

# 中国矿业大学计算机学院

## 2020 级本科生课程报告

课程名称 Linux 操作系统

报告时间 2022. 10. 10

学生姓名 周子欣

学 号 12204202

专 业 信息安全

任课教师 杨勇

## 《Linux 操作系统》评分表

编号	毕业要求	课程教学目标	考查方式与考查点	占比	得分
1	通过本课程的学习，使学生了解操作系统设计的总体知识结构，了解当前主流操作系统的基本架构。能够熟练在 Linux 平台下进行相关命令操作、编写脚本程序进行系统管理。能够熟练掌握 Linux 平台下的 C 语言程序的写法，会用编译器 GCC 及调试器 GDB。初步掌握 Linux 内核代码的阅读与调试能力。培养学生辩证唯物主义的历史发展观、深入领会实践出真知的道理、磨炼学生滴水穿石的科研精神。	目标 1：通过选定一研究主题，让学生去深入探索该主题的相关知识。能够运用图书馆及互联网的有价值资源，提高自学与探索能力。	平时作业针对核心知识，让学生做延伸探索，加深对知识的理解和掌握。	50	
2		目标 2：能够针对实际应用问题，合理的组建团队去解决。通过 CVS 协同开发，加强团队合作能力的培养。为学生以后进入公司进行大项目的开发培养经验。	期末考试，以学生自由组团队方式完成。通过共同完成一程序设计项目，提高团队见的沟通与协作能力。	50	
总分				100%	

评阅人：

日 期：

## 目录

实验一	Linux 命令	5
1.	目录操作命令	5
1.1	ls 命令	5
1.2	cd 命令	7
1.3	pwd 命令	7
1.4	mkdir 命令	8
1.5	rmdir 命令	8
2.	文件操作命令	9
2.1	touch 命令	9
2.2	cat 命令	10
2.3	cp 命令	10
2.4	rm 命令	12
2.5	mv 命令	14
2.6	chmod 命令	15
3.	文件处理命令	16
3.1	grep 命令	16
3.2	head 命令	18
3.3	tail 命令	19
3.4	wc 命令	20
3.5	sort 命令	21
3.6	find 命令	21
3.7	which 命令	22
3.8	whereis 命令	22
3.9	locate 命令	23
4.	压缩解压命令	23
4.1	bzip2 命令和 bunzip2 命令	23
4.2	gzip 命令	24
4.3	unzip 命令	25
4.4	zcat 命令和 bzcat 命令	26
4.5	tar 命令	26
5.	磁盘操作命令	28
5.1	mount 命令	28
5.2	umount 命令	29
5.3	df 命令	29
5.4	du 命令	30
6.	关机重启命令	31
6.1	shutdown 命令	31
6.2	halt 命令	31
6.3	poweroff 命令	31
6.4	reboot 命令	32
6.5	init 命令	32
7.	变换用户身份命令	33

---

7.1 su 命令 .....	33
7.2 sudo 命令 .....	33
7.3 useradd 命令 .....	33
7.4 passwd 命令 .....	34
8. 网络操作命令 .....	35
8.1 ifconfig 命令 .....	35
8.2 ping 命令 .....	35
8.3 netstat 命令 .....	36
8.4 ftp 命令和 bye 命令 .....	37
9. 其他命令 .....	38
9.1 echo 命令 .....	38
9.2 more 命令和 less 命令 .....	38
9.3 help 命令和 man 命令 .....	38
9.4 cal 命令 .....	38
9.5 date 命令 .....	39
10. 管道符 .....	40
10.1 作用及语法 .....	40
10.2 测试 .....	40
实验二 shell 编程 .....	40
1. Linux 下如何执行 shell 脚本 .....	40
2. 编写脚本 .....	41
2.1 通过 shell 脚本封请求数异常的 IP .....	41
2.2 猜拳游戏 .....	42
实验三 在 Linux 下搭建 Apache 服务器 .....	44
1. 搭建 apache 服务 .....	44
2. 制作个人介绍网页并部署到 apache 服务器上，并能通过浏览器访问 ..	46

# 实验一 Linux 命令

## 1. 目录操作命令

### 1.1 ls 命令

#### 1.1.1 作用及语法

ls 命令与在 MS-DOS 系统下的 dir 命令类似，用户可以利用 ls 命令查看某个目录下的所有内容。默认情况下，显示的条目按字母顺序排列。

功能：列出当前目录下的所有内容。

格式：ls [选项] [文件名或目录名]

选项：

选项	作用
-s	显示每个文件的大小
-S	按文件的大小排序
-a	显示目录中全部文件，包括隐藏文件
-l	使用长列表格式，显示文件详细信息
-t	按文件修改的时间排序显示
-F	显示文件类型描述符。*为可执行的普通文件，/为目录文件

#### 1.1.2 测试

(1) 仅列出当前目录可见文件

ls

```
[chongyan@localhost ~]$ ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
```

(2) 列出当前目录可见文件详细信息

ls -l

```
[chongyan@localhost ~]$ ls -l
总用量 32
lrwxrwxrwx. 1 root root 7 8月 29 06:31 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 8月 29 06:37 boot
drwxr-xr-x 19 root root 3280 10月 12 04:47 dev
drwxr-xr-x. 145 root root 8192 10月 12 04:47 etc
drwxr-xr-x. 4 root root 36 10月 9 09:44 home
lrwxrwxrwx. 1 root root 7 8月 29 06:31 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 8月 29 06:31 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 10 2018 media
drwxr-xr-x. 2 root root 6 4月 10 2018 mnt
drwxr-xr-x. 3 root root 16 8月 29 06:34 opt
dr-xr-xr-x 225 root root 0 10月 12 04:47 proc
dr-xr-xr-x. 4 root root 283 10月 11 05:52 root
drwxr-xr-x 44 root root 1300 10月 12 04:49 run
lrwxrwxrwx. 1 root root 8 8月 29 06:31/sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 10 2018 srv
dr-xr-xr-x 13 root root 0 10月 12 04:47 sys
drwxrwxrwt. 58 root root 8192 10月 12 04:49 tmp
drwxr-xr-x. 13 root root 155 8月 29 06:31 usr
drwxr-xr-x. 22 root root 4096 10月 10 05:54 var
[chongyan@localhost ~]$
```

(3) 列出所有文件（包括隐藏）的详细信息

`ls -al`

```
[chongyan@localhost ~]$ ls -al
总用量 32
dr-xr-xr-x. 17 root root 244 10月 12 04:47 .
dr-xr-xr-x. 17 root root 244 10月 12 04:47 ..
-rw-r--r--. 1 root root 0 10月 12 04:47 .autorelabel
lrwxrwxrwx. 1 root root 7 8月 29 06:31 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 8月 29 06:37 boot
drwxr-xr-x 19 root root 3280 10月 12 04:47 dev
drwxr-xr-x. 145 root root 8192 10月 12 04:47 etc
drwxr-xr-x. 4 root root 36 10月 9 09:44 home
lrwxrwxrwx. 1 root root 7 8月 29 06:31 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 8月 29 06:31 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 10 2018 media
drwxr-xr-x. 2 root root 6 4月 10 2018 mnt
drwxr-xr-x. 3 root root 16 8月 29 06:34 opt
dr-xr-xr-x 225 root root 0 10月 12 04:47 proc
dr-xr-xr-x. 4 root root 283 10月 11 05:52 root
drwxr-xr-x 44 root root 1300 10月 12 04:49 run
lrwxrwxrwx. 1 root root 8 8月 29 06:31/sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 10 2018 srv
dr-xr-xr-x 13 root root 0 10月 12 04:51 sys
drwxrwxrwt. 58 root root 8192 10月 12 04:49 tmp
drwxr-xr-x. 13 root root 155 8月 29 06:31 usr
drwxr-xr-x. 22 root root 4096 10月 10 05:54 var
[chongyan@localhost ~]$
```

(4) 显示当前目录下包括隐藏文件在内的所有文件列表

`ls -a`

```
[chongyan@localhost ~]$ ls -a
. .autorelabel boot etc lib media opt root/sbin sys usr
.. bin dev home lib64 mnt proc run srv tmp var
[chongyan@localhost ~]$
```

## 1.2 cd 命令

### 1.2.1 作用及语法

功能：切换用户当前工作目录

格式：cd [选项] [参数]

选项：

选项	作用
-L	（默认值）如果要切换到的目标目录是一个符号连接，那么切换到符号连接的目录
-P	如果要切换到的目标目录是一个符号连接，那么切换到它指向的物理位置目录
-	当前工作目录将被切换到环境变量 OLDPWD 所表示的目录，也就是前一个工作目录

### 1.2.2 测试

- （1）进入根目录 cd /
- （2）进入用户主目录 cd ~
- （3）返回上级目录 cd ..
- （4）返回上两级目录 cd ../../..

```
[ chongyan@localhost ~/]$ cd home/chongyan
[ chongyan@localhost ~]$ cd /
[ chongyan@localhost ~]$ cd ~
[ chongyan@localhost ~]$ cd ..
[ chongyan@localhost home]$ cd ../../..
[ chongyan@localhost ~]$
```

## 1.3 pwd 命令

### 1.3.1 作用及语法

功能：显示当前工作目录的绝对路径

格式：pwd [选项]

选项：

选项	作用
-L	打印环境变量“\$PWD”的值，可能为符号链接

-P	(默认值) 打印当前工作目录的物理位置
----	---------------------

### 1.3.2 测试

pwd

```
[ chongyan@localhost ~ ] $ pwd
/home/chongyan
```

## 1.4 mkdir 命令

### 1.4.1 作用及语法

功能：创建一个新的目录。Mkdir 即 make directory。

格式：mkdir [选项] 目录名

选项：

选项	作用
-m 权限	对新建目录设置存取权限。权限 777, 744, 755 等
-p	一次性建立多级目录，即以递归形式建立目录

### 1.4.2 测试

创建一个新目录

mkdir new

```
[ chongyan@localhost ~ ] $ ls
test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
[ chongyan@localhost ~ ] $ mkdir new
[ chongyan@localhost ~ ] $ ls
new test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
[ chongyan@localhost ~ ] $
```

## 1.5 rmdir 命令

### 1.5.1 作用及语法

功能：删除一个空目录

格式：rmdir [-p] 目录名

### 1.5.2 测试

删除一个空文件夹



```
[ chongyan@localhost ~]$ ls
empty new test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
[ chongyan@localhost ~]$ rmdir empty
[ chongyan@localhost ~]$ rmdir new
rmdir: 删除 "new" 失败: 目录非空
[ chongyan@localhost ~]$ ls
new test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
```

## 2. 文件操作命令

### 2.1 touch 命令

#### 2.1.1 作用及语法

功能：创建文件，如果文件名不存在则创建一个新的空文件，且该文件不包含任何格式，大小为 0 字节。

格式：touch 文件名

选项：

选项	作用
-a 或--time=atime 或--time=access 或--time=use	只更改存取时间
-c 或--no-create	不建立任何文件
-d<时间日期>	使用指定的日期时间，而非现在的时间
-f	此参数将忽略不予处理，仅负责解决 BSD 版本 touch 指令的兼容性问题
-m 或--time=mtime 或--time=modify	只更改变动时间
-r<参考文件或目录>	把指定文件或目录的日期时间，统统设成和参考文件或目录的日期时间相同
-t<日期时间>	使用指定的日期时间，而非现在的时间

#### 2.1.2 测试

创建新的空文件

touch 111.txt

```
[ chongyan@localhost ~]$ ls
new test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
[ chongyan@localhost ~]$ touch 111.txt
[ chongyan@localhost ~]$ ls
111.txt new test.cpp 公共 模板 视频 图片 文档 下载 音乐 桌面
[ chongyan@localhost ~]$
```

## 2.2 cat 命令

### 2.2.1 作用及语法

cat 用法很多。cat 是 concatenate（连锁）的简写。基本作用是合并文件，并在屏幕上显示整个文件的内容。

功能：显示某文件的内容。

格式：cat [选项] 文件名

选项：

选项	作用
-A	等价于"-vET"组合选项
-b	只对非空行编号，从 1 开始编号，覆盖 "-n"选项
-e	等价于"-vE"组合选项
-E	在每行的结尾显示'\$'字符
-n	对所有行编号，从 1 开始编号
-s	压缩连续的空行到一行
-t	等价于"-vT"组合选项
-T	使用'^I'表示 TAB（制表符）
-u	POSIX 兼容性选项，无意义
-v	使用'^'和'M-'符号显示控制字符，除了 LFD（line feed，即换行符'\n'）和 TAB（制表符）

### 2.2.2 测试

显示文件内容

cat aaa.txt

```
[ chongyan@localhost new]$ cat aaa.txt
testtesttest
[chongyan@localhost new]$
```

## 2.3 cp 命令

### 2.3.1 作用及语法

功能：实现文件复制。cp 是 copy 缩写。

格式：cp [选项] <源文件> <目标>

选项:

选项	作用
-a	此参数的效果和同时指定“-dpR”参数相同
-d	当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录
-f	强行复制文件或目录，不论目标文件或目录是否已存在
-i	覆盖既有文件之前先询问用户
-l	对源文件建立硬连接，而非复制文件
-p	保留源文件或目录的属性
-R/r	递归处理，将指定目录下的所有文件与子目录一并处理
-s	对源文件建立符号连接，而非复制文件
-u	使用这项参数后只会在源文件的更改时间较目标文件更新时或是名称相互对应的目标文件并不存在时，才复制文件
-S	在备份文件时，用指定的后缀“SUFFIX”代替文件的默认后缀
-b	覆盖已存在的文件目标前将目标文件备份
-v	详细显示命令执行的操作

说明：复制的三种情况：①目标文件是目录，就会把源文件复制到该目录中；②目标文件是普通文件，就会询问是否覆盖；③目标文件不存在，就直接执行正常的复制操作。其中-p 参数表示保留原始文件的属性；-r 参数表示递归持续复制（用于目录）；-i 参数表示若目标文件存在则询问是否覆盖。

### 2.3.2 测试

#### (1) 复制一个文件

```
cp aaa.txt bbb.txt
```

```

[ chongyan@localhost new] $ ls
aaa.txt
[ chongyan@localhost new] $ cp aaa.txt bbb.txt
[ chongyan@localhost new] $ ls
aaa.txt  bbb.txt
[ chongyan@localhost new] $

```

(2) 复制文件到一个目录下

```
cp aaa.txt /home/chongyan
```

```

[ chongyan@localhost new] $ cp aaa.txt /home/chongyan
[ chongyan@localhost new] $ cd /home/chongyan
[ chongyan@localhost ~] $ ls
111.txt  new      公共  视频  文档  音乐
aaa.txt  test.cpp 模板  图片  下载  桌面
[ chongyan@localhost ~] $

```

(3) 复制文件夹到另一个文件目录下

```
cp -r /home/chongyan/new /home/chongyan/new2
```

```

[ chongyan@localhost ~] $ cp -r /home/chongyan/new /home/chongyan/new2
[ chongyan@localhost ~] $ cd news
bash: cd: news: 没有那个文件或目录
[ chongyan@localhost ~] $ cd new2
[ chongyan@localhost new2] $ ls
new
[ chongyan@localhost new2] $ cd new
[ chongyan@localhost new] $ ls
aaa.txt  bbb.txt
[ chongyan@localhost new] $

```

## 2.4 rm 命令

### 2.4.1 作用及语法

功能：删除指定的文件。

格式：rm [选项] [文件名或目录名]

选项：

选项	作用
-i	以安全询问的方式进行删除操作
-r 或 -R	递归处理，将指定目录下的所有文件及子目录一并处理。

-f	强制删除文件或目录。
-v	显示指令执行过程
-d	直接把欲删除的目录的硬连接数据删成 0，删除该目录。

### 2.4.2 测试

#### (1) 删除一个文件

```
rm ccc.txt
```

```
[ chongyan@localhost new] $ ls
aaa.txt  bbb.txt  ccc.txt
[ chongyan@localhost new] $ rm ccc.txt
[ chongyan@localhost new] $ ls
aaa.txt  bbb.txt
[ chongyan@localhost new] $
```

#### (2) 删除一个文件目录

```
rm -r new
```

```
[ chongyan@localhost new2] $ ls
new
[ chongyan@localhost new2] $ rm -r new
[ chongyan@localhost new2] $ ls
[ chongyan@localhost new2] $
```

#### (3) 以安全询问的方式删除文件

```
rm -i bbb.txt
```

```
[ chongyan@localhost new] $ ls
aaa.txt  bbb.txt
[ chongyan@localhost new] $ rm -i bbb.txt
rm: 是否删除普通文件 "bbb.txt"? y
[ chongyan@localhost new] $ ls
aaa.txt
[ chongyan@localhost new] $
```

#### (4) 结合通配符“\*”实现删除所有 txt 文件

```
rm *.txt
```

```
[ chongyan@localhost ~] $ ls
111.txt  new  test.cpp  模板  图片  下载  桌面
aaa.txt  new2  公共  视频  文档  音乐
[ chongyan@localhost ~] $ rm *.txt
[ chongyan@localhost ~] $ ls
new  new2  test.cpp  公共  模板  视频  图片  文档  下载  音乐  桌面
[ chongyan@localhost ~] $
```

## (5) 一次性删除多个文件目录

```
[ chongyan@localhost ~]$ ls
new  new2  test.cpp  公共  模板  视频  图片  文档  下载  音乐  桌面
[ chongyan@localhost ~]$ rm -r new new2
[ chongyan@localhost ~]$ ls
test.cpp  公共  模板  视频  图片  文档  下载  音乐  桌面
[ chongyan@localhost ~]$
```

## 2.5 mv 命令

### 2.5.1 作用及语法

功能：实现文件的移动，实现文件夹或文件的重命名。

格式：mv 文件名 路径名 或者 mv 目录名或文件名 新目录名或文件名

选项：

选项	作用
--backup=<备份模式>	若需覆盖文件，则覆盖前先行备份
-b	当文件存在时，覆盖前，为其创建一个备份
-f	若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录
-i	交互式操作，覆盖前先行询问用户，如果源文件与目标文件或目标目录中的文件同名，则询问用户是否覆盖目标文件。用户输入“y”，表示将覆盖目标文件；输入“n”，表示取消对源文件的移动。这样可以避免误将文件覆盖
--strip-trailing-slashes	删除源文件中的斜杠“/”
-S<后缀>	为备份文件指定后缀，而不使用默认的后缀
--target-directory=<目录>	指定源文件要移动到目标目录
-u	当源文件比目标文件新或者目标文件不存在时，才执行移动操作

### 2.5.2 测试

#### (1) 重命名文件

```
mv aaa.txt fff.txt
```

```
[chongyan@localhost test]$ ls
aaa.txt bbb.txt ccc.txt ddd.txt
[chongyan@localhost test]$ mv aaa.txt fff.txt
[chongyan@localhost test]$ ls
bbb.txt ccc.txt ddd.txt fff.txt
[chongyan@localhost test]$ █
```

## (2) 移动文件

```
mv ddd.txt ../test2
```

```
[chongyan@localhost ~]$ cd test2
[chongyan@localhost test2]$ ls
[chongyan@localhost test2]$ cd ../test
[chongyan@localhost test]$ ls
bbb.txt ccc.txt ddd.txt fff.txt
[chongyan@localhost test]$ mv ddd.txt ../test2
[chongyan@localhost test]$ cd ../test2
[chongyan@localhost test2]$ ls
ddd.txt
```

## 2.6 chmod 命令

### 2.6.1 作用及语法

功能：修改文件的权限和文件的属性。是 change modify 的缩写。

在 Linux 下，可以使用“ls -l”命令查看某路径下的所有内容的详细内容。

格式：chmod [<文件使用者>+/-/=<权限类型>] 文件名 1 文件名 2 ...

选项：

选项	作用
a+rw	为所有用户增加读、写的权限
a-rwx	为所有用户取消读、写、执行的权限
g+w	为组群用户增加写权限
o-rwx	取消其他人的所有权限（读、写、执行）
ug+r	为所有者和组群用户增加读权限
g=rx	只允许组群用户读、执行。并删除其他的权限

### 2.6.2 测试

对文件给所有用户加读写执行权限

```
chmod 777 ddd.txt
```

```
[ chongyan@localhost test2]$ ls -l
总用量 0
-rw-rw-r-- 1 chongyan chongyan 0 10月 13 08:01 ddd.txt
[ chongyan@localhost test2]$ chmod 777 ddd.txt
[ chongyan@localhost test2]$ ls -l
总用量 0
-rwxrwxrwx 1 chongyan chongyan 0 10月 13 08:01 ddd.txt
[ chongyan@localhost test2]$
```

### 3.文件处理命令

#### 3.1 grep 命令

##### 3.1.1 作用及语法

功能：在指定的文件中查找某个特定的字符串。

格式：grep [选项] 关键字 文件名

选项：

选项	作用
-a	不要忽略二进制数据
-b	在显示符合范本样式的那一行之外，并显示该行之前的内容
-c 或--count	计算符合范本样式的列数
-d<进行动作> --directories=<动作>	指定要查找的是目录而非文件时，必须使用这项参数，否则 grep 命令将回报信息并停止动作
-e<范本样式> --regexp=<范本样式>	指定字符串作为查找文件内容的范本样式
-f<范本文件> --file=<规则文件>	指定范本文件，其内容有一个或多个范本样式，让 grep 查找符合范本条件的文件内容，格式为每一列的范本样式
-G--basic-regexp	将范本样式视为普通的表示法来使用
-h --no-filename	在显示符合范本样式的那一列之前，不标示该列所属的文件名称
-H	在显示符合范本样式的那一列之前，标示该列的文件名称



-i	忽略字符大小写的差别
-l	列出文件内容符合指定的范本样式的文件名称
-n	在显示符合范本样式的那一列之前，标示出该列的编号
-p	PATTERN 是一个 Perl 正则表达式
-q	不显示任何信息
-r	此参数的效果和指定“-d recurse”参数相同
-s	不显示错误信息
-v	反转查找
-w	只显示全字符合的列
-x	只显示全列符合的列
-y	此参数效果跟“-i”相同

说明：命令区分大小写，执行结果是显示含有要查找的字符串的每一行原文。

### 3.1.2 测试

#### (1) 在文本中搜索

```
grep yan dd.txt
```

```
[ chongyan@localhost test2] $ cat dd.txt
hhhhh
heloe
```

```
chongyan
```

```
zzx
```

```
www
```

```
yanchong
```

```
[ chongyan@localhost test2] $ grep yan dd.txt
chongyan
yan chong
```

#### (2) 在多级目录中对文本进行递归搜索

```

[chongyan@localhost test2]$ grep "txt" .. -r -n
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-cak@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-cs@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-cy@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-en-CA@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-es-CL@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-ms@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-rm@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-sk@firefox.mozilla.org.xpi
匹配到二进制文件 ../mozilla/firefox/8w2i20zc.default-default/extensions/langpack-uk@firefox.mozilla.org.xpi
匹配到二进制文件 ../cache/tracker/meta.db-wal
匹配到二进制文件 ../cache/gstreamer-1.0/registry.x86_64.bin
匹配到二进制文件 ../cache/mozilla/firefox/8w2i20zc.default-default/startupCache/startupCache.8.little
匹配到二进制文件 ../cache/mozilla/firefox/8w2i20zc.default-default/startupCache/scriptCache-current.bin
匹配到二进制文件 ../local/share/tracker/data/tracker-store.journal
../viminfo:32:'0 12 0 ~/test2/dd.txt
../viminfo:44:'- 12 0 ~/test2/dd.txt
../viminfo:45:'- 1 0 ~/test2/dd.txt
../viminfo:63:'- 1 11 ~/new/aaa.txt
../viminfo:74:'- 1 11 ~/new/aaa.txt
../viminfo:85:'- 1 11 ~/new/aaa.txt
../viminfo:96:'- 1 11 ~/new/aaa.txt
../viminfo:107:'- 1 11 ~/new/aaa.txt
../viminfo:118:'- 1 11 ~/new/aaa.txt
../viminfo:129:'- 1 11 ~/new/aaa.txt
../viminfo:146:> ~/test2/dd.txt
../viminfo:281:> ~/new/aaa.txt
[chongyan@localhost test2]$

```

## 3.2 head 命令

### 3.2.1 作用及语法

功能：查看文件的开头部分的内容

格式：head [选项] 文件名

选项：

选项	作用
-c	显示前 NUM 字节；如果 NUM 前有"-", 那么会打印除了文件末尾的 NUM 字节以外的其他内容
-n	显示前 NUM 行而不是默认的 10 行；如果 NUM 前有"-", 那么会打印除了文件末尾的 NUM 行以外的其他行
-q	不打印文件名行
-v	总是打印文件名行
-z	行终止符为 NUL 而不是换行符

说明：如果不加数字选项，默认只显示文件最初的 10 行。

### 3.2.2 测试

(1) 显示文件开头（未指定行数默认前 10 行）head dd.txt

(2) 显示文件前两行 head dd.txt -n 2

```
[ chongyan@localhost test2]$ head dd.txt
hhhhh
heloe

chongyan

ZZX

www

yanchong
[ chongyan@localhost test2]$ head dd.txt -n 2
hhhhh
heloe
[ chongyan@localhost test2]$
```

### 3.3 tail 命令

#### 3.3.1 作用及语法

功能：查看文件的结尾部分的内容

格式：tail [选项] 文件名

选项：

选项	作用
--retry	在 tail 命令启动时，文件不可访问或者文件稍后变得不可访问，都始终尝试打开文件。使用此选项时需要与选项“--follow=name”连用
-c<N>或--bytes=<N>	输出文件尾部的 N（N 为整数）个字节内容
-f<name/descriptor>或； --follow<namelndescript>	显示文件最新追加的内容。“name”表示以文件名的方式监视文件的变化。“-f”与“-fdescriptor”等效
-F	与选项 “-follow=name”和“--retry”连用时功能相同
-n<N>或--line=<N>	输出文件的尾部 N（N 位数字）行内容
--pid=<进程号>	与 “-f” 选项连用，当指定的进程号的进程终止后，自动退出 tail 命令
-q 或--quiet 或--silent	当有多个文件参数时，不输出各个文件名
-s<秒数>或--sleep-interval=<秒数>	与 “-f” 选项连用，指定监视文件变化时间间隔的秒数

-v 或--verbose	当有多个文件参数时，总是输出各个文件名
---------------	---------------------

说明：如果不加数字选项，默认只显示文件结尾的 10 行。

### 3.3.2 测试

(1) 显示文件尾部内容（未指定行数默认前 10 行）`tail dd.txt`

(2) 显示文件后两行 `tail dd.txt -n 2`

```
[ chongyan@localhost test2]$ tail dd.txt
chongyan
zzx
www
yanchong

[chongyan@localhost test2]$ tail dd.txt -n 2

[chongyan@localhost test2]$ █
```

## 3.4 wc 命令

### 3.4.1 作用及语法

功能：对文件的行数、单词数、字符数进行统计。`wc` 命令是一个对文件进行统计的相当实用的命令。

格式：`wc [选项] 文件名`

选项：

选项	作用
-c	统计字节数，或--bytes 或--chars：只显示 Bytes 数
-l	统计行数，或--lines：只显示列数
-m	统计字符数，这个标志不能与 -c 标志一起使用
-w	统计字数，或--words：只显示字数。一个字被定义为由空白、跳格或换行字符分隔的字符串
-L	打印最长行的长度

### 3.4.2 测试

(1) 显示文件的字节数、字数、行数 `wc dd.txt`

(2) 显示最长行的长度 `wc -L dd.txt`

```
[ chongyan@localhost test2]$ wc dd.txt
12  6 44 dd.txt
[ chongyan@localhost test2]$ wc -L dd.txt
8 dd.txt
```

### 3.5 sort 命令

#### 3.5.1 作用及语法

功能：对文件或查询结果进行排序。

格式：`sort [选项] 文件名`

选项：

选项	作用
-f	忽略大小写
-r	反向排序
-t	指定分隔符
-i	只考虑可以打印的字符，忽略任何非显示字符

#### 3.5.2 测试

给文件文本排序

`sort dd.txt`

```
[ chongyan@localhost test2]$ cat dd.txt
hhhhh
heloe

chongyan

zzx

www

yanchong

[ chongyan@localhost test2]$ sort dd.txt

chongyan
heloe
hhhhh
www
yanchong
zzx
[ chongyan@localhost test2]$ █
```

### 3.6 find 命令

#### 3.6.1 作用及语法

功能：查找文件或目录

格式：find 文件名（或目录名）

### 3.6.2 测试

列出当前目录及子目录下所有文件和文件夹

```
[ chongyan@localhost test2] $ find
.
./ddd.txt
./dd.txt
[ chongyan@localhost test2] $ find ../test
../test
../test/bbb.txt
../test/ccc.txt
../test/fff.txt
[ chongyan@localhost test2] $
```

## 3.7 which 命令

### 3.7.1 作用及语法

功能：按 PATH 变量所规定的路径进行查找相应的命令，显示该命令的绝对路径。

格式：which 命令名

### 3.7.2 测试

显示命令的绝对路径

which ls

which cp

which which

```
[ chongyan@localhost test2] $ which ls
alias ls='ls --color=auto'
/usr/bin/ls
[ chongyan@localhost test2] $ which cp
/usr/bin/cp
[ chongyan@localhost test2] $ which which
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
/usr/bin/alias
/usr/bin/which
```

## 3.8 whereis 命令

### 3.8.1 作用及语法

功能：不但能查询出命令，而且还能查询出 Ubuntu 资料库里记载的文件。

格式：whereis [选项] 文件名

选项:

选项	作用
-b	只查找二进制的文件
-w	只查找说明文件 manual 路径下的文件

说明：与 which 不同是，whereis 不但能找到可执行的命令，而且从资料库中寻找所有包含文件名字符串的文件全部查找出来。

### 3.8.2 测试

寻找所有包含文件名字符串的文件

```
[chongyan@localhost test2]$ whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.gz
```

## 3.9 locate 命令

### 3.9.1 作用及语法

功能：将所有与被查询的文件名相同的文件查找出来。

格式：locate 文件名

说明：locate 命令的使用方式比较简单，直接加上要查询的文件即可。

### 3.9.2 测试

搜索带有指定关键词的文件

```
locate xinxin
```

```
[chongyan@localhost test2]$ locate xinxin
/home/xinxin
/var/spool/mail/xinxin
```

## 4. 压缩解压命令

### 4.1 bzip2 命令和 bunzip2 命令

#### 4.1.1 作用及语法

功能：bzip2 和 bunzip2 是一对压缩和解压命令。

格式：

压缩文件：bzip2 文件名 1 [文件名 2] ...

解压文件：bunzip2 文件名 1 [文件名 2] ...

说明：在利用 bzip2 进行文件压缩后，压缩前地原始文件消失，系统会生成一个新的压缩文件，文件名的后缀名为.bz2。另外，利用 bzip2 压缩后的压缩文件必须利用 bunzip2 命令才能实现解压。

#### 4.1.2 测试

压缩文件和解压缩文件

```
bzip2 bbb.txt ccc.txt
```

```
bzip2 bbb.txt ccc.txt
```

```
[chongyan@localhost test]$ ls
bbb.txt  ccc.txt  fff.txt
[chongyan@localhost test]$ bzip2 bbb.txt ccc.txt
[chongyan@localhost test]$ ls
bbb.txt.bz2  ccc.txt.bz2  fff.txt
[chongyan@localhost test]$ bunzip2 bbb.txt ccc.txt
bunzip2: Can't open input file bbb.txt: No such file or directory.
bunzip2: Can't open input file ccc.txt: No such file or directory.
[chongyan@localhost test]$ bunzip2 bbb.txt.bz2 ccc.txt.bz2
[chongyan@localhost test]$ ls
bbb.txt  ccc.txt  fff.txt  -
```

## 4.2 gzip 命令

### 4.2.1 作用及语法

功能：不仅能够压缩文件，而且能够实现文件的解压。

格式：gzip [-选项] 文件名

选项：

选项	作用
-d	解压
-n	指定压缩级别。注 n 的范围是 1—9

说明：利用 gzip 命令可以将普通文件压缩.gz 为后缀的压缩文件。压缩完成后，原始文件消失。在压缩时还可以指定压缩级别，该命令的压缩级别范围是 1-9 级，默认的级别是 6。另外，1 的压缩比最差，速度最快；9 的压缩比最好，速度较慢。

### 4.2.2 测试

压缩文件和解压缩文件

```
gzip fff.txt
```

```
gzip -d fff.txt
```



```
[ chongyan@localhost test]$ gzip fff.txt
[ chongyan@localhost test]$ ls
bbb.txt  ccc.txt  fff.txt.gz
[ chongyan@localhost test]$ gzip -d fff.txt
[ chongyan@localhost test]$ ls
bbb.txt  ccc.txt  fff.txt
[ chongyan@localhost test]$
```

### 4.3 unzip 命令

#### 4.3.1 作用及语法

功能：解压.zip 文件。它能够将经过 winzip 压缩的文件进行解压操作。

格式：unzip [选项] 文件名.zip

选项：

选项	作用
-d	将文件解压到指定目录中
-v	查看文件目录列表但不解压
-n	不覆盖原来已经存在的文件
-o	以默认方式覆盖已经存在的文件

#### 4.3.2 测试

(1) 只查看压缩文件里的文件目录，但不解压

```
unzip -v Latex_Template.zip
```

```
[ chongyan@localhost 桌面]$ unzip -v Latex_Template.zip
Archive:  Latex_Template.zip
Length  Method      Size  Cmpr   Date       Time    CRC-32   Name
-----  -
  47104  Defl:N        10065  79% 03- 06- 2013  18:11 f88a49bf Latex_Template/ams2la.
sty
48738078 Defl:N       515542  99% 05- 26- 2016  14:39 3764ca11 Latex_Template/Circui
board.eps
 181793  Defl:N        85997  53% 05- 25- 2016  19:15 666d7d18 Latex_Template/Circui
board.psd
1723272  Defl:N       134732  92% 05- 26- 2016  14:38 d4a9ea78 Latex_Template/Circui
board.tif
  47158  Defl:N        10111  79% 02- 20- 2014  15:10 39dc5de2 Latex_Template/ctexte
p_ams2la.sty
   685   Defl:N         327  52% 02- 20- 2014  15:10 1ad02ac4 Latex_Template/ctexte
p_headrule.sty
326910   Defl:N       84790  74% 08- 10- 2015  18:01 8bc12fd6 Latex_Template/E1- 1. ej
```

(2) 将 zip 文件在/home/chongyan/test/new 目录中进行

```
unzip Latex_Template.zip -d /home/chongyan/test/new
```

```
[chongyan@localhost 桌面]$ ls
Latex_Template.zip
[chongyan@localhost 桌面]$ unzip Latex_Template.zip -d /home/chongyan/test/new
Archive:  Latex_Template.zip
  inflating: /home/chongyan/test/new/Latex_Template/ams2la.sty
  inflating: /home/chongyan/test/new/Latex_Template/Circuitboard.eps
  inflating: /home/chongyan/test/new/Latex_Template/Circuitboard.psd
  inflating: /home/chongyan/test/new/Latex_Template/Circuitboard.tif
  inflating: /home/chongyan/test/new/Latex_Template/ctextemp_ams2la.sty
  inflating: /home/chongyan/test/new/Latex_Template/ctextemp_headrule.sty
  inflating: /home/chongyan/test/new/Latex_Template/E1-1.eps
  inflating: /home/chongyan/test/new/Latex_Template/E1-2.eps
  inflating: /home/chongyan/test/new/Latex_Template/HEADRULE.STY
  inflating: /home/chongyan/test/new/Latex_Template/Mathformule.eps
  inflating: /home/chongyan/test/new/Latex_Template/Mathformule.tif
  inflating: /home/chongyan/test/new/Latex_Template/-ru.dvi
  inflating: /home/chongyan/test/new/Latex_Template/-ru.pdf
  inflating: /home/chongyan/test/new/Latex_Template/-ru.ps
  inflating: /home/chongyan/test/new/Latex_Template/-ru.tex
  inflating: /home/chongyan/test/new/Latex_Template/-ru.tex.bak
[chongyan@localhost 桌面]$
```

## 4.4 zcat 命令和 bzcat 命令

### 4.4.1 作用及语法

功能：都用来查看压缩文件内容的，即不用解压缩文件就可以查看里面的内容。

格式：zcat 文件名

        bzcat 文件名

说明：zcat 命令专门针对.gz 后缀的压缩文件进行查看，而 bzcat 是针对.bz2 后缀的压缩文件进行查看

### 4.4.2 测试

```
[chongyan@localhost test]$ ls
bbb.txt.gz  ccc.txt.bz2  fff.txt  new
[chongyan@localhost test]$ zcat bbb.txt.gz
g
flag
!
[chongyan@localhost test]$ bzcat ccc.txt.bz2
!!!!1
[chongyan@localhost test]$
```

## 4.5 tar 命令

### 4.5.1 作用及语法

功能：对文件或目录进行打包备份或解包操作。

格式：

tar [选项] [备份包的文件名] [要打包（或要解包）的文件或目录]

说明：打包备份只是把多个不同的文件放在一个大文件中，并没有压缩，这个大文件的后缀名为.tar。利用该命令打包后，原始文件不消失。

选项：

选项	作用
-c	创建新的打包文件
-x	抽取.tar 文件里的内容
-z	打包后直接用 gzip 命令进行压缩，或者解压文件
-j	打包后直接用 bzip2 命令进行压缩，或者解压文件
-t	查看一个打包文件里的文件目录
-f	使用文件或设备
-v	在打包压缩或解包解压后将文件的详细清单显示出来

#### 4.5.2 测试

(1) 使用 gzip 压缩格式对某个目录进行打包操作，显示压缩过程，压缩包规范后缀为.tar.gz

```
tar czvf new.tar.gz test
```

```
[chongyan@localhost ~]$ tar czvf new.tar.gz test
test/
test/fff.txt
test/new/
test/new/Latex_Template/
test/new/Latex_Template/ams2la.sty
test/new/Latex_Template/Circuitboard.eps
test/new/Latex_Template/Circuitboard.psd
test/new/Latex_Template/Circuitboard.tif
test/new/Latex_Template/ctextemp_ams2la.sty
test/new/Latex_Template/ctextemp_headrule.sty
test/new/Latex_Template/E1- 1.eps
test/new/Latex_Template/E1- 2.eps
test/new/Latex_Template/HEADRULE.STY
test/new/Latex_Template/Mathformule.eps
test/new/Latex_Template/Mathformule.tif
test/new/Latex_Template/-r u.dvi
test/new/Latex_Template/-r u.pdf
test/new/Latex_Template/-r u.ps
test/new/Latex_Template/-r u.tex
test/new/Latex_Template/-r u.tex.bak
test/ccc.txt.bz2
test/bbb.txt.gz
[chongyan@localhost ~]$
```

(2) 使用 bzip2 压缩格式对某个目录进行打包操作，显示压缩过程，压缩包规范后缀为.tar.bz2

```
tar cjvf new2.tar.bz2 test2
```

```
[chongyan@localhost ~]$ tar cjvf new2.tar.bz2 test2
test2/
test2/ddd.txt
test2/dd.txt
```

(3) 解压某个压缩包到当前工作目录

```
tar xvf new2.tar.bz2
```

```
tar: Error: is not recoverable: exiting now
[chongyan@localhost ~]$ tar xvf new2.tar.bz2
test2/
test2/ddd.txt
test2/dd.txt
```

## 5. 磁盘操作命令

### 5.1 mount 命令

#### 5.1.1 作用及语法

功能：挂载设备，包括光盘镜像文件、移动硬盘、U 盘等。

格式

```
mount [-t vfstype] [-o 选项] device dir
```

选项：[-t vfstype]指定文件系统的类型，通常不必指定。mount 命令自动选择正确的类型。vfstype 中常用选项类型有：

vfstype 中的各个选项	作用
iso9660	光盘或光盘镜像
msdos	DOS fat16 文件系统
vfat	Windows 9x fat32 文件系统
ntfs	Windows NT ntfs 文件系统
smbfs	Mount Windows 文件网络共享
nfs	UNIX(LINUX) 文件网络共享

#### 5.1.2 测试

查看当前系统中已有的文件系统信息

```
mount
```

```

chongyan@localhost ~]$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=482100k,nr_inodes=120525,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_prio,net_cls)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/sda3 on / type xfs (rw,relatime,attr2,inode64,noquota)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=33,pgpr=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=14010)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/dev/sda1 on /boot type xfs (rw,relatime,attr2,inode64,noquota)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=99568k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
chongyan@localhost ~]$

```

## 5.2 umount 命令

### 5.2.1 作用及语法

功能：卸载一个文件系统，它的使用权限是超级用户。

格式：umount <挂载点|设备>

## 5.3 df 命令

### 5.3.1 作用及语法

功能：查看当前硬盘的分区信息。

格式：df [选项]

选项：

选项	作用
-a	把全部的文件系统和各分区的磁盘使用情况列出来。
-i	列出 i-nodes 结点
-k	把各分区的大小和挂上来的文件分区的大小以 kb 为单位表示
-h	把各分区的大小和挂上来的文件分区的大小以 GB 为单位表示
-t	列出某一个文件系统的所有分区磁盘空间使用量
-x	和列出不是某个文件系统的所有分区的使用量选项相反
-T	列出每个分区所属文件系统的名称

### 5.3.2 测试

(1) 查看带有容量单位的显示系统全部磁盘使用情况

df -h

```
[chongyan@localhost ~]$ df -h
文件系统      容量  已用  可用  已用% 挂载点
devtmpfs      471M    0  471M    0% /dev
tmpfs         487M    0  487M    0% /dev/shm
tmpfs         487M   8.6M  478M    2% /run
tmpfs         487M    0  487M    0% /sys/fs/cgroup
/dev/sda3      18G   5.4G   13G   31% /
/dev/sda1      297M  152M  145M   52% /boot
tmpfs          98M   28K   98M    1% /run/user/1000
```

(2) 显示系统中所有文件系统格式为 xfs 的磁盘分区使用量情况

```
df -t xfs
```

```
[chongyan@localhost ~]$ df -t xfs
文件系统      1K-块  已用  可用  已用% 挂载点
/dev/sda3     18555904 5652460 12903444  31% /
/dev/sda1      303780 155536  148244  52% /boot
```

## 5.4 du 命令

### 5.4.1 作用及语法

功能：查看当前目录下所有文件及目录的信息。

格式：du [选项]

选项：

选项	作用
-a	列出所有文件及目录的大小
-h	以 Gb 或 Mb 等形式显示文件或目录的大小
-b	显示目录和文件的大小，以 B 为单位
-c	最后加上一个总计
-s	只列出各文件大小的总和
-x	值计算属于同一个文件系统的文件

### 5.4.2 测试

(1) 以易读的容量格式显示指定目录内各个文件的大小信息

```
du -h test
```

```
[chongyan@localhost ~]$ du -h test
136M  test/new/Latex_Template
136M  test/new
136M  test
```

(2) 以易读的容量格式显示指定目录内总文件的大小信息

```
du -sh test
```

```
[ chongyan@localhost ~]$ du -sh test
136M    test
```

## 6. 关机重启命令

### 6.1 shutdown 命令

#### 6.1.1 作用及语法

功能：安全关机。

格式：shutdown [选项] [时间] [警告信息]

#### 6.1.2 测试

系统在 2min 后关机

```
shutdown -h +2
```

### 6.2 halt 命令

#### 6.2.1 作用及语法

功能：关机

格式：halt [选项]

选项：常用选项-f，用于控制强行关机。

说明：halt 命令单独使用时，等于执行 shutdown -h 命令。停掉系统服务后安全关机。执行-f 选项后，不调用 shutdown 命令直接进行强行关机。

#### 6.2.2 测试

强制关机

```
halt -f
```

### 6.3 poweroff 命令

#### 6.3.1 作用及语法

功能：关机。

格式：poweroff

#### 6.3.2 测试

```
poweroff
```

## 6.4 reboot 命令

### 6.4.1 作用及语法

功能：重启系统

格式：reboot

### 6.4.2 测试

重启

reboot

## 6.5 init 命令

### 6.5.1 作用及语法

功能：切换 Ubuntu 的运行级别。

格式：init [运行级别]

说明：init 共有 7 个运行级别

0-停机

1-单用户模式

2-多用户，没有 NFS

3-完全多用户模式（标准的运行级）

4-没有用到

5-X11 (x window)

6-重新启动

init 进程是系统启动进程，它是系统启动后的由内核创建的第一个进程，进程号为 1。

### 6.5.2 测试

(1)关闭服务器

init 0

(2)切换单用户模式

init 1



## 7. 变换用户身份命令

### 7.1 su 命令

#### 7.1.1 作用及语法

功能：切换用户。

格式：su 用户名

#### 7.1.2 测试

(1) 切换到 root su

(2) 切换到 chongyan 用户 su chongyan

```
[chongyan@localhost ~]$ su
密码：
[root@localhost chongyan]# su chongyan
[chongyan@localhost ~]$ su -
密码：
上一次登录：日 10月  9 09:30:07 PDT 2022pts/0 上
[root@localhost ~]#
```

### 7.2 sudo 命令

#### 7.2.1 作用及语法

功能：该命令的含义是“super do”，是指以超级管理员地身份执行某种操作。

格式：sudo 命令

#### 7.2.2 测试

以超级管理员身份执行 ls 命令

sudo ls

```
[chongyan@localhost ~]$ sudo ls
[sudo] chongyan 的密码：
new2.tar.bz2  shell  test2      公共  视频  文档  音乐
new.tar.gz   test  test.cpp   模板  图片  下载  桌面
[chongyan@localhost ~]#
```

### 7.3 useradd 命令

#### 7.3.1 作用及语法

功能：创建一个新用户。

格式：useradd 新用户名

说明：root 用户操作

### 7.3.2 测试

添加用户 xinxin

useradd xinxin

```
passwd u.  
[root@localhost home] # ls  
chongyan  
[root@localhost home] # useradd xinxin  
[root@localhost home] # ls  
chongyan xinxin
```

## 7.4 passwd 命令

### 7.4.1 作用及语法

功能：修改用户密码。

格式：passwd 用户名

说明：root 用户执行。

### 7.4.2 测试

（1）为新建的 xinxin 用户设置密码

passwd xinxin --stdin

```
passwd: all authentication tokens updated successfully.  
[root@localhost ~]# passwd xinxin --stdin  
Changing password for user xinxin.  
xinxin  
passwd: all authentication tokens updated successfully.
```

（2）修改 xinxin 用户密码

passwd

```
passwd  
[xinxin@localhost ~]$ passwd  
Changing password for user xinxin.  
Changing password for xinxin.  
(current) UNIX password:  
New password:  
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word  
New password:  
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[xinxin@localhost ~]$
```

## 8. 网络操作命令

### 8.1 ifconfig 命令

#### 8.1.1 作用及语法

功能：查看 IP 地址

格式：ifconfig

#### 8.1.2 测试

查看本机 IP 地址 ifconfig

```
[chongyan@localhost ~]$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.83.137 netmask 255.255.255.0 broadcast 192.168.83.255
    inet6 fe80::1cee:f654:beb0:a6f2 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:cd:ca:41 txqueuelen 1000 (Ethernet)
    RX packets 17838 bytes 1755423 (1.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2352 bytes 226397 (221.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 72 bytes 6252 (6.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 72 bytes 6252 (6.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:b9:71:12 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

### 8.2 ping 命令

#### 8.2.1 作用及语法

功能：测试网络是否畅通。

格式：ping [选项] 主机名或 IP 地址

#### 8.2.2 测试

(1) 测试与指定网站服务器之间的网络连通性，发送请求包限定为 4 次

ping -c 4 www.linuxcool.com

```
[chongyan@localhost ~]$ ping -c 4 www.linuxcool.com
PING www.linuxcool.com.w.kunlunar.com (223.111.24.102) 56(84) bytes of data.
64 bytes from 223.111.24.102 (223.111.24.102): icmp_seq=1 ttl=128 time=29.5 ms
64 bytes from 223.111.24.102 (223.111.24.102): icmp_seq=2 ttl=128 time=58.8 ms
64 bytes from 223.111.24.102 (223.111.24.102): icmp_seq=3 ttl=128 time=44.8 ms
64 bytes from 223.111.24.102 (223.111.24.102): icmp_seq=4 ttl=128 time=20.6 ms

--- www.linuxcool.com.w.kunlunar.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 20.690/38.480/58.846/14.587 ms
[chongyan@localhost ~]$
```

(2) 测试与指定主机之间的网络连通性，发送 3 次请求包，每次间隔 0.2 秒，最长等待时间为 3 秒

```
ping -c 3 -i 0.2 -W 3 192.168.133.1
```

```
[chongyan@localhost ~]$ ping -c 3 -i 0.2 -W 3 192.168.133.1
PING 192.168.133.1 (192.168.133.1) 56(84) bytes of data.
64 bytes from 192.168.133.1: icmp_seq=1 ttl=128 time=0.612 ms
64 bytes from 192.168.133.1: icmp_seq=2 ttl=128 time=0.850 ms
64 bytes from 192.168.133.1: icmp_seq=3 ttl=128 time=0.871 ms

--- 192.168.133.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 414ms
rtt min/avg/max/mdev = 0.612/0.777/0.871/0.121 ms
[chongyan@localhost ~]$
```

## 8.3 netstat 命令

### 8.3.1 作用及语法

功能：检测网络端口的链接情况，是监控 TCP/IP 网络的有效工具。

格式：netstat [选项]

选项：

选项	作用
-a	显示所有有效的连接信息。包括已经建立的连接和正在监听的连接
-r	显示路由的信息
-i	显示 interface 的内容
-n	使用网络 IP 地址代替名称，显示网络的连接情况
-o	显示计时器
-c	持续列出网络状态
-t	显示 TCP 协议的连接情况
-u	显示 UDP 协议的连接情况
-v	显示版本信息
-w	显示 RAW 传输协议的连接情况

### 8.3.2 测试

#### (1) 显示系统网络状态中的所有连接信息

netstat -a

```
[chongyan@localhost ~]$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0: sunrpc          0.0.0.0:*               LISTEN
tcp        0      0 localhost.locald: domain 0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0: ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost: ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost: smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]: sunrpc             [::]:*                  LISTEN
tcp6       0      0 [::]: http               [::]:*                  LISTEN
tcp6       0      0 [::]: ssh                [::]:*                  LISTEN
tcp6       0      0 localhost: ipp           [::]:*                  LISTEN
tcp6       0      0 localhost: smtp          [::]:*                  LISTEN
udp        0      0 localhost: 323            0.0.0.0:*               7
udp        0      0 localhost.locald: domain 0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: bootps          0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: bootpc          0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: 33364           0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: sunrpc          0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: xfr              0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0: mdns             0.0.0.0:*               LISTEN
udp6       0      0 localhost: 323            [::]:*                  LISTEN
udp6       0      0 [::]: sunrpc             [::]:*                  LISTEN
udp6       0      0 [::]: xfr                 [::]:*                  LISTEN
raw6       0      0 [::]: ipv6-icmp           [::]:*                  LISTEN

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State      I-Node   Path
unix   2      [ ACC ] STREAM    LISTENING   34711    /tmp/.esd-1000/socket
unix   2      [ ACC ] STREAM    LISTENING   151568   /var/run/avahi-daemon/socket
```

#### (2) 显示系统网络状态中的 UDP 连接端口号使用信息

```
[chongyan@localhost ~]$ netstat -ap
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
udp        0      0 localhost: 323            0.0.0.0:*               -
udp        0      0 localhost.locald: domain 0.0.0.0:*               -
udp        0      0 0.0.0.0: bootps          0.0.0.0:*               -
udp        0      0 0.0.0.0: bootpc          0.0.0.0:*               -
udp        0      0 0.0.0.0: 33364           0.0.0.0:*               -
udp        0      0 0.0.0.0: sunrpc          0.0.0.0:*               -
udp        0      0 0.0.0.0: xfr              0.0.0.0:*               -
udp        0      0 0.0.0.0: mdns             0.0.0.0:*               -
udp6       0      0 localhost: 323            [::]:*                  -
udp6       0      0 [::]: sunrpc             [::]:*                  -
udp6       0      0 [::]: xfr                 [::]:*                  -
[chongyan@localhost ~]$
```

## 8.4 ftp 命令和 bye 命令

### 8.4.1 作用及语法

功能：登录 FTP 服务器。该命令允许用户使用 FTP 协议进行文件传输，实现文件的上传和下载。

格式：ftp 主机名/IP 地址

### 8.4.2 测试

ftp www.linuxcool.com

## 连接 FTP 服务器

```
chongyan@localhost ~]$ ftp www.linuxcool.com
Trying 223.111.24.105...
ftp: connect to address 223.111.24.105拒绝连接
```

## 9. 其他命令

### 9.1 echo 命令

#### 9.1.1 作用及语法

功能：显示命令行中的字符串。主要用于输出提示信息。

格式：echo [选项] [字符串]

选项：-n 表示输出字符串后，光标不换行。

#### 9.1.2 测试

### 9.2 more 命令和 less 命令

#### 9.2.1 作用及语法

功能：对文件内容或者查询结果分屏显示

格式：more [选项] 文件名

less [选项] 文件名

#### 9.2.2 测试

### 9.3 help 命令和 man 命令

#### 9.3.1 作用及语法

功能：显示每个命令的格式用法。

格式：help 命令名

man 命令名

说明：help 命名专门用于显示系统内部命令的格式用法。man 命令可以显示系统内部命令和外部命令的解释信息。

#### 9.3.2 测试

### 9.4 cal 命令

#### 9.4.1 作用及语法

功能：显示日历。

格式：cal [选项] [月份[年份]]

#### 9.4.2 测试

显示当前月份的日历

cal

```
[chongyan@localhost ~]$ cal
      十月 2022
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
```

### 9.5 date 命令

#### 9.5.1 作用及语法

功能：显示及设定系统的日期和时间。

格式：date [选项] 显示时间格式（以+号开始，后边加格式）

date 设定时间格式

选项：

选项	作用
-u	使用格林尼治时间
-r	最后一次修改文件的时间

#### 9.5.2 测试

(1) 以默认格式输出系统当前的日期与时间信息

date

```
30 31
[chongyan@localhost ~]$ date
2022年 10月 13日 星期四 11:46:26 PDT
```

(2) 按照“小时:分钟:秒”的指定格式输出系统当前的时间信息

date "+%Y-%m-%d"

```
[chongyan@localhost ~]$ date "+%Y-%m-%d"
2022-10-13
```

## 10. 管道符

### 10.1 作用及语法

作用：是用于两个命令或者多个命令相链接，将前边的命令的执行结果传递到后边的命令

语法：bash 命令 | 管道命令

### 10.2 测试

(1) 统计 home 里一共有多少文件

ls -al /home | wc -l

```
[chongyan@localhost ~]$ ls -al /home | wc -l
5
```

(2) 查看当前系统登录用户的数量

who | wc -l

```
[chongyan@localhost ~]$ who | wc -l
2
```

(3) 获取 ip 地址在哪一行

ifconfig ens33 | grep netmask

```
[chongyan@localhost ~]$ ifconfig ens33 | grep netmask
    inet 192.168.83.137 netmask 255.255.255.0 broadcast 192.168.83.255
```

(4) 显示 passwd 中以 root 为开头的行

cat /etc/passwd | grep "^root"

```
[chongyan@localhost ~]$ cat /etc/passwd | grep "^root"
root:x:0:0:root,,123456,123456:/root:/bin/bash
```

## 实验二 shell 编程

### 1. Linux 下如何执行 shell 脚本

1.1 用 shell 程序执行脚本：根据 shell 脚本的类型选择程序，常用的有：sh、bash、tcsh 等，然后输入命令：bash shell.sh



1.2 直接执行脚本：首先加上可执行权限，执行：chmod +x shell.sh

然后执行：./shell.sh 或者 sh shell.sh

## 2. 编写脚本

### 2.1 通过 shell 脚本封请求数异常的 IP

#### 2.1.1 找到 apache 日志文件，并查看格式

```
[root@localhost httpd]# cat access_log
127.0.0.1 - - [10/Oct/2022:06:02:56 - 0700] "GET / HTTP/1.1" 403 4897 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [10/Oct/2022:06:02:56 - 0700] "GET /noindex/css/bootstrap.min.css HTTP/1.1" 200 19341 "http://127.0.0.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [10/Oct/2022:06:02:56 - 0700] "GET /noindex/css/open-sans.css HTTP/1.1" 200 5081 "http://127.0.0.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
127.0.0.1 - - [10/Oct/2022:06:02:56 - 0700] "GET /images/poweredby.png HTTP/1.1"
```

#### 2.1.2 编写脚本如下

```
#!/bin/bash
#定义封IP方法
function ban_ip()
{
    t=`date -d "- 1 min" +%Y:%H:%M`
    log=/var/log/httpd/access_log
    #当IP前一分钟请求数大于100,记录到临时文件
    egrep "$t:[0-9]+" $log|awk '{print $1}'|sort -n|uniq -c|sort -n|awk '$1>100 {print $2}' >/tmp/ban_ip.list
    #当记录IP的临时文件不为空时,开始封IP
    ban_n=`wc -l /tmp/ban_ip.list|awk '{print $1}'`
    if [ $ban_n -ne 0 ]
    then
        for ip in `cat /tmp/ban_ip.list`
        do
            iptables -I INPUT -s $ip -j REJECT
        done
    fi
}
#定义解封IP的方法
function allow_ip()
{
    #将发送数据包小于5的被封IP记录到临时文件
    iptables -nvL INPUT|sed '1d'|awk '$1<5 {print $8}' >/tmp/allow_ip.list
    #当临时文件不为空时,开始解封IP
    allow_n=`wc -l /tmp/allow_ip.list|awk '{print $1}'`
    if [ $allow_n -ne 0 ]
    then
        for ip in `cat /tmp/allow_ip.list`
        do
            iptables -D INPUT -s $ip -j REJECT
        done
    fi
    #清空iptables计数器(记录数据包那一列的值将被清零)
    iptables -Z
}
#执行封IP操作,但每半小时才执行一次解封IP操作
d=`date +%M`
if [ $d == "00" ]||[ $d == "30" ]
then
    allow_ip
    ban_ip
else
    ban_ip
fi
```

#### 2.1.3 运行

```
[chongyan@localhost shell]$ sh ban_ip.sh
grep: /var/log/httpd/access_log: 权限不够
[chongyan@localhost shell]$ su
密码:
[root@localhost shell]# sh ban_ip.sh
[root@localhost shell]# cd /tmp
[root@localhost tmp]# ls
ban_ip.list

[root@localhost tmp]# cat ban_ip.list
[root@localhost tmp]# █
```

由于没有一分钟访问次数大于 100 次的 ip, 故 ban\_ip.list 为空

## 2.2 猜拳游戏

2.2.1 编写代码如下:

```
#!/bin/bash

game=(石头 剪刀 布)
num=$((RANDOM%3))
computer=$((game[$num]))

echo "请根据下列提示选择您的出拳手势"
echo " 1. 石头"
echo " 2. 剪刀"
echo " 3. 布"

cnt=0

for i in {1..3}
do
    read -p "请选择 1-3 : " person
    case $person in
    1)
        if [ $num -eq 0 ]
        then
            echo "平局"
        elif [ $num -eq 1 ]
        then
            echo "你赢"
            cnt=$((cnt+1))
        else
            echo "计算机赢"
        fi;;
    2)
        if [ $num -eq 0 ]
        then
            echo "计算机赢"
        elif [ $num -eq 1 ]
        then
            echo "平局"
        else
            echo "你赢"
            cnt=$((cnt+1))
        fi;;
    3)
        if [ $num -eq 0 ]
        then
            echo "你赢"
            cnt=$((cnt+1))
        elif [ $num -eq 1 ]
        then
            echo "计算机赢"
        else
            echo "平局"
        fi;;
    *)
        echo "必须输入1-3 的数字"
    esac
done
```

### 2.2.2 运行

```
[root@localhost shell]# sh caiquan.sh
请根据下列提示选择您的出拳手势
1. 石头
2. 剪刀
3. 布
请选择 1- 3 : 2
你赢
请选择 1- 3 : 1
计算机赢
请选择 1- 3 : 3
平局
[root@localhost shell]#
```

## 实验三 在 Linux 下搭建 Apache 服务器

### 1.搭建 apache 服务

(1) 检查是否安装过 httpd

http -v

```
[chongyan@localhost ~]$ httpd -v
bash: httpd: 未找到命令...
[chongyan@localhost ~]$
```

(2) 安装 httpd

yum install httpd

```
[chongyan@localhost ~]$ yum install httpd
已加载插件：fastestmirror, langpacks
您需要 root 权限执行此命令。
[chongyan@localhost ~]$ su
密码：
[root@localhost chongyan]# yum install httpd
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.lzu.edu.cn
* extras: mirror.lzu.edu.cn
* updates: mirror.lzu.edu.cn
base
extras
updates
(1/2): extras/7/x86_64/primary_db
(2/2): updates/7/x86_64/primary_db
正在解决依赖关系
--> 正在检查事务
---> 软件包 httpd.x86_64.0.2.4-6-97.el7.centos.5
--> 正在处理依赖关系 httpd-tools = 2.4.6-97.el7.c
ntos.5.x86_64 重要
```

作为依赖被安装：

apr.x86\_64 0:1.4.8-7.el7

httpd-tools.x86\_64 0:2.4.6-97.el7.centos.5

完毕！

```
[root@localhost chongyan]# █
```

(3) 启动 httpd 服务并添加到开机启动项中

```
systemctl start httpd
```

```
systemctl enable httpd
```

```
[root@localhost chongyan]# systemctl start httpd
[root@localhost chongyan]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib
/systemd/system/httpd.service.
[root@localhost chongyan]#
```

(4) 访问 <http://127.0.0.1>, 看到正常的 httpd 服务程序的默认界面



(5) 将 chongyan'wold 写入 /www/html/index.html, 访问 http://127.0.0.1 看到 index.html 界面




```
echo "chongyan'wold" >/var/www/html/index.html
```

```
chongyan@localhost html]$ echo "chongyan'wold" >/var/www/html/index.html
bash: /var/www/html/index.html: 权限不够
chongyan@localhost html]$ su
密码:
root@localhost html# echo "chongyan'wold" >/var/www/html/index.html
root@localhost html# █
```

## 2.制作个人介绍网页并部署到 apache 服务器上, 并能通过浏览器访问

### (1) 制作个人介绍网页

先在本地制作好网页

 assets	2022/10/11 15:50	文件夹	
 images	2022/10/11 18:49	文件夹	
 index.html	2022/10/11 21:05	SLBrowser HTM...	7 KB

主要的文件为 index.html

代码如下：

```
1. <!DOCTYPE HTML>
2.
3. <html>
4.     <head>
5.         <title>周子欣的个人简介站</title>
6.         <meta charset="utf-8" />
7.         <meta name="viewport" content="width=device-width, initial-
            scale=1, user-scalable=no" />
8.         <link rel="stylesheet" href="assets/css/main.css" />
9.         <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></n
            oscript>
10.    </head>
11.    <body class="is-preload">
12.
13.        <!-- Page Wrapper -->
14.        <div id="page-wrapper">
15.
16.            <!-- Header -->
17.            <header id="header" class="alt">
18.                <h1><a href="index.html">首页</a></h1>
19.                <nav>
20.                    <a href="#menu">目录</a>
21.                </nav>
22.            </header>
23.
24.            <!-- Menu -->
25.            <nav id="menu">
26.                <div class="inner">
27.                    <h2>Menu</h2>
28.                    <ul class="links">
29.                        <li><a href="index.html">首页</a></li>
30.                        <li><a href="https://c1ongyan.github.io/">个人
                            博客</a></li>
31.                    </ul>
32.                    <a href="#" class="close">Close</a>
33.                </div>
```

```

34.         </nav>
35.
36.         <!-- Banner -->
37.         <section id="banner">
38.             <div class="inner">
39.                 <div class="logo"><span class="icon fa-
gem"></span></div>
40.                 <h2>        大家好，这里是周子欣个人简介站</h2>
41.                 <p>如果你想要和我讨论问题，这是我的博客地址
<a href="https://c1ongyan.github.io/"><strong> Click here</strong></a>或者 发
邮件给我<strong>（2480838239@qq.com）</strong>。 欢迎联系我一起讨论问题 :)</p>
42.             </div>
43.         </section>
44.         <!-- Wrapper -->
45.         <section id="wrapper">
46.
47.             <!-- One -->
48.             <section id="one" class="wrapper spotlight style1
">
49.                 <div class="inner">
50.                     <a href="images/pic01.jpg" class="image">
</a>
51.                 <div class="content">
52.                     <h2 class="major">个人简介</h2>
53.                     <p>本人热爱学习，学习成绩突出，在校绩点排
名长期保持在前列，本学年的平均绩点排名位列前 10。在 20-21 学年获得校级二等奖，已经通过
CET-6。不论是选修还是必修都力图学懂学通，始终充满学习的热情。
54.             本人不仅在课程学习方面名列前茅，更是在各类实践竞赛里取得优秀的成绩。比如第五届全
国强网杯中获得优胜奖，在第一届深育杯中获得团体三等奖的成绩，也在校内举办的网络安全竞
赛中多次获得一等奖，二等奖的成绩。
55.             在创新科研方面，本人已和其他优秀的同学一起成功申请了“基于机器学习的图像检索系统”
的项目，并且正以“3D 空间中智能位置隐私保护方法研究”项目申报 2022 年挑战杯。
56. 本人乐于助人，有集体荣誉感。在学院担任女篮领队，带领计算机女篮参加 2021 年新生赛和
2022 年行健杯，参与比赛的同时协调后勤及比赛训练期间的具体安排，获得 2022 年行健杯女篮
乙级三等奖的成绩；在班级担任组织委员，负责班级的日常管理，组织各类志愿活动，获得了组
织委员之星的荣誉。
57. 本人爱好广泛。喜欢篮球，羽毛球，跑步，定向越野。在中国矿业大学定向越野比赛中代表学院
获得第七名，获得 2021 年行健杯女篮乙级三等奖和 2022 年行健杯女篮三等奖，也参与了 2020
年新生赛，以领队的身份参加 2021 年新生赛。也喜欢阅读，喜欢读法律和科普方面的书籍，比如
罗翔老师的《刑法学讲义》《圈圈正义》，曹天元的《上帝掷骰子吗》等等。也喜欢手工制作，
在中国矿业大学泥塑比赛中获得个人奖三等奖。
58. </p>
59.         </div>
60.     </div>

```



```

61.                                     </section>
62.
63.                                     <!-- Two -->
64.                                     <section id="two" class="wrapper alt spotlight st
65.                                     <div class="inner">
66.                                     <a href="images/pic02.jpg" class="image">
67.                                     </a>
68.                                     <div class="content">
69.                                     <h2 class="major">教育背景</h2>
70.                                     <p>2020.9-至今 就读于中国矿业大学信息安
71.                                     全专业 </p>
72.                                     </div>
73.                                     </div>
74.                                     </section>
75.                                     <!-- Three -->
76.                                     <section id="three" class="wrapper spotlight styl
77.                                     <div class="inner">
78.                                     <a href="images/pic03.jpg" class="image">
79.                                     </a>
80.                                     <div class="content">
81.                                     <h2 class="major">奖项经历</h2>
82.                                     <p>
83.                                     <li>2020-2021 学年获校二等奖学金
84.                                     </li>
85.                                     <li>2021-2022 学年获第一届深育杯三等
86.                                     奖</li>
87.                                     <li>2021-2022 学年校级大创 1 项
88.                                     </li>
89.                                     <li>2021-2022 学年行健杯乙级三等奖，
90.                                     泥塑大赛个人奖三等奖</li>
91.                                     <li>2020-2021 学年获强网青少赛第 19
92.                                     名</li>
93.                                     <li>2020-2021 学年获强网杯线下第 15
94.                                     名获得优秀奖</li>
95.                                     <li>2020-2021 学年获中国矿业大学网络
96.                                     安全技能大赛入门赛一等奖</li>
97.                                     <li>2020-2021 学年获中国矿业大学安全
98.                                     技能大赛岁末赛三等奖</li>
99.                                     <li>2020-2021 学年获中国矿业大学新春
100.                                    赛二等奖</li>

```

```

91.                <li>2020-2021 学年获中国矿业大学初夏
    赛二等奖</li>
92.                <li>2020-2021 学年获中国矿业大学秋季
    赛二等奖</li>
93.                <li>2020-2021 学年获中国矿业大学秋季
    赛二等奖</li>
94.                </p>
95.
96.                </div>
97.                </div>
98.                </section>
99.
100.               <!-- Four -->
101.               <section id="four" class="wrapper alt style1">
102.               <div class="inner">
103.               <h2 class="major">技能</h2>
104.               <p>
105.               <li>熟练使用 burpsuite、appscan、
    awvs、nessus、metasploit、nmap、sqlmap 等渗透测试工具</li>
106.               <li>熟悉 C++,GO,了解 python 等语言
    </li>
107.               <li>熟悉 linux 操作系统</li>
108.               <li>具有较好的工作习惯及较强的文档、报
    告、方案编写能力</li>
109.               </p>
110.
111.               </div>
112.               </section>
113.
114.
115.               <!-- Footer -->
116.               <section id="footer">
117.               <div class="inner">
118.               <a href="images/2.jpg" class="image"><im
    g src="images/2.jpg" alt="" /></a>
119.               <div class="content">
120.               <h2 class="major">个人博客</h2>
121.               <p>
122.               记录学习收获
123.               <a href="https://c1ongyan.github
    .io/"><strong> 去到博客</strong></a>
124.               </p>
125.
126.               </div>

```

```

127.                                </div>
128.                                </section>
129.
130.
131.                                </div>
132.
133.                                <!-- Scripts -->
134.                                <script src="assets/js/jquery.min.js"></script>
135.                                <script src="assets/js/jquery.scrollex.min.js"></script>
136.                                <script src="assets/js/browser.min.js"></script>
137.                                <script src="assets/js/breakpoints.min.js"></script>
138.                                <script src="assets/js/util.js"></script>
139.                                <script src="assets/js/main.js"></script>
140.
141.                                </body>
142. </html>

```

复制文件到 centos 桌面上

(2) 将制作好的网页部署到 linux 服务器上

在/var/www/html 下创建文件夹 xin

```

[ root@localhost html] # mkdir xin
[ root@localhost html] # ls
index.html  xin
[ root@localhost html] #

```

将本地写好的网页文件复制到/var/www/html 中

```

[ root@localhost 桌面] # mv assets /var/www/html/
[ root@localhost 桌面] # mv images /var/www/html/
[ root@localhost 桌面] # mv index.html /var/www/html/
[ root@localhost 桌面] #

```

访问 <http://127.0.0.1> 但是发现显示没有权限访问，将所有相关文件权限都设置成，但仍然显示没有权限

```
[chongyan@localhost html]$ chmod 777 index.html
[chongyan@localhost html]$ ls -l
总用量 8
drwxrwxr-x. 4 chongyan chongyan 27 10月 11 04:09 assets
drwxrwxrwx. 2 chongyan chongyan 97 10月 11 03:49 images
-rwxrwxrwx. 1 chongyan chongyan 6757 10月 11 03:50 index.html
drwxrwxrwx. 2 root root 6 10月 11 05:10 xin
[chongyan@localhost html]$ chmod 777 images
[chongyan@localhost html]$ chmod 777 assets
[chongyan@localhost html]$ ls -l
总用量 8
drwxrwxrwx. 4 chongyan chongyan 27 10月 11 04:09 assets
drwxrwxrwx. 2 chongyan chongyan 97 10月 11 03:49 images
-rwxrwxrwx. 1 chongyan chongyan 6757 10月 11 03:50 index.html
drwxrwxrwx. 2 root root 6 10月 11 05:10 xin
```

最后，找到方法，将/etc/sysconfig/selinux 中 selinux 的值改为 disabled  
即 selinux=disabled，再输入执行输入执行：setenforce 0 即可

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
[chongyan@localhost html]$ sudo vim /etc/sysconfig/selinux
[sudo] chongyan 的密码 :
[chongyan@localhost html]$ setenforce 0
setenforce: setenforce() failed
[chongyan@localhost html]$ sudo setenforce 0
```

```
[chongyan@localhost html]$ sudo vim /etc/sysconfig/selinux
[sudo] chongyan 的密码 :
[chongyan@localhost html]$ setenforce 0
setenforce: setenforce() failed
[chongyan@localhost html]$ sudo setenforce 0
[chongyan@localhost html]$ root
bash: root: 未找到命令...
[chongyan@localhost html]$ ls
assets images index.html xin
[chongyan@localhost html]$
```

访问 <http://127.0.0.1>，可以看到个人简介页面

浏览效果如下：

