

《Linux 基础教程》

- 讲师: 魏明择
- 时间: 2019

目录

《Linux 基础教程》

目录

文本编辑器

vi/vim 文件编辑器

visual studio code 文本编辑器

文本文件查看相关的命令

cat 命令

more命令:

less命令:

head 命令

tail 命令

文件管理命令

cp 命令

mv 命令

which 命令

whereis 命令

file 命令

find 查找命令

grep 命令

wc 命令

文件数据管理命令

gzip命令

数据压缩的种类:

gunzip 命令:

tar 命令

tar 练习

用户权限管理

sudo 命令

exit 命令

whoami 命令

UNIX/Linux组管理及命令

groups 命令

groupadd 命令

groupdel 命令

groupmod 命令

UNIX/Linux用户管理及命令

useradd 命令

userdel 命令

passwd 命令

usermod 命令

su 命令

文件的权限管理及命令

chmod 命令

chown 命令

chgrp 命令

Linux 磁盘管理及命令

df命令

lsblk命令

fdisk 命令

UNIX/Linux进程管理及命令

ps 命令

kill命令:

top 命令:

输出重定向 >

文本编辑器

- vim
- vscode
- pycharm

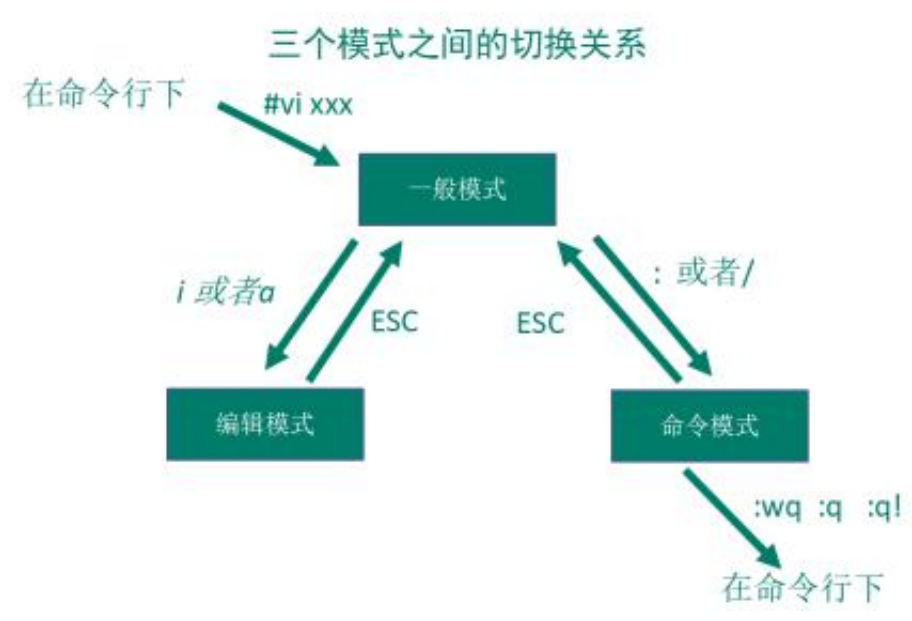
vi/vim 文件编辑器

- 编辑文本文件的软件
- 启动命令

```
vi
vi  文件名
vim
vim 文件名
```

- 退出 vim
:q<回车> :是英文，输入冒号是必须关闭中文输入法

- vim 三种常用的模式：
如图：



- vim的常用命令:

命令	功能
:q	退出
:q!	不保存，强制退出
:w	保存
:w 文件名	保存到指定文件(另存...)
:wq	保存并退出
:w!	强制保存
:x	保存并退出

- vim 练习
 - 在用户主目录下创建目录：myfile
 - 在myfile目录下创建文件：Linux.txt 、 Python-base.txt
 - 在Linux.txt中写入如下3行内容后保存退出
常用的Linux操作系统有Ubuntu、CentOS、Redhat(红帽)
Linux系统中文件夹叫做目录,文件叫做文件
pwd命令为显示当前所在路径
 - 继续在Linux.txt中末尾添加如下4行内容保存退出
rmdir只能删除空目录
rm -rf 命令能删除所有的文件或目录,使用时尽量避免写绝对路径
mkdir -p选项能递归创建目录
touch 命令如果文件已存在,则更新文件的创建时间
 - 继续在Linux.txt中末尾添加如下内容,添加完成后不保存直接退出
vi的使用流程：vi 文件名 -> a -> 写入内容 -> ESC -> shift+: -> wq/q!

visual studio code 文本编辑器

- vs code 启动命令:

```
code<回车>  
code 文件名<回车>
```

- vs code 的快捷操作

1. vs code 切换侧边栏:

查看 -> 外观 -> 切换侧边栏

快捷键: ctrl+b

2. 放大缩小: ctrl + 鼠标滑轮

快捷键: ctrl + "=" / ctrl + "-"

3. vs code 创建和取消多个光标:

创建多个光标: ctrl+鼠标左键单击

取消多个光标: 按ESC键 或 单击鼠标左键

4. sublime 批量多选相同的选择块

快捷键: ctrl+d 重复此操作选下一个相同块

5. 多行注释/取消注释:

快捷键: ctrl + /

6. 复制粘帖光标所在行

ctrl + c 复制光标所在的行

ctrl + v 在光标的上一行粘贴复制的行

7. 剪删除光标所在行:

ctrl + x

8. 新建文件:

ctrl + n

9. 交替换行???:

ctrl + shift + 上下键

10. 回退/撤销回退

ctrl + z

ctrl + r

11. 保存文件(重要)

快捷键: ctrl + s

12. 改变字体大小:

文件 / Code? -> 首选项 -> 设置

找到:

"editor.fontSize": 12,

将其复制到: 右侧的用户设置并修改字号如下:

{

"editor.fontSize": 16,

}

13. 改变颜色主题:

文件 / Code? -> 首选项 -> 颜色主题

快捷键: ctrl+k ctrl+t

14. 设置默认换行:

文件 / Code? -> 首选项 -> 设置

在右侧的用户设置添加如下:

```
{
  "editor.wordWrap": "on",
  # 或
  "[plaintext]": {
    "editor.wordWrap": "on"
  },
}
```

15. 安装二进制插件

hexdump for vscode插件

文本文件查看相关的命令

cat 命令

- 作用:
将文件内容作为标准输出显示
- 格式:

```
cat 文件1 文件2 ...
```

- 示例:

```
cat a.txt
```

more命令:

- 作用:
分页显示文本文件内容
- 格式:

```
more 文件名
```

- 示例:

```
more day03.txt
```

- 基本操作:
空格 下翻一页
q 退出
<回车> 下翻一行
- 练习

1. 复制/etc/passwd文件到用户主目录下
2. 将用户主目录/home/tarena/passwd文件重命名为 new_passwd
3. 在new_passwd的末尾添加如下内容：
good morning!
4. 用more命令查看主目录下new_passwd文件的内容
5. 删除用户主目录下的 new_passwd文件

less命令:

- 作用:
分页显示文本文件内容(可以上下回滚, 翻页)
- 格式:
less 文件名
- 基本操作:
q 退出
j 下翻
k 上翻
空格 下翻一页

head 命令

- 作用
查看文件头几行内容
- 格式

```
head [-n] 文件名
```

- 注: n为整数数字

- 示例

```
$ head /etc/passwd  
$ head -5 /etc/passwd
```

tail 命令

- 作用
查看文件末尾几行内容
- 格式

```
tail [-n] 文件名  
注: n为整数数字(默认为10)
```

- 示例

```
$ tail /etc/passwd
$ tail -5 /etc/passwd
```

- 练习
 1. 在用户主目录下 vi A1.txt,写入/etc/passwd文件中的所有内容
 2. 将A1.txt重命名为A2.txt
 3. 查看A2.txt内容的前5行
 4. 查看A2.txt内容的后10行
 5. 删除文件A2.txt

文件管理命令

cp 命令

- 作用:
复制文件或文件夹

- 格式:

```
cp [选项] 源文件名或文件夹 目标文件或文件夹
```

- 示例:

```
cp a.txt a/
```

- 练习:
 1. 复制 /etc/passwd 文件到用户主目录下的aid1803目录下
 2. 在aid1803目录下创建两个目录: dir1. dir2
 3. 在dir1下创建文件: file1.txt
 4. 在file1.txt中写入如下内容:
仰天大笑出门去,我辈岂是蓬蒿人!
 5. 将文件file1.txt复制一份到dir2目录下,名字为new_file1.txt
 6. 将目录dir2复制,放到dir1目录下
 7. 删除dir1下的dir2目录

mv 命令

- 作用
文件搬移或者更名

- 格式

```
mv 源文件或文件夹 目标文件或文件夹
```

- 练习:

1. 在用户主目录下创建aid1709
 2. 在aid1709下创建两个文件夹:
linux
pbase
 3. 在linux文件夹下创建三个文件夹: day01 day02
 4. 在day02下创建文件:
a.txt b.txt
 5. 将/etc/passwd 复制到day02下
- 练习:
 1. 创建一个文件夹: 我的电脑
 2. 在"我的电脑"内创建两个文件夹:c , d
 3. 在"我的电脑/d"里创建如下几个文件夹:
音乐, 电影, 笔记, python程序
 4. 在笔记文件夹内用文本编辑器创建两个文件:
目录.txt,和 日记.txt
 5. 在目录.txt中写入:
day01 linux 命令
day02 python
day03 web前端
 6. 在 日记.txt 中写:
2018年 3月12日
 7. 把之前的创始的所有.mp3结尾的文件放入"音乐"文件
 8. 把"我的电脑" 文件夹复制到桌面上

which 命令

- 显示命令所在的可执行文件的完整路径
- 命令示例

```
$ which 命令名
```

whereis 命令

- 作用:
显示一个二进制文件、源码或man的位置
- 格式:

```
whereis 命令名
```

file 命令

- 作用

查看文件的类型

find 查找命令

- 作用：
根据文件名等信息查找指定的文件
- 格式：

```
find 路径 -name "文件名"
```

- 示例：

```
# 根据名字进行查找文件
冰雨.mp3
$ find /home/weimz -name "冰雨.mp3"
find ~ -name "*.txt"
练习查找passwd这个文件在哪儿
find / -name "passwd"
```

grep 命令

- 作用：
查找文件中相应内容,及文本信息
- 格式：

```
grep "内容" [选项] 文件名或路径
```

- 常用选项：

选项	说明
-n	显示行号
-R/-r/--recursive	递归搜索文件夹内的文件

- 将错误重定向到 /dev/null

```
grep "weimz" * # 显示太多的错误信息
grep "weimz" * 2> /dev/null
```

- 强制终止程序的执行：
ctrl + c

- 练习：

1. 查找 group 这个文件在哪儿个文件夹下
命令：
find /etc -name "group"

2. 查找/etc下哪儿个文件里含有 "tedu" 这个字符串
命令：

```
grep "tedu" -nr /
```

- grep 练习

1. 在用户主目录下创建目录 PYTHON，在python中创建5个文件：
AA.TXT BB.TXT CC.DOC DD.DOC EE.TXT
2. 在AA.TXT中添加如下内容：
黑夜给了我黑色的眼睛,我要用它来寻找光明!
find是查找文件或者目录
grep是从文件中筛选内容
3. 在用户主目录下查找文件名为DD.TXT的文件
4. 在python目录中查找以.DOC结尾的文件
5. 在AA.TXT中筛选带有 find 这个单词的行

wc 命令

- 作用：
统计文件中字节数、字数、行数

- 格式：

```
wc 文件名
```

- 常用选项
-l 只显示行号
-c 只显示字节数

- 示例：

```
wc /etc/passwd  
wc -l /etc/passwd  
wc -c /etc/passwd
```

- 练习

1. 统计AA.TXT文件一共有多少行
2. 删除python目录中所有的 .DOC文件
3. 将AA.TXT复制到用户主目录下
4. 将BB.TXT重命名为BB2.TXT

```

    ~~~
    -R 递 归 recursive
      门
    /  |  \
客厅 卫生门 厨房
  |  \
  |  \
小客厅 阳台
  /  \
卧1   卧2
    ~~~

```

- 练习： 用命令创建 一个文件夹day01
在day01内创建文件a.txt, b.txt c.txt
在day01内创文件夹 python
在./day01/python/下创建文件hello.py并写入
aaaa bbbb abcdef
再复制/etc/passwd 到 day01下

文件数据管理命令

- 压缩和解压缩命令gzip/gunzip
- tar 打包和解包的命令

gzip命令

- 作用：
用zip压缩算法对文件进行压缩,生成压缩后的.gz文件
- 格式:

```
gzip 文件名
```

- 说明
压缩后的文件名后缀通常为 .gz

数据压缩的种类:

- 有损压缩
mp3, jpg, mp4,
- 无损压缩
zip, gz, png, gif, rar, .xz

gunzip 命令:

- 作用：
对用zip压缩算法压缩.gz格式的数据文件进行解压缩

- 格式：

```
gunzip 文件名
```

tar 命令

- 作用：
对文件或文件夹进行打包和解包的操作
- 格式：

```
tar [选项] 文件名 [文件名或路径]
```

- 常用选项:

选项	说明
-C	创建包
-X	解包
-f 文件名	操作的文件名
-v	显示操作的文件细节
-z	用gzip/gunzip对包进行压缩和解压缩
-C 路径	改变解压缩的路径(只对解包有效)

- 示例：

```
# 打包不压缩
tar -cf day01.tar day01/
# 解包
tar -xf day01.tar

# 常用的打包命令：
tar -czvf day02.tar.gz day02
# 解包：
tar -xzvf day02.tar.gz
```

tar 练习

1. 在用户主目录下创建目录mydir1、mydir2
2. 在mydir1下创建文件 myfile1.txt、myfile2.txt、myfile3.txt
3. 将mydir打包压缩为 mydir1.tar.gz,放在用户主目录下
4. 将mydir1.tar.gz剪切到mydir1目录下
5. 将mydir1.tar.gz解压缩,放到mydir2目录下

用户权限管理

sudo 命令

- 作用：
用超级用户root权限来执行这些命令
- 格式：

```
sudo 命令 [选项] [参数]
```

- 常用选项：
-i 选项，切换到root用户
如：

```
$ sudo -i # 进入root用户
```

exit 命令

- 作用：
退出用户登陆

whoami 命令

- 作用
查看当前登录用户名
- 示例：

```
$ whoami  
tarena
```

UNIX/Linux组管理及命令

- 组相关的配置文件
/etc/group

groups 命令

- 显示当前用户在哪儿些组中
- 格式

```
groups
```

groupadd 命令

- 作用
创建一个组
- 格式

```
groupadd 组名
```

groupdel 命令

- 作用
删除一个组
- 格式

```
groupdel 组名
```

groupmod 命令

- 作用
修改一个组的信息
- 格式

```
groupmod 组名
```

UNIX/Linux用户管理及命令

- 用户相关的信息
用户的名称
用户的ID (UID)
用户的主组
用户的主目录
用户的密码
用户的登陆shell
- 用户相关的配置文件
 - 用户信息存储文件
/etc/passwd
 - 用户密码信息存储文件
/etc/shadow

useradd 命令

- 作用：
添加一个新用户
- 格式：

```
$ useradd [ 选项 ] 新用户名
```

- 常用选项

选项	说明
-g 组名或GID	指定主组
-G 组列表	将用户添加到其它组中，列表用逗号分开。
-p 密码	在创建时指定密码
-m	创建用户主目录
-d 用户主目录	指定其它的用户主目录
-s SHELL程序路径	指定新用户的登shell

- 示例：

```
$ useradd weimz -m -s /bin/bash -p 123456
$ useradd weimingze -g 1001 -m -s /bin/bash
```

userdel 命令

- 作用：
删除一个已存在的用户
- 格式:

```
$ userdel [ 选项 ] 用户名
```

- 常用选项
-r 删除用户的主目录和相关文件
- 示例：

```
$ sudo userdel -r weimingze
```

passwd 命令

- 作用：
修改用户密码
- 格式:

```
$ passwd [ 选项 ] 用户名
```

- 常用选项
-d 删除用户密码

usermod 命令

- 作用：
修改用户的相关信息
- 格式:

```
$ usermod [ 选项 ] 用户名
```

- 常用选项

选项	说明
-l 新登陆名	重新指定新的登陆名
-g 组名或GID	强制指定新主组
-G 组列表	新的附加组列，列表用逗号分开。
-p 密码	指定新密码
-d 用户主目录	指定其它的用户主目录
-s SHELL程序路径	指定新用户的登shell

su 命令

- 作用：
切换到指定的用户
- 格式:

```
$ su 用户名
```

- 说明
如不指定参数，默认切换到root用户
- 练习(创建帐户):
 1. 在你的Ubuntu 虚拟机内创建你名字的用户 \$ sudo useradd weimz -m -s /bin/bash
 2. 用passwd给你名字的用户设置初始密码是123456
 3. 用su 命令切换到你名字的那个用户, 创建一个notes.txt文件，并用vi 写入你的姓名
 4. 退出以你名字命名的用户,回到tarena 用户
 5. 查看刚才你写过的文件
 6. 尝试修改你刚才写过的文件
 7. 查看/etc/shadow 下你名字帐户对应的密码值
 8. 修改你名字帐户的密码,再看看/etc/shadow里有什么变化
 9. 删除你名字的帐户（可以考虑不做此步）

文件的权限管理及命令

• 查看文件的权限

```
ls -l
```

• Unix/Linux文件类型

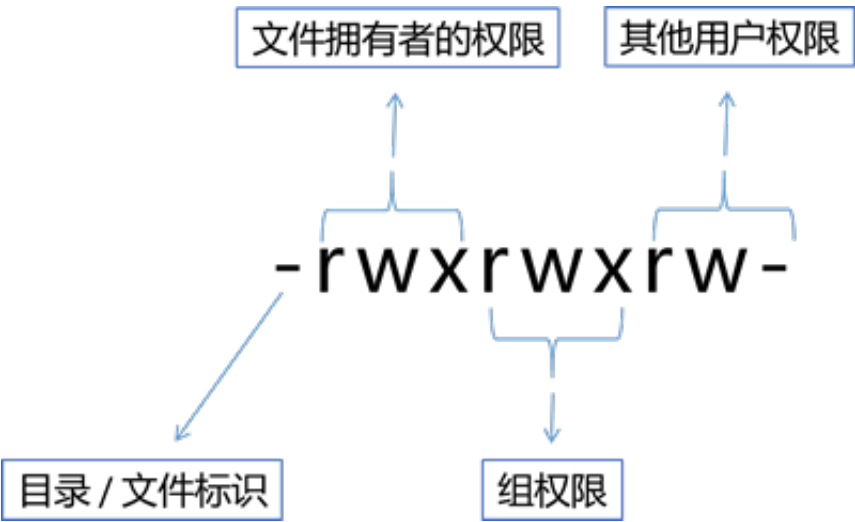
符号	类型
d	文件夹(d)
-	普通文件(-)
c/b	设备文件(c/b)
p	管道文件(p)
l	链接文件（Unix中是替身文件）(l)
s	套接字文件(s)

- Linux系统上对文件的权限有着严格的控制，用于如果相对某个文件执行某种操作，必须具有对应的权限方可执行 成功。

权限类型：权限类型一般包括读，写，执行，对应字母为 r、w、x

权限粒度：文件拥有者（创建文件的用户）、组（创建文件用户所在的组）、其它组三。

所以，一个文件的权限表示为9中状态，分别指拥有这、组、其它用户对该文件是否具有读、写、执行的权限



在rwx三个权限中，从右至左依次用值1,2,4来表示，所以拥有某个文件的读、写、执行权限也可以用三个权限值累加的方式来表示，如下表所示：

• 文件的权限类型

◦ r 读权限

文件，具有读取文件内容的权限。

文件夹，具有读取文件夹内文件目录的权限

◦ w 写权限

文件，具有新增、修改、删除文件内容的权限。

文件夹，具有新建、删除、修改文件名、移动目录内文件的权限。

◦ x 执行权限

文件，具有执行文件的权。

文件夹，具有进入目录的权限。

◦ - 无权限

- 文件的权限的分组
用户权限(user) 组权限(group) 其它(other)

- 最高权限

```
rwxrwxrwx
```

- 最低权限

```
-----
```

- 文件权限是可以改的

```
r--r--r--
-w--w----
----w----
```

- 属主

- Unix/Linux系统中，每一个文件都必然属于某一用户。我们把这个用户叫做所有者，也叫属主
文件的所有者（属主）才有权利修改文件的权限。

- 属组

- Unix/Linux系统中，每一个文件都必然属于某一组。我们把这个组叫 属组

- 其他

- 不属于属主也不属于属组的用户，叫做其它用户（other）

chmod 命令

- 作用

用来修改文件的权限

- 格式

```
chmod 权限 文件名/文件夹名
```

- 权限

- u 用户（所有者）
- g 同组用户
- o 其他用户
- a 所有用户
- + 添加权限
- - 去限权限

- 示例

```
chmod u+r yy.txt
chmod o-rwx yy.txt
chmod g+rw yy.txt
chmod g-w yy.txt
chmod a+x yy.txt
chmod a-r yy.txt
chmod 777 yy.txt
chmod 000 yy.txt
```

- 以数字方式去修改文件权限

r:4

w:2

x:1

```
chmod 764 file.txt
chmod 664 file.txt
chmod 777 file.txt
```

chown 命令

- 作用：
修改文件的属主（需要管理员权限）
- chown命令的格式

```
$ chown [owner][:group] [选项] FILES
```

- 常用选项
-R 文件名 递归对所有文件进行相同的操作

chgrp 命令

- 作用：
修改文件的属组
- chgrp命令的格式

```
$ chgrp GROUP [选项] FILES
```

- 常用选项

-R 文件名 递归对所有文件进行相同的操作

- 示例

```
#将文件afile属组设置为weimz
$ chgrp weimz afile
等同于
$ chown :weimz afile
```

- 练习：
 1. 用tarena用户创建一个文件: myprog.sh
 2. 编辑 myprog.sh 添入以下内容:

```
#!/bin/bash

ls /
cd ~
mkdir "xxx来过"
```

3. 修改此文件的权限,让tarena用户有可执行此文件的权限
4. 用tarena用户执行此文件:
\$./myprog.sh

- 练习
 - 操作文件权限:
创建两个文件 a.txt b.txt写入一些内容
让其它用户不能读取a.txt文件
让用组户和其它用户不能读取b.txt
让用户自身对所有文件有执行权限

Linux 磁盘管理及命令

df命令

- 作用
用于显示目前在Linux系统上的文件系统的磁盘使用情况统
- 示例

```
$ df
文件系统              1K-块      已用      可用  已用%  挂载点
udev                  1971520          0   1971520    0%  /dev
/dev/sda1             205373416  18106860  176764516   10%  /
/dev/loop0             144128     144128          0  100%  /snap/gnome-3-26-
1604/97
/dev/loop2             15104      15104          0  100%  /snap/gnome-
characters/359
```

- 示例2

```
$ df -h
文件系统          容量  已用  可用 已用% 挂载点
udev              1.9G   0    1.9G   0% /dev
/dev/sda1         196G   18G  169G   10% /
/dev/loop0        141M  141M    0  100% /snap/gnome-3-26-1604/97
/dev/loop2        15M   15M    0  100% /snap/gnome-characters/359
```

lsblk命令

- 作用
用来查看磁盘(block设备)的信息.
- 示例:

```
$ lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0      7:0    0 140.7M  1 loop /snap/gnome-3-26-1604/97
loop1      7:1    0  14.8M  1 loop /snap/gnome-characters/367
sda        8:0    0   200G  0 disk
└─sda1     8:1    0   200G  0 part /
sr0       11:0    1  1024M  0 rom
```

fdisk 命令

- 作用
用来查看磁盘分区或为磁盘分区
- 常用选项
-l 查看磁盘分区
- 示例:

```
sudo fdisk -l
[sudo] tarena 的密码:
Disk /dev/sda: 200 GiB, 214748364800 字节, 419430400 个扇区
单元: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x1d3d809a

设备          启动  起点      末尾      扇区  大小  Id  类型
/dev/sda1    *      2048 419428351 419426304  200G  83  Linux
```

UNIX/Linux进程管理及命令

ps 命令

- 作用：
查看进程
- 格式: ps [选项]
- 常用选项
-aux 查看当前系统内的所有进程的详细信息
- 示例：

```
ps          # 查看当前环境的进程
ps aux      # 查看当前系统内的所有进程的详细信息
```

kill命令:

- 作用：
杀死进程
- 格式:

```
kill 进程的pid号
```

- 注: 在终端运行时 可以用 ctrl+c 终止进程

top 命令:

- 作用:
查看程序运行状态
- 退出
Q 键

输出重定向 >

- 作用： 把输出到终端的内容输出到指定文件
- 格式：

```
命令      >      文件
```

- 示例：

```
ls > a.txt
cat passwd > a.txt
head -5 passwd > a.txt
echo "Python" > a.txt
```

- 注意

1. 如果文件存在，则覆盖原文件
 2. 如果文件不存在，则创建一个新文件
- 练习
 1. 在用户主目录下创建文件 echo.txt
 2. 用vi在echo.txt中添加如下内容
人生难得起起落落,还是要坚强地生活!
 3. 用echo命令将echo.txt中的内容变为：我是echo过来覆盖你们的
 4. 在用户主目录下的文件以长格式的方式重定向输入到echo.txt中
 5. 查看echo.txt中的内容