

Linux文件管理（上）

学习目标

- 1、了解文件命名规则和工作中的建议命名规则
- 2、会创建和删除目录mkdir/rmdir
- 3、会创建和删除文件touch/rm
- 4、了解复制cp和移动mv的区别会使用tar命令进行压缩和解压缩
- 5、掌握VIM的保存退出wq和不保存强制退出q!掌握VIM的快捷键yy,dd,gg,G,u
- 6、会使用tail命令查看文件
- 7、会使用find命令按文件名称查找文件

一、文件命名规则

1、可以使用哪些字符？

除了字符“/”之外，所有的字符都可以使用，但是要注意，在目录名或文件名中，不建议使用某些特殊字符，例如，<、>、?、*等，尽量避免使用。

如果一个文件名中包含了特殊字符，例如空格，那么在访问这个文件时就需要使用引号将文件名括起来。

建议文件命名规则：

由于linux严格区分大小写，所以尽量都用小写字母

如果必须对文件名进行分割，建议使用“_”，例如：itheima_bj_2020.log

a.txt

001.txt

002.txt

tongxunlu.txt

tongxunlu_bj_caiwu.txt

2、文件名的长度

目录名或文件名的长度不能超过 255 个字符

3、Linux文件名大小写

Linux目录名或文件名是区分大小写的。如 itheima、ITheima、yunwei 和 Yunwei，是互不相同的目录名或文件名。

不要使用字符大小写来区分不同的文件或目录。

建议文件名==一律使用小写字母==

4、Linux文件扩展名

Linux文件的扩展名对 Linux 操作系统没有特殊的含义，Linux 系统并不以文件的扩展名开分区文件类型。例如，itheima.exe 只是一个文件，其扩展名 .exe 并不代表此文件就一定是可执行的。

在Linux系统中，文件扩展名的用途为了==使运维人员更好的区分不同的文件类型==。

二、文件管理命令

在日常工作中，我们经常需要对Linux的文件或目录进行操作，常见操作包括新建，删除，更改，查看，复制，移动等。

1、目录创建/删除

在实际应用中，与目录相关的操作主要有两个：创建目录与删除目录

① mkdir创建目录

命令：mkdir (make directory, 创建目录)

作用：创建目录

语法：#mkdir [参数选项] 路径（包含目录名）

常见参数：

-p：递归创建所有目录,如果想创建多层不存在的路径，可以使用-p参数实现。-p表示parents，父级的意思

用法一：mkdir 不加参数，路径（需要包含目录名称）

示例代码：

```
#mkdir /usr/local/nginx
```

含义：在/usr/local目录下，创建一个文件夹名为nginx

特别注意：mkdir命令默认不能隔级创建目录，必须要求要创建的目录所在的目录一定要存在

用法二：mkdir 加-p参数，路径（需要包含目录名称）

示例代码：

```
#mkdir -p /usr/local/itheima/tomcat
```

含义：在/usr/local目录下，同时创建itheima文件夹和itheima下的子文件夹tomcat

```
File Edit View Search Terminal Help
[root@localhost local]# mkdir /usr/local/itheima/tomcat
mkdir: cannot create directory '/usr/local/itheima/tomcat': No such file or directory
[root@localhost local]# mkdir -p /usr/local/itheima/tomcat
[root@localhost local]# ls
bin  games  itheima  lib64  nginx  share  VMwareTools-10.0.5-3228253.tar.gz
etc  include lib  libexec sbin  src  vmware-tools-distrib
[root@localhost local]# cd itheima/
[root@localhost itheima]# ls
tomcat
[root@localhost itheima]#
```

错误信息：cannot create directory '/usr/local/itheima/tomcat':No such file or directory

含义：不能建立目录'XXX'：没有文件或文件夹

问题：为什么会有这个错误信息？

答：

用法三：mkdir 路径1 路径2 路径3

示例代码：

```
#mkdir /usr/local/a /usr/local/b /usr/local/c
```

含义：在/usr/local目录下，同时创建a,b,c文件夹

```
File Edit View Search Terminal Help
[root@localhost local]# ls
bin  games  itheima  lib64  nginx  share  VMwareTools-10.0.5-3228253.tar.gz
etc  include lib  libexec sbin  src  vmware-tools-distrib
[root@localhost local]# mkdir /usr/local/a /usr/local/b /usr/local/c
[root@localhost local]# ls
a  c  include  lib64  sbin  VMwareTools-10.0.5-3228253.tar.gz
b  etc  itheima  libexec share  vmware-tools-distrib
bin  games  lib  nginx  src
[root@localhost local]#
```

mkdir命令总结

```
#mkdir /mydata    在根目录下建立mydata文件夹
```

```
#mkdir -p /itcast/tbd    一次性建立多级文件夹/itcast/tbd
```

```
#mkdir /tbd /jinyanlong /shunyi
```

问题：一台刚刚安装好的Centos系统，小明想建立/xiaoming/zuoye/shuxue，应该使用mkdir还是mkdir -p？

答案A mkdir

答案B mkdir -p

② 删除目录

命令：rmdir (remove directory缩写)

作用：删除==空==目录

语法：#rmdir [参数选项] 路径（包含目录名）

常见参数：

-p：递归删除所有空目录

用法一：rmdir 不加参数，路径（需要包含目录名称）

示例代码：

```
#rmdir /usr/local/nginx
```

含义：删除/usr/local下的空目录nginx

用法二：rmdir 路径1 路径2 路径3

示例代码：

```
#rmdir /usr/local/a /usr/local/b /usr/local/c
```

含义：同时删除a,b,c三个空目录

```
File Edit View Search Terminal Help
[root@localhost local]# ls
a   etc      lib      share
b   games   lib64    src
bin include  libexec  VMwareTools-10.0.5-3228253.tar.gz
c   itheima  sbin     vmware-tools-distrib
[root@localhost local]# rmdir /usr/local/a /usr/local/b /usr/local/c
[root@localhost local]# ls
bin      itheima  sbin                                vmware-tools-distrib
etc      lib      share
games    lib64    src
include  libexec  VMwareTools-10.0.5-3228253.tar.gz
[root@localhost local]#
```

用法三：rmdir -p 路径（需要包含目录名称）

示例代码：

```
#rmdir -p itheima/tomcat
```

含义：递归删除目录，删除tomcat和itheima目录

首先删除子目录，删除成功后，删除上级目录，直至结束。

问题：rmdir -p itheima/tomcat 中，“itheima/tomcat”是绝对路径还是相对路径？

答：A 绝对

B ==相对==

错误信息：rmdir: failed to remove directory '/usr/local/itheima': Directory not empty

含义：

rmdir命令总结

```
#rmdir /itcast    删除根目录下的itcast目录

#rmdir /itcast /jinyanlong /shunyi    同时删除根目录下的itcast jinyanlong hunyi

#rmdir -p /itcast/heima    一次性删除heima和他的上级目录itcast
```

问题：能不能使用绝对路径rmdir -p /usr/local/itheima/tomcat？为什么？

答：

2、文件创建/删除

在实际应用中，与文件相关的操作主要有两个：创建文件与删除文件

① 创建文件

命令：touch

作用：创建文件

语法：# touch 文件路径 [文件路径2 文件路径3 ...]

用法一：touch 路径（包含文件名）

示例代码：

```
#touch readme.txt
```

含义：在当前路径下创建一个文件 readme.txt

```
File Edit View Search Terminal Help
[root@localhost local]# ls
bin  games  lib  libexec  share  VMwareTools-10.0.5-3228253.tar.gz
etc  include lib64 sbin  src  vmware-tools-distrib
[root@localhost local]# touch readme.txt
[root@localhost local]# ls
bin      lib      sbin      vmware-tools-distrib
etc      lib64    share
games    libexec  src
include  readme.txt  VMwareTools-10.0.5-3228253.tar.gz
[root@localhost local]#
```

用法二：touch 路径1（包含文件名） 路径2（包含文件名） 路径3（包含文件名）

示例代码：

```
#touch 1.txt 2.txt 3.txt
```

含义：在当前路径下创建1.txt 2.txt 3.txt三个文件

用法三（了解）：`touch 路径1（包含文件名）{1..n}`

示例代码：

```
#touch {1..5}.txt
```

含义：在当前路径下创建1.txt 2.txt 3.txt 4.txt 5.txt三个文件
其中

1表示开始的数字

..表示连续的

5表示结束的数字

touch命令总结

```
#touch readme.txt
```

 在 当前 目录下创建文件

```
#touch /readme.txt
```

 在 根目录 下创建文件

```
#touch 1.txt 2.txt 3.txt
```

 在 当前 同时创建 1.txt 2.txt 3.txt

```
#touch /usr/local/1.txt /usr/local/2.txt /usr/local/3.txt
```

在/usr/local下创建1.txt 2.txt 3.txt

```
#touch /usr/local/1.txt 2.txt 3.txt
```

```
#touch {1..6}.log
```

在对当前目录下，创建 1到6.txt

1.txt 2.txt 3.txt 4.txt 5.txt 6.txt

思考题

```
#cd /usr
```

```
#touch /usr/local/a.txt b.txt c.txt
```

问题：执行以上两条命令后，请列出a.txt,b.txt,c.txt的绝对路径

② 删除文件

命令：rm（remove缩写）

作用：删除文件或文件夹

语法：rm [参数选项] 文件或文件夹

选项：-r：递归删除，主要用于删除目录，可删除指定目录及包含的所有内容，包括所有子目录和文件

-f：强制删除，不提示任何信息。操作前一定要慎重！！

用法一：rm 路径（包含文件名）

示例代码：

```
#rm readme.txt
```

含义：删除当前路径下文件 readme.txt

提示信息：remove regular empty file 'readme.txt'？

含义：是否要删除普通的空文件'readme.txt'？

用法二：rm -r 路径（通常为目录名）

示例代码：

```
#rm -r itheima
```

含义：删除当前文件夹下的itheima文件夹及其包含的所有子文件夹/文件

提示信息：descend into directory 'itheima/'？

含义：是否进入itheima目录？

提示信息：remvove directory 'itheima/'？

含义：是否删除itheima文件夹？

用法三：rm -rf 路径（通常为目录名）

示例代码：

```
#rm -rf itheima
```

含义：强制删除当前文件夹下的itheima文件夹及其包含的所有子文件夹/文件

```
File Edit View Search Terminal Help
[root@localhost local]# mkdir /usr/local/itheima
[root@localhost local]# touch /usr/local/itheima/a.txt /usr/local/itheima/b.txt
[root@localhost local]# ls /usr/local/
1.txt  etc      lib      share
2.txt  games    lib64    src
3.txt  include  libexec  VMwareTools-10.0.5-3228253.tar.gz
bin    itheima  sbin     vmware-tools-distrib
[root@localhost local]# rm -rf /usr/local/itheima/
[root@localhost local]# ls /usr/local/
1.txt  bin      include  libexec  src
2.txt  etc      lib      sbin     VMwareTools-10.0.5-3228253.tar.gz
3.txt  games    lib64    share    vmware-tools-distrib
[root@localhost local]#
```

rm命令总结

#rm readme.txt 带确认的删除文件

#rm -r shop 带确认的删除目录

#rm -rf shop 不带确认删除目录及其所有子目录和文件

问题：rm -r /usr/local/itheima/ 是否会删除itheima的父文件夹local?

答：A克隆

B快照

安全性和易用性

3、复制与剪切

① 复制操作

命令：cp (copy缩写，复制操作)

作用：复制文件/文件夹到指定的位置

语法：#cp [参数选项] 源路径（含文件名） 目标路径（如不指定文件名，则文件名不变）

常见参数：-r: recursion, 递归，用于复制目录

用法一：cp 源路径 目标路径（不指定文件名）

示例代码：

```
#cp /root/itheima.txt /usr/local/
```

含义：将/root目录下的itheima.txt文件拷贝到/usr/local下，文件名不变

用法二：cp 源路径 目标路径（指定文件名）

示例代码：

```
#cp /root/itheima.txt /usr/local/heimayunwei.txt
```

含义：将/root目录下的itheima.txt文件拷贝到/usr/local下，重命名为heimayunwei.txt

```
File Edit View Search Terminal Help
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  initial-setup-ks.cfg  Music  Public  Videos
Desktop          Downloads  itheima.txt           Pictures  Templates
[root@localhost ~]# cp /root/itheima.txt /usr/local/
[root@localhost ~]#
[root@localhost ~]# ls /usr/local/
1.txt  etc  lib  share
2.txt  games  lib64  src
3.txt  include  libexec  VMwareTools-10.0.5-3228253.tar.gz
bin  itheima.txt  sbin  vmware-tools-distrib
[root@localhost ~]# cp /root/itheima.txt /usr/local/heimayunwei.txt
[root@localhost ~]# ls /usr/local/
1.txt  games  lib64  VMwareTools-10.0.5-3228253.tar.gz
2.txt  heimayunwei.txt  libexec  vmware-tools-distrib
3.txt  include  sbin
bin  itheima.txt  share
etc  lib  src
```


用法三: `cp -r` 源路径 (包含目录名) 目标路径

示例代码:

```
#cp -r /root/chuanzhiboke /usr/local/
```

含义: 将/root目录下的chuanzhiboke目录完整拷贝到/usr/local下

cp命令总结

```
#cp readme.txt /usr/local/ 当前目录下的readme.txt 拷贝到/usr/local下
```

```
#cp readme.txt /usr/local/readme.txt
```

```
#cp readme.txt /usr/local/readhis.txt
```

```
#cp readme.txt readme.aaa
```

```
#cp -r /root/chuanzhiboke /usr/local/
```

问题: 小明在/root下面建立了一个文件, 叫做xiaoming.txt, 小明想把这个文件拷贝到/usr/local下, 文件名不变, 应该用什么命令? 文件名变成xiaohong.txt, 用什么命令

答:

② 剪切操作

命令: `mv` (move, 移动, 剪切)

作用: 可以在不同的目录之间==移动==文件或目录, 也可以对文件和目录进行==重命名==

语法: `#mv [参数] 源文件 目标路径` (不指定文件名)

`mv`与`cp`的区别:

☆ `mv` 与 `cp` 命令不一样, 不管是针对文件还是针对文件夹都不需要加类似 `-r` 的选项。

☆ 默认在移动的过程中文档名称是不变的, 变的是路径

用法一: `mv` 源文件 目标路径 (不指定文件名)

示例代码:

```
#mv readme.txt /usr/local/
```

含义: 将当前目录下的readme.txt文件移动到/usr/local下, 文件名不变

```
File Edit View Search Terminal Help
[root@localhost ~]# pwd
/root
[root@localhost ~]# touch readme.txt
[root@localhost ~]# ls
anaconda-ks.cfg  Documents          itheima.txt  Public  Videos
chuanzhiboke     Downloads          Music       readme.txt
Desktop         initial-setup-ks.cfg Pictures      Templates
[root@localhost ~]# mv readme.txt /usr/local/
[root@localhost ~]# ls
anaconda-ks.cfg  Documents          itheima.txt  Public
chuanzhiboke     Downloads          Music       Templates
Desktop         initial-setup-ks.cfg Pictures      Videos
[root@localhost ~]# ls /usr/local/
1.txt      etc      lib      share
2.txt      games    lib64     src
3.txt      heimayunwei.txt libexec    VMwareTools-10.0.5-3228253.tar.gz
bin        include  readme.txt vmware-tools-distrib
chuanzhiboke itheima.txt sbin
```

用法二：mv 源目录 目标路径（不指定目录名）

示例代码：

```
#mv shop /usr/local/
```

含义：将当前目录下的shop目录移动到/usr/local下，目录名不变

③ 重命名操作

命令：mv（move，移动，剪切）

语法：#mv [参数] 源文件 目标路径（指定文件名）

用法三：mv 当前文件名 新文件名

示例代码：

```
#mv itheima.txt heimacxy.txt
```

含义：将当前目录下的itheima.txt文件重命名成 heimacxy.txt

前面我们说过，默认情况下文件名称不变，但如果我们指定目标文件名称，就变成了重命名操作。

```
File Edit View Search Terminal Help
[root@localhost ~]# ls
anaconda-ks.cfg  Documents          itheima.txt  Public
chuanzhiboke     Downloads          Music       Templates
Desktop         initial-setup-ks.cfg Pictures      Videos
[root@localhost ~]# mv itheima.txt heimacxy.txt
[root@localhost ~]# ls
anaconda-ks.cfg  Documents          initial-setup-ks.cfg  Public
chuanzhiboke     Downloads          Music                Templates
Desktop         heimacxy.txt      Pictures             Videos
```

用法四：mv 当前目录名 新目录名

示例代码：

```
#mv shop shopping
```

含义：将当前目录下的shop目录重命名成 shopping

在Linux 中重命名的命令也是mv，语法和移动语法一样。区别在于重命名的话一般是路径不变，名称改变。而移动是名字不变，路径变。

mv命令总结

```
#mv readme.txt /tmp/      移动 当前 目录下的readme.txt 到 /tmp目录下，文件名不变

#mv /root/shop /tmp      移动root目录下的 shop目录 到 /tmp目录下

#mv hello.txt readme.txt

#mv /root/shop1 /usr/local/shop2
```

新建目录的时候，不要加点，或者说，不要建立一个类似 shop.txt这样的目录

4、tar打包压缩与解压缩

打包，指的是一个文件或目录的集合，而这个集合被存储在一个文件中。归档文件没有经过压缩，占用的空间是其中**所有文件和目录的总和**。

1.txt 5MB

2.txt 10MB

3.txt 20MB

123.tar 35MB

压缩文件也是一个文件和目录的集合，且这个集合也被存储在一个文件中，但它们的不同之处在于，压缩文件所占用的磁盘空间**比集合中所有文件大小的总和要小**。

123.tar.gz 不是35MB，是压缩后的大小。

1) 打包

命令：tar

作用：将多个文件打包成一个文件

语法：tar 选项 打包文件名 要打包的文件或目录

常见参数：-c, create 创建的意思

-v, 显示打包文件过程

-f, 指定打包的文件名，此参数是必须加的。

-u, update缩写，更新原打包文件中的文件（了解）

-t, 查看打包的文件内容（了解）

注意：

在使用 tar 命令指定选项时可以不在选项前面输入“-”。例如，使用“cvf”选项和“-cvf”起到的作用一样。

使用 tar 命令归档的包通常称为 tar 包（tar 包文件都是以“.tar”结尾的）。

.tar给谁看的？

A centos

B 运维人员

用法一：tar -cvf 文件名 文件1 文件2 文件3

示例代码：

#tar -cvf abc.tar a.txt b.txt c.txt

含义：将当前目录下的a.txt b.txt c.txt 打包成abc.tar文件，大小是三个文件的总合

重点掌握CVF就可以了

```
File Edit View Search Terminal Help
[root@localhost local]# ls -lh
total 104M
-rw-r--r--. 1 root root    0 Feb  8 16:09 1.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 2.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 3.txt
-rw-r--r--. 1 root root  5.0M Feb  9 11:48 a.txt
drwxr-xr-x. 2 root root    6 Apr 11 2018 bin
-rw-r--r--. 1 root root  10M Feb  9 11:48 b.txt
drwxr-xr-x. 2 root root   36 Feb  8 17:19 chuanzhiboke
-rw-r--r--. 1 root root  20M Feb  9 11:48 c.txt
```

```
File Edit View Search Terminal Help
[root@localhost local]# tar -cvf abc.tar a.txt b.txt c.txt
a.txt
b.txt
c.txt
[root@localhost local]# ls -lh
total 139M
-rw-r--r--. 1 root root    0 Feb  8 16:09 1.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 2.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 3.txt
-rw-r--r--. 1 root root  36M Feb  9 11:53 abc.tar
-rw-r--r--. 1 root root  5.0M Feb  9 11:48 a.txt
```

用法二：tar -uf 现有包文件名 要追加的文件

示例代码：

#tar -uf abc.tar d.txt

含义：将当前目录下的d.txt 追加到abc.tar文件，大小是四个文件的总合

```
File Edit View Search Terminal Help
[root@localhost local]# tar -uf abc.tar d.txt
[root@localhost local]# ls -lh
total 233M
-rw-r--r--. 1 root root    0 Feb  8 16:09 1.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 2.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 3.txt
-rw-r--r--. 1 root root  66M Feb  9 11:58 abc.tar
```

用法三: `tar -tf` 包文件名
示例代码:
`#tar -tf abc.tar`
含义: 查看`abc.tar`文件内容

```
File Edit View Search Terminal Help
[root@localhost local]# tar -tf abc.tar
a.txt
b.txt
c.txt
d.txt
[root@localhost local]#
```

2) 打包并压缩 (重点)

Linux下, 常用的压缩工具有很多, 比如 `gzip`、`zip`、`bzip2`、`xz` 等

`tar` 在打包的时候, 是支持压缩的, `gzip`、`bzip2`、`xz` 压缩工具都可以在 `tar` 打包文件中使用。

命令: `tar`

作用: 将多个文件打包并压缩成一个文件, ==其实就是`tar`命令的三个压缩参数==

语法: `tar` 选项 打包文件名 要压缩的文件或目录==

常见参数: ==`-z`, 压缩为`.gz`格式==

==`-j`, 压缩为`.bz2`格式==

==`-J`, 压缩为`.xz`格式==

`-c`, create 创建的意思

==`-x`, 解压缩==

`-v`, 显示打包文件过程

`-f`, file指定打包的文件名, 此参数是必须加的。

`-u`, update缩写, 更新原打包文件中的文件 (了解)

`-t`, 查看打包的文件内容 (了解)

替换c

替换f

```
File Edit View Search Terminal Help
[root@localhost local]# tar -zcvf abc.tar.gz a.txt b.txt c.txt
a.txt
b.txt
c.txt
[root@localhost local]# ls -lh
total 134M
-rw-r--r--. 1 root root 0 Feb 8 16:09 1.txt
-rw-r--r--. 1 root root 0 Feb 8 16:09 2.txt
-rw-r--r--. 1 root root 0 Feb 8 16:09 3.txt
-rw-r--r--. 1 root root 36K Feb 9 13:27 abc.tar.gz
```

注意: 此处打包后的文件名叫做 `abc.tar.gz`, 其中`.gz`表示使用`gzip`压缩的`tar`文件, 目的是方便运维人员识别文件

用法二: `tar -jcvf 文件名 文件1 文件2 文件3`

示例代码:

`#tar -jcvf abc.tar.bz2 a.txt b.txt c.txt`

含义: 将当前目录下的a.txt b.txt c.txt 使用bz2压缩打包成abc.tar.bz2文件, 是压缩后的大小

```
File Edit View Search Terminal Help
[root@localhost local]# tar -jcvf abc.tar.bz2 a.txt b.txt c.txt
a.txt
b.txt
c.txt
[root@localhost local]# ls -lh
total 134M
-rw-r--r--. 1 root root    0 Feb  8 16:09 1.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 2.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 3.txt
-rw-r--r--. 1 root root 174 Feb  9 13:38 abc.tar.bz2
-rw-r--r--. 1 root root 36K Feb  9 13:27 abc.tar.gz
```

用法三: `tar -Jcvf 文件名 文件1 文件2 文件3`

示例代码:

`#tar -Jcvf abc.tar.xz a.txt b.txt c.txt`

含义: 将当前目录下的a.txt b.txt c.txt 使用xz压缩打包成abc.tar.xz文件, 是压缩后的大小

```
File Edit View Search Terminal Help
[root@localhost local]# tar -Jcvf abc.tar.xz a.txt b.txt c.txt
a.txt
b.txt
c.txt
[root@localhost local]# ls -lh
total 134M
-rw-r--r--. 1 root root    0 Feb  8 16:09 1.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 2.txt
-rw-r--r--. 1 root root    0 Feb  8 16:09 3.txt
-rw-r--r--. 1 root root 174 Feb  9 13:38 abc.tar.bz2
-rw-r--r--. 1 root root 36K Feb  9 13:27 abc.tar.gz
-rw-r--r--. 1 root root 5.5K Feb  9 13:41 abc.tar.xz
```

注意: bz2,gzip,xz三种工具的压缩比不同, 实际工作中最常用的是gzip, 换句话说, 大家==最常见到的压缩打包文件是 .tar.gz==

3) 解压

解压的时候, 把压缩命令中的 ==c== 换成 ==x== 即可

用法一: `tar -zxvf 文件名`

示例代码:

`#tar -zxvf abc.tar.gz`

含义: 将abc.tar.gz文件, 解压缩

用法二: `tar -jxvf 文件名`

示例代码:

`#tar -jxvf abc.tar.bz2`

含义: 将abc.tar.bz2文件, 解压缩

用法三: `tar -Jxvf 文件名`
示例代码:
`#tar -Jxvf abc.tar.xz`
含义: 将`abc.tar.xz`文件, 解压缩

通用解压缩参数(记住这个)

用法四: `tar -xvf 文件名`
示例代码:
`#tar -xvf abc.tar.xz`
含义: 系统将自动识别压缩格式, 并自动选择相应工具, 解压缩

```
[root@localhost local]# tar -xvf abc.tar.bz2
a.txt
b.txt
c.txt
[root@localhost local]# tar -xvf abc.tar.gz
a.txt
b.txt
c.txt
[root@localhost local]# tar -xvf abc.tar.xz
a.txt
b.txt
c.txt
```

思考题:

如果我把`abc.tar.gz` 文件名改成`abc.tar.1`

`tar -xvf abc.tar.1`

能不能正确的解压缩

5、zip压缩与解压缩（了解）

1) zip压缩

命令: `zip`

作用: 兼容类unix与windows, 可以压缩多个文件或目录

语法: `# zip [参数] 压缩后的文件 需要压缩的文件(可以是多个文件)`

参数选项: `-r` 递归压缩 (压缩文件夹)

注意: zip压缩默认压缩后的格式就是.zip, 当然也可以加后缀.zip, 一般都加上

==用法一: 文件压缩==

用法一：zip 压缩后的文件名 要压缩的文件（一个或多个）

示例代码：

```
#zip 1.zip 1.txt
```

含义：将1.txt 压缩成1.zip

```
#zip 1dao4.zip 1.txt 2.txt 3.txt 4.txt
```

含义：将1.txt,2.txt,3.txt,4.txt四个文件压缩成一个1dao4.zip文件

```
cndws@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@localhost ~]# zip readme.zip readme.txt  
adding: readme.txt (stored 0%)
```

==用法二：文件夹压缩==

2) unzip解压缩

命令：unzip

作用：解压文件

语法：unzip 要解压的压缩文件 [-d] 解压目录

选项：-d, directory缩写，代表解压文件到指定目录下

==用法一：解压到当前目录==

```
cndws@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@localhost ~]# unzip readme.zip  
Archive:  readme.zip  
extracting: readme.txt
```

==用法二：解压到指定目录==

zip命令总结

zip是压缩

unzip是解压缩

```
#zip 3.zip 3.txt
```

```
#unzip 3.zip
```

```
#unzip 3.zip -d /tmp/
```


三、VIM文件编辑器概述

Vim文本编辑器，是由vi发展演变过来的文本编辑器，使用简单、功能强大、是Linux众多发行版的默认文本编辑器

1、vi编辑器

vi (visual editor) 编辑器通常被简称为vi，它是Linux和Unix系统上最基本的文本编辑器，类似于Windows系统下的notepad（记事本）编辑器。

2、vi与Vim编辑器

Vim(Vi improved)是vi编辑器的加强版, 比vi更容易使用。vi的命令几乎全部都可以在vim上使用。

3、Vim编辑器安装

Centos通常都已经默认安装好了 vi 或 Vim 文本编辑器

当命令行中输入“Vim”显示如下所示的画面时，视为 Vim 安装成功

```
File Edit View Search Terminal Help
```

VIM - Vi IMproved

version 7.4.1099

by Bram Moolenaar et al.

Modified by <bugzilla@redhat.com>

Vim is open source and freely distributable

Help poor children in Uganda!

type :help iccf<Enter> for information

type :q<Enter> to exit

type :help<Enter> or <F1> for on-line help

type :help version7<Enter> for version info

如果在命令行模式下输入“vim”，输出结果为“Command not found”，则表示此系统中未安装Vim。

```
File Edit View Search Terminal Help
[root@localhost local]# vim
bash: vim: command not found...
Similar command is: 'vim'
[root@localhost local]# vim
```

错误信息: command not found...

含义：找不到这条命令，通常表示没有安装这条命令或者可能敲错了命令

如果没有安装，可通过以下命令进行安装

```
#yum install vim
```

后面会详细介绍关于yum的使用，这里大家默认都已经安装了vim。

4、Vim编辑器四种工作模式

Vim 中存在四种模式：

命令模式
编辑模式（输入/插入模式）
可视化模式
末行模式（尾行模式）

①命令模式：使用VIM编辑器时，==默认处于命令模式。==在该模式下可以移动光标位置，可以通过==快捷键==对文件==内容==进行复制、粘贴、删除等操作。

②编辑模式：在该模式下可以对文件的内容进行编辑

③末行模式：可以在==末行输入命令==来对文件进行查找、替换、保存、退出等操作

④可视化模式：可以做一些列选操作

四种模式之间的关系：

```
File Edit View Search Terminal Help
[root@localhost ~]# vim 1.txt
```

四、Vim使用

1、Vim打开文件

命令：vim

作用：编辑文件

语法：vim 文件名

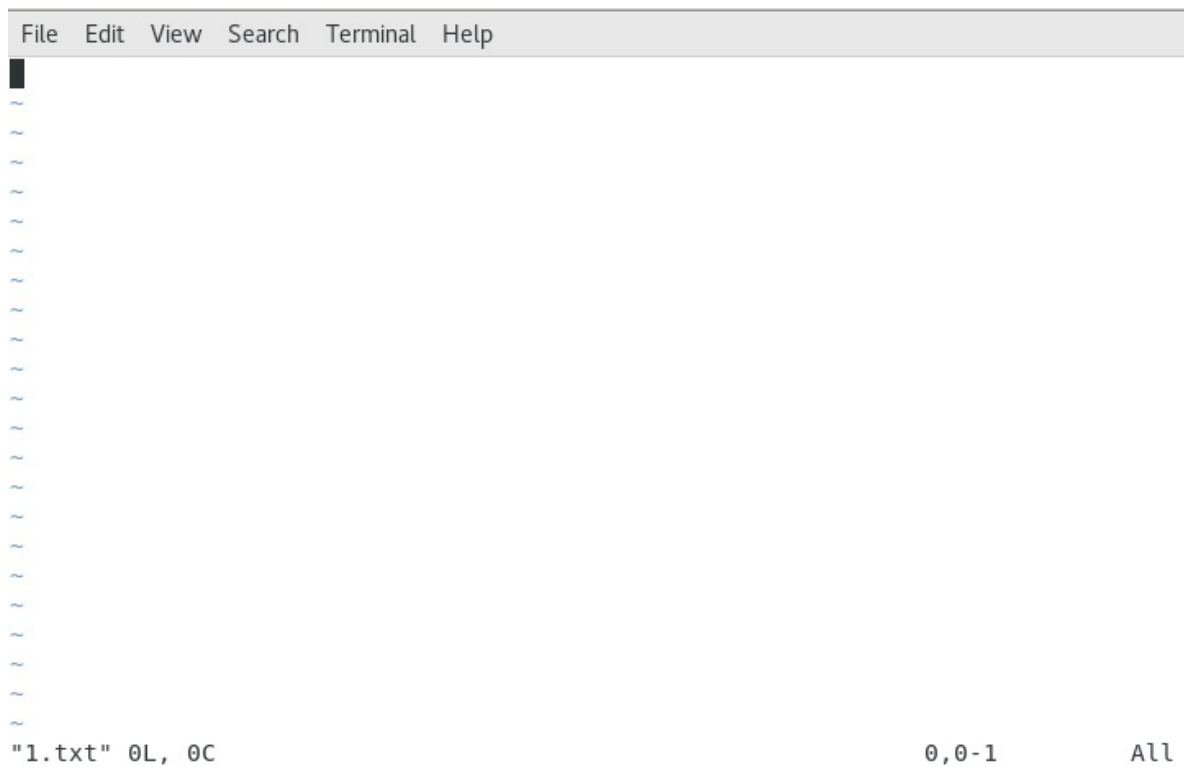
用法一：vim 文件名

示例代码：

```
#vim 1.txt
```

含义：用vim编辑器，打开1.txt文件，如果1.txt文件不存在，则新建一个空文件1.txt，保存退出编辑器时会自动创建这个文件

```
File Edit View Search Terminal Help
[root@localhost ~]# vim 1.txt
```



2、Vim保存文件并退出

在任何模式下，都可以按两下ESC回到命令模式，在命令模式输入:wq 按回车键



3、Vim不保存文件并退出

在任何模式下，都可以按两下ESC回到命令模式，在命令模式输入:q! 按回车键



4、Vim命令模式操作（重点）

4.1、进入命令模式

问：如何进入命令模式？

答：

4.2、光标快速移动操作

☆ 光标移动到首与尾

光标移动到文件第一行的行首，按键： `gg`

光标移动到文件最后一行的行首，按键： `G` [Capslk 再加 `G` 键] / [Shift + `G` 键]

```
#vim /var/log/boot.log-20200219
```

☆ 翻页

向上 翻页，按键： `ctrl + b` (before) 或 `PgUp`

向下 翻页，按键： `ctrl + f` (after) 或 `PgDn`

向上翻半屏，按键： `ctrl + u` (up)

向下翻半屏，按键： `ctrl + d` (down)

☆ 快速定位到指定行（重点）

数字 + `G`

150G

注意：常用于错误定位

```
#vim /var/log/boot.log-20200219
```

4.3、复制

① 复制光标所在行

按键： `yy`

粘贴：在想要粘贴的地方按下 `p` 键【将粘贴在光标所在行的下一行】，如果想粘贴在光标所在行之前，则使用 `P` 键

② 以光标所在行为准（包含当前行），向下复制指定的行数

按键：数字 `yy`，如 `5yy`

```
#vim 1.txt 在其中输入如下，可以用来测试5yy，比较清晰
```

```
File Edit View Search Terminal Help
1
2
3
4
5
a
b
c
d
e
_
```

4.4、剪切/删除

① 剪切/删除光标所在行

按键：dd（删除之后下一行上移）

注意：dd 严格意义上说是剪切命令，但是如果剪切了不粘贴就是删除的效果。

② 剪切/删除光标所在行为准（包含当前行），向下删除/剪切指定的行

按键：数字dd（删除之后下一行上移）

③ 剪切/删除光标所在的当前行（光标所在位置）之后的内容，但是删除之后下一行不上移

按键：D（删除之后当前行会变成空白行）

4.5、撤销/恢复

撤销：u（undo）

恢复：ctrl + r 恢复（取消）之前的撤销操作【重做，redo】

5、Vim末行模式操作（重点）

5.1、进入末行模式

进入方式：由命令模式进入，按下 `:` 或者 `/`（表示查找）即可进入末行模式

退出方式：

① 按下Esc键

② 连按 2 次Esc键

③ 删除末行全部输入字符

那末行模式有哪些作用？我们能使用末行模式做什么呢？

5.2、末行模式相关功能

① 保存操作 (write)

输入: `:w` 保存文件 (了解)

输入: `:w 路径` 另存为 (了解)

```
~  
~  
:w /root/heima.txt
```

② 退出 (quit)

输入: `:q` 退出文件 (了解)

默认情况下, 退出的时候需要对已经进行修改的文件进行保存 `:w`, 然后才能退出

③ 保存并退出 (掌握, 常用)

输入: `:wq` 保存并且退出

```
~  
~  
~  
~  
~  
~  
:wq
```

④ 强制 (!) (掌握, 常用)

输入: `:q!` 表示强制退出, 刚才做的修改操作不做保存

备注: 以后我们在更改系统配置文件时, 很多时候不想保存之前的更改, 甚至我们只想查看, 没想更改。这时候一律使用`q!`退出, 可以保证我们的文件不被误更改

```
~  
~  
~  
~  
~  
~  
:wq
```

⑤ 搜索/查找

输入: `/关键词`, 再按下回车 【按下/也是进入末行模式的方式之一】

在搜索结果中切换上/下一个结果: `N/n` (大写N代表上一个结果, 小写n代表next)

如果需要取消高亮, 则需要末行模式中输入: `:noh` 【no highlight】

⑥ 替换 (了解)

通常在修改配置文件的时候, 我个人很少使用批量替换, 避免替换了一些自己不知道的内容

`:s/搜索的关键词/新的内容` 替换==光标所在行==的第一处符合条件的内容(只替换1次)

```
~
~
:s/l/yi
```

`:s/搜索的关键词/新的内容/g` 替换==光标所在行==的全部符合条件的内容

```
~  搜索的关键词
~  要替换成什么
~
:s/l/yi/g 有g表示替换光标所在行所有的，没有g只替换第一个
```

`:%s/搜索的关键词/新的内容` 替换==整个文档中每行==第一个符合条件的内容

`:%s/搜索的关键词/新的内容/g` 替换==整个文档中所有==符合条件的内容

⑦ 显示行号

输入: `:set nu`, nu代表number

```
File Edit View Search Terminal Help
1 e:yiiyiiyiiyiiyiiyiiyiiyiiyiiyi
2
3 4444444444
4 55555555
5 aaaaaaaa
6 bbbbbbbb
7 cccccccccc
8 dddddddd
9 eeeeeeeee
10
~
~
~
~
:set nu                                     1,2      All
```

如果想取消显示，则输入: `:set nonu`

```
File Edit View Search Terminal Help
eyiiyiiyiiyiiyiiyiiyiiyiiyi
4444444444
55555555
aaaaaaa
bbbbbbb
cccccccc
ddddddd
eeeeeeee
~
~
~
~
:set nonu                                 1,1      All
```

⑧ set paste与set nopaste（了解）

为什么要使用paste模式？

问题：在终端Vim中粘贴代码时，发现插入的代码会有多余的缩进，而且会逐行累加。原因是终端把粘贴的文本存入键盘缓存（Keyboard Buffer）中，Vim则把这些内容作为用户的键盘输入来处理。导致在遇到换行符的时候，如果Vim开启了自动缩进，就会默认地把上一行缩进插入到下一行的开头，最终使代码变乱。

在粘贴数据之前，输入下面命令开启paste模式

```
:set paste
```

粘贴完毕后，输入下面命令关闭paste模式

```
:set nopaste
```

6、编辑模式操作

6.1进入和退出编辑模式

按字母i进入编辑模式，按ESC键退出编辑模式，回到命令模式

7、可视化模式下复制

按键：ctrl + v（可视块）或V（可视行）或v（可视），然后按下↑ ↓ ← →方向键来选中需要

复制的区块，按下y 键进行复制（不要按下yy），最后按下p 键粘贴

退出可视模式按下Esc

1) .添加多行注释：（重点）

A 每行前面都加#

B 把所有行都删了

步骤1：首先按esc进入命令行模式下，按下Ctrl + v，进入列（也叫区块）模式；


```
File Edit View Search Terminal Help
1111111111
222222222222
33333333
4444444444
55555555
aaaaaaa
bbbbbbb
ccccccccc
dddddddd
eeeeeeeeee

~
~
~
~
-- VISUAL BLOCK --                                1,1      All
```

步骤2：在行首使用上下键选择需要注释的多行；

```
1111111111
222222222222
33333333
4444444444
55555555
aaaaaaa
bbbbbbb
ccccccccc
dddddddd
eeeeeeeeee

~
~
~
~
-- VISUAL BLOCK --                                10,1     All
```

步骤3：按下键盘（大写）“I”键，进入插入模式；

```
File Edit View Search Terminal Help
1111111111
222222222222
33333333
4444444444
55555555
aaaaaaa
bbbbbbb
ccccccccc
dddddddd
eeeeeeeeee

~
~
~
~
-- INSERT --                                       1,1      All
```

步骤4：然后输入注释符（“#”）；

```
File Edit View Search Terminal Help
#1111111111
222222222222
33333333
44444444444
55555555
aaaaaaa
bbbbbbb
ccccccccc
ddddddd
eeeeeeeeee
~
~
~
~
-- INSERT --
```

1,2 All

步骤5: 最后按 两下“Esc”键。

```
File Edit View Search Terminal Help
#1111111111
#222222222222
#33333333
#44444444444
#55555555
#aaaaaaa
#bbbbbbb
#ccccccccc
#ddddddd
#eeeeeeeeee
~
~
~
~
1,1 All
```

2) .删除多行注释: (重点)

步骤1: 首先按esc进入命令行模式下, 按下Ctrl + v, 进入列模式;

步骤2: 选定要取消注释的多行的第一列

```
File Edit View Search Terminal Help
#1111111111
#222222222222
#33333333
#44444444444
#55555555
#aaaaaaa
#bbbbbbb
#ccccccccc
#ddddddd
#eeeeeeeeee
~
~
~
~
-- VISUAL BLOCK --
```

ctrl+v进入可视化模式
选中要删除#的多行的第一列, 直接按del键

第一列

10,1 All

步骤3：按del键即可

8、Vim的一些实用功能

8.1、代码着色

之前说过vim 是vi 的升级版，其中比较典型的区别就是vim 更加适合coding，因为vim比vi 多一个代码着色的功能，这个功能主要是为程序员提供编程语言的语法显示效果，如下：

```
#vim index.php
```

在文件中添加以下内容：

```
<?php
    echo 'hello world';
?>
```

末行模式输入:syntax off和syntax on可看到效果



在实际应用中，我们如何控制着色显示与否？

syntax：语法，临时调整

开启显示：:syntax on

关闭显示：:syntax off

8.2、异常退出（重点）

什么是异常退出：在编辑文件之后并没有正常的去wq（保存退出），而是遇到突然关闭终端或者断电的情况，则会显示下面的效果，这个情况称之为异常退出：

```
File Edit View Search Terminal Help

E325: ATTENTION
Found a swap file by the name ".1.php.swp"
    owned by: root    dated: Wed Feb 26 18:20:01 2020
    file name: ~root/1.php
    modified: YES
    user name: root    host name: yunweisserv01
    process ID: 8854
While opening file "1.php"

(1) Another program may be editing the same file.  If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes.  Quit, or continue with caution.
(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r 1.php"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".1.php.swp"
    to avoid this message.

Swap file ".1.php.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort:
```

==解决办法：将交换文件（在编程过程中产生的临时文件）删除掉即可【在上述提示界面按下D 键，或者使用rm 指令删除交换文件】==

```
File Edit View Search Terminal Help

[root@yunweisserv01 ~]# rm -rf .1.php.swp
```

8.3、退出方式

回顾：在vim中，退出正在编辑的文件可以使用 `:q` 或者 `:wq`

除了上面的这个语法之外，vim 还支持另外一个保存退出(针对内容)方法 `:x`

说明：

① `:x` 在文件没有修改的情况下，表示直接退出（等价于`:q`），在文件修改的情况下表

示保存并退出（`:wq`）

② 如果文件没有被修改，但是使用`wq` 进行退出的话，则文件的修改时间会被更新；但是如果文件没有被修改，使用`x` 进行退出的话，则文件修改时间不会被更新的；主要是会混淆用户对文件的修改时间的认定。

问：我们该用`x`还是`wq`或者`q!`退出编辑器？

答：

A X

B WQ/Q!