

## 重定向

--把终端执行命令结果 保存在目标文件中  
> 覆盖原文件内容,---'w'操作  
>> 追加至文件末尾 ---'a'操作

## 查看文件内容

cat 文件名称	more 查看大型文件 操作键 回车---显示下一行 b---显示上一屏 f---显示下一屏 q---退出	管道 左边命令的输出作
grep 查看/搜索文件内容 [文件] -v 显示不包含匹配文本的所有行 -n 显示行号 -i 忽略大小写	结合正则表达式: ^str 以指定字符串开头 str\$ 以指定字符串结尾 str.str匹配一个非换行符的字符	

例子: **grep ^h hello.py -ni** ---查找有h开头的文件内容行,忽略大小写  
vim **显示行号 set number**

## find /root -type f -name "value" 文件-----在指定目录下查找复合条件的文件

通配符!! ---- \* 代表任意个数的字符  
---- ? 仅代表任意一个字符  
---- 任何查找命令都能用通配符,但是只有find需要添加' '--双引号

## 链接命令--软链接(建议绝对链接)

ln -s 相当于windows快捷方式  
格式: ln -s 文件路径 [快捷方式名称]  
ln 硬链接  
格式一样,查看时,链接数目为1,不显示链接路径,删除原文件也不影响 --跟软链接相反

## 压缩和解压缩文件

tar (命令选项) (包名称) (打包文件) (解压目录) ---压缩, 解压缩		
格式: .gz ---对于大文件更少空间 .bz2	命令选项 -z -j	tar命令选项: -c 创建打包 -x 解压 -v 显示打包或解包详细信息 -f 指定压缩/解包文件名称 -C(大写) 解压缩到指定目录

**压缩 tar -zcvf test.tar.gz a.txt -----压缩a.txt ,压缩包为test,格式为gz**  
**解压 tar -zxvf test.tar.gz -C . -----解压test包,压缩到.当前路径**  
**总结: 命令选项 一定写 -zx(c)vf ,压缩加上文件名称,解压要指定路径**

## 软件安装—离线安装dpkg，在线安装apt-get 软件卸载—离线安装包卸载, 在线安装包卸载

sudo dpkg -i deb安装包      sudo apt-get install deb安装包    可以不需要记下载名字,直接打,联网时候  
sudo dpkg -r deb安装包名      sudo apt-get remove 安装包名

## 文件权限

文件信息 ls -lh

-rw-rw-r-- 1 python python 0 4月 15 10:41 e.txt

-rw-rw-r-- 第一位置: -为文件 d为目录 ①rw为用户权限 ②rw为用户组权限 ③r为其他用户

r --可读 w --可写 x --可执行 -表示没权限

①python 是当前用户

②python 为所属用户组

chmod 修改文件权限

字母法: chmod (u用户/g用户组/o其他用户/a所有用户)(+/-/=(权限rwx-))文件名

数字法: chmod 权限值 文件名 r-4 w-2 x-1 --0

chmod 777 a.txt ----rwxrwxrwx 数字从左至右

sudo 获取管理员权限

sudo -s 切换到root用户,获取管理员权限

sudo+其他命令 一次性

whoami 查看当前用户

who 查看所有的登录用户

exit 退出登录用户

passwd 修改用户密码

which 文件 ---查看文件位置

组管理与用户管理

/etc 目录存放的都是系统配置文件,group是组管理

changeproup -R(递归) 组名 文件/目录 ---修改文件/目录所属组

/etc/passwd 是保存用户信息

常用格式: sudo useradd -m -g Liu liukaitao

sudo password liukaitao --修改liukaitao用户密码

ssh 用户名@ip地址 远程登录 -----linux ip地址获取 ifconfig

scp (-r)本地文件 远程服务器用户名@远程服务器ip地址:指定copy到远程服务器的路径

scp (-r)远程服务器用户名@远程服务器ip地址:远程服务器文件 指定copy到本地的路径

---一般使用软件 OpenSSH

提示: 创建组/删除组 的终端命令都需要通过 sudo 执行

序号	命令	作用
01	<u>groupadd 组名</u>	添加组
02	<u>groupdel 组名</u>	删除组
03	<u>cat /etc/group</u>	确认组信息
04	chgrp 组名 文件/目录名	修改文件/目录的所属组

3.1 创建用户/设置密码/删除用户

序号	命令	作用	说明
01	useradd -m -g 组 新建用户名	添加新用户	<ul style="list-style-type: none"><li>-m 自动建立用户家目录</li><li>-g 指定用户所在的组，否则会建立一个和同名的组</li></ul>
02	passwd 用户名	设置用户密码	如果是普通用户，直接用 passwd 可以修改自己的账户密码
03	userdel -r 用户名	删除用户	-r 选项会自动删除用户家目录
04	cat /etc/passwd   grep 用户名	确认用户信息	新建用户后，用户信息会保存在 /etc/passwd 文件中

1\重定向:>

```
python@ubuntu: ~/Desktop
python@ubuntu:~/Desktop$ ls /
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
boot dev  home lib32 libx32 media  opt  root /sbin  srv  tmp  var
python@ubuntu:~/Desktop$ ls / > info.txt
python@ubuntu:~/Desktop$
```

>>

```
python@ubuntu: ~/Desktop/test
python@ubuntu:~/Desktop/test$ ls
jerry tom
python@ubuntu:~/Desktop/test$ ls >> ../msg.txt
python@ubuntu:~/Desktop/test$
```

2\查看文件内容

less命令和more命令用法差不多一样，也是查看大型文件的  
唯一的区别是less命令查看文件不会自动退出，一定要在：后面输入q

find命令格式

find 搜索文件路径 -name 文件名  
文件名可以用通配符表示，但是以通配符表示的文件名表示的文件名一定要加引号

grep

选项	说明
-v	显示不包含匹配文本的所有行
-n	显示匹配行号
-i	忽略大小写

## **-i 选项:**

```
python@ubuntu:~/Desktop/A/B/C$ cat hello.py
AAaaA
BB
CCC
python@ubuntu:~/Desktop/A/B/C$ grep 'a' hello.py
AAaaA
python@ubuntu:~/Desktop/A/B/C$ grep 'a' hello.py -i
AAaaA
python@ubuntu:~/Desktop/A/B/C$
```

## **-n 选项:**

```
python@ubuntu:~/Desktop/A/B/C$ cat hello.py
AAaaA
BB
CCC
python@ubuntu:~/Desktop/A/B/C$ grep "B" hello.py
BB
python@ubuntu:~/Desktop/A/B/C$ grep "B" hello.py -n
2:BB
python@ubuntu:~/Desktop/A/B/C$
```

## **-v 选项:**

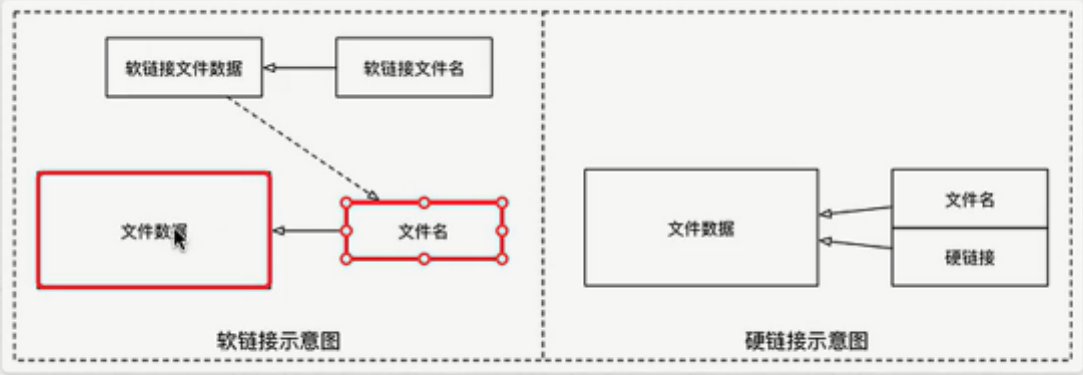
```
python@ubuntu:~/Desktop/A/B/C$ cat hello.py
AAaaA
BB
CCC
python@ubuntu:~/Desktop/A/B/C$ grep 'c' hello.py
ccc
python@ubuntu:~/Desktop/A/B/C$ grep 'c' hello.py -v
AAaaA
BB
python@ubuntu:~/Desktop/A/B/C$ grep 'c' hello.py -vn
1:AAaaA
2:BB
python@ubuntu:~/Desktop/A/B/C$
```

## 3\链接

```
python@ubuntu: ~/Desktop/A
python@ubuntu:~/Desktop$ ln -s A/B/C/hello.py hello-s1.py
python@ubuntu:~/Desktop$ ln -s /home/python/Desktop/A/B/C/hello.py hello-s2.py
python@ubuntu:~/Desktop$ ls -l
总用量 8
drwxrwxr-x 3 python python 4096 11月 11 16:37 A
lrwxrwxrwx 1 python python 14 11月 11 16:41 hello-s1.py -> A/B/C/hello.py
lrwxrwxrwx 1 python python 35 11月 11 16:42 hello-s2.py -> /home/python/Desktop/A/B/C/hello.py
drwxrwxr-x 5 python python 4096 11月 11 16:19 资料
python@ubuntu:~/Desktop$ mv hello-s* A
python@ubuntu:~/Desktop$ cd A
python@ubuntu:~/Desktop/A$ ls -l
总用量 4
drwxrwxr-x 3 python python 4096 11月 11 16:37 B
lrwxrwxrwx 1 python python 14 11月 11 16:41 hello-s1.py -> A/B/C/hello.py
lrwxrwxrwx 1 python python 35 11月 11 16:42 hello-s2.py -> /home/python/Desktop/A/B/C/hello.py
python@ubuntu:~/Desktop/A$
```

## 软链接,硬链接原理

文件软硬链接的示意图



在 Linux 中，文件名 和 文件的数据 是分开存储的

4\在linux里面查看当前进程的命令

ps aux 或者 ps -ef

5\改变权限命令

chown 改变拥有者 chgrp改变组

7.1 chmod 字母法

格式: chmod u/g/o/a+/-/=rwx 文件名

操作符	说明
+	增加权限
-	撤销权限
=	设置权限
权限	说明
r	可读
w	可写
x	可执行
-	无任何权限

角色	说明
u	user 表示该文件的所有者
g	group 表示用户组
o	other 表示其他用户
a	all 表示所有用户

## 7.2 chmod 数字法

格式: chmod 权限值 文件名

权限	说明
r	可读, 权限值是4
w	可写, 权限值是2
x	可执行, 权限值是1
-	无任何权限, 权限值是0

## 7.2 chmod 数字法

```
python@ubuntu:~/Desktop/test$ ls -l
总用量 0
-rw-rw-r-- 1 python python 0 11月 12 13:22 1.txt
python@ubuntu:~/Desktop/test$ chmod 666 1.txt
python@ubuntu:~/Desktop/test$ ls -l
总用量 0
-rw-rw-rw- 1 python python 0 11月 12 13:22 1.txt
python@ubuntu:~/Desktop/test$ chmod 777 1.txt
python@ubuntu:~/Desktop/test$ ls -l
总用量 0
-rwxrwxrwx 1 python python 0 11月 12 13:22 1.txt
python@ubuntu:~/Desktop/test$ chmod 000 1.txt
python@ubuntu:~/Desktop/test$ ls -l
总用量 0
----- 1 python python 0 11月 12 13:22 1.txt
python@ubuntu:~/Desktop/test$ chmod 664 1.txt
python@ubuntu:~/Desktop/test$ ls -l
总用量 0
-rw-rw-r-- 1 python python 0 11月 12 13:22 1.txt
python@ubuntu:~/Desktop/test$
```

---

## 8\用户管理



3.1 创建用户/设置密码/删除用户

序号	命令	作用	备注
01	useradd -m -g 组 新建用户名	添加新用户	<ul style="list-style-type: none"><li>-m 自动建立用户家目录</li><li>-g 指定用户所在的组，否则会建立一个和同名的组</li></ul>
02	passwd 用户名	设置用户密码	如果是普通用户，直接用 passwd 可以修改自己的账户密码
03	userdel -r 用户名	删除用户	-r 选项会自动删除用户家目录
04	cat /etc/passwd   grep 用户名	确认用户信息	新建用户后，用户信息会保存在 /etc/passwd 文件中

序号	命令	作用
01	id [用户名]	查看用户 UID 和 GID 信息
02	who	查看当前所有登录的用户列表
03	whoami	查看当前登录用户的账户名

--UID用户id GID组ID

passwd 文件

```
python@ubuntu:~$ id lisi
uid=1002(lisi) gid=1001(dev) 组=1001(dev)
python@ubuntu:~$ cat -n /etc/passwd | grep lisi
50  lisi:x:1002:1001:::/home/lisi:
```

passwd 文件

/etc/passwd 文件存放的是用户的信息，由 6 个分号组成的 7 个信息，分别是

1. 用户名
2. 密码 (x, 表示加密的密码)
3. UID (用户标识)
4. GID (组标识)
5. 用户全名或本地帐号
6. 家目录
7. 登录使用的 Shell，就是登录之后，使用的终端命令，ubuntu 默认是 dash

9\主组与附加组——就相当于让附加用户也可以使用主组权限

```
python@ubuntu:~$ cat -n /etc/group | grep python
5  adm:x:4:syslog,python
18 cdrom:x:24:python
21 sudo:x:27:python,zhangsan
23 dip:x:30:python
35 plugdev:x:46:python
52 lpadmin:x:113:python
67 python:x:1000:
68 sambashare:x:128:python
python@ubuntu:~$
```

让张三也拥有sudo执行权限 usedmod -G sudo zhangsan

## usermod

- `usermod` 可以用来设置 用户 的主组 / 附加组 和 登录 Shell, 命令格式如下:
- 主组: 通常在新建用户时指定, 在 `etc/passwd` 的第 4 列 `GID` 对应的组
- 附加组: 在 `etc/group` 中最后一列表示该组的用户列表, 用于指定 用户的附加权限

提示: 设置了用户的附加组之后, 需要重新登录才能生效!

```
# 修改用户的主组 (passwd 中的 GID)
usermod -g 组 用户名

# 修改用户的附加组
usermod -G 组 用户名

# 修改用户登录 Shell
usermod -s /bin/bash
```

注意: 默认使用 `useradd` 添加的用户是没有权限使用 `sudo` 以 `root` 身份执行命令的, 可以使用以下命令, 将用户添加到 `sudo` 附加组中

```
usermod -G sudo 用户名
```

关于

`usermod -s /bin/bash` --修改用户登录shell ,ubuntu中shell使用dash win使用bash,易造成不便

10\which --可以查看命令所在位置

## which (重要)

提示

- `/etc/passwd` 是用于保存用户信息的文件
- `/usr/bin/passwd` 是用于修改用户密码的程序
- `which` 命令可以查看执行命令所在位置, 例如:

```
which ls

# 输出
# /bin/ls

which useradd

# 输出
# /usr/sbin/useradd
```

5\软件安装与卸载

```
python@ubuntu: ~/Desktop
python@ubuntu:~/Desktop$ sl
程序“sl”尚未安装。 您可以使用以下命令安装:
sudo apt install sl
python@ubuntu:~/Desktop$ sudo apt instal
```

可以不需要记下载名字,直接打,联网时候

查看当前系统所有已安装的安装包的命令

`dpkg -l`

`dpkg -l|grep atom`



