第一章 Linux的基本认识和常见操作

1.基本认知

2.常见操作

3.卸载软件

1.基本认知

1.Linux由来

> Linux操作系统是基于UNIX操作系统的， 其内核主要是由C程序编写。Linux是自由和开放的，任何组织和个人只要遵循GNU通用公共许可证协议都可以自由免费地使用Linux的所有底层源代码，并可以自由地修改和分发。

2.Linux的目录结构

> Linux和Windows最大的不同之处在于Linux的目录结构的设计，在Linux中，任何文件，目录和设备都在根目录“/”之下。Linux把所有文件和设备都当作文件来管理，这些文件都在根目录下，同时Linux中的文件名区分大小写。

3.命令提示符

`[root@budong ~]#`

格式：`[用户@主机名 当前目录]#`

`root`是`Linux`管理员，也称为超级用户

这里的`budong`是主机名

`~`是当前用户的家目录，家目录就相当于我们`window`系统中的桌面。

`#`是超级用户的提示符，`$`是普通用户的提示符。

命令：`pwd` 当前目录 、`hostname` 主机名、`whoami` 当前用户

```
[root@budong ~]# pwd
/root
[root@budong ~]# hostname
budong
[root@budong ~]# whoami
root
```

4.Linux的用户

在Linux中`root`用户具有超级权限，可以操作任何文件，日常使用中应该避免使用它。这就需要我们在平常使用的过程中使用普通用户。

在Linux中有三种用户，超级用户、系统用户和普通用户，超级用户就是root用户；系统用户是系统正常使用时使用的账户，如bin、mail等，但是系统用户不能够登录；普通用户是普通使用者，能够使用Linux大部分资源，但是一些特定的权限受到控制。

在Linux中可以使用`cat /etc/passwd`查看当前的用户

```
root:x:0:0:root:/root:/bin/bash
#用户名称:用户密码：用户标记号：组标记号：相关注释：主目录：使用的Shell
#root用户可以使用 cat /etc/shadow 查看加密后的用户密码
```

5.Linux的用户管理

1.添加用户

```
[root@budong ~]# useradd budong
[root@budong ~]# passwd budong
更改用户 budong 的密码 。
新的 密码：
无效的密码： 它基于字典单词
无效的密码： 过于简单
重新输入新的 密码：
passwd： 所有的身份验证令牌已经成功更新。
[root@budong ~]#
##在添加用户之后，给用户改密码
```

2.为普通用户添加sudo权限

```
[root@budong ~]# visudo
#在root ALL=(ALL)       ALL  #这行下面添加如下
budong  ALL=(ALL)       ALL
```

`visudo`打开文件后，将`username ALL=(ALL)  ALL`加入到文件中。具体步骤：

1.visudo进入文件。

2.按上下键找到`root    ALL=(ALL)       ALL`这一行内容。

3.按 i 键进入插入模式，然后输入`username ALL=(ALL)  ALL`。

4.输入完成后，按Esc，然后shift+；,末行出现冒号后输入wq回车来保存退出。

在完成上面的步骤之后我们就可以使用普通用户登陆，xshell这里也是，可以是用普通用户，不用担心root权限过大造成关系文件的误操作。

2.常见操作

1.目录跳转`cd`

语法：`cd (选项) (参数)`

常见用法：

```
cd 绝对路径
cd 相对路径
cd .. 回到上一级目录
cd / 跳到根目录
cd ~ 回到家目录
cd   回到家目录
cd . 当前目录
cd - 回到上一次目录
```

查看目录内容的命令配合着cd命令一起学习

```
ls 查看当前目录中的内容
ll 详细列出当前目录中的内容
```

目录结构：

```
~家目录
/根目录  从逻辑上说系统中的所有一切都隶属于它
/bin            --存放所有用户都能执行的命令（二制文件）
/boot           --存放启动文件/内核的相关文件，一般独立成为一个分区。
/dev            --存放物理设备的目录
/etc            --存放配置文件
/home           --用户的家目录
/lib            --32位库文件
/lost+found     --分区修复时找回来的文件会存放在这里，
                存放一些系统不正常关机的的文件残片
/media          --专门用于挂载的目录
/misc           --autofs备用文件夹
/mnt            --专门用于挂载的目录
/opt            --用于存放第三方软件可选目录
/proc           --当前内核的映射，一个虚拟的文件系统
/root           --管理root的家目录
/sbin           --管理员才能够执行的命令  root
/selinux        --selinux安全策略相关的文件
/sys            --内核在内存中的映像文件
/tmp            --临时目录，建议独立划成分区
/usr            --用于存放第三方软件
/var            --存放日志或者频繁修改的文件
```

2.查看目录下的文件`ls`

语法：`ls（选项）（参数）`

选项

-a：显示所有档案及目录（ls内定将档案名或目录名称为“.”的视为影藏，不会列出）
-C：多列显示输出结果。这是默认选项；
-l：与“-C”选项功能相反，所有输出信息用单列格式输出，不输出为多列；
-F：在每个输出项后追加文件的类型标识符，具体含义：“*”表示具有可执行权限的普通文件，“/”表示目录，“@”表示符号链接，“|”表示命令管道FIFO，“=”表示sockets套接字。当文件为普通文件时，不输出任何标识符；
-b：将文件中的不可输出的字符以反斜线“”加字符编码的方式输出；
-c：与“-lt”选项连用时，按照文件状态时间排序输出目录内容，排序的依据是文件的索引节点中的ctime字段。与“-l”选项连用时，则排序的一句是文件的状态改变时间；
-d：仅显示目录名，而不显示目录下的内容列表。显示符号链接文件本身，而不显示其所指向的目录列表；
-f：此参数的效果和同时指定“aU”参数相同，并关闭“lst”参数的效果；
-i：显示文件索引节点号（inode）。一个索引节点代表一个文件；
--file-type：与“-F”选项的功能相同，但是不显示“*”；
-k：以KB（千字节）为单位显示文件大小；
-l：以长格式显示目录下的内容列表。输出的信息从左到右依次包括文件名，文件类型、权限模式、硬连接数、所有者、组、文件大小和文件的最后修改时间等；
-m：用“,”号区隔每个文件和目录的名称；
-n：以用户识别码和群组识别码替代其名称；
-r：以文件名反序排列并输出目录内容列表；
-s：显示文件和目录的大小，以区块为单位；
-t：用文件和目录的更改时间排序；
-L：如果遇到性质为符号链接的文件或目录，直接列出该链接所指向的原始文件或目录；
-R：递归处理，将指定目录下的所有文件及子目录一并处理；

常见用法

```
[root@budong ~]# ls
[root@budong ~]# ll
[root@budong ~]# ls -lrt  #最新更改的文件在最下面
[budong@budong /]$ ls -a
```

3.创建/删除目录`mkdir` `rmdir`

语法：`mkdir (选项)(参数)` `rmdir(选项)(参数)`

```
[budong@budong ~]$ mkdir test  #创建文件夹 test
[budong@budong ~]$ ls
[budong@budong ~]$ cd test/  #进入文件夹
[budong@budong test]$ mkdir a  #创建文件夹  a
[budong@budong test]$ ls
[budong@budong test]$ mkdir b
[budong@budong test]$ rmdir b  #删除文件夹
[budong@budong test]$ ls
[budong@budong test]$ cd ..
[budong@budong ~]$ rmdir test
rmdir: 删除 "test" 失败: 目录非空   #test文件夹下有a文件夹，所以不能直接删除
```

4.创建/删除文件`touch` `rm`

语法：touch(选项)(参数)

touch命令有两个功能：一是用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来；二是用来创建新的空文件

选项

```
-a：或--time=atime或--time=access或--time=use  只更改存取时间；
-c：或--no-create  不建立任何文件；
-d：<时间日期> 使用指定的日期时间，而非现在的时间；
-f：此参数将忽略不予处理，仅负责解决BSD版本touch指令的兼容性问题；
-m：或--time=mtime或--time=modify  只更该变动时间；
-r：<参考文件或目录>  把指定文件或目录的日期时间，统统设成和参考文件或目录的日期时间相同；
-t：<日期时间>  使用指定的日期时间，而非现在的时间；
```

文件类型:

```
b       块文件也叫设备文件也叫特殊文件
c       字符文件
d       目录文件
p       管道文件
f(-)    普通文件 / 文本文件
l       链接文件
s(socket)       unix/类unix套接字
```

注意：linux上文件的后缀名只是给我们自己看的，并不能表示文件的类型。

语法：`rm (选项)(参数)`

`rm` 删除文件或目录

选项

```
-d：直接把欲删除的目录的硬连接数据删除成0，删除该目录
-f：强制删除文件或目录
-i：删除已有文件或目录之前先询问用户
-r或-R：递归处理，将指定目录下的所有文件与子目录一并处理
```

```
rm -i 删除前提示用户进行确认
rm -r 删除指定目录及目录下的所有文件
rm -f 强制删除，没有提示确认
```

5.复制/移动文件`cp` `mv`

`cp` 复制文件或目录，默认情况下，cp命令不能复制目录，如果要复制目录，则必须使用-r选项；

`mv` 对文件/目录重命名或移动文件

`cat`获取文件内容

```
[budong@budong ~]$ mkdir test
[budong@budong ~]$ touch a.py

[budong@budong ~]$ cp a.py b.py
[budong@budong ~]$ ls
a.py  b.py  test

[budong@budong ~]$ cp a.py test/
[budong@budong ~]$ cd test/
[budong@budong test]$ ls
a.py

[budong@budong ~]$ cp a.py test/b.py
[budong@budong ~]$ cd test/
[budong@budong test]$ ls
a.py  b.py

[budong@budong ~]$ cp /etc/passwd test/

[budong@budong ~]$ mv a.py aa.py
[budong@budong ~]$ ls
aa.py  b.py  test
[budong@budong ~]$ mv aa.py test/
[budong@budong ~]$ mv test haha
[budong@budong ~]$ ls
b.py  haha
```

6.查看帮助

`help` 简单帮助

`command(out) --help` 外部命令

`help command(build_in)` 内部命令

安装man命令: `sudo yum install man`

`man` 命令，查看帮助信息时和`less`命令 查看文档一样

```
less 命令使用技巧：
直接上下键到跳行
下一行： e
上一行： y
下一页： 空格键 或 f 或 z
上一页： b 或 w
/string ： 向下搜寻string这个字符串
? string ： 向上搜寻string这个字符串
n,N ： n 继续下一个搜寻，N进行反向搜寻
帮助信息：h
退出 ： q
```

3：卸载软件的命令

```
方法一、如果你知道要删除软件的具体名称，可以使用

sudo apt-get remove --purge 软件名称
sudo apt-get autoremove --purge 软件名称
1
2
方法二、如果不知道要删除软件的具体名称，可以使用

dpkg --get-selections | grep ‘软件相关名称’
```

第二章 Linux常见文件操作命令

`20:30上课！！！`

1.查找命令

2.管道

3.输入输出重定向

4.vim 编辑器

5.关机

6.文件传输命令 sz, rz

1.查找命令

1.命令搜索

`whereis` 搜索命令的位置和帮助文档的位置
`which` 搜索位置和命令的别名

```
[budong@budong ~]$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
[budong@budong ~]$ which ls
alias ls='ls --color=auto'
    /bin/ls
```

2.文件查找

1. `find` 命令格式： `find [-path] -options`

```
path :要查找的目录，默认是当前目录
option:
-name 按文件名的某种规则的查找
-type 按文件类型查找  f 普通文件   l 符号连接   d 目录
-size 按文件大小查找
-exec<执行指令>：假设find指令的回传值为True，就执行该指令；
-print 假设find指令的回传值为Ture，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称前皆有“./”字符串

通配符：
*匹配任意内容
?匹配任意一个字符
[]匹配任意一个中括号内的字符
```

```
[budong@budong /]$ find ./ -name '*.py'   #查找根目录下所有后缀为py的文件
[budong@budong bin]$ find -name 'se*'
./setfont
./sed
[budong@budong bin]$ find -name 'se?'
./sed

[budong@budong bin]$ find -size -6k  #查找小于6k的文件
[budong@budong bin]$ find -size +6k  #查找大于6k的文件，不写的时候就是等于

[budong@budong ~]$ find ./ -name '*.py' -exec rm -rf {} \;  #删除当前目录下所有的py文件
#{} 用于与-exec选项结合使用来匹配所有文件，然后会被替换为相应的文件名

[budong@budong ~]$ find ./ -name 'a' -type d  #查找当前目录下所有为a的文件夹

[budong@budong ~]$ find ./ -name 'haha*' -type d
./tmp/haha

[budong@budong ~]$ find ./ -name 'haha*' -type f
./haha.txt

[budong@budong ~]$ find ./ -name '*.py'
./test/a/b.py
[budong@budong ~]$ find ./ ! -name '*.py'  #查找所有当前目录下所有不是py的文件

[budong@budong ~]$ find  -name '*.py' -print
```

`locate`命令

安装：`sudo yum install mlocate`

locate 是在数据库中按文件名搜索，要查找的文件名中含有的字符串，搜索数据速度更快。搜索的数据库是：/var/lib/mlocate/mlocate.db ，这个数据库，每天自动更新一次，在使用locate之前，可以使用updatedb命令，手动更新数据库。

初始化： sudo updatedb

```
[budong@budong ~]$ locate b.py
[budong@budong ~]$ locate haha.py
[budong@budong ~]$ touch haha.py
[budong@budong ~]$ ls
haha.py  test
[budong@budong ~]$ locate haha.py
[budong@budong ~]$ sudo updatedb
[budong@budong ~]$ locate haha.py
/home/budong/haha.py
```

`grep`命令

文件过滤分割与合并,grep是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。

选项

```
-c 计算符合范本样式的列数。
-E 将范本样式为延伸的普通表示法来使用，意味着使用能使用扩展正则表达式。
-i 忽略字符大小写的差别。
-n 在显示符合范本样式的那一列之前，标示出该列的编号。
-s 不显示错误信息。
-v 反转查找。
-w 只显示全字符合的列。
-x 只显示全列符合的列。
-o 只输出文件中匹配到的部分。
```

```
[budong@budong ~]$ grep -c root /etc/passwd
2
[budong@budong ~]$ grep  root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

[budong@budong ~]$ grep  root /etc/passwd
[budong@budong ~]$ grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
11:operator:x:11:0:operator:/root:/sbin/nologin

[budong@budong ~]$ grep -i Root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

[budong@budong ~]$ grep -x root /etc/passwd
[budong@budong ~]$ grep -w root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

[budong@budong ~]$ grep -o root /etc/passwd
```

2.管道

将一个程序或命令的的输出作为另一个程序或命令的的输入，有两种方法：一种是通过一个临时文件将两个命令或程序结合在一起；另一种就是管道。

管道可以把一系列命令连接起来，可以将前面的命令的输出作为后面命令的输入。使用管道符'|'来建立一个管道行。

```
[budong@budong ~]$ find ./ -name '*.py'|grep test
./test/a/b.py
#先找到所有txt文件，然后再在这些文件里面找包含test的文件
```

3.输入输出重定向

1.输入重定向

输入重定向是指把命令或可执行程序的的标准输入重定向到指定的文件中。也就是输入可以不来自键盘，而是来自一个指定的文件。输入重定向主要用于改变一个命令的的输入源。

输入重定向的一般形式为"命令<文件名"

```
[budong@budong ~]$ cat < /etc/passwd
```

2.输出重定向

输出重定向是指把命令或可执行程序的标准输出或标准错误输出重新定向到指定文件中。命令中的输出不显示在屏幕上，而是写入到指定的文件中，以便完成以后的问题定位或其他的用途。

输出重定向的一般形式为"命令>文件名"

标准输出`echo`

```
[budong@budong ~]$ echo python
python

[budong@budong ~]$ echo python > py.txt
[budong@budong ~]$ ls
haha.py  py.txt  test
```

4.vim编辑器

安装vim

`sudo yum install vim`

工作模式：命令模式、输入模式、末行模式

模式之间切换

多次按ESC可以进入命令模式
在命令模式下，按 i或o或a进入输入模式
在命令模式下，按shift+; ，末行出现:冒号则进入末行模式
按ESC回到命令模式

进入与退出

vim filename 进入
当打开一个文件时处于命令模式

在末行模式下输入q退出文件
wq 保存退出
q! 不保存退出

移动光标

命令模式和编辑模式下都可以用上下左右键（或者h,j,k,l）

gg 到文件第一行

G 到文件最后一行

^非空格行首

0 行首(数字0)

$行尾

```
w  #移动到下一个单词开始
b  #移动到上一个单词开始
e  #移动到当前单词词尾
```

输入文本

在命令模式下
按 i 从光标所在位置前面开始插入

按I在当前行首插入

按 a 从光标所在位置后面开始输入

按A在当前行尾插入

按 o 在光标所在行下方新增一行并进入输入模式

按O在当前上面一行插入

进入输入模式后，在最后一行会出现--INSERT--的字样

复制与粘贴

在命令模式下
yy 复制整行内容到vi缓冲区
yw 复制当前光标到单词尾内容到vi缓冲区
y$ 复制当前光标到行尾的内容到vi缓冲区
y^ 复制当前光标到行首的内容到vi缓冲区

p 读取vi缓冲区的内容，并粘贴到光标当前的位置

```
6yy  #复制6行
```

删除与修改

命令模式下
dd 删除光标所在行
x 删除光标所在字符
u 撤销上一次操作

dw 删除一个单词

CTRL + r 撤销u

CTRL + v 块操作

v 块操作

```
#怎样注释
1.命令模式下，按住CTRL+v（小写）
2.方向选择，选择要注释的内容
3.安装shift+i
4.输入#
5.按esc
```

保存文档

:q 结束编辑不保存退出，如果有修改不会退出
:q! 放弃所做的更改强制退出
:w 保存更改
:wq 保存更改并退出

x 等于wq

查找

命令模式下：

向前搜索 `/`

向后搜索`?`

n　　重复上一条/或？命令，搜索方向相同

N　　重复上一条/或？命令，搜索方向相反

5.关机

`shutdown` 命令安全地将系统关机。

```
-t    多久之后关机
-r    重启计算机
-h    关机后关闭电源
```

最简单的关机命令`halt`

`halt`命令就是调用`shutdown  -h`

重启

`reboot`命令用于重启系统

6.文件传输命令 sz, rz

linux传输文件命令： rz 和 sz

一. 概述

rz，sz是Linux/Unix同Windows进行ZModem文件传输的命令行工具。

优点就是不用再开一个sftp工具登录上去上传下载文件。

Zmodem协议是针对modem的一种错误校验协议。利用Zmodem协议，可以在modem上发送512字节的数据块。如果某个数据块发生错误，接受端会发送“否认”应答，因此，数据块就会被重传。它是Xmodem 文件传输协议的一种增强形式，不仅能传输更大的数据，而且错误率更小。包含一种名为检查点重启的特性，如果通信链接在数据传输过程中中断，能从断点处而不是从开始处恢复传输。

二. 相关命令

1. 【安装命令】：yum install lrzsz
1. 【从linux服务器发送文件 filename 到本地 wndows】：sz filename。 这时会弹出窗口让你选择将文件保存到本地的位置，如下图所示：
1. 【从本地 wndows 上传文件到 linux 服务器】：rz 。 这时会弹出窗口让你选择上传的文件，如下图所示：

第三章 Linux常见命令

1.alias别名

2.压缩解压

3.文件打包

4.链接命令

5.文件权限

6.磁盘管理

1.alias别名

`alias`命令用来设置指令的别名

查看别名可以用`alias`或者`type`

```
[budong@budong ~]$ alias ll
alias ll='ls -l --color=auto'
[budong@budong ~]$ type ll
ll is aliased to `ls -l --color=auto'
```

使用方法：`alias 新的命令='原命令 -选项/参数'`

```
[budong@budong ~]$ alias la='ls -a'
```

取消别名`unalias`

```
[budong@budong ~]$ unalias la
[budong@budong ~]$ la
-bash: la: command not found
```

这种定义别名的方式只在当次登录有效，如果要永久定义生效，可以修改用户（非全部用户）自己的`alias`，修改~/.bashrc文件，在文件中加上自己定义的`alias`。

这个修改要在下次登录才能生效，如果要立即生效则输入`source ~/.bashrc`。

```
[budong@budong ~]$ vim .bashrc
# User specific aliases and functions
alias la='ls -a'
[budong@budong ~]$ source .bashrc
[budong@budong ~]$ la
```

来点有意思的别名

```
[budong@budong ~]$ alias cd='rm -rfv'
[budong@budong ~]$ alias sudo='sudo halt'
[budong@budong ~]$ alias cp='mv'
[budong@budong ~]$ alias vim="vim +q"
#郑重声明！！！这些自己玩玩就好，就不要整蛊同事啦，万一友尽可不能怪我
```

2.压缩解压

1.`zip` / `unzip`

`zip`命令可以用来解压缩文件，或者对文件进行打包操作

`unzip`命令用于解压缩由zip命令压缩的“.zip”压缩包

这两个不是Linux自带的，需要安装

```
sudo yum install zip
sudo yum install unzip
```

选项

```
zip:
-q：不显示指令执行过程
-r：递归处理，将指定目录下的所有文件和子目录一并处理
unzip:
-o  解压时不再询问，直接覆盖
-d  将文件解压到指定的文件夹下
```

压缩

```
[budong@budong ~]$ zip -q -r /tmp/test.zip test #讲当前目录下的test文件夹打包压缩，放到/tmp下
[budong@budong ~]$ zip -q -r test.zip test
[budong@budong ~]$ ls
haha.py  py.txt  test  test.zip   #放到当前目录下
```

解压

```
[budong@budong ~]$ unzip test.zip
[budong@budong ~]$ unzip -o test.zip
[budong@budong ~]$ unzip test.zip -d /tmp/a
```

2. `gzip` / `gunzip`

`gzip`命令用来压缩文件。`gzip`是个使用广泛的压缩程序，文件经它压缩过后，其名称后面会多处`.gz`扩展名。

`gunzip`命令用来解压缩文件。`gunzip`是个使用广泛的解压缩程序，它用于解开被`gzip`压缩过的文件，这些压缩文件预设最后的扩展名为`.gz`。事实上`gunzip`就是`gzip`的`硬连接`，因此不论是压缩或解压缩，都可通过`gzip`指令单独完成。

选项

```
gzip:
-d 对压缩的文件进行解压
-r 递归式压缩指定目录以及子目录下的所有文件
-l 显示压缩文件的压缩信息
gunzip:
-c 把解压后的文件输出到标准输出设备
-f 强行解开压缩文件
-q 不显示警告信息
-r 递归处理
-v 显示命令执行过程
```

压缩

```
[budong@budong ~]$ gzip haha.py
[budong@budong ~]$ gzip -c py.txt > py.txt.gz  #保留源文件
[budong@budong ~]$ gzip -r -l test
[budong@budong ~]$ cd test
[budong@budong test]$ ls
a.py.gz  b.py.gz
```

解压

```
[budong@budong test]$ gzip -d a.py.gz
[budong@budong test]$ ls
a.py  b.py.gz

[budong@budong test]$ gunzip b.py.gz
[budong@budong test]$ ls
a.py  b.py

[budong@budong ~]$ gunzip -f -q -v haha.py.gz
haha.py.gz: -40.0% -- replaced with haha.py
```

3. `bzip2` / `bunzip2`

`bzip2`命令用于创建和管理（包括解压缩）`.bz2`格式的压缩包,它是Linux下的一款压缩软件，比传统的gzip或zip的压缩效率更高，但是它的压缩速度较慢。

`bunzip2`命令解压缩由`bzip2`指令创建的`.bz2`压缩包

选项

```
-c  将压缩与解压缩结果送到标准输出
-d  执行解压缩
-f  文件同名时，预设不会覆盖现有文件,使用这个会覆盖
-k  bizp2 在压缩或解压缩后，会删除原始文件，使用这个不会删除
-s  降低程序执行时内存的使用量
-v  压缩或解压缩文件时，显示详细的信息
```

压缩

```
[budong@budong ~]$ bzip2 haha.py
[budong@budong ~]$ bzip2 -c py.txt > py.txt.bz2  #保留源文件
```

解压

```
[budong@budong ~]$ bzip2 -d -f py.txt.bz2
[budong@budong ~]$ bunzip2 -v -s -k haha.py.bz2
```

3.文件打包

`tar`命令用于将文件打包或解包，扩展名一般为`.tar`,指定特定参数可以调用`gzip`或`bzip2`制作压缩包或解开压缩包

选项

```
-c 建立新的压缩包
-x 解压压缩包
-f 使用压缩包的名字，f参数之后不能再加参数
-i 忽略存档中的0字块
-v 处理过程中输出相关信息
-z 调用gzip来压缩归档文件，与-x联用时调用gzip完成解压缩
-j 调用bzip2压缩或解压
-p 使用源文件的原来属性
```

打包和压缩

```
#仅打包，不压缩
[budong@budong ~]$ tar -cvf /tmp/test.tar test
test/
test/b.py
test/a.py
#打包并使用gzip压缩
[budong@budong ~]$ tar -zcvf test.tar.gz test
#打包并使用bzip2压缩
[budong@budong ~]$ tar -jpcvf test.tar.bz2 test
```

解压

```
[budong@budong ~]$ tar -zxvf test.tar.gz
[budong@budong ~]$ tar -jxvf test.tar.bz2
[budong@budong ~]$ tar -jpxvf ./test.tar.bz2 -C /tmp #解压到/tmp目录下去
```

综合使用

```
[budong@budong tmp]$ find / -name '*.py' >> /tmp/a.list  #>> 输出重定向 追加
[budong@budong tmp]$ tar -T a.list -czvf a.tar.gz
#在上面把要打包的东西都放在a.list文件里面，然后再根据文件内容去打包
```

4.链接命令

`ln`命令用来为文件创件链接，链接类型分为`硬链接`和`符号链接`两种，默认的链接类型是`硬链接`。如果要创建`符号链接`必须使用`-s`选项

注意：`符号链接`文件不是一个独立的文件，它的许多属性依赖于`源文件`，所以给`符号链接`文件设置`存取权限`是没有意义的

`软链接`只会在目的位置生成一个文件的`链接文件`，实际不会占用磁盘空间，相当于Windows中的快捷方式。`硬链接`会在目的位置上生成一个和源文件大小相同的文件。无论`软链接`和`硬链接`，文件保持同步变化。

选项

```
-i 覆盖既有文件之前先询问用户
-s 创建符号(软)链接而不是硬链接
```

示例

```
#ln 源文件  目标文件(快捷方式)
[budong@budong ~]$ ln py.txt p.hard  #创建硬链接
[budong@budong ~]$ ln -s py.txt p.soft  #创建软链接
[budong@budong ~]$ ll

[budong@budong ~]$ cat py.txt
[budong@budong ~]$ cat p.hard
[budong@budong ~]$ cat p.soft
[budong@budong ~]$ echo 'this' >> py.txt
[budong@budong ~]$ cat py.txt
[budong@budong ~]$ cat p.hard
[budong@budong ~]$ cat p.soft     #链接会随着文件内容一起变化

[budong@budong ~]$ rm py.txt
[budong@budong ~]$ cat p.hard
[budong@budong ~]$ cat p.soft   #删除源文件之后，硬链接可以继续使用，软链接不行

[budong@budong ~]$ vim py.txt
[budong@budong ~]$ ln -i py.txt p.hard  #可以重新覆盖之前的链接
ln：是否替换"p.hard"？ y
[budong@budong ~]$ vim p.hard
[budong@budong ~]$ cat py.txt    #改变硬链接内容，文件也会随着一起变化
```

5.文件权限

`ls -l` 查看文件信息，别名为`ll`

```
[budong@budong bin]$ ll
总用量 5660
-rwxr-xr-x. 1 root root  23408 3月  23 02:46 arch
lrwxrwxrwx. 1 root root      4 8月   5 22:16 awk -> gawk
-rwxr-xr-x. 1 root root  22484 3月  23 02:46 basename
-rwxr-xr-x. 1 root root 872372 3月  23 08:11 bash

#第1列表示文件类型
#第2列表示文件权限
#第3列为硬链接个数
#第4列表示文件所有者，就是文件属于那个用户
#第5列表示文件所属的组
#第6列表示文件大小
#第7列表示文件的修改时间
#第8列表示文件名或目录名
```

`-rwxr-xr-x`这10个字符的确定了文件类型和用户对文件的权限

第1个字符代表文件类型：- 表示普通文件

后面9位每3位为一组 （rwx），读（r），写（w），执行（x）

第1组是`u`(user)所有者的权限：`rwx`代表文件的所有者`root`用户有读、写和执行的权限。

第2组是`g`(group)所属组的权限：`r-x` 代表与文件所有者在同一组的用户有读和执行的权限

第3组是`o`(other)其他人的权限：`r-x`代表其他的用户有读和执行权限。

在Linux中，文件有3种属性：`可读`、`可写`和`可执行`。每个文件都有自己的`属主`，每个用户有自己的`用户组`，这样文件权限就有`属主权限`、`同组用户权限`和`不同组用户权限`。

1.文件的权限

`读权限`：对应标志为`r`，表示具有读取文件或目录的权限，对应的使用者可以查看文件内容

`写权限`：对应标志为`w`，对应使用者可以变更此文件，如：删除，移动等。写权限依赖于该文件父目录的权限设置，既父目录没有写权限，就算文件有写权限也不行。

`执行权限`：对应标志为`x`，一些可执行文件必须有可执行权限才可以运行，如：py文件。对于目录而言，可执行权限表示其他用户可以进入此目录，如目录没有可执行权限，则其他用户不能进入此目录。

Linux中通过`符号`表示权限之外，也可以通过`数字`来表示权限
`r`对应数字`4`，`w`对应数字`2`，`x`对应数字`1`。那么rwx就是数字`7`,`0`表示没有任何权限

```
[budong@budong bin]$ ll vi
-rwxr-xr-x. 1 root root 845932 12月 22 2016 vi

[budong@budong bin]$ rm vi
rm：是否删除有写保护的普通文件 "vi"？y
rm: 无法删除"vi": 权限不够
#无法删除没有权限的文件
```

2.改变文件的权限

`chmod`可以改变文件或目录的权限

```
[budong@budong ~]$ ll py.txt
-rw-rw-r--. 1 budong budong 0 8月   9 08:17 py.txt

[budong@budong ~]$ chmod u+x  py.txt  #属主既budong增加执行权限
[budong@budong ~]$ chmod g+x  py.txt  #同组增加执行权限
[budong@budong ~]$ chmod o-r  py.txt  #其他用户去掉可读权限
[budong@budong ~]$ ll py.txt
-rwxrwx---. 1 budong budong 0 8月   9 08:17 py.txt
#使用数字改变权限
[budong@budong ~]$ chmod 664 py.txt
[budong@budong ~]$ ll py.txt
-rw-rw-r--. 1 budong budong 0 8月   9 08:17 py.txt
```

`chown`将文件变更成新的属主或属组

```
[budong@budong tmp]$ ll vi
-rwxr-xr-x. 1 budong budong 845932 8月   9 08:03 vi  #属主和组都是不动

[root@budong tmp]# chown -R root /tmp/vi  #登陆root用户，改成属主为root用户
[root@budong tmp]# ll vi
-rwxr-xr-x. 1 root budong 845932 8月   9 08:03 vi  #属主已改，组没改
```

`chgrp`改变文件或目录所属的用户组

```
[root@budong tmp]#  ll vi
-rwxr-xr-x. 1 root budong 845932 8月   9 08:03 vi #属组为budong
[root@budong tmp]# chgrp root /tmp/vi
[root@budong tmp]# ll vi
-rwxr-xr-x. 1 root root 845932 8月   9 08:03 vi   #属组已改成root
```

6.磁盘管理

`df`命令查看硬盘空间的使用情况

```
[budong@budong /]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_budong-lv_root
                       18G  736M   16G   5% /
tmpfs                 504M     0  504M   0% /dev/shm
/dev/sda1             477M   26M  426M   6% /boot
#-h 可读的方式显示当前磁盘空间
```

`du`查看磁盘或某个目录占用的磁盘空间

```
[budong@budong bin]$ du -chs
5.6M    .
5.6M    总用量
#c 最后再加上总计  h 打印可识别的格式，如：1KB，500MB，1GB s只显示各档案大小的总和
[budong@budong bin]$ du -ah  #显示全部目录和其次目录下的每个档案所占的磁盘空间
```

第四章 Linux基础

1.Linux进程管理

2.程序安装与管理

1.Linux进程管理

1.进程的概念

计算机实际上可以做的事情实质上非常简单，比如计算两个数的和，再比如在内存中寻找到某个地址等等。这些最基础的计算机动作被称为指令 (instruction)。所谓的程序(program)，就是这样一系列指令的所构成的集合。通过程序，我们可以让计算机完成复杂的操作。程序大多数时候被存储为可执行的文件。这样一个可执行文件就像是一个菜谱，计算机可以按照菜谱作出可口的饭菜。

进程是程序的一个具体实现。大家可以简单的理解为进程就是正在进行中的程序。一般来说，一个程序就是一个进程，但是很多情况下，程序为啦完成更多的事情，往往一个程序有多个进程。

2.进程监视

`ps`提供进程的一次性查看

选项

```
u 按用户和启动时间的顺序来显示进程
a 显示所有用户的所有进程
x 显示无终端控制的进程
f 列出进程全部相关信息，通常和其他选项联用
e 所有进程
```

演示

```
[budong@budong etc]$ ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   2900  1400 ?        Ss   Aug08   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    Aug08   0:00 [kthreadd]

[budong@budong etc]$ ps -ef|grep budong
#USER: 启动进程的用户
#PID: 进程的ID号
#%CPU: 进程占用的CPU百分比
#%MEM: 进程占用的物理内存百分比
#VSN: 进程使用的虚拟内存总量，单位KB
#RSS:该进程占用实际物理内存的大小，单位KB
#TTY: 该进程在哪个终端中运行
#STAT: 进程状态
#START: 启动进程的时间
#TIME: 进程消耗CPU的时间
#COMMAND: 产生此进程的命令名
#STAT常见状态：
#R 运行， S 睡眠，T 停止，s 包含子进程，+ 位于后台 Z僵尸进程  < 优先级比较高的进程
```

`top` 监视系统实时状态

选项

```
-d 设置更新的时间间隔
-n 显示更新的次数，然后退出
-u 只显示指定用户的进程信息
-s 安全模式运行，禁用一些交互命令
```

演示

```
[budong@budong ~]$ top -d 2
top - 09:36:12 up 17:34,  2 users,  load average: 0.00, 0.00, 0.00
Tasks:  76 total,   1 running,  75 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1030372k total,   326728k used,   703644k free,    33636k buffers
Swap:  2064380k total,        0k used,  2064380k free,   216784k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND

    1 root      20   0  2900 1400 1204 S  0.0  0.1   0:01.28 init

    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
#第1行分别是：系统时间、系统启动了17小时34分、登录的用户有2个、 “load average” 代表：最近1分钟，5分钟，15分钟的系统负载值，如超过CPU个数的2倍说明高负载，需处理

#第2行Tasks表示：有76个进程在内存中，1个正在运行，75个睡眠，0个停止，0个处于僵尸状态

#第3行Cpu(s)表示：user上花费的时间，sys上花费的时间，nice优先级调整上花费的时间，idle：CPU空闲时间，iowait等待系统io时间，hi硬件中断时间，si软中断时间，steal被虚拟机偷掉的时间

#第4行Mem表示：总内存、已经使用的内存、空闲的内存、用于缓存文件系统的内存、交换内存的总容量，这内存还可以用free命令来查看

#第5行Swap表示：交换空间总大小、使用的交换内存空间、空闲的交换空间、用于缓存文件内容的交换空间
```

`top`命令的交互模式当中可以执行的命令

```
h  显示交互模式的帮助
P  以CPU使用率排序，由大到小，默认
M  按内存占有大小排序，由大到小
N  以进程ID大小排序，由大到小
q  退出top程序
```

大家可以使用`htop`来查看，界面上更加好看

`yum -y install epel-release`

`yum install htop`

Linux 的进程分为前台进程和后台进程，`前台进程`占用`终端窗口`，而`后台进程`不占用`终端窗口`。要启动一个前台进程，只需要在命令行输入启动进程的命令即可，要让一个程序在后台运行，只需要在启动进程时，在命令后加上`&`符号即可。

`Ctrl + Z` 让正在前台执行的进程暂停

`jobs` 获取当前的后台作业号

`fg` 将进程从后台调到前台执行
`bg` 将进程放到后台执行

`kill` 删除执行中的程序

常用的信号量：

```
HUP  1  终端断线
INT  2  中断（同 Ctrl + C）
QUIT 3  退出（同 Ctrl + \）
TERM 15 终止
KILL 9  强制终止
CONT 18 继续（与STOP相反， fg/bg命令）
STOP 19 暂停（同 Ctrl + Z）
```

`kill -9 进程id` 杀死进程

一般在命令之后加“&”，就可以放到后台执行。

2.程序安装与管理

`RPM`包管理机制最早是有`Red Hat`公司研制，然后由开源社区维护，是`centos`上主要的软件包管理机制。

`yum` 基于`rpm`的软件包管理器，可以自动处理依赖关系

如用`yum`安装`python3`

```
[budong@budong ~]$ python -V
Python 2.6.6
[budong@budong ~]$ sudo yum -y install epel-release  #添加源
[budong@budong ~]$ sudo yum -y install python34  #安装python3.4
#如果想卸载安装的软件可以使用 sudo yum remove python34  来移除安装的软件
```

另外使用`yum list`查看所有能用`yum`安装的包,使用`yum list|grep python`可以查看和`python`有关的包。

使用`yum`可以很方便的安装的软件，如`MySQL`等等，但是`yum`源一般比较的旧，不是特别的新，我们可以使用163源，这样下载的时候会快一些，软件版本也会高一点。

```
[root@budong /]# mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup
#先备份
[root@budong yum.repos.d]# yum install wget
[root@budong yum.repos.d]# cd /etc/yum.repos.d
[root@budong yum.repos.d]# wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
[root@budong yum.repos.d]# yum clean all
[root@budong yum.repos.d]# yum makecache
#按照以上步骤就可安装163源
```

`yum`源很方便，但是有时候我们需要安装高版本的软件的时候，`yum`安装就满足不了我们要求，这个时候我们可以去下载`rpm`包，然后用`rpm`包去安装。

我们用这个方法来安装`MySQL5.7`

```
[budong@budong tools]$ rpm -qa|grep mariadb*    #首先删除已有的mariadb，mysql数据库，

[root@budong tools]# rpm -e --nodeps mysql-libs-5.1.73-8.el6_8.i686
#使用rpm -e --nodeps mariadb-...   删除安装的数据库

[budong@budong tools]$ wget https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.19-
1.el6.i686.rpm-bundle.tar
#即下载群文件进阶软件10，在虚拟机里面下载很慢，大家可以打开这个链接用迅雷去下,下载地址：
https://dev.mysql.com/downloads/mysql/

[budong@budong tools]$ sudo yum install lrzsz
#大家从群文件里面下载之后，安装这个，就可以直接从主机拖到虚拟机里面

[budong@budong tools]$ tar -xvf 10mysql-5.7.19-1.el6.i686.rpm-bundle.tar  #解压

[budong@budong tools]$ sudo yum -y install numactl #安装一个依赖

#必须要按照下面的顺序来，不然就会安装不上
[budong@budong tools]$ sudo rpm -ivh mysql-community-common-5.7.19-1.el6.i686.rpm
[budong@budong tools]$ sudo rpm -ivh mysql-community-libs-5.7.19-1.el6.i686.rpm
[budong@budong tools]$ sudo rpm -ivh mysql-community-client-5.7.19-1.el6.i686.rpm
[budong@budong tools]$ sudo rpm -ivh mysql-community-server-5.7.19-1.el6.i686.rpm

#切换root身份，数据库初始化，也可以一开始就切换成root
[root@budong tools]# mysqld --initialize --user=mysql

#查看root密码  xormlHYlC1&o  就是自己生成的root密码
[root@budong tools]# cat /var/log/mysqld.log | grep 'password'
2017-08-09T12:49:51.364886Z 1 [Note] A temporary password is generated for root@localhost:
xormlHYlC1&o

#启动MySQL服务
[root@budong tools]# service mysqld start

#登陆
[root@budong tools]# mysql -u root -p

#修改 root 密码
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'qwe123';

#退出
mysql> \q

#修改配置文件
[root@budong tools]# vim /etc/my.cnf
[mysqld]

datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
character_set_server = utf8

symbolic-links=0
```

```
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[client]
default-character-set=utf8
socket=/var/lib/mysql/mysql.sock

[mysql]
default-character-set = utf8

#重启服务
[root@budong tools]# service mysqld restart

#清理安装
[root@budong tools]# mysql_secure_installation
此时输入 root 密码，接下来，为了安全，MySQL 会提示你密码为级别，重置 root 密码，移除其他用户账号，禁用 root 远程登录，移除 test 数据库，重新加载 privilege 表格等，你只需输入 y 继续执行即可。如果root密码改啦的话就是输no，其他的地方一直y就可以啦。至此，整个 MySQL 安装完成。
#Press y|Y for Yes, any other key for No: y  确定后面的密码级别
#Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0 最小级别，如果是后面的级别的话，以后新建用户的时候设置密码必须要满足选定的级别
#Change the password for root ? ((Press y|Y for Yes, any other key for No) : n 是不是改root密码，我这里就不改啦
#接下来全部都是y


#设置开机启动
[root@budong tools]# chkconfig --levels 235 mysqld on

#再次登陆MySQL
[root@budong tools]# mysql -u root -p

#查看字符集
mysql> SHOW VARIABLES LIKE 'character%';

#创建管理员用户 因为root用户限制不能远程登陆
mysql> CREATE USER 'admin'@'%' IDENTIFIED BY 'rootqwe123';

#给这个用户授予所有的远程访问的权限。这个用户主要用于管理整个数据库、备份、还原等操作。
mysql> GRANT ALL  ON *.* TO 'admin'@'%';

#使更改立即生效
mysql> FLUSH PRIVILEGES;

#创建普通用户
mysql> CREATE USER 'develop'@'%' IDENTIFIED BY 'QWEqwe123';  #这里的%代表可以远程登陆

#给这个用户授予 SELECT,INSERT,UPDATE,DELETE 的远程访问的权限，这个账号一般用于提供给实施的系统访问
mysql> GRANT SELECT,INSERT,UPDATE,DELETE  ON *.* TO 'develop'@'%';

#使更改立即生效

mysql> FLUSH PRIVILEGES;
```

```
#退出
mysql> \q

#以上就是在centos6.9上用rpm安装MySQL5.7.19的方法，注意这个方法适用5.7.5之后的版本
```

`rpm`总结

```
rpm -qa  |grep mysql  #查看已安装的某个软件
rpm -e mysql-...  #删除某个安装包
rpm -ivh mysql...  #安装某个安装包

#选项
-a 显示安装的所有软件列表
-e 从系统中移除指定的软件包
-h 安装软件时输出hash记号：#
-i 安装软件时显示软件包的相关信息
-l 显示软件包中的列表
-v 安装软件时显示命令的执行过程
-q 使用询问模式，当遇到任何问题时，rpm指令会先询问用户
-p 查询软件包的文件
```

这个再给大家用yum安装MySQL的方法，大家参考一下，但是yum安装的版本比较低。

```
linux上的mysqld的yum安装，yum安装的是MySQL5.1的版本，参考一下

#查看已经安装的mysql文件
rpm -qa | grep mysql

#查看可以按的RPM包
yum list | grep ^mysql

#安装mysql开发包及mysql服务端
sudo yum install mysql-devel mysql-server

#启动mysql：
sudo service mysqld start
#会出现非常多的信息，目的是对mysql数据库进行初始化操作

#查看mysql服务是否开机自启动
chkconfig --list | grep mysqld

#如果没有启动，可以通过下面语句来启动：
sudo chkconfig mysqld on

#通过命令给root账号设置密码为root
#(注意：这个root账号是mysql的root账号，非Linux的root账号)
mysqladmin -u root password 'root'

#通过 mysql -u root -p 命令来登录我们的mysql数据库了

#进入mysql,查看编码集
SHOW VARIABLES LIKE '%char%';

#查看校对集
SHOW VARIABLES LIKE '%colla%';

#改配置文件：
#查看服务的状态
service mysqld status
#关闭服务
sudo service mysqld stop

#进入文件
sudo vi /etc/my.cnf

# 往里面加进进去的内容
# 服务端
[mysqld]
character-set-server=utf8
collation-server=utf8_general_ci

# 客户端：
[client]
default-character-set=utf8


#重启服务
```

```
sudo service mysqld restart
```

在`Linux`上除了`yum`和`rpm`安装之外，还有源码包的安装，这里以`python3`的源码安装为例

```
#首先卸载之前安装好的python34
[budong@budong tools]$ sudo yum remove python34

#首先安装一些依赖，为后面的编译做准备
[budong@budong tools]$ sudo yum install -y openssl-static
[budong@budong tools]$ sudo yum install -y gcc
[budong@budong tools]$ sudo yum groupinstall "Development tools"
sudo yum -y install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-
devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel

#下载python源码包
[budong@budong tools]$ wget http://python.org/ftp/python/3.6.1/Python-3.6.1.tar.xz
#下载会有点慢，可以先在Windows上用迅雷下好，然后像上面MySQL的包一样，直接拖进来也可以

#解压文件
[budong@budong tools]$ tar -xvf Python-3.6.1.tar.xz

#进入解压后的文件夹
[budong@budong tools]$ cd Python-3.6.1

#配置 配置文件
[budong@budong Python-3.6.1]$ ./configure --prefix=/usr/local/python3

#编译
[budong@budong Python-3.6.1]$ make

#安装
[budong@budong Python-3.6.1]$ sudo make install

#添加python3的符号链接
[budong@budong Python-3.6.1]$ sudo ln -s /usr/local/python3/bin/python3 /usr/bin/python3

#添加pip3的符号链接
[budong@budong Python-3.6.1]$ sudo ln -s /usr/local/python3/bin/pip3 /usr/bin/pip3

#看看效果
[budong@budong Python-3.6.1]$ pip3 -V
pip 9.0.1 from /usr/local/python3/lib/python3.6/site-packages (python 3.6)

#编译安装完成
```

在Linux上，不是所有的有的软件都能通过`yum`或者`rpm`等其他类似的包管理工具就能安装好，很多时候需要自己去编译安装，各种软件的编译安装所需要的依赖也不一样，但是在网上都可以找到资料，大家安装的时候可以先找找资料然后多操作几次就行。

补充一点python内容：

这里我们用python的时候默认是python2，使用python3需要每次都加个3，虽然也可以重新链接，让python指向的是python3，但是这样会影响系统的一些程序运行，这是时候我们可以建立一个python3的虚拟环境，在虚拟环境里面使用python就是python3，具体操作如下：

```
#Linux上的python2不自带pip，需要安装操作如下：
[budong@budong ~]$ wget https://bootstrap.pypa.io/get-pip.py
[budong@budong ~]$ sudo python get-pip.py
#这样pyhton2的pip就安装上啦，yum安装python3时，python3 get-pip.py就会安装python3的pip

#下载python的虚拟环境安装包
[budong@budong ~]$ sudo pip install virtualenv

#创建虚拟环境
[budong@budong ~]$ virtualenv  -p /usr/bin/python3  py3env

#进入虚拟环境
[budong@budong ~]$ source py3env/bin/activate
(py3env) [budong@budong ~]$ python

#退出虚拟环境
(py3env) [budong@budong ~]$ deactivate

#把  source py3env/bin/activate  添加到  .bashrc  中，可以一登陆就进入啦python3的虚拟环境

#如果要添加python2的虚拟环境只需把上面的python3改成python2即可，或者如下
[budong@budong ~]$ virtualenv  env_py2
#因为系统默认是python2，所以可以不用添加命令的路径
```

第五章 Linux系统修改PATH环境变量方法

Linux系统修改PATH环境变量方法

在Linux安装一些软件通常要添加路径环境变量PATH.PATH环境变量通俗的讲就是把程序的路径"备案"到系统中,这样执行这些程序时就不需要输入完整路径,直接在bash输入程序名就可以执行.比如常用的ls命令就是添加好了环境变量才可以直接执行ls

0查看PATH环境变量

终端输入echo $PATH返回如下,各路径用：隔开.符号用于展开变量的值.

```
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

1添加路径到PATH

添加路径可以直接给PATH赋值

```
export PATH=$PATH:新添加的路径
```

$PATH是当前的路径1:路径2:路径3 ,在后面追加:新添加的路径,然后把这个新的路径集赋值给PATH本身,相当于覆盖了原PATH变量.export命令是导出变量,相当于更新PATH

2删除PATH中的路径

删除PATH中的某个路径也是用以上重新给PATH赋值的方法,比如当前PATH是

```
echo $PATH
路径1:路径2:路径3:路径4
```

要删除路径4,只需要copy上面的路径1:路径2:路径3

然后赋值给PATH

```
export PATH=路径1:路径2:路径3
```

3在系统文件修改环境变量

修改环境变量常用3种方法:

1.直接在命令行输入语句

2.在文本文件(常命名为为.sh脚本文件,也可无后缀)中写语句,source这个文件使修改生效

3.在系统文件(如HOME下的.bashrc)中写语句,source或重启生效

用命令行修改PATH,只针对当前shell有用,关了终端就失效了

普通文本文件修改PATH每次登陆需要source,常用在安装软件和交叉编译

系统文件中修改相当于每次登陆系统自动source,参考添加环境变量到系统级或用户级的文件中

(推荐用root账户修改系统文件,设置root账户的方法)

登陆root,在家目录(root)的.bashrc文件添加新路径到PATH

```
export PATH=$PATH:新增路径
```

更直接的方法:在.bashrc写明PATH的所有路径,注意先要echo $PATH然后copy系统自带的路径

```
export PATH=路径1:路径2:路径n
```

这样增加和删除路径直接改写.bashrc即可

要立即生效只需要source一下.bashrc文件,之后无需再source

```
source .bashrc
```

.bashrc中的修改对于当前账户的每次登陆都有效

要恢复系统默认PATH,删掉.bashrc中的修改语句即可