

常用Linux命令的基本使用

序号	命令	对应英文	作用
01	ls	list	查看当前文件夹下的内容
02	pwd	print wrok directory	查看当前所在文件夹
03	cd [目录名]	change directory	切换文件夹
04	touch [文件名]	touch	如果文件不存在，新建文件
05	mkdir [目录名]	make directory	创建目录
06	rm [文件名]	remove	删除指定的文件名
07	clear	clear	清屏

加空格

08 rmdir 删除目录

小技巧

- $Ctrl+Shift+=$ 放大终端窗口的字体显示
- $Ctrl+-$ 缩小终端窗口的字体显示

实践过程

打错命令
运行失败

```
syq@syq-None:~$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面 snap
syq@syq-None:~$ pwd
/home/syq
syq@syq-None:~$ cd 桌面
syq@syq-None:~/桌面$ pwd
/home/syq/桌面
syq@syq-None:~/桌面$ ls
syq@syq-None:~/桌面$ ls
p16
syq@syq-None:~/桌面$ touch p16.1
syq@syq-None:~/桌面$ ls
p16 p16.1
syq@syq-None:~/桌面$ mkdir p16.2
syq@syq-None:~/桌面$ rm p16
rm: 无法删除 'p16': 是一个目录
syq@syq-None:~/桌面$ rm p16.1
syq@syq-None:~/桌面$ rmdir p16
找不到命令 "rmdir"，您的意思是：
"rmdir" 命令来自 Debian 软件包 coreutils (9.1-1ubuntu2.23.10.1)
尝试 sudo apt install <deb name>
syq@syq-None:~/桌面$ rmdir p16
syq@syq-None:~/桌面$
```

终端命令格式

command [-options] [parameter]

说明：

command：命令名，相应回能的英文单词或单词的缩写

[-options]：选项，可用来自对命令进行控制，也可以省略

parameter：传给命令的参数，可以是零个、一个或者多个

[]代表可选(可有可无)

ls
pwd
clear

touch 文件名称
cd 目录名称
mkdir 目录名称
rm 文件名称

实践过程

```
syq@syq-None:~$ cd
syq@syq-None:~$ cd 桌面
syq@syq-None:~/桌面$ ls
p16.2
syq@syq-None:~/桌面$ rm p16.2
rm: 无法删除 'p16.2': 是一个目录
syq@syq-None:~/桌面$ rm -r p16.2
syq@syq-None:~/桌面$
```

查阅命令帮助信息

- 现阶段只需要知道通过以下两种方式可以查询命令的帮助信息
- 先学习常用命令及常用选项的使用即可，工作中如果遇到问题可以借助网络搜索

1 --help

用法: command --help

2 man

用法: man command

说明:

man是manual的缩写，是Linux提供的一个手册，包含了绝大部分的命令、函数的详细使用说明

使用man时的操作键:

syq@syq-None:~/桌面\$ mkdir --help

用法: mkdir [选项]... 目录...

若 <目录> 不存在，则创建 <目录>。

长选项的必选参数对于短选项也是必选的。

-m, --mode=模式	设置文件模式（格式同 chmod），而不是 a=rwx - umask
-p, --parents	需要时创建目标目录的父目录，但即使这些目录已存在也不视为错误，且其文件模式也不受 -m 选项影响。
-v, --verbose	每次创建新目录时，打印一条消息
-Z	将每个创建的目录的 SELinux 安全上下文设置为默认类型
--context[=上下文]	类似 -Z，但如果指定了 <上下文>，则将 SELinux 或 SMACK 安全上下文设置为 <上下文>
--help	显示此帮助信息并退出
--version	显示版本信息并退出

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>

请向 <http://translationproject.org/team/zh_CN.html> 报告任何翻译错误
完整文档 <<https://www.gnu.org/software/coreutils/mkdir>>

或者在本地使用: info '(coreutils) mkdir invocation'

syq@syq-None:~/桌面\$ man mkdir

syq@syq-None:~/桌面\$

mkdir的用法

下方“q to quit”为退出

操作键	功能
空格键	显示手册页的下一屏
Enter 键	一次滚动手册页的一行
b	回滚一屏
f	前滚一屏
q	退出
/word	搜索 word 字符串

文件和目录常用命令

查看目录内容 ls

切换目录 cd

创建和删除操作 touch ; rm ;

mkdir

拷贝和移动文件 cp ; mv

查看文件内容 cat ; more ; grep

其他 echo ; 重定向 > 和 >> ; 管道 |

终端实用技巧

1> 自动补全

在敲出文件/目录/命令的前几个字母之后，按下 tab 键

● 如果输入的没有歧义，系统会自动补全

● 如果还存在其他文件/目录/命令，再按一下 tab 键，系统会提示可能存在命令

2> 曾经使用过的命令

按上/下光标键可以在曾经使用过的命令之间来回切换

如果想要退出选择，并且不想执行当前选中的命令，可以按 ctrl + c

ls命令说明

· ls是英文单词list的简写，其功能为列出目录的内容，是用户最常用的命令之一，类似于DOS下的dir命令

LINUX下文件和目录的特点

- LINUX文件或者目录名称最长可以有256个字符
- 以.开头的文件为隐藏文件，需要用-a参数才能显示
- ..代表当前目录
- ..代表上一级目录

ls -a相当于la

实践过程

```
syq@syq-None:~$ cd 桌面
syq@syq-None:~/桌面$ touch p21
syq@syq-None:~/桌面$ touch .p21.1
syq@syq-None:~/桌面$ ls
p21
syq@syq-None:~/桌面$ ls -a
ls -a: 未找到命令
syq@syq-None:~/桌面$ ls -a
. .. p21 .p21.1
syq@syq-None:~/桌面$ la
p21 .p21.1
syq@syq-None:~/桌面$ rm p21 .p21.1
syq@syq-None:~/桌面$ la
syq@syq-None:~/桌面$ ls -a
. ..
syq@syq-None:~/桌面$ pwd
/home/syq/桌面
syq@syq-None:~/桌面$ cd .
syq@syq-None:~/桌面$ pwd
/home/syq/桌面
syq@syq-None:~/桌面$ cd ..
syq@syq-None:~$ pwd
/home/syq
syq@syq-None:~$
```

ls的通用选项

参数	含义
-a	显示指定目录下所有子目录与文件，包括隐藏文件
-l	以列表方式显示文件的详细信息
-h	配合-l以人性化的方式显示文件大小

说明：

- -l与-l -h(相当于-hl)显示的文件大小的表达方式不同
- 仅用ls -a只显示名称
- -ahl可显示所有文件的信息

```
syq@syq-None:~$ ls -l
总计 36
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 公共
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 模板
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 视频
drwxr-xr-x 3 syq syq 4096 10月 4 17:38 图片
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 文档
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 下载
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 音乐
drwxr-xr-x 2 syq syq 4096 10月 5 23:06 桌面
-rw-rw-r-- 1 syq syq 0 10月 5 23:10 p22.2
drwx----- 4 syq syq 4096 9月 29 00:00 snap
syq@syq-None:~$ ls -l -h
总计 36K
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 公共
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 模板
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 视频
drwxr-xr-x 3 syq syq 4.0K 10月 4 17:38 图片
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 文档
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 下载
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 音乐
drwxr-xr-x 2 syq syq 4.0K 10月 5 23:06 桌面
-rw-rw-r-- 1 syq syq 0 10月 5 23:10 p22.2
drwx----- 4 syq syq 4.0K 9月 29 00:00 snap
```

计算机中文件大小的表达方式

单位	英文	含义
字节	B (Byte)	在计算机中作为一个数字单元，一般为8位二进制数
千	K (Kibibyte)	1 KB = 1024 B，千字节 (1024 = 2 ¹⁰)
兆	M (Mebibyte)	1 MB = 1024 KB，百万字节
千兆	G (Gigabyte)	1 GB = 1024 MB，十亿字节，千兆字节
太	T (Terabyte)	1 TB = 1024 GB，万亿字节，太字节
拍	P (Petabyte)	1 PB = 1024 TB，千万亿字节，拍字节
艾	E (Exabyte)	1 EB = 1024 PB，百亿亿字节，艾字节
泽	Z (Zettabyte)	1 ZB = 1024 EB，十万亿亿字节，泽字节
尧	Y (Yottabyte)	1 YB = 1024 ZB，一亿亿亿字节，尧字节

通配符的使用

通配符	含义
*	代表任意个数个字符
?	代表任意一个字符，至少 1 个
[]	表示可以匹配字符组中的任一个
[abc]	匹配 a、b、c 中的任意一个
[a-f]	匹配从 a 到 f 范围内的任意一个字符

```
syq@syq-None:~/桌面$ ls  
102.txt 123.txt 257.txt 537.txt 822.txt 863.txt p22.1  
syq@syq-None:~/桌面$ ls ?*  
102.txt 123.txt  
syq@syq-None:~/桌面$ ls *?  
ls: 无法访问 '*?': 没有那个文件或目录  
syq@syq-None:~/桌面$ ls ?*?  
ls: 无法访问 '?*': 没有那个文件或目录  
syq@syq-None:~/桌面$ ls ?3*  
537.txt
```

实践过程

```
syq@syq-None:~/桌面$ ls [12345]23.txt  
123.txt 223.txt 323.txt 423.txt 523.txt  
syq@syq-None:~/桌面$ ls [1-5]23.txt  
123.txt 223.txt 323.txt 423.txt 523.txt  
syq@syq-None:~/桌面$
```

cd

cd 是英文单词 change directory 的简写，其功能为更改当前的工作目录，也是用户最常用的命令之一

命令	含义
cd	切换到当前用户的主目录(/home/用户名)
cd ~	切换到当前用户的主目录(/home/用户名)
cd .	保持在当前目录不变
cd ..	切换到上级目录
cd -	可以在最近两次工作目录之间来回切换

```
syq@syq-None:~/桌面$ cd  
syq@syq-None:~$ pwd  
/home/syq  
syq@syq-None:~$ cd ..  
syq@syq-None:/home$ cd ..  
syq@syq-None:/$ pwd  
/  
syq@syq-None:/$ cd -  
/home  
syq@syq-None:/home$ cd -  
/
```

~ 表示 home 目录
/ 表示根目录

```
syq@syq-None:~$ cd 桌面  
syq@syq-None:~/桌面$ cd ~/下载  
bash: cd: ~/下载: 没有那个文件或目录  
syq@syq-None:~/桌面$ cd ~/下载  
syq@syq-None:~/下载$ cd -  
/home/syq/桌面  
syq@syq-None:~/桌面$
```

相对路径和绝对路径

● 相对路径 在输入路径时，最前面不是 / 或者 ~，表示相对当前目录所在的目录位置

● 绝对路径 在输入路径时，最前面是 / 或者 ~，表示从根目录/家目录开始的具体目录位置

逐层列出

touch

创建文件或修改文件时间

● 如果文件不存在，可以创建一个空白文件

● 如果文件已经存在，可以修改文件的末次修改日期

```
syq@syq-None:~/桌面$ ls -lh  
总计 0  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 102.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:25 123.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 1abc3.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:34 223.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:25 257.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 323.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 423.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 523.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 537.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 822.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 863.txt  
-rw-rw-r-- 1 syq syq 0 10月 5 23:06 p22.1  
syq@syq-None:~/桌面$ touch 523.txt
```

```
syq@syq-None:~/桌面$ ls -lh  
总计 0  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 102.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:25 123.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 1abc3.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:34 223.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:25 257.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 323.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:35 423.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 15:15 523.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 537.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 822.txt  
-rw-rw-r-- 1 syq syq 0 10月 6 12:26 863.txt  
-rw-rw-r-- 1 syq syq 0 10月 5 23:06 p22.1  
syq@syq-None:~/桌面$
```

实践过程

mkdir

创建一个新的目录

选项 含义

-p	可以递归创建目录
----	----------

```
syq@syq-None:~/桌面$ mkdir a
syq@syq-None:~/桌面$ cd a
syq@syq-None:~/桌面/a$ mkdir b
syq@syq-None:~/桌面/a$ cd b
syq@syq-None:~/桌面/a/b$ mkdir c
syq@syq-None:~/桌面/a/b$ cd c
syq@syq-None:~/桌面/a/b/c$ mkdir d
syq@syq-None:~/桌面/a/b/c$ cd d
syq@syq-None:~/桌面/a/b/c/d$ cd ~/桌面
syq@syq-None:~/桌面$ mkdir -p a1/b1/c1/d1/
syq@syq-None:~/桌面$ cd a1/b1/c1/d1/
```

新建目录的名称不能与当前目录中已有的目录或文件同名

```
syq@syq-None:~/桌面/a1/b1/c1/d1$ cd ~/桌面
syq@syq-None:~/桌面$ touch 123
syq@syq-None:~/桌面$ mkdir 123
mkdir: 无法创建目录 "123": 文件已存在
```

rm

删除文件或目录

使用 rm 命令要小心，因为文件删除后不能恢复

参数 含义

-f	强制删除，忽略不存在的文件，无需提示
----	--------------------

-r	递归地删除目录下的内容，删除文件夹时必须加此参数
----	--------------------------

没有不删不提示

```
syq@syq-None:~/桌面$ ls
102.txt 123.txt 223.txt 323.txt 523.txt 822.txt a p22.1
123 1abc3.txt 257.txt 423.txt 537.txt 863.txt a1
syq@syq-None:~/桌面$ rm -f *
rm: 无法删除 'a': 是一个目录
rm: 无法删除 'a1': 是一个目录
syq@syq-None:~/桌面$ ls
a a1
```

```
syq@syq-None:~/桌面$ ls
102.txt 123.txt 223.txt 323.txt 523.txt 822.txt a p22.1
123 1abc3.txt 257.txt 423.txt 537.txt 863.txt a1
syq@syq-None:~/桌面$ rm abc
rm: 无法删除 'abc': 没有那个文件或目录
syq@syq-None:~/桌面$ rm -f abc
```

rm -f * 只能删除桌面上的所有文件
rm -r * 能删除桌面上的所有文件和目录
均不能恢复

拷贝和移动文件

序号	命令	对应英文	作用
01	tree [目录名]	tree	以树状图列出文件目录结构
02	cp 源文件 目标文件	copy	复制文件或者目录
03	mv 源文件 目标文件	move	移动文件或者目录/文件或者目录重命名

tree

tree命令可以以树状图列出文件目录结构

```
syq@syq-None:~/桌面$ touch a/b/123
syq@syq-None:~/桌面$ touch a/b/1234
syq@syq-None:~/桌面$ touch a/b/c/12345
syq@syq-None:~/桌面$ tree
.
```



选项	含义
-d	只显示目录

tree安装命令为: sudo apt-get install tree

. 表示当前目录(即桌面)

实践过程

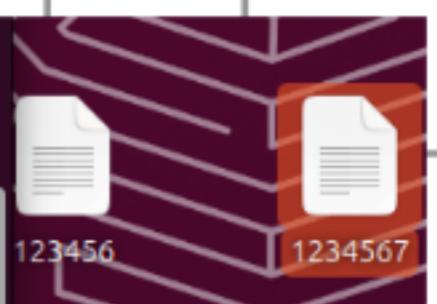
cp

cp命令的功能是将给出的文件或目录复制到另一个文件或目录中，相当于DOS下的copy命令

选项	含义
-f	已经存在的目标文件直接覆盖，不会提示
-i	覆盖文件前提示
-r	若给出的源文件是目录文件，则cp将递归复制该目录下的所有子目录和文件，目标文件必须为一个目录名

实践过程

```
syq@syq-None:~/桌面$ cp 123456
cp: 在 '123456' 后缺少要操作的目标文件
请尝试执行 "cp --help" 来获取更多信息。
syq@syq-None:~/桌面$ cp ~/下载/123456 ./123456
syq@syq-None:~/桌面$ cp ~/下载/123456 ./1234567
```



文件名称不改变使，目标文件可不写改文件名称，只写./(cp ~/下载/123456 ./)

- 改变复制到桌面的文件，再次用cp ~/下载/123456 ./时，下载目录中会弹出丢弃更改并重新载入的按钮，点击则覆盖桌面上改变过的文件

```
syq@syq-None:~/桌面$ cp -i ~/下载/123456 .
cp: 是否覆盖 './123456'? n
syq@syq-None:~/桌面$ cp -i ~/下载/123456 .
cp: 是否覆盖 './123456'? y
syq@syq-None:~/桌面$
```

n为no

y为yes，会弹出丢弃更改并重新载入

MV

MV命令可以用来移动文件或目录，也可以给文件或目录重命名

选项	含义
-i	覆盖文件前提示

```
syq@syq-None:~/桌面$ mv 123456 ~/下载/
```

移动文件
可用tree查看目录结构

```
syq@syq-None:~/桌面$ mv 123456 654321
```

- 如果桌面上仅有123456这个文件，此命令可以让它重命名为654321
如果两者都有123456，将完全覆盖654321的内容，123456将消失

查看文件内容

序号	命令	对应英文	作用
01	cat 文件名	concatenate	查看文件内容、创建文件、文件合并、追加文件内容等功能
02	more 文件名	more	分屏显示文件内容
03	grep 搜索文本文件名	grep	搜索文本文件内容

cat

cat命令可以用来查看文件内容、创建文件、文件合并、追加文件内容等功能
cat会一次显示所有的内容，适合查看内容较少的文本文件

选项	含义
-b	对非空输出行编号
-n	对输出的所有行编号

Linux中还有一个nl的命令和cat -b的效果等价

more

more命令可以用于分屏显示文件内容，每次只显示一页内容
适合于查看内容较多的文本文件

文件內容

实践过程

more命令显示结果
按空格可显示剩余

使用 more 的操作键：

操作键	功能
空格键	显示手册页的下一屏
Enter 键	一次滚动手册页的一行
b	回滚一屏
f	前滚一屏
q	退出
/word	搜索 word 字符串

grep

Linux系统中grep命令是一种强大的文本搜索工具

grep允许对文本文件进行模式查找,所谓模式查找又被称为正则表达式。

选项	含义
-n	显示匹配行及行号
-v	显示不包含匹配文本的所有行（相当于求反）
-i	忽略大小写

单词中间有空格的话搜索
内容要加引号

点Ctrl+f呼出文件搜索栏

常用的两种模式查找

参数	含义
^a	行首, 搜索以 a 开头的行
ke\$	行尾, 搜索以 ke 结束的行

```
syq@syq-None:~/桌面$ grep -n ^df 1234.txt
4:dfbjnxkzmxmddjknvckzzxdf
5:dfbjnxkzmxmddjknvckzzxdf
7:dfbjnxkzmxmddjknvckzzxdf
8:dfbjnxkzmxmddjknvckzzxdf
10:dfbjnxkzmxmddjknvckzzxdf
11:dfbjnxkzmxmddjknvckzzxdf
13:dfbjnxkzmxmddjknvckzzxdf
14:dfbjnxkzmxmddjknvckzzxdf
16:dfbjnxkzmxmddjknvckzzxdf
17:dfbjnxkzmxmddjknvckzzxdf
19:dfbjnxkzmxmddjknvckzzxdf
20:dfbjnxkzmxmddjknvckzzxdf
22:dfbjnxkzmxmddjknvckzzxdf
23:dfbjnxkzmxmddjknvckzzxdf
```

```
kaishi
4:dfbjnxkzmxmddjknvckzzxdf
5:dfbjnxkzmxmddjknvckzzxdf
7:dfbjnxkzmxmddjknvckzzxdf
8:dfbjnxkzmxmddjknvckzzxdf
10:dfbjnxkzmxmddjknvckzzxdf
11:dfbjnxkzmxmddjknvckzzxdf
13:dfbjnxkzmxmddjknvckzzxdf
14:dfbjnxkzmxmddjknvckzzxdf
16:dfbjnxkzmxmddjknvckzzxdf
17:dfbjnxkzmxmddjknvckzzxdf
19:dfbjnxkzmxmddjknvckzzxdf
20:dfbjnxkzmxmddjknvckzzxdf
22:dfbjnxkzmxmddjknvckzzxdf
23:dfbjnxkzmxmddjknvckzzxdf
syq@syq-None:~/桌面$ grep -v nx 1234.txt
-----
kaishi

syq@syq-None:~/桌面$ grep -vn nx 1234.txt
1:
2:kaishi
3:
6:
9:
12:
15:
18:
21:
```

实践过程

echo文字内容

echo会在终端中显示参数指定的文字,通常会和重定向联合使用
重定向>和>>

- Linux允许将命令执行结果重定向到一个文件
- 将本应显示在终端上的内容输出/追加到指定文件中其中
 - >表示输出,会覆盖文件原有的内容
 - >> 表示追加,会将内容追加到已有文件的末尾

```
syq@syq-None:~/桌面$ echo hello tianjin >1234.txt
syq@syq-None:~/桌面$ echo hello tianjin >234.txt
syq@syq-None:~/桌面$ cat 234.txt
hello tianjin
syq@syq-None:~/桌面$ ls -hl
总计 20K
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4.0K 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 14 10月 6 22:48 234.txt
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:23 a1
syq@syq-None:~/桌面$ ls -hl >a
bash: a: 是一个目录
syq@syq-None:~/桌面$ ls -hl >234.txt
syq@syq-None:~/桌面$ echo hello tianjin >>234.txt
syq@syq-None:~/桌面$ echo hello tianjin >> 234.txt
syq@syq-None:~/桌面$ echo hello tianjin >> 234.txt
syq@syq-None:~/桌面$ cat 234.txt
总计 16K
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4.0K 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 0 10月 6 22:49 234.txt
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:23 a1
hello tianjin
hello tianjin
hello tianjin
```

```
123456 cby 654321 1234.txt
总计 16K
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4.0K 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 0 10月 6 22:49 234.txt
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4.0K 10月 6 15:23 a1
```

使用echo修改文件内容
后,文件时间会改变

重命名可用来保存终端命令的结果

```
syq@syq-None:~/桌面$ echo hello >c
```

可用来创建文件 ↴

管道

Linux允许将一个命令的输出可以管道做为另一个命令的输入
可以理解现实生活中的管道,管道的一头塞东西进去,另一头取出来,
这里|的左右分两端,左端塞东西(写),右端取东西(读)

常用的管道命令有:

more: 分屏显示内容

grep: 在命令执行结果的基础上查询指定的文本

```

syq@syq-None:~/桌面$ ls -lha ~
总计 84K
drwxr-x--- 16 syq syq 4.0K 10月 6 16:32 .
drwxr-xr-x 3 root root 4.0K 9月 28 23:36 ..
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 公共
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 模板
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 视频
drwxr-xr-x 3 syq syq 4.0K 10月 4 17:38 图片
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 文档
drwxr-xr-x 2 syq syq 4.0K 10月 7 12:13 下载
drwxr-xr-x 2 syq syq 4.0K 9月 28 23:39 音乐
drwxr-xr-x 5 syq syq 4.0K 10月 6 22:56 桌面
-rw----- 1 syq syq 754 10月 6 13:14 .bash_history
-rw-r--r-- 1 syq syq 220 1月 7 2023 .bash_logout
-rw-r--r-- 1 syq syq 3.7K 1月 7 2023 .bashrc
drwx----- 12 syq syq 4.0K 10月 5 21:14 .cache
drwx----- 14 syq syq 4.0K 10月 6 16:54 .config
drwx----- 2 syq syq 4.0K 10月 4 13:48 .gnupg
-rw----- 1 syq syq 20 10月 5 21:08 .lessht
drwx----- 4 syq syq 4.0K 9月 28 23:39 .local
-rw-r--r-- 1 syq syq 807 1月 7 2023 .profile
drwx----- 4 syq syq 4.0K 9月 29 00:00 snap
drwx----- 2 syq syq 4.0K 9月 28 23:38 .ssh
-rw-r--r-- 1 syq syq 0 10月 6 16:32 .sudo_as_admin_successful
syq@syq-None:~/桌面$ ls -lha ~ | grep ba
-rw----- 1 syq syq 754 10月 6 13:14 .bash_history
-rw-r--r-- 1 syq syq 220 1月 7 2023 .bash_logout
-rw-r--r-- 1 syq syq 3.7K 1月 7 2023 .bashrc

```

截图软件推荐 snipaste

实践过程

远程管理常用命令

关机/重启

序号	命令	对应英文	作用
01	shutdown 选项 时间	shutdown	关机/重新启动

shutdown

● shutdown命令可以安全关闭或者重新启动系统

提示:

- 不指定选项和参数，默认表示1分钟之后关闭电脑
- 远程维护服务器时，最好不要关闭系统，而应该重新启动系统

```

syq@syq-None:~/桌面$ shutdown
Shutdown scheduled for Mon 2024-10-07 12:27:34 CST, use 'shutdown -c' to cancel.
syq@syq-None:~/桌面$ shutdown -c
syq@syq-None:~/桌面$ shutdown -r

```

选项	含义
-r	重新启动

常用命令示例

```

# 重新启动操作系统，其中 now 表示现在
$ shutdown -r now

# 立刻关机，其中 now 表示现在
$ shutdown now

# 系统在今天的 20:25 会关机
$ shutdown 20:25

# 系统再过十分钟后自动关机
$ shutdown +10

# 取消之前指定的关机计划
$ shutdown -c

```

查看或配置网卡信息

序号	命令	对应英文	作用
01	ifconfig	configure a network interface	查看/配置计算机当前的网卡配置信息
02	ping ip地址	ping	检测到目标 ip地址 的连接是否正常

网卡和IP地址

网卡

网卡是一个专门负责网络通讯的硬件设备

IP地址是设置在网卡上的地址信息

我们可以把电脑比作电话，网卡相当于SIM卡，IP地址相当于电话号码

IP地址

每台联网的电脑上都有IP地址，是保证电脑之间正常通讯的重要设置

注意：每台电脑的IP地址不能相同，否则会出现IP地址冲突，并且没有办法正常通讯

IPv4地址 192.168.19.128

IPv6地址 fe80::20c:29ff:fe5c:f136

ifconfig

ifconfig可以查看/配置计算机当前的网卡配置信息

安装ifconfig: sudo apt install net-tools

安装时出现无法定位用 sudo apt update

提示：

一台计算机中有可能会有一个物理网卡和多个虚拟网卡，在Linux中物理网卡的名字通常以ensxx表示

127.0.0.1被称为本地回环/环回地址，一般用来测试本机网卡是否正常

```
# 查看网卡配置信息  
$ ifconfig  
  
# 查看网卡对应的 IP 地址  
$ ifconfig | grep inet
```

可通过管道快速查找IP地址
inet为ipv4, inet6为ipv6

```
syq@syq-None:~/桌面$ ifconfig | grep inet  
inet 192.168.19.128 netmask 255.255.255.0 broadcast 192.168.19.255  
inet6 fe80::20c:29ff:fe5c:f136 prefixlen 64 scopeid 0x20<link>  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>
```

实践过程

ping

ping一般用于检测当前计算机到目标计算机之间的网络是否通畅，数值越大速度越慢

ping的工作原理与潜水艇的声纳相似，ping这个命令就是取自声纳的声音

网络管理员之间也常将ping用作动词——ping一下计算机X，看他是否开着

```
# 检测到目标主机是否连接正常  
$ ping IP地址
```

```
# 检测本地网卡工作正常  
$ ping 127.0.0.1
```

原理：网络上的机器都有唯一的IP地址，我们给目标IP地址发送一个数据包，对方就要返回一个数据包，根据返回的数据包以及时间，我们可以确定目标主机的存在。

提示：在Linux中，想要终止一个终端程序的执行，绝大多数都可以使用CTRL + C

```
syq@syq-None:~/桌面$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.088 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.091 ms
^C
--- 127.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.041/0.071/0.091/0.020 ms
```

发送6个数据包
最小、最大和平均时间
给本地计算机发送了一个56字节的数据包，本地网卡接收之后回了一个64字节的数据包，网络请求时间为time，有正常的回馈时间，说明工作正常

```
syq@syq-None:~/桌面$ ping www.baidu.com
PING www.a.shifen.com (180.101.50.188) 56(84) bytes of data.
64 bytes from 180.101.50.188 (180.101.50.188): icmp_seq=1 ttl=128 time=68.4 ms
64 bytes from 180.101.50.188 (180.101.50.188): icmp_seq=2 ttl=128 time=77.0 ms
64 bytes from 180.101.50.188 (180.101.50.188): icmp_seq=3 ttl=128 time=42.3 ms
64 bytes from 180.101.50.188 (180.101.50.188): icmp_seq=4 ttl=128 time=54.8 ms
64 bytes from 180.101.50.188 (180.101.50.188): icmp_seq=5 ttl=128 time=52.7 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 42.263/59.052/77.038/12.260 ms
```

使用ping命令时，

Ubuntu给目标电脑（例如：Windows XP）发送数据包（ttl）在接收回执，time记录所用时间



180.101.50.188也能进入百度网页，（与180.101.50.188:80一样）

远程登录和复制文件

序号	命令	对应英文	作用
01	ssh 用户名@ip	secure shell	关机/重新启动
02	scp 用户名@ip:文件名或路径 用户名@ip:文件名或路径	secure copy	远程复制文件

SSH基础（重点）

在Linux中SSH是非常常用的工具，通过SSH客户端我们可以连接到运行了SSH服务器的远程机器上。

数据传输是加密的，可以防止信息泄漏
数据传输是压缩的，可以提高传输速度



● SSH客户端是一种使用Secure Shell (SSH) 协议连接到远程计算机的软件程序。

● SSH是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。

● 利用SSH协议可以有效防止远程管理过程中的信息泄露
● 通过SSH协议可以对所有传输的数据进行加密，也能够防止DNS欺骗和IP欺骗。

● SSH的另一项优点是传输的数据可以是经过压缩的，所以可以加快传输的速度。

域名和端口号

域名

- 由一串用点分隔的名字组成，例如：www.itcast.cn

- 是IP地址的别名，方便用户记忆

端口号

- IP地址：通过IP地址找到网络上的计算机

- 端口号：通过端口号可以找到计算机上运行的应用程序

SSH服务器的默认端口号是22，如果是默认端口号，在连接的时候，可以省略

常见服务端口号列表：

序号	服务	端口号
01	SSH 服务器	22
02	Web 服务器	80
03	HTTPS	443
04	FTP 服务器	21

```
python@ubuntu:~$ cd .ssh
python@ubuntu:~/ssh$ ls -alh
总用量 8.0K
drwx----- 2 python python 4.0K 5月 8 18:00 .
drwxr-xr-x 33 python python 4.0K 5月 8 2017 ..
python@ubuntu:~/ssh$ ssh -p 22 itheima@172.16.140.1
The authenticity of host '172.16.140.1 (172.16.140.1)' can't be established.
ECDSA key fingerprint is SHA256:2hDyqoSgaBc/IIjFJRLDWSQbBsXig+xCk4kx8hIXhgg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.140.1' (ECDSA) to the list of known hosts.
Password:
liufan-mbp15:~ itheima$ ls
Applications Documents GitBook Movies Pictures
Desktop Downloads Library Music Public
liufan-mbp15:~ itheima$ cd Desktop/
liufan-mbp15:Desktop itheima$ ls
Python基础
liufan-mbp15:Desktop itheima$ exit
logout
Connection to 172.16.140.1 closed.
```

SSH客户端的简单使用

ssh [-p port] user@remote

user是在远程机器上的用户名，如果不指定的话默认认为当前用户
remote是远程机器的地址，可以是IP/域名，或者是后面会提到的别名

port是SSH Server监听的端口，如果不指定，就为默认值22

提示：使用exit退出当前用户的登录

注意：

SSH这个终端命令只能在Linux或者UNIX系统下使用

如果在Windows系统中安装PUTTY或者xshell客户端软件即可

提示：

在工作中，SSH服务器的端口号很有可能不是22，如果遇到这种情况，就需要使用-p选项，指定正确的端口号，否则无法正常连接到服务器

```
liufan-mbp15:~ itheima$ ls
Applications Documents GitBook Movies Pictures
Desktop Downloads Library Music Public
liufan-mbp15:~ itheima$ ssh -p 22 python@172.16.140.138
The authenticity of host '172.16.140.138 (172.16.140.138)' can't be established.
ECDSA key fingerprint is SHA256:/00d2GZ+OZ85kT+GachP5WrgpMeJwyWBOSfk1xTidXQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.140.138' (ECDSA) to the list of known hosts.
python@172.16.140.138's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-77-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

0 个可升级软件包。
0 个安全更新。
```

实践过程

→ Mac终端

远程链接拒绝，执行命令：

sudo apt get install openssh-server
(试过没用)

```
python@ubuntu:~$ ls
dbs  Documents  dump.rdb  Music  Public  Videos
Desktop  Downloads  examples.desktop  Pictures  Templates  workspace
python@ubuntu:~$ cd Desktop/
python@ubuntu:~/Desktop$ ls
123.txt  a  aaa  b  demo
python@ubuntu:~/Desktop$ touch 01.py
python@ubuntu:~/Desktop$ cat 01.py
print("hello python")
python@ubuntu:~/Desktop$ rm b
python@ubuntu:~/Desktop$ shutdown -r now
Failed to set wall message, ignoring: Interactive authentication required.
Failed to reboot system via logind: Interactive authentication required.
Failed to start reboot.target: Interactive authentication required.
See system logs and 'systemctl status reboot.target' for details.
Failed to open /dev/initctl: 权限不够
Failed to talk to init daemon.
python@ubuntu:~/Desktop$ sudo shutdown -r now
[sudo] python 的密码:
```

Ubuntu学习的命令，Mac终端可用

权限不够执行sudo shutdown -r now的命令

Windows 下 SSH 客户端的安装

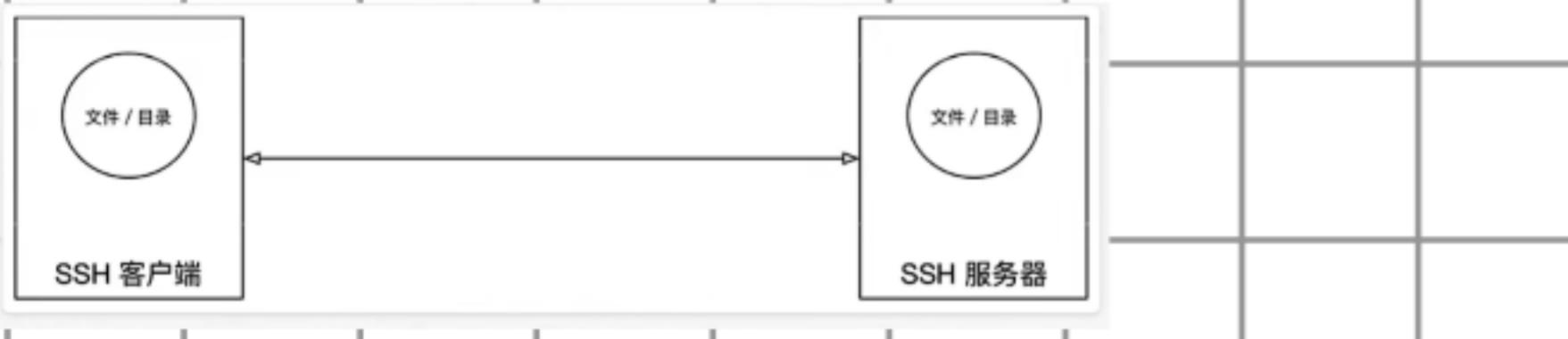
Putty <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

xShell <http://xshellcn.com>

SCP (掌握)

SCP 就是 secure copy, 是一个在 Linux 下用来进行远程拷贝文件的命令

它的地址格式与 ssh 基本相同, 需要注意的是, 在指定端口时用的是大写的 -P 而不是小写的 -p



```
# 把本地当前目录下的 01.py 文件 复制到 远程 家目录下的 Desktop/01.py  
# 注意: `:` 后面的路径如果不是绝对路径, 则以用户的家目录作为参照路径  
scp -P port 01.py user@remote:Desktop/01.py
```

```
# 把远程 家目录下的 Desktop/01.py 文件 复制到 本地当前目录下的 01.py  
scp -P port user@remote:Desktop/01.py 01.py
```

```
# 加上 -r 选项可以传送文件夹
```

```
# 把当前目录下的 demo 文件夹 复制到 远程 家目录下的 Desktop  
scp -r demo user@remote:Desktop
```

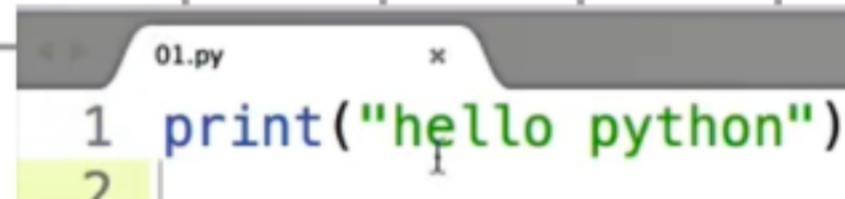
```
# 把远程 家目录下的 Desktop 复制到 当前目录下的 demo 文件夹
```

```
scp -r user@remote:Desktop demo
```

实践过程

```
liufan-mbp15:Desktop itheima$ scp -P 22 python@172.16.140.138:Desktop/01.py .  
python@172.16.140.138's password:  
01.py          100%   22    64.7KB/s  00:00  
liufan-mbp15:Desktop itheima$ scp -P 22 01.py python@172.16.140.138:Desktop  
python@172.16.140.138's password:  
01.py          100%   42    31.1KB/s  00:00  
liufan-mbp15:Desktop itheima$ scp -P 22 -r python@172.16.140.138:Desktop demo  
python@172.16.140.138's password:  
123.txt        100%  313   242.0KB/s  00:00  
322.txt        100%    0    0.0KB/s  00:00  
321.txt        100%    0    0.0KB/s  00:00  
123 (复件)    100%    0    0.0KB/s  00:00  
123            100%    0    0.0KB/s  00:00  
01.py          100%   42   165.4KB/s  00:00  
a              100%  479   2.0MB/s  00:00  
liufan-mbp15:Desktop itheima$
```

将乌班图桌面 01.py 复制到 mac 桌面
修改后再复制回乌班图桌面, 名称不变
的话, 将覆盖原文件



注意:

SCP 这个终端命令只能在 Linux 或者 UNIX 系统下使用

如果在 Windows 系统中, 可以安装 Putty, 使用 pscp 命行工具或者安装 Filezilla 使用 FTP 进行文件传输

Filezilla

官方网站: <https://www.filezilla.cn/download/client>

Filezilla 在传输文件时, 使用的是 FTP 服务而不是 SSH 服务, 因此端口号应该设置为 21

SSH 高级 (知道)

提示: 有关 SSH 配置信息都保存在用户家目录下的 .ssh 目录下

免密码登录

步骤

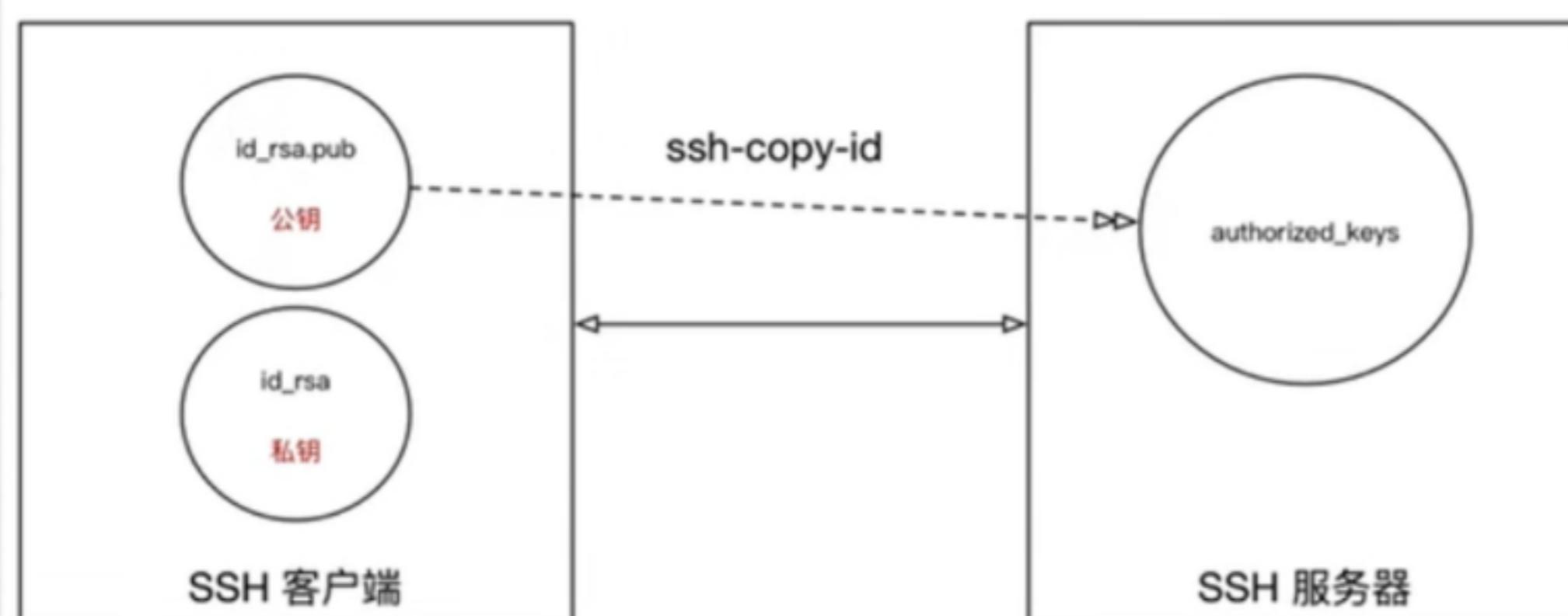
① 配置公钥

执行 ssh-keygen 即可生成 SSH 钥匙, 一路回车即可

② 上传公钥到服务器

执行 ssh-copy-id -p port user@remote, 可以让远程服务器记住我们的公钥

示意图



本地使用私钥对数据
进行加密/解密
服务器使用公钥对数
据进行加密/解密

```

python@ubuntu:~$ cd .ssh
python@ubuntu:~/ssh$ ls -alh
总用量 8.0K
drwx----- 2 python python 4.0K 5月 8 18:00 .
drwxr-xr-x 33 python python 4.0K 5月 8 2017 ..
python@ubuntu:~/ssh$ ssh -p 22 itheima@172.16.140.1
The authenticity of host '172.16.140.1 (172.16.140.1)' can't be established.
ECDSA key fingerprint is SHA256:2hDyqoSgaBc/IIjFJRLDWSQbBsXig+xCk4kx8hIXhqg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.140.1' (ECDSA) to the list of known hosts.
Password:
liufan-mbp15:~ itheima$ ls
Applications Documents GitBook Movies Pictures
Desktop Downloads Library Music Public
liufan-mbp15:Desktop itheima$ ls
Python基础
liufan-mbp15:Desktop itheima$ exit
logout
Connection to 172.16.140.1 closed.
python@ubuntu:~/ssh$ ls
known_hosts
python@ubuntu:~/ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/python/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/python/.ssh/id_rsa.
Your public key has been saved in /home/python/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ktHawKOAbOElGsm6hKF3sBk0Pte3vBdRfSFt6aZH4Xo python@ubuntu
The key's randomart image is:
+---[RSA 2048]---
|o=o. .o..o|
|*=*. o . o=|
|*+=. * o . +..|
|=++.. X . . =|
|.o .. + S . =|
| . . . o E |
| . . o |
| . |
+---[SHA256]---+
python@ubuntu:~/ssh$ ls
id_rsa id_rsa.pub known_hosts
python@ubuntu:~/ssh$ ssh-copy-id itheima@172.16.140.1
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'itheima@172.16.140.1'"
and check to make sure that only the key(s) you wanted were added.

python@ubuntu:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCA+MuVMywsFXBsF5WtRMA4g4ZmvpE7vEhx/CqmqpGAX
8npf+SRzwfZxtjW7latqOlm/JVPua4/H7cEsqenaKYe1i+bPkzV1IHcv0AVqht+IYqIjMuQMVQuFC0o
BEDt61Mbq7z5Liolibz6NGzw2PqvphfEJ/wLLlq75pFvTg0ycl9tDg6m8p87VPpz85IT0BWIVtkIBw0T
ywYgr6YZnlhviPIxSWkVCL8Fp0L8Ni0JIic0v4lOpZ1RaFdhFL0wQmPfHzacLud24SF2lgW9PuT0wvLe
LI7kpKMwId44c6zyU4Mg5Erjc8/V8/aFV31pTcv5Lbrfc1uSBkk7vimiRRUz python@ubuntu

```

非对称加密算法

使用公钥加密的数据，需要使用私钥解密

使用私钥加密的数据，需要使用公钥解密。

第一次远程链接一台主机时，授权信息输入yes后，Ubuntu上会出现一个known_hosts文件记录该主机地址

免密码登录

```

python@ubuntu:~/ssh$ ssh itheima@172.16.140.1
Last login: Fri Jan 1 09:31:13 2038 from 172.16.140.149
liufan-mbp15:~ itheima$ ls
Applications Documents GitBook Movies Pictures
Desktop Downloads Library Music Public
liufan-mbp15:~ itheima$ exit
logout
Connection to 172.16.140.1 closed.
python@ubuntu:~/ssh$ ssh itheima@172.16.140.1
Last login: Fri Jan 1 09:36:38 2038 from 172.16.140.149
liufan-mbp15:~ itheima$ exit
logout
Connection to 172.16.140.1 closed.
python@ubuntu:~/ssh$ cd
python@ubuntu:$ cd Desktop/
python@ubuntu:~/Desktop$ ls
01.py 123.txt a aaa demo
python@ubuntu:~/Desktop$ scp 01.py itheima@172.16.140.1:Desktop
01.py                                         100%   42    0.0KB/s  00:00
python@ubuntu:~/Desktop$
```

配置别名

每次都输入ssh -p port user@remote，时间久了会觉得很麻烦，特别是当user, remote和port都得输入，而且还不好记忆。而配置别名可以让我们进一步偷懒，譬如用：ssh mac来替代上面这么一长串，那么就在~/.ssh/config里面追加以下内容：

Host mac

HostName ip地址

User itheima

Port 22

指定别名名称

服务器的ip地址

远程服务器的名称

端口

保存之后，即可用ssh mac实现远程登录了，SCP同样可以使用

用户权限相关命令

用户和权限的基本概念

用户是Linux系统工作中重要的一环,用户管理包括用户与组管理

在Linux系统中,不论是由本机或是远程登录系统,每个系统都必须拥有一个账号,并且对于不同的系统资源拥有不同的使用权限

对文件/目录的权限包括:

序号	权限	英文	缩写	数字代号
01	读	read	r	4
02	写	write	w	2
03	执行	excute	x	1
04	无权限		-	0

要在目录下执行终端命令就必须有可执行权限,如果没有连cd进入目录的权限都没有

组

在实际应用中,可以预先针对组设置好权限,然后将不同的用户添加到对应的组中,从而不用依次为每一个用户设置权限

ls -l 扩展

ls -l 可以查看文件夹下文件的详细信息,从左到右依次是:

权限,第1个字符如果是d表示目录

硬链接数,通俗地讲,就是有多少种方式,可以访问到当前目录/文件

拥有者,家目录下文件/目录的拥有者通常都是当前用户

组,在Linux中,很多时候会出现组名和用户名相同的情况

大小

时间

名称

```
syq@syq-None:~/桌面$ ls -l
总计 24
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
-rw-rw-r-- 1 syq syq 6 10月 6 22:56 c
```

第一列syq是用户名,对应第一部分权限

第二列syq是组的名称,对应第二部分

第三部分对应其他用户

硬链接数,子目录越多,硬链接数越多,查找方法越多,例如a目录

绝对路径

a目录下cd.

b目录下cd..

文件权限示例	目录			拥有者权限			组权限			其他用户权限		
	-	r	w	-	r	w	-	r	-	-	-	-
目录权限示例	d	r	w	x	r	w	x	r	-	x		

chmod简单使用(重要)

chmod可以修改用户/组对文件/目录的权限

命令格式如下: chmod +|-rwx文件名|目录名

提示:以上方式会一次性修改拥有者/组权限

超级用户

- Linux系统中的root账号通常用于系统的维护和管理,对操作系统的所有资源具有所有访问权限
- 在大多数版本的Linux中,都不推荐直接使用root账号登录系统
- 在Linux安装的过程中,系统会自动创建一个用户账号,而这个默认的用户就称为“标准用户”

SUDO

SU是substitute user的缩写,表示使用另一个用户的身份

SUDO命令用来以其他身份来执行命令,预设的身份为root

用户使用SUDO时,必须先输入密码,之后有5分钟的有效期限,超过期限则必须重新输入密码

若其未经授权的用户企图使用SUDO,则会发出警告邮件给管理员

```
syq@syq-None:~/桌面 $ chmod +x python
```

```
syq@syq-None:~/桌面 $ ls -l
```

总计 24

```
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt  
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt  
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt  
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a  
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1  
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b  
-rwxrwxr-x 1 syq syq 50 10月 8 23:26 python
```

```
syq@syq-None:~/桌面 $ ./python
```

hello

hello

打开(O) ▾

123456

#!/usr/bin/python3

```
print("hello")  
print("hello")
```

python

~/桌面

c

执行一个文件可执行
命令./文件名

组管理和终端命令

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

提示:

组信息保存在/etc/group文件中

/etc目录是专门用来保存系统配置信息的目录

在实际应用中,可以预先针对组设置好权限,然后将不同的用户添加到对应的组中,从而不用依次为每一个用户设置权限

```
syq@syq-None:~$ groupadd dev  
groupadd: Permission denied.  
groupadd: 无法锁定 /etc/group, 请稍后再试。  
syq@syq-None:~$ sudo groupadd dev  
[sudo] syq 的密码:
```

```
syq@syq-None:~$ cat /etc/group
```

```
geodude:x:127:  
polkitd:x:994:  
rtkit:x:125:  
colord:x:126:  
gdm:x:127:  
nm-openvpn:x:128:  
lxde:x:129:  
gamemode:x:993:  
gnome-initial-setup:x:992:  
syq:x:1000:  
dev:x:1001:
```

实践过程

演练目标

1. 在syq用户的桌面文件夹下创建Python学习目录
2. 新建dev组
3. 将Python学习目录的组修改为dev

```

syq@syq-None:~$ mkdir Python学习
syq@syq-None:~$ mv Python学习 ./桌面
syq@syq-None:~$ sudo groupadd dev
syq@syq-None:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,syq
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:syq
floppy:x:25:
tape:x:26:
sudo:x:27:syq
audio:x:29:
dip:x:30:syq
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:syq
staff:x:50:
games:x:60:
users:x:100:syq
nogroup:x:65534:

```

```

rtkit:x:125:
colord:x:126:
gdm:x:127:
nm-openvpn:x:128:
lxde:x:129:
gamenode:x:993:
gnome-initial-setup:x:992:
syq:x:1000:
dev:x:1001:
syq@syq-None:~$ ls -l
总计 36
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 公共
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 模板
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 视频
drwxr-xr-x 3 syq syq 4096 10月 4 17:38 图片
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 文档
drwxr-xr-x 2 syq syq 4096 10月 7 12:13 下载
drwxr-xr-x 2 syq syq 4096 9月 28 23:39 音乐
drwxr-xr-x 6 syq syq 4096 10月 8 23:37 桌面
drwx----- 5 syq syq 4096 10月 7 13:24 snap
syq@syq-None:~$ cd 桌面
syq@syq-None:~/桌面$ ls -l
总计 28
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
-rwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 syq syq 4096 10月 8 23:36 Python学习
syq@syq-None:~/桌面$ chgrp -R dev Python学习/
chgrp: 更改 'Python学习/' 的所属组: 不允许的操作
syq@syq-None:~/桌面$ sudo chgrp -R dev Python学习/
syq@syq-None:~/桌面$ ls -l
总计 28
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
-rwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 syq dev 4096 10月 8 23:36 Python学习

```

用户管理终端命令

提示: 创建用户/删除用户/修改其他用户密码的终端命令都需要通过 **SUDO** 执行

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"> -m 自动建立用户家目录 -g 指定用户所在的组，否则会建立一个和同名的组
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

用户代号的信息保存在 etc 下
passwd 文件中
组代号信息保存在 etc 下 group
文件下

提示:

- 创建用户时，如果忘记添加-m 选项指定新用户的家目录——最简单的方法就是删除用户，重新创建
- 创建用户时，默认会创建一个和用户名同名的组名
- 用户信息保存在 /etc/passwd 文件中

```

syq@syq-None:~$ sudo useradd -m -g dev cby
[sudo] syq 的密码:
syq@syq-None:~$ ls -l /home
ls: 不适用的选项 -- /
请尝试执行 "ls --help" 来获取更多信息。
syq@syq-None:~$ ls -l /home
总计 8
drwxr-x--- 2 cby dev 4096 10月 8 23:47 cby
drwxr-x--- 16 syq syq 4096 10月 8 23:37 syq
syq@syq-None:~$ sudo passwd cby
新的密码:
无效的密码: 密码少于 8 个字符
重新输入新的密码:
抱歉, 密码不匹配。
新的密码:
无效的密码: 密码未通过字典检查 - ??????????/????????
重新输入新的密码:
passwd: 已成功更新密码

```

不指定密码就没办法实现远程实践过程

查看用户信息

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

如果在Windows XP是登录
Ubuntu的一个账户，使用
WHO也能显示出来

```
syq@syq-None:~$ id cby
uid=1001(cby) gid=1001(dev) 组=1001(dev)
syq@syq-None:~$ cat -n /etc/passwd | grep cby
 51 cby:x:1001:1001::/home/cby:/bin/sh
syq@syq-None:~$ cat -n /etc/group | grep dev
 34 plugdev:x:46:syq
 60 netdev:x:117:
 77 dev:x:1001:
```

```
syq@syq-None:~$ id
uid=1000(syq) gid=1000(syq) 组=1000(syq),4(adm),24(cdrom),27(sudo),30(dip),46(pl
ugdev),100(users),118(lpadmin)
syq@syq-None:~$ cat -n /etc/passwd | grep syq
 49 syq:x:1000:1000:syq:/home/syq:/bin/bash
syq@syq-None:~$ cat -n /etc/group | grep syq
 5 adm:x:4:syslog,syq
 18 cdrom:x:24:syq
 21 sudo:x:27:syq
 23 dip:x:30:syq
 34 plugdev:x:46:syq
 37 users:x:100:syq
 61 lpadmin:x:118:syq
 76 syq:x:1000:
```

```
syq@syq-None:~$ whoami
syq
syq@syq-None:~$ who
syq      seat0          2024-10-07 15:50 (login screen)
syq      tty2           2024-10-07 15:50 (tty2)
```

冒号分隔信息

用户名：x表示有密码；用户代号
1001：cby归属的组：cby用户的全
名：cby的家目录：登录的shell

adm表示管理员，syq用户具有管理
员这个组的权限

cdrom表示syq可以访问系统的光驱
sudo表示syq可以用sudo以root这
个身份执行系统的管理和维护等

时间是上次启动的时间

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

Ubuntu上默认的是dash，但是在
Windows XP上有缺陷，要用bash
(方便远程连接使用)

usermod

- usermod可以用来设置用户的主组/附加组和登录Shell
 - 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！

命令格式如下：

```
# 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 用户名
```

```
syq@syq-None:~$ id  
uid=1000(syq) gid=1000(syq) 组=1000(syq),4(adm),24(cdrom),27(sudo),30(dip),46(pl  
ugdev),100(users),118(lpadmin)  
syq@syq-None:~$ cat -n /etc/passwd | grep syq  
49 syq:x:1000:1000:syq:/home/syq:/bin/bash  
syq@syq-None:~$ cat -n /etc/group | grep syq  
5 adm:x:4:syslog,syq  
18 cdrom:x:24:syq  
21 sudo:x:27:syq  
23 dip:x:30:syq  
34 plugdev:x:46:syq  
37 users:x:100:syq  
61 lpadmin:x:118:syq  
76 syq:x:1000:
```

除了uid,gid,组,剩下的都是附加组,没有附加组就没有相对于的权限,比如cby不能执行sudo的命令(上上方实践过程显示)

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

```
syq@syq-None:~$ sudo usermod -G sudo cby  
[sudo] syq 的密码:  
syq@syq-None:~$ cat -n /etc/passwd | grep cby  
51 cby:x:1001:1001::/home/cby:/bin/sh  
syq@syq-None:~$ cat -n /etc/group | grep cby  
21 sudo:x:27:syq,cby
```

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:

```
syq@syq-None:~$ which passwd  
/usr/bin/passwd  
syq@syq-None:~$ ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 64152 2月 6 2024 /usr/bin/passwd  
syq@syq-None:~$ ls -l /usr/passwd  
ls: 无法访问 '/usr/passwd': 没有那个文件或目录  
syq@syq-None:~$ ls -l /etc/passwd  
-rw-r--r-- 1 root root 2925 10月 8 23:47 /etc/passwd
```

```
which ls  
# 输出  
# /bin/ls  
  
which useradd  
# 输出  
# /usr/sbin/useradd
```

etc下的不能执行

bin存储普通的可执行文件

sbin存储系统管理相关的文件

bin和sbin

● 在Linux中,绝大多数可执行文件都是保存在 /bin、/sbin、/usr/bin、/usr/sbin

● /bin (binary)是二进制执行文件目录,主要用于具体应用

● /sbin (system binary)是系统管理员专用的二进制代码存放目录,主要用于系统管理

● /usr/bin (user commands for applications)后期安装的一些软件

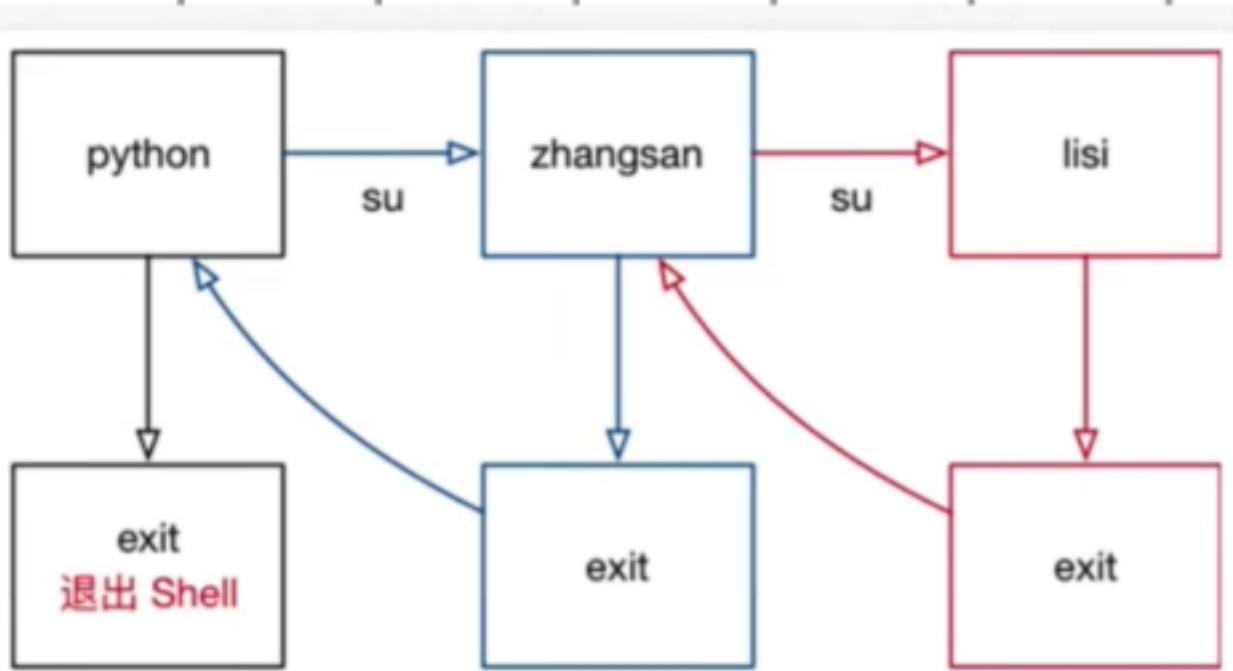
● /usr/sbin (super user commands for applications)超级用户的一些管理程序

提示:cd这个终端命令是内置在系统内核中的,没有独立的文件,因此用which无法找

切换用户

序号	命令	作用	说明
01	su - 用户名	切换用户，并且切换目录	- 可以切换到用户家目录，否则保持位置不变
02	exit	退出当前登录账户	

SU不接用户名,可以切换到root,但是不推荐使用,因为不安全
exit示意图如下:



```
python@ubuntu:~$ su - zhangsan
密码:
zhangsan@ubuntu:~$ pwd
/home/zhangsan
zhangsan@ubuntu:~$ exit
注销
python@ubuntu:~$ pwd
/home/python
python@ubuntu:~$ su -
密码:
root@ubuntu:~# pwd
/root
root@ubuntu:~# whoami
root
root@ubuntu:~# useradd -m -g dev zhaoliu
root@ubuntu:~# cat -n /etc/passwd
```

实践过程

没有密码无法切换,可以用passwd用户名修改密码

修改文件权限

序号	命令	作用
01	chown	修改拥有者
02	chgrp	修改组
03	chmod	修改权限

命令格式如下

直接修改文件|目录的 读|写|执行 权限,但是不能精确到 拥有者|组|其他
chmod +/ -rwx 文件名|目录名

chmod在设置权限时,可以简单地使用三个数字分别对应 拥有者 / 组 和 其他用户的权限

7表示文件或者目录拥有者的权限
5表示文件或目录组用户的权限
3表示文件或目录其他用户的权限

拥有者	组	其他						
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

4	2	1	7	rw-
4	2	0	6	rw-
4	0	1	5	r-x
4	0	0	4	r--
0	2	1	3	-wx
0	2	0	2	-w-
0	0	1	1	--x
0	0	0	0	---

```
syq@syq-None:~/桌面$ sudo chown cby Python学习/
syq@syq-None:~/桌面$ ls -l
总计 28
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
-rw-rwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby dev 4096 10月 8 23:36 Python学习
syq@syq-None:~/桌面$ id
uid=1000(syq) gid=1000(syq) 组=1000(syq),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),100(users),118(lpadmin)
syq@syq-None:~/桌面$ sudo chgrp cby Python学习/
chgrp: 无效的组: "cby"
syq@syq-None:~/桌面$ sudo chmod syq Python学习/
syq@syq-None:~/桌面$ ls -l
总计 28
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 6 17:57 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
-rw-rwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby syq 4096 10月 8 23:36 Python学习
```

见数字组合有 (u表示用户/g表示组/o表示其他):

777 ==> u=rwx, g=rwx, o=rwx

755 ==> u=rwx, g=rwx, o=r

644 ==> u=rw, g=r, o=r

chmod演练目标

1. 将 01.py 的权限修改为 u=rwx, g=rwx, o=r

2. 将 123.txt 的权限修改为 u=rw, g=r, o=-

3. 将 test 自录以及目录下的所有文件权限修改为

u=rwx, g=rwx, o=rwx

```

syq@syq-None:~/桌面$ chmod 754 01.py
syq@syq-None:~/桌面$ ls -l
总计 32
-rwxr-xr-- 1 syq syq 0 10月 10 00:04 01.py
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxrwxr-x 2 syq syq 4096 10月 10 00:04 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
drwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby syq 4096 10月 8 23:36 Python学习
drwxrwxr-x 2 syq syq 4096 10月 10 00:05 test
syq@syq-None:~/桌面$ chmod 640 123.txt
syq@syq-None:~/桌面$ chmod -R 775 test
syq@syq-None:~/桌面$ ls -l
总计 32
-rwxr-xr-- 1 syq syq 0 10月 10 00:04 01.py
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drwxr----- 2 syq syq 4096 10月 10 00:04 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 a
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
drwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby syq 4096 10月 8 23:36 Python学习
drwxrwxr-x 2 syq syq 4096 10月 10 00:05 test

```

系统信息相关命令

本节内容主要是为了方便通过远程终端维护服务器时，查看服务器上当前系统日期和时间/磁盘空间占用情况/程序执行情况

本小节学习的终端命令基本都是查询命令，通过这些命令对系统资源的使用情况有个了解

时间和日期

序号	命令	作用
01	cal	查看日历，-y选项可以查看一年的日历
02	date	查看系统时间

磁盘信息

序号	命令	作用
01	df -h	disk free 显示磁盘剩余空间
02	du -h [目录名]	disk usage 显示目录下的文件大小

```

syq@syq-None:~/桌面$ df
文件系统 1K的块 已用 可用 已用% 挂载点
tmpfs 396320 1920 394400 1% /run
/dev/sda2 20463184 10889940 8508440 57% /
tmpfs 1981584 0 1981584 0% /dev/shm
tmpfs 5120 8 5112 1% /run/lock
tmpfs 396316 88 396228 1% /run/user/1000
/dev/sr0 5052730 5052730 0 100% /media/syq/Ubuntu 23.10.1 amd64
syq@syq-None:~/桌面$ du -h
4.0K ./123.txt
4.0K ./a/b/c/d
8.0K ./a/b/c
12K ./a/b
16K ./a
4.0K ./test
4.0K ./a1/b1/c1/d1
8.0K ./a1/b1/c1
12K ./a1/b1
16K ./a1
4.0K ./Python学习
60K .

```

du -h显示所有子目录
60K是总数

syq@syq-None:~\$ date	2024年 10月 10日 星期四 13:15:26 CST
syq@syq-None:~\$ cal	十月 2024
	日 一 二 三 四 五 六
	1 2 3 4 5
	6 7 8 9 10 11 12
	13 14 15 16 17 18 19
	20 21 22 23 24 25 26
	27 28 29 30 31
syq@syq-None:~\$ cal -y	2024
	一月 二月 三月
	日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
	1 2 3 4 5 6 1 2 3 4 5 6 1 2
	7 8 9 10 11 12 13 4 5 6 7 8 9 10 3 4 5 6 7 8 9
	14 15 16 17 18 19 20 11 12 13 14 15 16 17 18 11 12 13 14 15 16
	21 22 23 24 25 26 27 18 19 20 21 22 23 24 25 17 18 19 20 21 22 23
	28 29 30 31 25 26 27 28 29 29 30 31 24 25 26 27 28 29 30
	31
	四月 五月 六月
	日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
	1 2 3 4 5 6 1 2 3 4 5 6 1 2
	7 8 9 10 11 12 13 5 6 7 8 9 10 11 2 3 4 5 6 7 8
	14 15 16 17 18 19 20 12 13 14 15 16 17 18 9 10 11 12 13 14 15
	21 22 23 24 25 26 27 19 20 21 22 23 24 25 16 17 18 19 20 21 22
	28 29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29
	38
	七月 八月 九月
	日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
	1 2 3 4 5 6 1 2 3 1 2 3 4 5 6 7
	7 8 9 10 11 12 13 4 5 6 7 8 9 10 8 9 10 11 12 13 14
	14 15 16 17 18 19 20 11 12 13 14 15 16 17 15 16 17 18 19 20 21 22
	21 22 23 24 25 26 27 18 19 20 21 22 23 24 22 23 24 25 26 27 28
	28 29 30 31 25 26 27 28 29 30 31 23 24 25 26 27 28 29
	38
	十月 十一月 十二月
	日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
	1 2 3 4 5 1 2 3 1 2 3 4 5 6 7
	6 7 8 9 10 11 12 3 4 5 6 7 8 9 8 9 10 11 12 13 14

实践过程

选项说明

参数	含义
-h	以人性化的方式显示文件大小

Windows XP右击-任务管理器可以看到进程

进程信息

所谓进程，通俗地说就是 当前正在执行的一个程序

序号	命令	作用
01	ps aux	process status 查看进程的详细状况
02	top	动态显示运行中的进程并且排序
03	kill [-9] 进程代号	终止指定代号的进程，-9 表示强行终止

ps默认只会显示当前用户通过终端启动的应用程序

ps选项说明功能

选项	含义
a	显示终端上的所有进程，包括其他用户的进程
u	显示进程的详细状态
x	显示没有控制终端的进程

如果用CPU或者内存占用率比较高，可以使
用kill终止进程。

提示：使用kill命令时，最好只终止由当前用户开启的进程，而
不要终止root身份开启的进程，否则可能导致系统崩溃

要退出top可以直接输入q

```

syq@syq-None:~/桌面$ ps
  PID TTY      STAT TIME COMMAND
 2555 pts/0    0:00:00 bash
 2791 pts/0    0:00:00 ps
syq@syq-None:~/桌面$ ps a
  PID TTY      STAT TIME COMMAND
1484 tty2    Ssl+  0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSIO
1494 tty2    Sl+   0:00 /usr/libexec/gnome-session-binary --session=ubuntu
 2555 pts/0    Ss    0:00 bash
 2792 pts/0    R+   0:00 ps a
syq@syq-None:~/桌面$ ps au
USER      PID %CPU %MEM   VSZ   RSS TTY      STAT START  TIME COMMAND
syq      1484  0.0  0.1 235968  6400 tty2    Ssl+ 13:11  0:00 /usr/libexec/gdm-
syq      1494  0.0  0.4 298704 16384 tty2    Sl+ 13:11  0:00 /usr/libexec/gnom
syq      2555  0.0  0.1 11252   5376 pts/0    Ss 13:12  0:00 bash
syq      2793  500  0.1 14196   5120 pts/0    R+ 13:22  0:00 ps au

```

```

syq@syq-None:~/桌面$ top
top - 13:26:39 up 15 min, 2 users, load average: 0.17, 0.32, 0.41
任务: 288 total, 1 running, 287 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.2 sy, 0.0 ni, 95.3 id, 0.5 wa, 0.0 hi, 0.2 sl, 0.0 st
MiB Mem : 3870.3 total, 1948.3 free, 1244.3 used, 954.6 buff/cache
MiB Swap: 3868.0 total, 3868.0 free, 0.0 used. 2626.0 avail Mem

进程号 USER PR NI VIRT RES SHR %CPU %MEM TIME+ COMMAND
1997 syq 20 0 463480 11392 9600 S 1.3 0.3 0:00.28 gvfs-udisks+
 1 root 20 0 178584 14524 9148 S 1.0 0.4 0:00.21 systemd
1664 syq 20 0 3926972 271372 142568 S 0.7 6.8 0:30.58 gnome-shell
 41 root 20 0 0 0 I 0.3 0.6 0:02.25 kworker/1:1+
326 root 19 -1 50224 17280 16000 S 0.3 0.4 0:01.23 systemd-jou+
360 root -51 0 0 0 S 0.3 0.0 0:00.36 irq/16-vmwg+
473 systemd+ 20 0 17076 7296 6528 S 0.3 0.2 0:02.29 systemd-oomd
919 root 20 0 469284 14164 11476 S 0.3 0.4 0:00.45 udisksd
1354 syq 20 0 20748 11984 8960 S 0.3 0.3 0:01.66 systemd
1920 syq 20 0 216040 39212 30216 S 0.3 1.0 0:03.86 vmtoolsd
2252 syq 20 0 536180 9216 7936 S 0.3 0.2 0:00.09 gvfsd-trash
2320 syq 39 19 717040 32632 21900 S 0.3 0.8 0:02.24 tracker-min+
2528 syq 20 0 787960 70472 55892 S 0.3 1.8 0:06.92 gnome-terminal
2926 syq 20 0 14600 5632 3456 R 0.3 0.1 0:00.72 top
 2 root 20 0 0 0 S 0.0 0.0 0:00.09 kthreadd
 3 root 0 -20 0 0 I 0.0 0.0 0:00.00 rCU_gp
 4 root 0 -20 0 0 I 0.0 0.0 0:00.00 rCU_par_gp
 5 root 0 -20 0 0 I 0.0 0.0 0:00.00 slab_flushwq
 6 root 0 -20 0 0 I 0.0 0.0 0:00.00 netns

```

其他命令

查找文件

find 命令功能强大，通常用在特定的目录下搜索符合条件的文件。

序号	命令	作用
01	find [路径] -name "*.py"	查找指定路径下扩展名是 .py 的文件，包括子目录

如果省略路径，表示在当前文件夹下查找
之前学习的通配符，在使用 find 命令时同时可用

演练目标

1. 搜索桌面 目录下，文件名包含 1 的文件 find -name "*1*"
2. 搜索桌面 目录下，所有以 .txt 为扩展名的文件 find -name "*.txt"

```

syq@syq-None:~$ tree 桌面/
桌面/
├── 01.py
├── 1234.txt
├── 123.txt
└── 234.txt
    └── a
        └── b
            ├── 123
            └── 1234
                └── c
                    └── 12345
            └── d
                └── 12345
        └── a1
            └── b1
                └── c1
                    └── d1
            └── a1
        └── python
        └── Python学习
        └── test
12 directories, 8 files
syq@syq-None:~$ find 桌面/ -name "*1*"
桌面/123.txt
桌面/a/b/c/12345
桌面/a/b/1234
桌面/a/b/123
桌面/1234.txt
桌面/01.py
桌面/a1
桌面/a1/b1
桌面/a1/b1/c1
桌面/a1/b1/c1/d1

```

```

syq@syq-None:~/桌面$ find -name "*.txt"
./123.txt
./1234.txt
./234.txt

```

软链接

序号	命令	作用
01	ln -s 被链接的源文件 链接文件	建立文件的软链接，用通俗的方式讲类似于 Windows 下的快捷方式

注意：1. 没有 -s 选项建立的是一个硬链接文件
两个文件占用相同大小的硬盘空间，工作中几乎不会建立文
件的硬链接

2. 原文件要使用绝对路径，不能使用相对路径，这样可以方便移
动链接文件后，仍然能够正常使用

bash 是正在运行的终端
ps 是正在运行的程序命令

ps aux 还包括在系统启动时同时启
动在后台工作的进程
COMMON 运行名称 CPU 占用率
PID 是进程代号

top 动态显示
可以查看哪一个进程异常

演练目标

- 将桌面目录下的 01.py 移动到 demo/b/c 目录下
- 在桌面目录下新建 01.py 的软链接 FirstPython
- 分别使用相对路径和绝对路径建立 FirstPython 的软链接
- 将 FirstPython 移动到 demo 目录下, 对比使用相对路径和绝对路径的区别

```
syq@syq-None:~/桌面$ mv 01.py demo/b/c
syq@syq-None:~/桌面$ tree
.
├── 1234.txt
├── 123.txt
├── 234.txt
├── a1
│   └── b1
│       └── c1
│           └── d1
└── b
    ├── demo
    │   └── b
    │       ├── 123
    │       ├── 1234
    │       └── c
    │           └── 01.py
    └── python
        └── Python学习
        └── test
2 directories, 8 files
```

```
syq@syq-None:~/桌面$ ls -l
总计 32
lrwxrwxrwx 1 syq syq 14 10月 10 23:08 01_xiangdui -> demo/b/c/01.py
-rw-rw-r-- 1 syq syq 364 10月 6 22:47 1234.txt
drw-r----- 2 syq syq 4096 10月 10 00:04 123.txt
-rw-rw-r-- 1 syq syq 288 10月 6 22:52 234.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 demo
-rwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby syq 4096 10月 8 23:36 Python学习
drwxrwxr-x 2 syq syq 4096 10月 10 00:05 test
```

```
syq@syq-None:~/桌面$ ln -s /home/syq/桌面/demo/b/c/01.py 01_juedui
```

```
syq@syq-None:~/桌面$ cat 01_juedui
```

```
#!/usr/bin/python3
```

```
print("hello")
```

```
print("hello")
```

```
syq@syq-None:~/桌面$ ls -l
```

```
总计 32
```

```
lrwxrwxrwx 1 syq syq 31 10月 10 23:15 01_juedui -> /home/syq/桌面/demo/b/c/01.py
lrwxrwxrwx 1 syq syq 14 10月 10 23:08 01_xiangdui -> demo/b/c/01.py
drw-r----- 1 syq syq 364 10月 6 22:47 1234.txt
drw-r----- 2 syq syq 4096 10月 10 00:04 123.txt
drwxrwxr-x 3 syq syq 4096 10月 6 15:23 a1
-rw-rw-r-- 1 syq syq 0 10月 6 22:56 b
drwxrwxr-x 3 syq syq 4096 10月 6 15:21 demo
-rwxrwxr-x 1 syq syq 50 10月 8 23:26 python
drwxrwxr-x 2 cby syq 4096 10月 8 23:36 Python学习
drwxrwxr-x 2 syq syq 4096 10月 10 00:05 test
```

```
print("hello")
```

```
print("hello")
```

```
syq@syq-None:~/桌面$
```

```
syq@syq-None:~/桌面$ mv 01* demo
syq@syq-None:~/桌面$ tree
.
├── 1234.txt
├── 123.txt
├── 234.txt
├── a1
│   └── b1
│       └── c1
│           └── d1
└── b
    ├── demo
    │   ├── 01_juedui -> /home/syq/桌面/demo/b/c/01.py
    │   └── 01_xiangdui -> demo/b/c/01.py
    └── python
        └── Python学习
        └── test
12 directories, 10 files
```

```
syq@syq-None:~/桌面$ cd demo
```

```
syq@syq-None:~/桌面/demo$ ls -l
```

```
总计 4
```

```
lrwxrwxrwx 1 syq syq 31 10月 10 23:15 01_juedui -> /home/syq/桌面/demo/b/c/01.py
```

```
lrwxrwxrwx 1 syq syq 14 10月 10 23:08 01_xiangdui -> demo/b/c/01.py
```

```
drwxrwxr-x 3 syq syq 4096 10月 6 16:40 b
```

```
syq@syq-None:~/桌面/demo$ cat 01_juedui
```

```
#!/usr/bin/python3
```

```
print("hello")
```

```
print("hello")
```

```
syq@syq-None:~/桌面/demo$
```

```
cat: 01_xiangdui: 没有那个文件或目录
```

剪头指向链接所在位置

用绝对路径建立软连接,无论移动到哪个目录都可以访问到它对应链接的文件
相对路径建立的软连接不能移动位置

硬链接简介(知道)

在使用 ln 创建链接时,如果没有 -s 选项,会创建一个硬链接

硬链接演练

- 在 ~/Desktop/demo 目录下建立 ~/Desktop/demo/b/c/01.py 的硬链接 01_hard
- 使用 ls -l 查看文件的硬链接数(硬链接—有多少种方式可以访问文件或者目录)
- 删除 ~/Desktop/demo/b/c/01.py,并且使用 tree 来确认 demo 目录下的三个链接文件

```
syq@syq-None:~/桌面/demo$ tree
.
├── 01_juedui -> /home/syq/桌面/demo/b/c/01.py
└── 01_xiangdui -> demo/b/c/01.py
.
├── b
│   ├── 123
│   ├── 1234
│   └── c
│       └── 01.py
│           └── 12345
└── d
4 directories, 6 files
```

```
syq@syq-None:~/桌面/demo$ ln /home/syq/桌面/demo/b/c/01.py 01_hard
```

```
syq@syq-None:~/桌面/demo$ tree
```

```
.
├── 01_hard
│   ├── 01_juedui -> /home/syq/桌面/demo/b/c/01.py
│   └── 01_xiangdui -> demo/b/c/01.py
└── b
    ├── 123
    ├── 1234
    └── c
        └── 01.py
            └── 12345
4 directories, 7 files
```

```
syq@syq-None:~/桌面/demo$ ls -l
```

```
总计 8
```

```
-rwxr-xr-- 2 syq syq 50 10月 10 23:09 01_hard
lrwxrwxrwx 1 syq syq 31 10月 10 23:15 01_juedui -> /home/syq/桌面/demo/b/c/01.py
lrwxrwxrwx 1 syq syq 14 10月 10 23:08 01_xiangdui -> demo/b/c/01.py
drwxrwxr-x 3 syq syq 4096 10月 6 16:40 b
syq@syq-None:~/桌面/demo$ cat 01_hard
#!/usr/bin/python3
```

```
syq@syq-None:~/桌面/demo$ tree
.
├── 01_hard
│   ├── 01_juedui -> /home/syq/桌面/demo/b/c/01.py
│   └── 01_xiangdui -> demo/b/c/01.py
└── b
    ├── 123
    ├── 1234
    └── c
        └── 01.py
        └── 12345
4 directories, 6 files
```

```
syq@syq-None:~/桌面/demo$ cat 01_hard
```

```
#!/usr/bin/python3
```

```
print("hello")
```

```
print("hello")
```

```
syq@syq-None:~/桌面/demo$ cat 01_xiangdui
```

```
cat: 01_xiangdui: 没有那个文件或目录
```

```
syq@syq-None:~/桌面/demo$ cat 01_juedui
```

```
cat: 01_juedui: 没有那个文件或目录
```

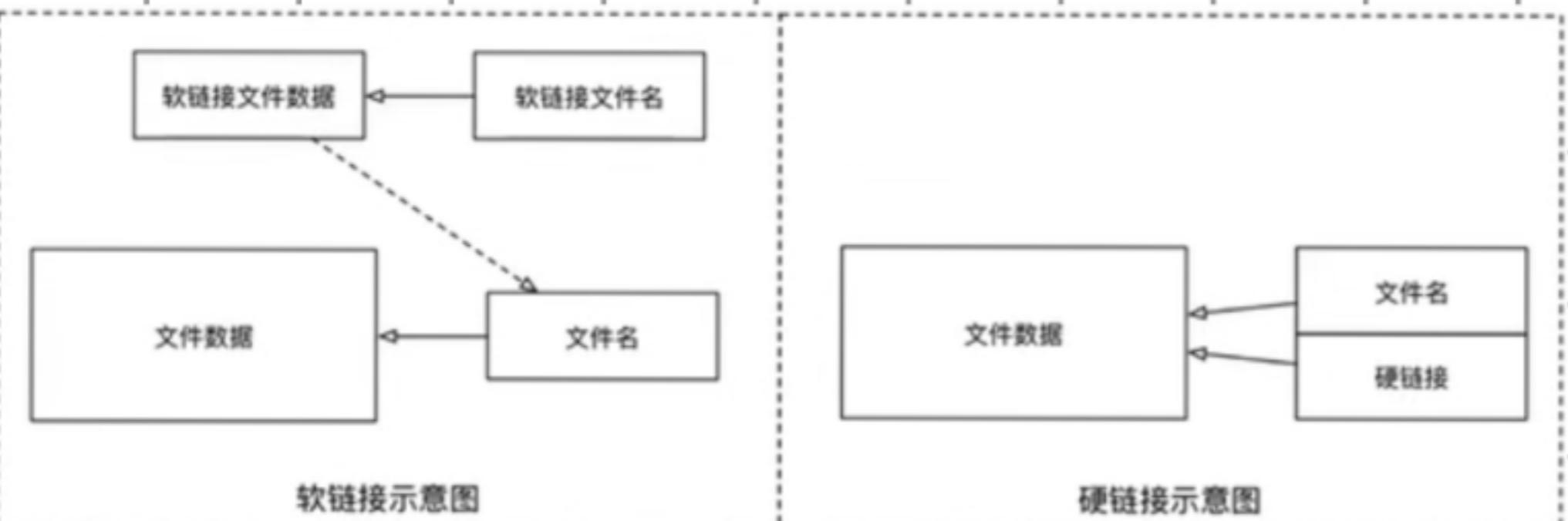
```
syq@syq-None:~/桌面/demo$ ls -l
```

```
总计 8
```

```
-rwxr-xr-- 1 syq syq 50 10月 10 23:09 01_hard
lrwxrwxrwx 1 syq syq 31 10月 10 23:15 01_juedui -> /home/syq/桌面/demo/b/c/01.py
lrwxrwxrwx 1 syq syq 14 10月 10 23:08 01_xiangdui -> demo/b/c/01.py
drwxrwxr-x 3 syq syq 4096 10月 6 16:40 b
```

删除原文件 硬链接内容不会受到影响
软链接内容会受影响
(直观差异)

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

提示：

在Linux中，只有文件的硬链接数 == 0 才会被删除

使用ls -l 可以查看一个文件的硬链接的数量

在日常工作中，几乎不会建立文件的硬链接，知道即可

打包压缩

打包压缩是日常工作中备份文件的一种方式

在不同操作系统中，常用的打包压缩方式是不同的

Windows 常用 rar

Mac 常用 zip

Linux 常用 tar.gz

打包 / 解包

tar 是 Linux 中最常用的备份工具，此命令可以把一系列文件打包到一个大文件中，也可以把一个打好的大文件恢复成一系列文件

tar 的命令格式如下：只负责打包不负责压缩

```
# 打包文件
tar -cvf 打包文件.tar 被打包的文件 / 路径...
# 解包文件
tar -xvf 打包文件.tar
```

选项说明

选项	含义
c	生成档案文件，创建打包文件
x	解开档案文件
v	列出归档解档的详细过程，显示进度
f	指定档案文件名称，f 后面一定是 .tar 文件，所以必须放选项最后

注意：f 选项必须放在最后，其他选项顺序可以随意

打包解包演练

1. 删除桌面下的所有内容

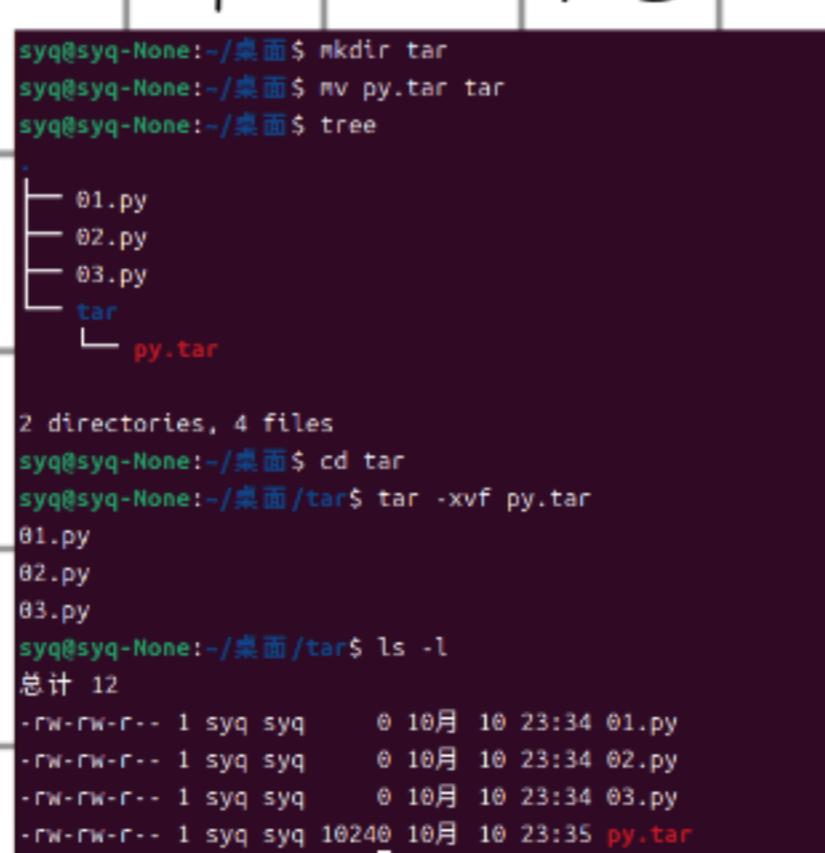
2. 在桌面下新建三个空白文件 01.py、02.py、03.py

3. 将这三个文件打一个 py.tar 的包

4. 新建 tar 目录，并且将 py.tar 移动到 tar 目录下

5. 解包 py.tar

```
syq@syq-None:~/桌面$ rm -r *
syq@syq-None:~/桌面$ ls -l
总计 0
syq@syq-None:~/桌面$ touch 01.py
syq@syq-None:~/桌面$ touch 02.py
syq@syq-None:~/桌面$ touch 03.py
syq@syq-None:~/桌面$ tar -cvf py.tar 01.py 02.py 03.py
01.py
02.py
03.py
syq@syq-None:~/桌面$ ls -l
总计 12
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 01.py
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 02.py
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 03.py
-rw-rw-r-- 1 syq syq 18248 10月 10 23:35 py.tar
syq@syq-None:~/桌面$ ls -lh
总计 12K
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 01.py
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 02.py
-rw-rw-r-- 1 syq syq 0 10月 10 23:34 03.py
-rw-rw-r-- 1 syq syq 10240 10月 10 23:35 py.tar
```



压缩/解压缩

gzip

tar与gzip命令结合可以使用实现文件打色和压缩

tar只负责打色文件,但不压缩

用gzip压缩tar打色后的文件,其扩展名一般用xxx.tar.gz

在Linux中最常见的压缩文件格式就是xxx.tar.gz

在tar命令中有一个选项-**v**可以调用gzip,从而可以方便的实现压缩和解压缩的功能
命令格式如下:

实践过程

```
# 压缩文件
tar -zcvf 打包文件.tar.gz 被压缩的文件 / 路径...
# 解压缩文件
tar -zxvf 打包文件.tar.gz
# 解压缩到指定路径
tar -zxvf 打包文件.tar.gz -C 目标路径
```

选项	含义
-C	解压缩到指定目录,注意:要解压缩的目录必须存在

```
syq@syq-None:~/桌面/tar$ tar -zcvf py.tar.gz 01.py
01.py
02.py
03.py
syq@syq-None:~/桌面/tar$ ls -l
总计 16
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 01.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 02.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 03.py
-rw-rw-r-- 1 syq syq 10240 10月 10 23:35 py.tar
-rw-rw-r-- 1 syq syq   133 10月 10 23:41 py.tar.gz
```

压缩后字节减少很多(133)

```
syq@syq-None:~/桌面/tar$ mkdir gz
syq@syq-None:~/桌面/tar$ mv py.tar.gz gz
syq@syq-None:~/桌面/tar$ tree
.
├── 01.py
├── 02.py
├── 03.py
└── gz
    └── py.tar.gz
py.tar
2 directories, 5 files
syq@syq-None:~/桌面/tar/gz$ tar -zcvf py.tar.gz 01.py
01.py
02.py
03.py
syq@syq-None:~/桌面/tar/gz$ ls -l
总计 4
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 01.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 02.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 03.py
-rw-rw-r-- 1 syq syq   133 10月 10 23:41 py.tar.gz
```

bzip2(two)

tar与bzip2命令结合可以使用实现文件打色和压缩(用法和gzip一样)

tar只负责打色文件,但不压缩,

用bzip2压缩tar打色后的文件,其扩展名一般用xxx.tar.bz2

在tar命令中有一个选项-**v**可以调用bzip2,从而可以方便的实现压缩和解压缩的功能

命令格式如下:

```
# 压缩文件
tar -jcvf 打包文件.tar.bz2 被压缩的文件 / 路径...
# 解压缩文件
tar -jxvf 打包文件.tar.bz2
```

```
syq@syq-None:~/桌面/tar/gz$ tar -jcvf py.tar.bz2 *.py
01.py
02.py
03.py
syq@syq-None:~/桌面/tar/gz$ ls -l
总计 8
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 01.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 02.py
-rw-rw-r-- 1 syq syq    0 10月 10 23:34 03.py
-rw-rw-r-- 1 syq syq 143 10月 10 23:49 py.tar.bz2
-rw-rw-r-- 1 syq syq   133 10月 10 23:41 py.tar.gz
```

压缩文件
和gzip压缩效果差不多

```
syq@syq-None:~/桌面$ tree
.
├── 01.py
├── 02.py
├── 03.py
└── bz2
    ├── tar
    │   ├── 01.py
    │   ├── 02.py
    │   ├── 03.py
    └── gz
        ├── 01.py
        ├── 02.py
        ├── 03.py
        └── py.tar.bz2
        └── py.tar.gz
        └── py.tar
4 directories, 12 files
syq@syq-None:~/桌面$ cd ./tar/gz
syq@syq-None:~/桌面/tar/gz$ tar -jxvf py.tar.bz2 -C /home/syq/桌面/bz2
01.py
02.py
03.py
```

```
syq@syq-None:~/桌面$ tree
.
├── 01.py
├── 02.py
├── 03.py
└── bz2
    ├── 01.py
    ├── 02.py
    ├── 03.py
    └── tar
        ├── 01.py
        ├── 02.py
        ├── 03.py
        └── gz
            ├── 01.py
            ├── 02.py
            ├── 03.py
            └── py.tar.bz2
            └── py.tar.gz
            └── py.tar
4 directories, 15 files
```

实践过程

解压缩文件

软件安装

通过apt安装/卸载软件

使用apt必须以sudo身份

apt是Advanced Packaging Tool,是Linux下的一款安装包管理工具
可以在终端中方便的安装/卸载/更新软件包

```
# 1. 安装软件  
$ sudo apt install 软件包  
  
# 2. 卸载软件  
$ sudo apt remove 软件名  
  
# 3. 更新已安装的包  
$ sudo apt upgrade
```

例如

```
# 一个小火车提示  
$ sudo apt install sl  
  
# 一个比较漂亮的查看当前进程排名的软件  
$ sudo apt install htop
```

效果都是动态的动画

配置软件源

如果希望在 Ubuntu 中安装软件，更加快速，可以通过设置镜像源，选择一个访问网速更快的服务器来提供软件下载安装服务。

提示：更换服务器之后，需要一个相对比较长时间的更新过程，需要耐心等待。更新完成后，再安装软件都会从新设置的服务器下载软件了。

所谓镜像源，就是所有服务器的内容是相同的（镜像），但是根据所在位置不同，国内服务器通常速度会更快一些！



Ubuntu 中设置镜像源通过图形界面操作
任务栏 - 系统设置 - 软件和更新 - 下载
自... (可以指定) - 重新载入

