.0	0.0	0	0	?	S<	15:00	0:00	[khelper]
root	28	0.0	0.0	0	0	?	S	15:00	0:00	[kdevtmpfs]


```
[root@localhost ~]# ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	15:00	?	00:00:12	/usr/lib/systemd/systemd --switched-root --system --deserialize 21
root	2	0	0	15:00	?	00:00:00	[kthreadd]
root	3	2	0	15:00	?	00:00:00	[ksoftirqd/0]
root	5	2	0	15:00	?	00:00:00	[kworker/0:0H]
root	7	2	0	15:00	?	00:00:01	[migration/0]
root	8	2	0	15:00	?	00:00:00	[rcu_bh]
root	9	2	0	15:00	?	00:00:04	[rcu_sched]
root	10	2	0	15:00	?	00:00:01	[watchdog/0]
root	11	2	0	15:00	?	00:00:00	[watchdog/1]
root	12	2	0	15:00	?	00:00:01	[migration/1]
root	13	2	0	15:00	?	00:00:00	[ksoftirqd/1]

8.3 终止进程

8.3.1 基本语法:

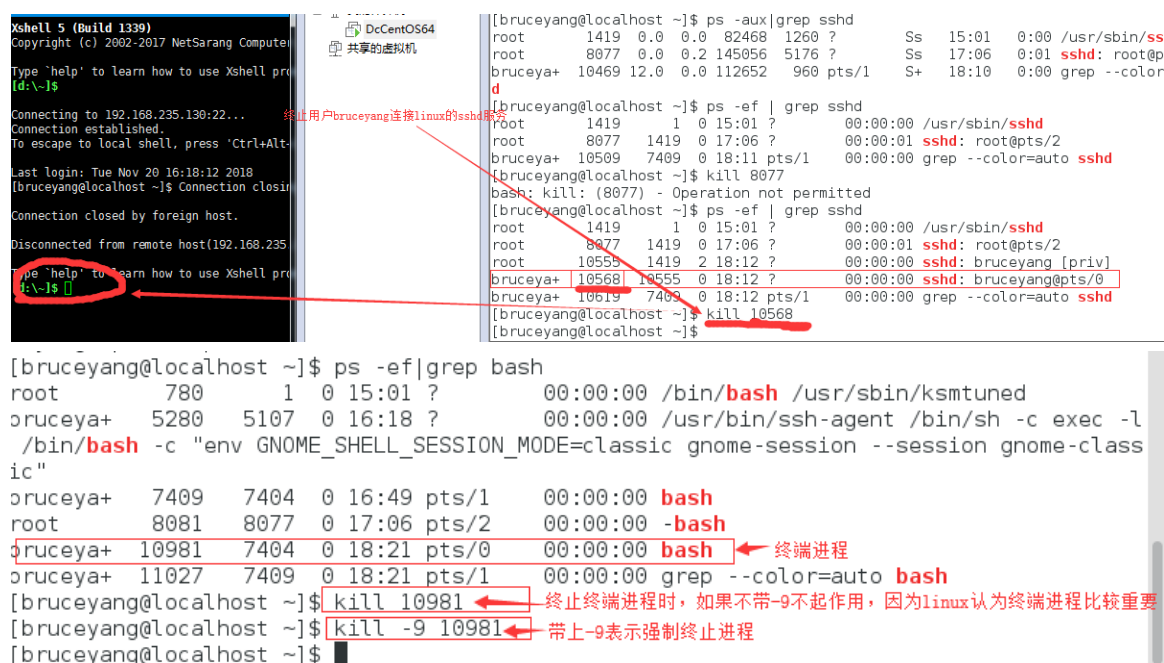
kill 选项 进程ID

killall 进程名称(支持通配符)

选项说明:

-9:表示强迫进程立即停止。

8.3.2 实操案例:



```
[bruceyang@localhost ~]$ ps -aux|grep sshd
root      1419   0.0  0.0  82468  1260 ?        Ss   15:01   0:00 /usr/sbin/sshd
root      8077   0.0  0.2  145056  5176 ?        Ss   17:06   0:01 sshd: root@pts/2
bruceya+ 10469  12.0  0.0  112652   960 pts/1    S+   18:10   0:00 grep --color sshd

[bruceyang@localhost ~]$ ps -ef | grep sshd
root      1419   1  0 15:01 ?        00:00:00 /usr/sbin/sshd
root      8077   1419  0 17:06 ?        00:00:01 sshd: root@pts/2
bruceya+ 10509   7409  0 18:11 pts/1    00:00:00 grep --color=auto sshd

[bruceyang@localhost ~]$ kill 8077
bash: kill: (8077) - Operation not permitted

[bruceyang@localhost ~]$ ps -ef | grep sshd
root      1419   1  0 15:01 ?        00:00:00 /usr/sbin/sshd
root      8077   1419  0 17:06 ?        00:00:01 sshd: root@pts/2
root      10555  1419  2 18:12 ?        00:00:00 sshd: bruceyang [priv]
bruceya+ 10568  10555  0 18:12 ?        00:00:00 sshd: bruceyang@pts/0
bruceya+ 10619   7409  0 18:12 pts/1    00:00:00 grep --color=auto sshd

[bruceyang@localhost ~]$ kill 10568
[bruceyang@localhost ~]$
```

```
[bruceyang@localhost ~]$ ps -ef|grep bash
root      780     1  0 15:01 ?        00:00:00 /bin/bash /usr/sbin/ksmtuned
bruceya+ 5280   5107  0 16:18 ?        00:00:00 /usr/bin/ssh-agent /bin/sh -c exec -l
/bin/bash -c "env GNOME_SHELL_SESSION_MODE=classic gnome-session --session gnome-classic"
bruceya+ 7409   7404  0 16:49 pts/1    00:00:00 bash
root      8081   8077  0 17:06 pts/2    00:00:00 -bash
bruceya+ 10981  7404  0 18:21 pts/0    00:00:00 bash ← 终端进程
bruceya+ 11027  7409  0 18:21 pts/1    00:00:00 grep --color=auto bash

[bruceyang@localhost ~]$ kill 10981 ← 终止终端进程时, 如果不带-9不起作用, 因为linux认为终端进程比较重要
[bruceyang@localhost ~]$ kill -9 10981 ← 带上-9表示强制终止进程
[bruceyang@localhost ~]$
```

8.4 服务管理

8.4.1 服务介绍

服务是支持Linux运行的一些必要程序, 本质上也是进程, 叫守护进程。守护进程通常默默地运行在后台, 为应用程序提供必要支撑, 比如sshd、防火墙等。

8.4.2 服务管理指令

8.4.2.1 基本语法:

systemctl [start、stop、restart、reload、status、enable] 服务名称

CentOS7 之前, 使用命令 service 服务名称 [start、stop、restart、reload、status、enable]

功能描述: 开启、关闭、重启、重新加载、查看服务状态、开机启动

8.4.2.2 实操案例:

```
[root@localhost ~]# systemctl restart firewalld
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-11-22 12:03:11 CST; 12s ago
     Docs: man:firewalld(1)
    Main PID: 4301 (firewalld)
    CGroup: /system.slice/firewalld.service
            └─4301 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Nov 22 12:03:11 localhost.localdomain firewalld[4301]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w2 -w --table filter --de
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# systemctl reload firewalld
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl enable firewalld
[root@localhost ~]#
```

8.4.3 查看网路服务(了解)

8.4.3.1 netstat 命令

netstat 命令查看系统网络服务情况

8.4.3.1.1 基本语法:

netstat 选项

选项说明:

- an 按一定顺序排列输出
- p 显示哪个进程在调用

8.4.3.1.2 实操案例:

```
[root@localhost ~]# netstat -anp | grep sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN     1625/sshd
tcp        0  52 192.168.235.130:22  192.168.235.1:2189  ESTABLISHED 3421/sshd: root@pts
tcp6       0      0 :::22              :::*                 LISTEN     1625/sshd
unix 2      [ ]   DGRAM              32887               3421/sshd: root@pts

[root@localhost ~]# netstat -anp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111         0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:122         0.0.0.0:*               LISTEN      2482/dnsmasq
tcp        0      0 0.0.0.0:22          0.0.0.0:*               LISTEN      1625/sshd
tcp        0      0 0.0.0.0:1631        0.0.0.0:*               LISTEN      1102/cupsd
tcp        0      0 0.0.0.0:125         0.0.0.0:*               LISTEN      2284/master
tcp        0  52 192.168.235.130:22  192.168.235.1:2189  ESTABLISHED 3421/sshd: root@pts
tcp6       0      0 :::111              :::*                    LISTEN      1/systemd
tcp6       0      0 :::22               :::*                    LISTEN      1625/sshd
tcp6       0      0 :::1631             :::*                    LISTEN      1102/cupsd
tcp6       0      0 :::125              :::*                    LISTEN      2284/master
udp        0      0 0.0.0.0:5353        0.0.0.0:*               *          702/avahi-daemon: r
udp        0      0 0.0.0.0:44754       0.0.0.0:*               *          702/avahi-daemon: r
udp        0      0 0.0.0.0:40914       0.0.0.0:*               *          3393/local
udp        0      0 0.0.0.0:122         0.0.0.0:*               *          2482/dnsmasq
```


8.4.3.2 ping 命令

8.4.3.2.1 基本语法:

ping 选项 目标主机

功能描述: 是一种网络检测工具, 它主要是用检测远程主机是否正常, 或是两部主机间的介质是否为断、网线是否脱落或网卡故障。

8.4.3.2.2 实操案例:

```
[root@localhost ~]# ping www.baidu.com
PING www.a.shifen.com (119.75.217.26) 56(84) bytes of data.
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=1 ttl=128 time=28.3 ms
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=2 ttl=128 time=4.67 ms
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=3 ttl=128 time=8.13 ms
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=4 ttl=128 time=11.2 ms
64 bytes from 119.75.217.26 (119.75.217.26): icmp_seq=5 ttl=128 time=5.59 ms
```

8.4.3.3 curl 命令(了解)

8.4.3.3.1 基本语法:

curl 选项 url

功能描述: 用来发送 HTTP 请求。

-X 参数: 指定请求方式

-v 参数: 显示响应结果

-u 参数: 携带用户名/密码

-H 参数: 携带请求消息头信息

8.4.3.3.2 实操案例

```
[root@192 hooks]# curl -X post -v -u admin:b8929794235b46f2b208610902cb4ea7 http://192.168.235.132:8080/jenkins/job/usermgr/build?token=USERMGR_TOKEN
```

9 Linux 软件包管理

9.1 RPM 包的管理(了解)

9.1.1 RPM 介绍

一种用于互联网下载包的打包及安装工具, 它包含在**某些Linux分发版**中。它生成具有.RPM扩展名的文件。RPM是RedHat Package Manager (RedHat软件包管理工具)的缩写, 类似windows的setup.exe, 这一文件格式名称虽然打

上了RedHat的标志，但理念是通用的。Linux的分发版本都有采用（suse,redhat,centos等等），可以算是公认的行业标准了。

9.1.2 RPM 包的查询指令

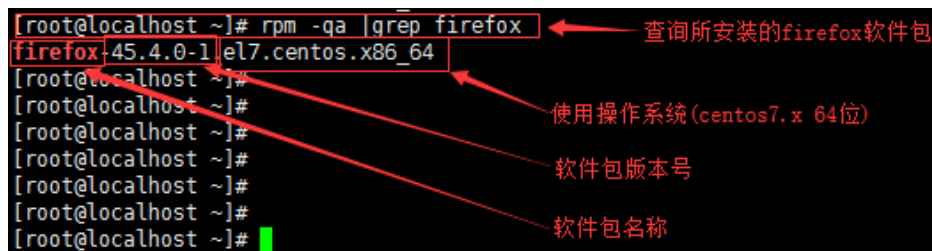
9.1.2.1 基本语法：

rpm -qa :查询所安装的所有软件包

rpm -qa|more: 分页查询所安装的所有软件包

rpm -qa|grep xxx: 过滤查询安装的指定软件包

9.1.2.2 实操案例：



```
[root@localhost ~]# rpm -qa | grep firefox
firefox-45.4.0-1.el7.centos.x86_64
```

Annotations:

- 查询所安装的firefox软件包 (points to the command)
- 使用操作系统(centos7.x 64位) (points to the architecture part of the output)
- 软件包版本号 (points to the version part of the output)
- 软件包名称 (points to the name part of the output)

9.1.3 RPM 包的卸载指令

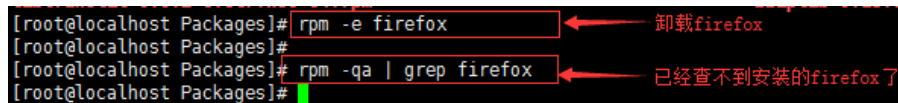
9.1.3.1 基本语法：

rpm -e 选项 软件包名称

选项说明：

--nodeps 表示强制删除，用于被删除的软件包有依赖的情况

9.1.3.2 实操案例：



```
[root@localhost Packages]# rpm -e firefox
[root@localhost Packages]# rpm -qa | grep firefox
```

Annotations:

- 卸载firefox (points to the command)
- 已经查不到安装的firefox了 (points to the empty output of the second command)

9.1.4 RPM 包的安装指令

9.1.4.1 基本语法：

rpm 选项 RPM 包全路径名

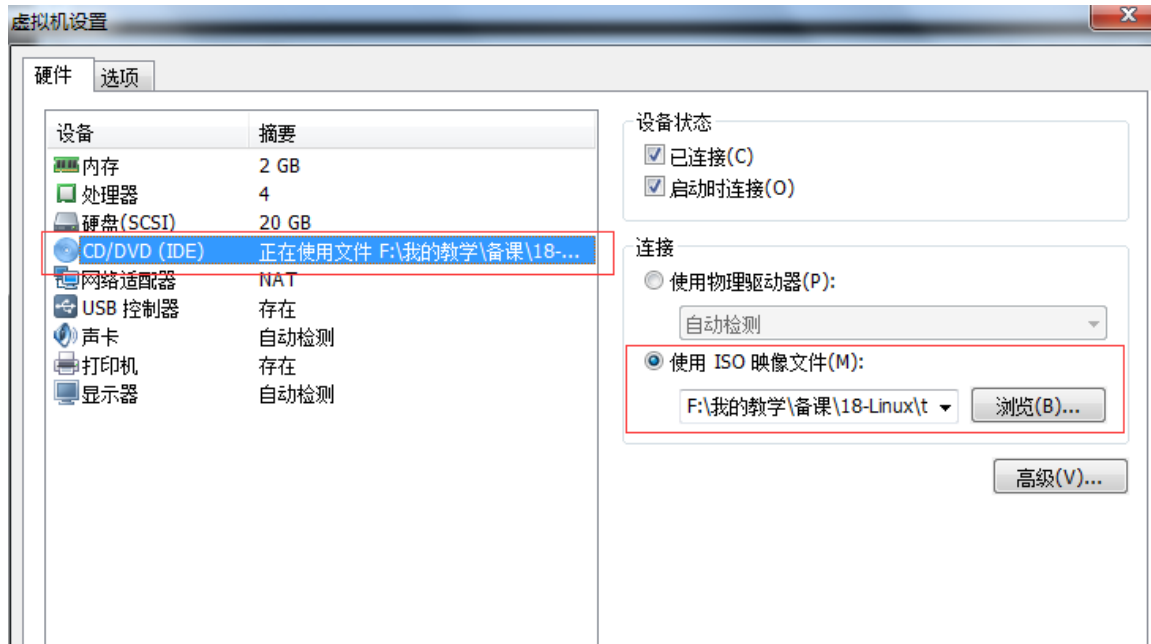
选项说明：

-i=install 安装

-v=verbose 提示
-h=hash 进度条

9.1.4.2 实操案例：

第一步：在虚拟机上挂载 centos 的 iso 文件



第二步：在目录/run/media/root/CentOS 7 x86_64/Packages 下找到 firefox 安装包：

```
[root@localhost ~]# cd /run/media/root/'CentOS 7 x86_64'/Packages
[root@localhost Packages]# ls
389-ds-base-1.3.5.10-11.el7.x86_64.rpm
389-ds-base-libs-1.3.5.10-11.el7.x86_64.rpm
abattis-cantarell-fonts-0.0.16-3.el7.noarch.rpm
abrt-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-ccpp-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-kerneloops-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-pstoreoops-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-python-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-vmcore-2.1.11-45.el7.centos.x86_64.rpm
abrt-addon-xorg-2.1.11-45.el7.centos.x86_64.rpm
abrt-cli-2.1.11-45.el7.centos.x86_64.rpm
abrt-console-notification-2.1.11-45.el7.centos.x86_64.rpm
```

第三步：拷贝安装包到/opt 目录下

```
[root@localhost Packages]# ls -l |grep firefox
-rw-rw-r--. 2 root root 80090076 Sep 21 2016 firefox-45.4.0-1.el7.centos.x86_64.rpm
[root@localhost Packages]# cp firefox-45.4.0-1.el7.centos.x86_64.rpm /opt
[root@localhost Packages]# ls -l |grep firefox
-rw-rw-r--. 2 root root 80090076 Sep 21 2016 firefox-45.4.0-1.el7.centos.x86_64.rpm
[root@localhost Packages]# ls -l /opt |grep firefox
-rw-rw-r--. 1 root root 80090076 Nov 22 14:22 firefox-45.4.0-1.el7.centos.x86_64.rpm
[root@localhost Packages]#
```

第三步：安装 firefox

```
[root@localhost opt]# rpm -ivh firefox-45.4.0-1.el7.centos.x86_64.rpm
Preparing...
Updating / installing...
 1:firefox-45.4.0-1.el7.centos
[root@localhost opt]# rpm -qa |grep firefox
firefox-45.4.0-1.el7.centos.x86_64
[root@localhost opt]#
```

9.2 YUM 包的管理

9.2.1 YUM 介绍

Yum 是一个Shell前端软件包管理器。基于RPM包管理，能够从指定的服务器(在公网上)自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包。

注意：使用YUM的前提是可以连接外网。

9.2.2 YUM 包的查询指令

9.2.2.1 基本语法：

`yum list[installed]|grep xx`

功能描述：查询YUM服务器上的[已经安装到本地的]软件包

9.2.2.2 实操案例：

```
[root@localhost opt]# yum list|grep firefox
firefox.x86_64      45.4.0-1.el7.centos @anaconda
firefox.i686       60.3.0-1.el7.centos updates
firefox.x86_64     60.3.0-1.el7.centos updates
```

← 查询YUM服务器上的firefox软件包

← YUM服务器上存在多个版本的firefox

9.2.3 YUM 包的安装指令：

9.2.3.1 基本语法：

`yum install 包名`

下载安装

9.2.3.2 实操案例：

```
[root@localhost opt]# rpm -e firefox
[root@localhost opt]# rpm -qa |grep firefox
[root@localhost opt]# yum install firefox
Loaded plugins: fastestmirror, langpacks
base
extras
updates
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
--> Package firefox.x86_64 0:60.3.0-1.el7.centos will be installed
--> Processing Dependency: nss >= 3.36.0 for package: firefox-60.3.0-1.el7.centos.x86_64
--> Processing Dependency: nspr >= 4.19.0 for package: firefox-60.3.0-1.el7.centos.x86_64
--> Processing Dependency: libssl3.so(NSS_3.28)(64bit) for package: firefox-60.3.0-1.el7.centos.x86_64
--> Processing Dependency: libssl3.so(NSS_3.27)(64bit) for package: firefox-60.3.0-1.el7.centos.x86_64
```

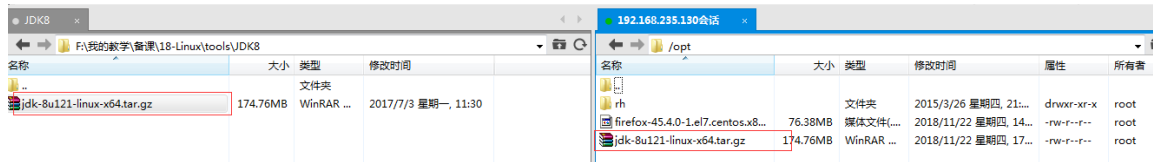
← yum安装firefox

YUM包的卸载：`yum remove firefox.x86_64`

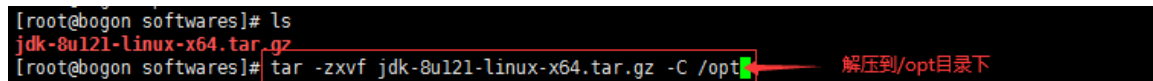
10 搭建 JAVAEE 开发环境

10.1 安装 JDK

第一步：先将软件通过xftp5 上传到/opt 下



第二步：解压缩到/opt目录下



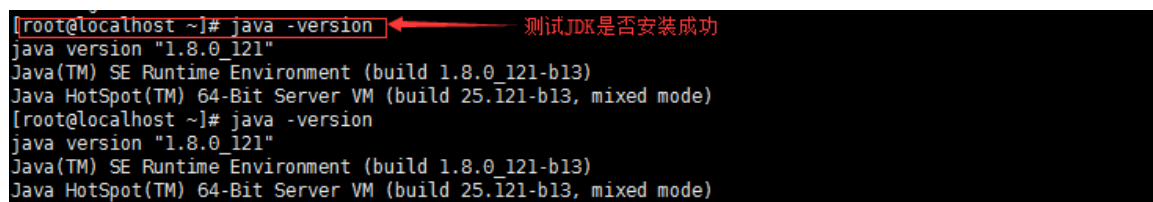
第三步：配置环境变量的配置文件vim /etc/profile

```
JAVA_HOME=/opt/jdk1.7.0_79
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=$JAVA_HOME/lib:.
export JAVA_HOME PATH CLASSPATH
```

第四步：使配置文件生效



第五步：测试安装成功

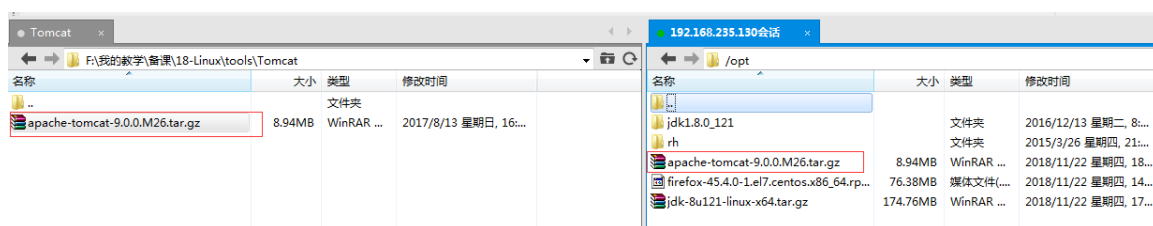


第五步：在目录/home/zhangsan 下编写 HelloWorld.java 编译并运行



10.2 安装 tomcat

第一步：先将软件通过xftp5 上传到/opt 下



第二步：解压到/opt

```
[root@bogon softwares]# ls
apache-tomcat-9.0.0.M26.tar.gz  jdk-8u121-linux-x64.tar.gz
[root@bogon softwares]# tar -zxvf apache-tomcat-9.0.0.M26.tar.gz -C /opt
```

解压到/opt目录下

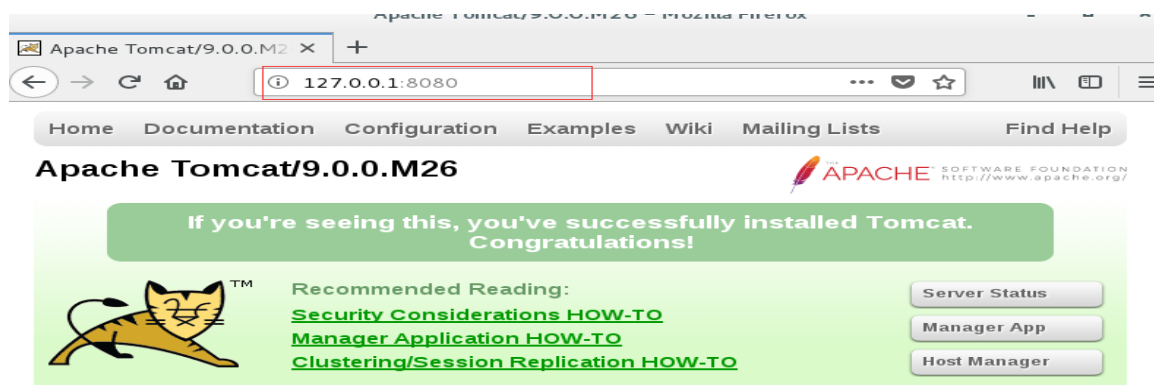
第三步：启动tomcat

```
[root@localhost bin]# pwd
/usr/local/apache-tomcat-8.0.53/bin
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.0.53
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.0.53
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.0.53/temp
Using JRE_HOME:        /usr/local/jdk1.8.0_121
Using CLASSPATH:       /usr/local/apache-tomcat-8.0.53/bin/bootstrap.jar:/usr/local/apache-tomcat-8.0.53/bin/tomcat-juli.jar
Tomcat started.
[root@localhost bin]#
```

进入tomcat的bin目录

启动tomcat

第四步：Linux上访问tomcat



第五步：windows上访问tomcat



注意：从其它机器上访问需要关闭linux的防火墙。

10.3 安装 mysql

第一步：查看是否已经安装了mariadb

检查linux是否安装了mariadb数据库，mariadb数据库是mysql的分支。是免费开源的。mariadb和mysql会有冲突。首先要检查安装了mariadb，卸载掉。

检查命令：yum list installed | grep mariadb

卸载命令：yum -y remove xxxx

*如果卸载不成功，需要去掉参数-y，手动确认卸载。

```
[root@localhost bin]# yum list installed | grep mariadb
mariadb-libs.x86_64 1:5.5.52-1.el7 @anaconda
[root@localhost bin]# yum -y remove mariadb-libs.x86_64
Loaded plugins: fastestmirror, langpacks
Resolving Dependencies
--> Running transaction check
--> Package mariadb-libs.x86_64 1:5.5.52-1.el7 will be erased
--> Processing Dependency: libmysqlclient.so.18()(64bit) for package: 2:postfix-2.10.1-6.el7.x86_64
--> Processing Dependency: libmysqlclient.so.18(libmysqlclient_18)(64bit) for package: 2:postfix-2.10.1-6.el7.x86_64
--> Running transaction check
```

查看是否安装了mariadb

卸载mariadb

第二步：上传 mysql 安装包到/opt 下

名称	大小	类型	修改时间	名称	大小	类型	修改时间	属性
mysql-5.7.18-linux-glibc2.5-x86_64.tar.gz	624.11MB	WinRAR ...	2017/7/3 星期一, 11:32	apache-tomcat-7.0.70		文件夹	2018/11/23 星期五, 13...	drwx
				apache-tomcat-8.0.53		文件夹	2018/11/23 星期五, 12...	drwx
				eclipse		文件夹	2015/2/19 星期四, 16...	drwx
				jdk1.8.0_121		文件夹	2016/12/13 星期二, 8...	drwx
				rh		文件夹	2015/3/26 星期四, 21...	drwx
				apache-tomcat-7.0.70.tar.gz	8.51MB	WinRAR ...	2018/11/23 星期五, 13...	-rw-r
				apache-tomcat-8.0.53.tar.gz	9.02MB	WinRAR ...	2018/11/23 星期五, 12...	-rw-r
				eclipse-jee-luna-SR2-linux-gtk-x86_64.tar.gz	252.68MB	WinRAR ...	2018/11/23 星期五, 12...	-rw-r
				firefox-45.4.0-1.el7.centos.x86_64.rpm	76.38MB	媒体文件(...	2018/11/22 星期四, 14...	-rw-r
				jdk-8u121-linux-x64.tar.gz	174.76MB	WinRAR ...	2018/11/22 星期四, 17...	-rw-r
				mysql-5.7.18-linux-glibc2.5-x86_64.tar.gz	624.11MB	WinRAR ...	2018/11/23 星期五, 16...	-rw-r

第四步：解压 mysql 安装包到目录/opt

```
[root@bogon softwares]# ls
apache-tomcat-9.0.0.M26.tar.gz  jdk-8u121-linux-x64.tar.gz  mysql-5.7.18-linux-glibc2.5-x86_64.tar.gz
[root@bogon softwares]# tar -zxvf mysql-5.7.18-linux-glibc2.5-x86_64.tar.gz -C /opt
```

第五步：修改解压后的根目录名

```
[root@bogon opt]# ls
apache-tomcat-9.0.0.M26  jdk1.8.0_121  mysql-5.7.18-linux-glibc2.5-x86_64  rh  softwares
[root@bogon opt]# mv mysql-5.7.18-linux-glibc2.5-x86_64/ mysql-5.7.18
[root@bogon opt]# ls
apache-tomcat-9.0.0.M26  jdk1.8.0_121  mysql-5.7.18  rh  softwares
[root@bogon opt]#
```

第六步：创建数据文件夹 data

data文件夹是mysql用来存放数据库文件的，数据库的表数据都放在data目录。
默认没有 data 目录，可以手工创建 data 目录，在 mysql-5.7.18 文件夹目录下创建一个 data 文件夹。

```
[root@localhost ~]# cd mysql-5.7.18/
[root@localhost mysql-5.7.18]# ls
bin  COPYING  docs  include  lib  man  README  share  support-files
[root@localhost mysql-5.7.18]# mkdir data
[root@localhost mysql-5.7.18]# ls
bin  COPYING  data  docs  include  lib  man  README  share  support-files
[root@localhost mysql-5.7.18]#
```

第七步：创建用来执行 mysqld 命令的 Linux 用户

创建mysql用户，用来执行MySQL的命令mysqld,此命令用来初始化mysql基础信息。可以使用其他用户，例如叫做 mydb.等，但不推荐。

```
[root@localhost mysql-5.7.18-linux-glibc2.5-x86_64]# groupadd mysql
[root@localhost mysql-5.7.18-linux-glibc2.5-x86_64]# useradd -g mysql mysql
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Creating mailbox file: File exists
[root@localhost mysql-5.7.18-linux-glibc2.5-x86_64]#
```

第七步：初始化 mysql

使用 mysql 的 mysqld 命令初始化数据库的基本信息。切换到 mysql-5.7.18/bin 目录下执行。
命令：./mysqld --initialize --user=mysql --datadir=/opt/mysql-5.7.18/data --basedir=/opt/mysql-5.7.18
参数说明：
--initialize 初始化 mysql，创建 mysql 的 root，随机生成密码。记住密码，登录 msyql 使用。
--user 执行 msyqld 命令的 linux 用户名
--datadir : mysql 数据文件的存放位置，目录位置参照本机的设置。
--basedir : msyql 安装程序的目录，目录位置参照本机的设置。

该命令执行后，会生成一个临时的 mysql 数据库 root 用户的密码，请先拷贝出来记住，后续第一次登录 mysql 需要使用


```
[root@bogon bin]# ./mysqld --initialize --user=mysql --datadir=/opt/mysql-5.7.18/data --basedir=/opt/mysql-5.7.18
2020-02-14T06:37:35.067697Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2020-02-14T06:37:35.297193Z 0 [Warning] InnoDB: New log files created, LSN=45790
2020-02-14T06:37:35.346323Z 0 [Warning] InnoDB: Creating foreign key constraint system tables. 随机生成的mysql root用户密码
2020-02-14T06:37:35.358913Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 7c5ba4c2-4ef4-11ea-bfe3-000c293fcf03.
2020-02-14T06:37:35.360355Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2020-02-14T06:37:35.362722Z 1 [Note] A temporary password is generated for root@localhost: LTt+nCxrX0xM
[root@bogon bin]#
```

密码: iGT#A:CfF7i(

第八步：启用安全功能

在服务器与客户机之间来回传输的所有数据进行加密。通过证书提供了身份验证机制，mysql 命令程序 mysql_ssl_rsa_setup 提供了开启数据加密功能，生成数字证书。

在mysql-5.7.18/bin目录下执行命令：

./mysql_ssl_rsa_setup --datadir=/opt/mysql-5.7.18/data

```
[root@bogon bin]# ./mysql_ssl_rsa_setup --datadir=/opt/mysql-5.7.18/data
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'ca-key.pem'
-----
Generating a 2048 bit RSA private key
....+++
writing new private key to 'server-key.pem'
-----
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'client-key.pem'
-----
```

第九步：修改 mysql 安装目录权限

mysql安装后，需要更改mysql-5.7.18整个文件夹目录权限，更改所属的用户和组为之前创建的mysql用户及其所在组。在mysql安装目录的上级 (/opt) 位置，执行命令chown。

例如：chown -R mysql:mysql /opt/mysql-5.7.18/

chmod 777 /opt/mysql-5.7.18/

```
[root@bogon opt]# chown -R mysql:mysql /opt/mysql-5.7.18/
[root@bogon opt]# ll
total 0
drwxr-xr-x. 9 root root 160 Feb 14 14:20 apache-tomcat-9.0.0.M26
drwxr-xr-x. 8 10 143 255 Dec 13 2016 jdk1.8.0_121
drwxr-xr-x. 10 mysql mysql 141 Feb 14 14:34 mysql-5.7.18
drwxr-xr-x. 2 root root 6 Mar 26 2015 rh
drwxr-xr-x. 2 root root 127 Feb 14 14:27 softwares
[root@bogon opt]# chmod 777 /opt/mysql-5.7.18/
[root@bogon opt]# ll
total 0
drwxr-xr-x. 9 root root 160 Feb 14 14:20 apache-tomcat-9.0.0.M26
drwxr-xr-x. 8 10 143 255 Dec 13 2016 jdk1.8.0_121
drwxrwxrwx. 10 mysql mysql 141 Feb 14 14:34 mysql-5.7.18
drwxr-xr-x. 2 root root 6 Mar 26 2015 rh
drwxr-xr-x. 2 root root 127 Feb 14 14:27 softwares
```

第十步：启动 mysql

启动MySQL服务，mysql-5.7.18/bin目录下执行命令：./mysqld_safe &（其中&符号表示后台启动），输入命令后按Enter。

```
[root@localhost bin]# ls
innobackupsum  myisampack  mysqlcheck  mysqld-debug  mysql_embedded  mysql_secure_installation  mysql_tzinfo_to_sql
lz4_decompress  my_print_defaults  mysql_client_test_embedded  mysqld_multi  mysqlimport  mysqlshow  mysql_upgrade
myisamchk  mysql  mysql_config  mysqld_safe  mysql_install_db  mysqlslap  mysqlxtest
myisam_ftdump  mysqladmin  mysql_config_editor  mysqldump  mysql_plugin  mysql_ssl_rsa_setup  perror
myisamlog  mysqlbinlog  mysql  mysqldumpslow  mysqlpump  mysqltest_embedded  replace
[2]+ Exit 127 ./mysqld_safe
[root@localhost bin]# ./mysqld_safe &
[2] 12710
[root@localhost bin]# Logging to '/usr/local/mysql-5.7.18/data/localhost.localdomain.err'.
2018-11-23T12:42:38.385339Z mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql-5.7.18/data
2018-11-23T12:42:49.003370Z mysqld_safe mysqld from pid file /usr/local/mysql-5.7.18/data/localhost.localdomain.pid ended
```

确认 mysql 是否启动，查看进程，使用 ps -ef | grep mysql


```
[root@localhost bin]# ps -ef|grep mysql
root      6804   6513   0 17:31 pts/2    00:00:00 /bin/sh ./mysqld_safe 已经启动mysql
mysql     6866   6804   0 17:31 pts/2    00:00:17 /usr/local/mysql-5.7.18-linux-glibc2.5-x86_64/bin/mysqld --based
.7.18-linux-glibc2.5-x86_64/data --plugin-dir=/usr/local/mysql-5.7.18-linux-glibc2.5-x86_64/lib/plugin --user=mysql
caldomain.err --pid-file=/usr/local/mysql-5.7.18-linux-glibc2.5-x86_64/data/localhost.localdomain.pid
root      12837   6513   0 20:44 pts/2    00:00:00 grep --color=auto mysql
[root@localhost bin]#
```

第十一步：客户端登录 mysql

登录进入mysql，mysql-5.7.18/bin目录下执行命令：./mysql -uroot -p

-u表示使用root用户登录系统，使用第7步生成的密码。

-p表示使用密码登录

```
[root@localhost bin]# ./mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.18

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

输入密码不显示，输入完成按Enter

mysql提示符：可以输入sql命令

第十步：修改 root 密码

第8步的root用户密码是临时密码，要修改才能使用。

执行sql语句 show databases; 第一次使用将会提示修改mysql的root用户密码：

```
mysql> show databases
-> ;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
mysql>
```

修改mysql的密码，命令语法：alter user '用户名'@'主机域名或ip' identified by '新密码'

例如：alter user 'root'@'localhost' identified by 'yf123';

```
mysql> alter user 'root'@'localhost' identified by 'yf123';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

第十一步：授权远程访问

授权远程访问，在没有授权之前只能在本机访问mysql，远程授权就是让其他计算机通过网络访问mysql（这样远程客户端才能访问）。

授权命令:grant

语法：grant all privileges on *.* to root@'%' identified by 'yf123';

参数：

其中*.* 的第一个*表示所有数据库名，第二个*表示所有的数据库表；

root@'%' 中的root表示用户名，%表示ip地址，%也可以指定具体的ip地址，比如root@localhost,root@192.168.235.130等。

执行授权命令：grant all privileges on *.* to root@'%' identified by 'yf123';

```
mysql> grant all privileges on *.* to root@'%' identified by '123456';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

更新权限信息，执行flush刷新权限: flush privileges;

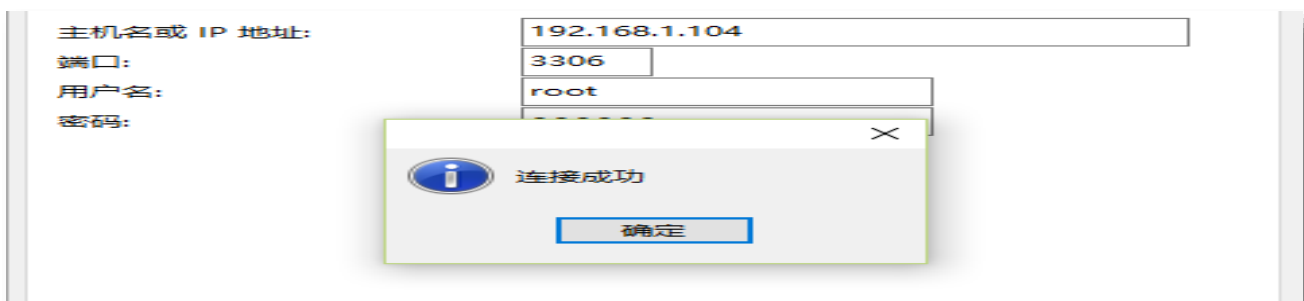
```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

关闭防火墙: systemctl stop firewalld

```
[root@localhost ~]#
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2017-08-16 17:01:45 CST; 14s ago
     Docs: man:firewalld(1)
    Main PID: 729 (code=exited, status=0/SUCCESS)
```

停用防火墙，再查看状态，变为非激活

远程连接数据库:



第十二步: 关闭mysql服务

mysql-5.7.18/bin目录下执行: ./mysqladmin -uroot -p shutdown 输入密码关闭

```
[root@localhost bin]#
[root@localhost bin]# pwd
/usr/local/mysql5.7.18/bin
[root@localhost bin]# ./mysqladmin -uroot -p shutdown
Enter password:
2017-08-16T09:09:53.508874Z mysqld_safe mysqld from pid file /usr/local/mysql5.7.18/data/localhost.localdomain.pid ended
[1]+  Done                  ./mysqld_safe
[root@localhost bin]#
```

查看 mysql 进程，已经没有 mysqld_safe

```
[root@localhost bin]# ps -ef | grep msyql
root      10587   7745  0 17:10 pts/0    00:00:00 grep --color=auto msyql
[root@localhost bin]#
```

这是grep命令

第十三步: 修改数据库编码

查看数据库编码: show variables where Variable_name like '%char%';

修改mysql的字符集: 在mysql客户端执行如下命令

```
set character_set_client=utf8;
set character_set_connection=utf8;
set character_set_database=utf8;
set character_set_results=utf8;
set character_set_server=utf8;
set character_set_system=utf8;
set collation_connection=utf8;
set collation_database=utf8;
set collation_server=utf8;
```

