
Linux 文件查找与编辑器使用

1、文件查找、grep、管道命令

1.1、which 命令

which 命令用于查找并显示给定命令的绝对路径，环境变量 **PATH** 中保存了查找命令时需要遍历的目录。**which** 指令会在环境变量 **\$PATH** 设置的目录里查找符合条件的文件。也就是说，使用 **which** 命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令

which/usr/bin/which#显示系统命令所在目录,绝对目录，不能查找文件

此命令会去搜索**\$PATH** 环境变量中的目录路径：可以使用 **echo \$PATH** 查看,如：

```
luohuihua@ubuntu:~$ echo $PATH
/home/luohuihua/bin:/home/luohuihua/.local/bin:/usr/local/sbin:/usr/local/bin
:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
luohuihua@ubuntu:~$
```

路径使用“:”号分隔，**which** 命令会在这些路径下去搜索。

语法: **which** [系统命令]

```
luohuihua@ubuntu:~$ which pwd
/bin/pwd

luohuihua@ubuntu:~$ which adduser
/usr/sbin/adduser

luohuihua@ubuntu:~$
```

说明： **which** 是根据使用者所配置的 **PATH** 变量内的目录去搜寻可运行档的！所以，不同的 **PATH** 配置内容所找到的命令当然不一样的！

扩展： 与 **which** 功能相似的还有一条命令 **whereis** 也可以查找到命令的绝对路径

与 `whereis` 不同, `which` 会列出这个命令的别名记录, 而 `whereis` 会显示出这个命令的帮助文档所在位置

语法: `whereis`(选项)(参数)

选项

- b: 只查找二进制文件;
- B<目录>: 只在设置的目录下查找二进制文件;
- f: 不显示文件名前的路径名称;
- m: 只查找说明文件;
- M<目录>: 只在设置的目录下查找说明文件;
- s: 只查找原始代码文件;
- S<目录>: 只在设置的目录下查找原始代码文件;
- u: 查找不包含指定类型的文件。

`whereis` 命令只能用于程序名的搜索, 如果省略选项, 则返回所有信息

```
luohuihua@ubuntu:~$ whereis pwd
pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz
luohuihua@ubuntu:~$ whereis -b pwd
pwd: /bin/pwd /usr/include/pwd.h
luohuihua@ubuntu:~$ whereis svn
svn:
luohuihua@ubuntu:~$
```

说明: `svn` 没安装, 找不出来

1.2、find 命令

`find` 命令用于: 在一个目录 (及子目录) 中搜索文件, 你可以指定一些匹配条件, 如按文件名、文件类型、用户等条件查找文件。

语法: `find` [搜索路径] [搜索选项] filename

path 路径, 表示从这个路径下开始查找

选项说明:

- name filename 查找名为 filename 的文件
- size +/-大小 按照文件大小来查找, +大于, -小于

-user username 按文件所属查找

通过时间值查找

-ctime -atime -mtime （以天为单位）

-cmin -amin -mmin （以分钟为单位）

-type 按文件的类型

-inum 根据 i 节点进行查找

-group 组名 按所属组来查找

-a and 逻辑与 -o or 逻辑或

-exec 或 -ok command {} \; 将查到的文件执行 command 操作,{} 和 \;之间有空格,固定格式。提示: 如果 find 命令省略路径不写,表示从当前路径开始查找。find 还可以结合通配符使用。

linux 下的通配符

字符	含义	实例
*	匹配 0 或多个字符	a*b a与b之间可以有任意长度的任意字符,也可以一个也没有,如aabcb, axyzb, a012b, ab。
?	匹配任意一个字符	a?b a与b之间必须也只能有一个字符,可以是任意字符,如aab, abb, acb, a0b。
[list]	匹配 list 中的任意单一字符	a[xyz]b a与b之间必须也只能有一个字符,但只能是 x 或 y 或 z,如: axb, ayb, azb。
[!list]	匹配 除list 中的任意单一字符	a[!0-9]b a与b之间必须也只能有一个字符,但不能是阿拉伯数字,如axb, aab, a-b。
[c1-c2]	匹配 c1-c2 中的任意单一字符 如 : [0-9] [a-z]	a[0-9]b 0与9之间必须也只能有一个字符如a0b, a1b... a9b。
{string1,string2,...}	匹配 string1 或 string2 (或更多)其一字符串	a{abc,xyz,123}b a与b之间只能是 abc或xyz或123这三个字符串之一。

1.2.1 -name 选项

find /etc -name 文件名 # -name 最常见的选项 find /etc -name init // 在目录/etc 中查找文件 init 文件

注意:

- 尽量缩小查找范围,不要在根目录下查找,不然:1、查找速度非常慢;2、占用大量系统资源。
- 占用系统资源越少越好,而且尽量在服务器压力较小时用 find 进行查找。

- `find .. -name` 查找结果与 Windows 不同，如:Windows 会把所有包含 `init` 关键词的文件全都列出来而 Linux 则只会匹配 `init` 关键词
- 使用通配符:
 - *: 用于匹配任意字符
 - `find /etc -name init*` #查找所有以 `init` 开头的文件
 - `find /etc -name *init*` #`init` 左右都没有空格，用于查找所有包含 `init` 关键词的文件
 - ?: 用于匹配单个字符
 - `find /etc -name init???` #这个文件会有七个字符。
 - `find /etc -name ?init??`

```
root@ubuntu:~# find /etc -name init
/etc/apparmor/init
/etc/init
root@ubuntu:~# find /etc -name init*
/etc/init.d
/etc/apparmor/init
/etc/kernel/postinst.d/initramfs-tools
/etc/kernel/postrm.d/initramfs-tools
/etc/initramfs-tools
/etc/initramfs-tools/scripts/init-bottom
/etc/initramfs-tools/scripts/init-premount
/etc/initramfs-tools/scripts/init-top
/etc/initramfs-tools/initramfs.conf
/etc/init
root@ubuntu:~# find /etc -name init??
/etc/init.d
root@ubuntu:~#
```

1.2.2 -size 选项

`find /etc -size` 文件大小

他是以数据块为单位的！512 字节 = 0.5KB, 1K = 2Blocks

100M=? blocks 100M = 102400K = 102400*2blocks

`find /etc -size +204800` #在/etc 下查找大于 80MB 大于 100MB 的文件

`find /etc -size -204800` #在/etc 下查找大于 80MB 小于 100MB 的文件

`find /etc -size 204800` #在/etc 下查找等于 100MB 的文件,不常用!

`find /etc -size +10k` #在/etc 下查找大于 10K 的文件

```
root@ubuntu:~# find /etc/ -size +100k
/etc/apparmor.d/cache/usr.bin.webbrowser-app
/etc/apparmor.d/cache/usr.bin.evince
/etc/ssh/moduli
/etc/vmware-tools/locations
/etc/ssl/certs/ca-certificates.crt
/etc/brltty/Contraction/zh-tw-ucb.ctb
/etc/brltty/Contraction/ko.ctb
/etc/brltty/Contraction/zh-tw.ctb
/etc/ImageMagick-6/mime.xml
root@ubuntu:~#
```

1.2.3 -user 选项

`find /etc -user username` 查找属于 `username` 的文件

`find /home/luohuihua/Programme/Data_Structure/ -user luohuihua`

```
root@ubuntu:/home/luohuihua# pwd
/home/luohuihua
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  下载  公共的  图片  文档  桌面  模板  视频  音乐
root@ubuntu:/home/luohuihua# touch test.txt
root@ubuntu:/home/luohuihua# ls -l test.txt
-rw-r--r-- 1 root root 0 6月 28 11:03 test.txt
root@ubuntu:/home/luohuihua# find /home/luohuihua -user root
/home/luohuihua/test.txt
root@ubuntu:/home/luohuihua#
```

1.2.4 时间 选项

以天为单位: `ctime`, `atime`, `mtime`

以分钟为单位: `cmin`, `amin`, `mmin` #更为常用

- `c-change` 改变: 表示文件的属性被修改过, 比如: 所有者, 所属组, 权限
- `access` 访问: 文件被浏览过
- `m-modify` 修改: 文件内容被修改过

- 多长时间之内

+ 超过多少时间

`find /etc -mtime -1 //查找/etc 目录下一天之内文件内容被修改过的文件`

`find /etc -amin -60` //查找/etc 目录下 60 分钟内被浏览过的文件

`find /home -cmin -120` //查找/home 目录下 2 小时内文件属性属性被修改过的文件

```
root@ubuntu:/home/luohuihua# mkdir test
root@ubuntu:/home/luohuihua# cd test
root@ubuntu:/home/luohuihua/test# touch test.txt
root@ubuntu:/home/luohuihua/test# ls
test.txt
root@ubuntu:/home/luohuihua/test# find ./ -cmin -10
./
./test.txt
root@ubuntu:/home/luohuihua/test# find ./ -cmin +10
root@ubuntu:/home/luohuihua/test#
```

1.2.5 -type 选项

`find /etc -type` 文件类型 #根据文件类型查找

文件类型:

1. f 二进制文件
2. l 软链接文件
3. d 目录

`find /etc -type f`

`find /etc -type l`

```
root@ubuntu:/home/luohuihua# pwd
/home/luohuihua
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  test  test.txt  下载  公共的  图片  文档  桌面
模板  视频  音乐
root@ubuntu:/home/luohuihua# find ./ -maxdepth 1 -type f
./bashrc
./bash_history
./sudo_as_admin_successful
./pam_environment
./xsession-errors
./profile
./ICEauthority
./bash_logout
./test.txt
./Xauthority
./dmmrc
./xsession-errors.old
./examples.desktop
./xinputrc
root@ubuntu:/home/luohuihua#
```

1.2.6 -inum 选项

-inum #根据 i 节点进行查找

touch abc #删除:rm abc

touch “a b” #删除:rm “a b”

find . -inum 159341

find . -inum 159341 -exec rm -f {} \; #找到 i 节点为 159341 的文件，并删除它

```
root@ubuntu:/home/luohuihua# pwd
/home/luohuihua
root@ubuntu:/home/luohuihua# ls -li
总用量 52
2361647 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 11:53 Downloads
2378249 -rw-r--r-- 1 luohuihua luohuihua 8980 4月 24 16:52 examples.desktop
2491109 drwxr-xr-x 2 root      root      4096 6月 28 11:25 test
2364019 -rw-r--r-- 1 root      root        0 6月 28 11:24 test.txt
2490875 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 下载
2490978 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 公共的
2491100 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 图片
2491098 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 文档
2490744 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 桌面
2490958 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 模板
2491101 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 视频
2491099 drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 音乐
root@ubuntu:/home/luohuihua# find . -inum 2361647
./Downloads
root@ubuntu:/home/luohuihua#
```

1.2.7 -group 选项

-group 组名 #根据用户组的名称查找

find . -group luohuihua #查找当前目录下属于 luohuihua 用户组的所有文件


```
root@ubuntu:/home/luohuihua# pwd
/home/luohuihua
root@ubuntu:/home/luohuihua# ls -l
总用量 52
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 11:53 Downloads
-rw-r--r-- 1 luohuihua luohuihua 8980 4月 24 16:52 examples.desktop
drwxr-xr-x 2 root      root      4096 6月 28 11:25 test
-rw-r--r-- 1 root      root        0 6月 28 11:24 test.txt
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 下载
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 公共的
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 图片
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 文档
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 桌面
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 模板
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 视频
drwxr-xr-x 2 luohuihua luohuihua 4096 5月 7 12:42 音乐
root@ubuntu:/home/luohuihua# find . -maxdepth 1 ! -name ".*" -group luohuihua
./Downloads
./音乐
./公共的
./图片
./视频
./文档
./桌面
./模板
./下载
./examples.desktop
root@ubuntu:/home/luohuihua#
```

1.2.8 逻辑与、逻辑或 选项

-a: and 逻辑与

-o: or 逻辑或

find /etc -size +163840 -a -size 204800 #查找>80M,<100M 的文件

find /etc -name init* -a -type f #查找名为 init 并且为二进制的文件,并不包含目录

```
root@ubuntu:/home/luohuihua# find /etc -maxdepth 1 -size +10k -a -size -20k
/etc
/etc/sensors3.conf
/etc/login.defs
/etc/ltrace.conf
root@ubuntu:/home/luohuihua#
```

1.2.9 -exec 选项

find ... -exec 命令 {} \; #固定格式,只能这样来写

{} :find 的查询结果

\ :转义符-使得符号命令使用本身的含义

;; :语句结束

find /etc -name inittab -exec ls -l {} \; #在/etc 下查找 inittab 文件并显示其详细信息

find /home -user sax -exec rm -rf {} \; #删除用户 sax 所有的文件

find /home -user sax -ok rm -rf {} \; # -ok 连接符删除用户 sax 所有的文件,他会提示你是否确认

find /etc -name init* -ok rm -rf {} \;

-ok 和-exec 行为一样, 不过它会给出提示, 是否执行相应的操作

```
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  test  test.txt  下载  公共的  图片  文档  桌面
模板  视频  音乐
root@ubuntu:/home/luohuihua# find . -name test.txt
./test/test.txt
./test.txt
root@ubuntu:/home/luohuihua# find . -name test.txt -exec ls -l {} \;
-rw-r--r-- 1 root root 0 6月 28 11:25 ./test/test.txt
-rw-r--r-- 1 root root 0 6月 28 11:24 ./test.txt
root@ubuntu:/home/luohuihua# find . -name test.txt -exec rm -rf {} \;
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  test  下载  公共的  图片  文档  桌面  模板  视频
音乐
root@ubuntu:/home/luohuihua#
```

1.3、locate 命令

locate[list file in databases]: /usr/bin/locate

说明:

- 是 Linux 所特有的命令,寻找文件或目录, 最好用于快速定位系统命令, 配置文件等
- 虽然搜索速度很快, 但有时候会找不到
- locate 是在文件数据库中查找的, 所以速度会很快
- 但是如果数据库没有包含这个文件的话, 他就会找不到;

格式:locate [搜索关键字] 配合:update db [update a database for mlocate]:
/usr/bin/update db 建立整个系统目录文件的数据库 注意: 其执行权限为: root!!!

```
luohuihua@ubuntu:~$ locate /etc/sh
/etc/shadow
/etc/shadow-
/etc/shells
luohuihua@ubuntu:~$
```

当在某些目录下创建文件, 然后更新数据库之后, 并不能用 locate 命令查找到

原因是系统在更新数据库的配置文件中, 设置了一些搜索限制, 所以搜索不到, 输入如下命令可以看到:

```
[root@localhost ~]# vi /etc/update db.conf
```

PRUNE_BIND_MOUNTS = “yes” 表示开启搜索限制, 如果为’ no’ 则表示不开启搜索限制;

PRUNEFS = 表示搜索时, 不搜索的文件系统;

PRUNENAMES = 表示搜索时, 不搜索的文件类型;

PRUNEPATHS = 表示搜索时, 不搜索的路径;

不只 locate 命令遵循搜索限制, whereis 与 which 也遵循

1.4、grep 命令

grep 命令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。

grep “python” filename 在 filename 文件中查找 python, 并且将结果打印出来

grep "python" filename filename1 filename2 在多个文件中查找 python, 并且将结果打印出来

-E 选项使用正则表达式

grep -E "[a-c]+" filename 加上-E 选项可以使用正则表达式。

```
root@ubuntu:/home/luohuihua# grep home /etc/passwd
syslog:x:104:108::/home/syslog:/bin/false
luohuihua:x:1000:1000:luohuihua,,,:/home/luohuihua:/bin/bash
root@ubuntu:/home/luohuihua# grep -E "(home)|(sbin)" /etc/passwd
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:104:108::/home/syslog:/bin/false
luohuihua:x:1000:1000:luohuihua,,,:/home/luohuihua:/bin/bash
sshd:x:121:65534::/var/run/sshd:/usr/sbin/nologin
root@ubuntu:/home/luohuihua#
```

在多个文件中查找 `grep "match_pattern" file_1 file_2 file_3 ...`

```
root@ubuntu:/home/luohuihua# grep luohuihua /etc/passwd /etc/group
/etc/passwd:luohuihua:x:1000:1000:luohuihua,,,:/home/luohuihua:/bin/bash
/etc/group:adm:x:4:syslog,luohuihua
/etc/group:cdrom:x:24:luohuihua
/etc/group:sudo:x:27:luohuihua
/etc/group:dip:x:30:luohuihua
/etc/group:plugdev:x:46:luohuihua
/etc/group:lpadmin:x:113:luohuihua
/etc/group:luohuihua:x:1000:
/etc/group:sambashare:x:128:luohuihua
root@ubuntu:/home/luohuihua#
```

输出包含匹配字符串的行数 `-n` 选项:

```
root@ubuntu:/home/luohuihua# grep luohuihua -n /etc/passwd /etc/group
/etc/passwd:40:luohuihua:x:1000:1000:luohuihua,,,:/home/luohuihua:/bin/bash
/etc/group:5:adm:x:4:syslog,luohuihua
/etc/group:18:cdrom:x:24:luohuihua
/etc/group:21:sudo:x:27:luohuihua
/etc/group:23:dip:x:30:luohuihua
/etc/group:35:plugdev:x:46:luohuihua
/etc/group:52:lpadmin:x:113:luohuihua
/etc/group:67:luohuihua:x:1000:
/etc/group:68:sambashare:x:128:luohuihua
```

统计文件或者文本中包含匹配字符串的行数 `-c` 选项:

```
root@ubuntu:/home/luohuihua# grep luohuihua -c /etc/passwd /etc/group
/etc/passwd:1
/etc/group:8
root@ubuntu:/home/luohuihua#
```

1.5、 | 管道符的使用

| 管道符的作用，将左边的输出当右边的输入

ps -ajx |grep ssh 将 ps 命令的输出结果当 grep 的输入过滤。

```
root@ubuntu:/home/luohuihua# ps -ajx | grep ssh
  1   1281   1281   1281 ?          -1 Ss      0   0:00 /usr/sbin/sshd -D
1281  3162  3162  3162 ?          -1 Ss      0   0:00 sshd: luohuihua [priv]
3162  3191  3162  3162 ?          -1 S      1000   0:03 sshd: luohuihua@pts/0
4403  7495  7494  3192 pts/0      7494 S+      0   0:00 grep --color=auto ssh
```

显示 /etc 目录下的文件或目录的总数量

```
root@ubuntu:/home/luohuihua# ll /etc | grep -E '".*"' -c
235
```

1.6、重定向 >,追加重定向 >>

> 将输出重定向到文件

>> 将输出重定向追加到文件

ls > test.txt 将 ls 命令的输出重定向到 test.txt 文件

ls >> test.txt 将 ls 命令的输出追加到 test.txt

```
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  test  下载  公共的  图片  文档  桌面  模板  视频
音乐
root@ubuntu:/home/luohuihua# ls > test.txt
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  test  test.txt  下载  公共的  图片  文档  桌面
模板  视频  音乐
root@ubuntu:/home/luohuihua#
```

test.txt 的内容如下：

```
root@ubuntu:/home/luohuihua# cat test.txt
```

Downloads

examples.desktop

test

test.txt

下载

公共的

图片

文档

桌面

模板

视频

音乐

追加内容

```
root@ubuntu:/home/luohuihua# echo "好的" >> test.txt
```

```
root@ubuntu:/home/luohuihua# cat test.txt
```

Downloads

examples.desktop

test

test.txt

下载

公共的

图片

文档

桌面

模板

视频

音乐

好的

输入重定向 <

```
root@ubuntu:/home/luohuihua# ls
```

Downloads examples.desktop test test.txt 下载 公共的 图片 文档 桌面
模板 视频 音乐

```
root@ubuntu:/home/luohuihua# cat test.txt
```

/etc

```
root@ubuntu:/home/luohuihua# ls < test.txt
```

Downloads examples.desktop test test.txt 下载 公共的 图片 文档 桌面
模板 视频 音乐

2、文件打包与压缩

2.1、文件打包

`tar -cvf` 打包文件名.tar 被打包的文件/路径

-c 创建一个新的归档

-v 详细地列出处理的文件

-f 使用归档文件

```
root@ubuntu:/home/luohuihua# tar -cvf luohuihua.tar ./
root@ubuntu:/home/luohuihua# ls
Downloads  examples.desktop  luohuihua.tar  test  test.txt  下载  公共的  图片  文档  桌面  模板  视频  音乐
```

2.2、解包文件

`tar -xvf` 打包过的文件.tar

-x 将打包过的文件解包

提示：-f 选项必须放在最后面

```
root@ubuntu:/home/luohuihua/test# pwd
/home/luohuihua/test
root@ubuntu:/home/luohuihua/test# ls
test.txt
root@ubuntu:/home/luohuihua/test# tar -cvf test.tar .
./
tar: ./test.tar: 文件是归档文件；未输出
./test.txt
root@ubuntu:/home/luohuihua/test# ls
test.tar  test.txt
root@ubuntu:/home/luohuihua/test# rm -rf test.txt
root@ubuntu:/home/luohuihua/test# ls
test.tar
root@ubuntu:/home/luohuihua/test# tar -xvf test.tar
./
./test.txt
root@ubuntu:/home/luohuihua/test# ls
test.tar  test.txt
```

2.3、gzip 压缩

gzip 一般跟 tar 一起使用，完成打包压缩

tar 只负责打包并为做压缩，使用 -z 选项可以调用 gzip 压缩，完成打包压缩。

使用 tar 打包压缩的文件名，一般命名成 xxx.tar.gz 区别与其他文件。

压缩文件：

```
tar -zcvf py.tar.gz a.txt b.txt c.txt
```

解压缩文件：

```
tar -zxvf py.tar.gz
```

解压缩到指定路径

```
tar -zxvf 打包文件.tar.gz -C 目标路径
```

-C 指定解压到那里，解压的目录必须存在


```
root@ubuntu:/home/luohuihua/test# pwd
/home/luohuihua/test
root@ubuntu:/home/luohuihua/test# ls
a.txt  b.txt  c.txt
root@ubuntu:/home/luohuihua/test# tar -zcvf test.tar.gz a.txt b.txt c.txt
a.txt
b.txt
c.txt
root@ubuntu:/home/luohuihua/test# ls
a.txt  b.txt  c.txt  test.tar.gz
```

```
root@ubuntu:/home/luohuihua/test# pwd
/home/luohuihua/test
root@ubuntu:/home/luohuihua/test# ls
test.tar.gz
root@ubuntu:/home/luohuihua/test# tar -zxvf test.tar.gz
a.txt
b.txt
c.txt
root@ubuntu:/home/luohuihua/test# ls
a.txt  b.txt  c.txt  test.tar.gz
```

2.4、bzip2 压缩

bzip2 使用方式跟 gzip 差不多，也是由 tar 去调用，使用-j 选项

bzip2 压缩的文件命名采用 xxx.tar.bz2

压缩文件：

```
tar -jcvf py.tar.bz2 a.txt b.txt c.txt
```

```
root@ubuntu:/home/luohuihua/test# pwd
/home/luohuihua/test
root@ubuntu:/home/luohuihua/test# ls
a.txt  b.txt  c.txt
root@ubuntu:/home/luohuihua/test# tar -jcvf test.tar.bz2 a.txt b.txt c.txt
a.txt
b.txt
c.txt
root@ubuntu:/home/luohuihua/test# ls
a.txt  b.txt  c.txt  test.tar.bz2
```

```
tar -jxvf test.tar.bz2
```

```
root@ubuntu:/home/luohuihua/test# ls
test.tar.bz2
root@ubuntu:/home/luohuihua/test# tar -jxvf test.tar.bz2
a.txt
b.txt
c.txt
root@ubuntu:/home/luohuihua/test# ls
a.txt b.txt c.txt test.tar.bz2
```

3、ubuntu 软件安装

3.1、apt 是 ubuntu 下的包管理工具，可很方便的安装、卸载、更新。

安装软件包：

```
sudo apt install 软件包
```

卸载软件包：

```
sudo apt remove 软件包
```

更新已安装的包：

```
sudo apt upgrade
```

小火车安装演示：

```
sudo apt install sl
```

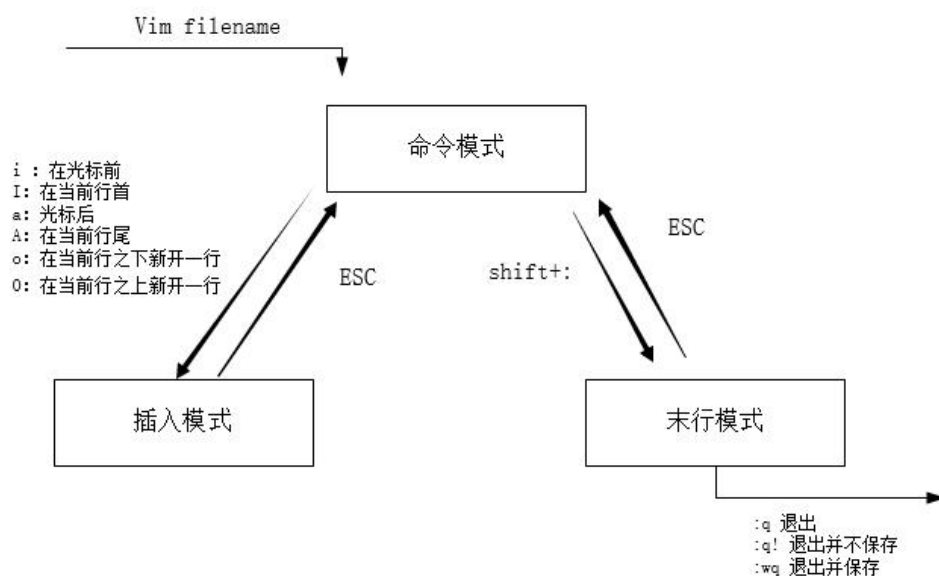
4、vim 编辑器的使用

vim linux 下的文本编辑器，可以执行输出、删除、查找、替换、块操作等众多文本操作。vim 没有菜单，只有命令，且命令繁多。

4.1、vim 有三种模式

- 命令模式
- 输入模式
- 末行模式

4.2 、三种模式切换与关系:



4.3、命令模式:

使用 `vim` 打开文件的时候，就进到命令模式。

`vim filename` : 打开或新建文件，并将光标置于第一行首

`vim+n filename` : 打开文件，并将光标置于第 `n` 行首

`vim + filename` : 打开文件，并将光标置于最后一行首

`vim +/pattern filename`: 打开文件，并将光标置于第一个与 `pattern` 匹配的串处

`vim -r filename` : 在上次正用 `vi` 编辑时发生系统崩溃，恢复 `filename`

`vim filename....filename` : 打开多个文件，依次进行编辑

命令模式下只能输入命令不能进行编辑，只有进入输入模式才能做文件编辑。

4.4、移动光标类命令:

`h` : 光标左移一个字符

`l` : 光标右移一个字符

`k` 或 `Ctrl+p`: 光标上移一行

j 或 Ctrl+n : 光标下移一行

w 或 W : 光标右移一个字至字首

b 或 B : 光标左移一个字至字首

0: (注意是数字零) 光标移至当前行首

\$: 光标移至当前行尾

Ctrl+f 向文件尾翻一屏

Ctrl+b 向文件首翻一屏

home 移动至行首

end 移动至行末

G 跳转至文档最末尾

gg 跳转至文档首行

nG n 是一个数字, 表示跳转至第几行。

4.5、删除复制命令

dd 删除光标所在行

do 删除光标所在行光标前面的内容(数字 0)

d\$ 删除光标所在行光标末尾的内容

3 dd 删除光标所在行包含当前行后面 3 行内容 (数字可以替换成自己想要删除的行数多少)

yy 复制

2 yy 复制光标所在行开始 2 行内容

p 粘贴, 删除的内容也可以直接用 p 粘贴

u 撤销

ctrl r 反撤销

4.6、进入输入模式

插入文本类命令:

i : 在光标前

I : 在当前行首

a: 光标后

A: 在当前行尾

o: 在当前行之下新开一行

O: 在当前行之上新开一行

4.7、进入末行模式

在命令模式下，用户按":"键即可进入末行模式

退出及保存：

:q 退出

:q! 退出并不保存

:w 保存

:wq 退出并保存

:x 退出并保存

查找替换：

/string 在文本中查找 string

n 下一个

N 上一个

:n1,n2s /word1/word2/g 将 n1 行到 n2 行中 word1 替换成 word2, g 表示全部替换，不加 g 则只替换匹配中的第一个

:1,\$s /word1/word2/g 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 (常用)

:1,\$s /word1/word2/gc 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 ！且在取代前显示提示字符给用户确认 是否需要替换(常用)

4.8、显示行号

: set nu 显示行号

: set nonu 取消显示行号

4.9、可视模式

v:按字符移动,选中文本

V:按行移动,选中文本可视模式可以配合 d,y,>>,<<实现对文本块的删除,复制,左右移动

[搜索选项]: