

基础知识

基础知识

- Terminal

- shell

- 命令

 - 命令细节

 - 命令名

 - 分隔符

 - 选项

 - 操作对象

 - 格式约定

- 程序与进程

- 路径

 - 绝对路径

 - 相对路径

 - 远程路径

 - 路径相关的命令

 - 特殊路径

- 软件

- 配置方式

- 隐藏文件

- 文件类型

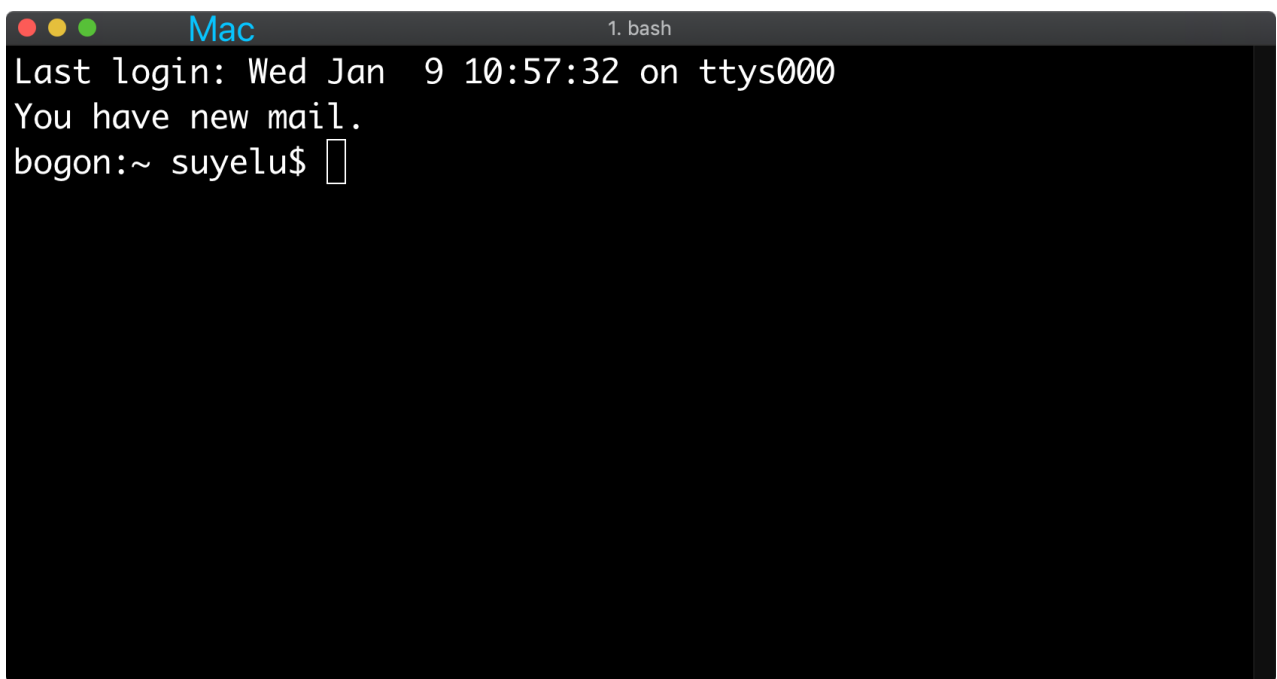
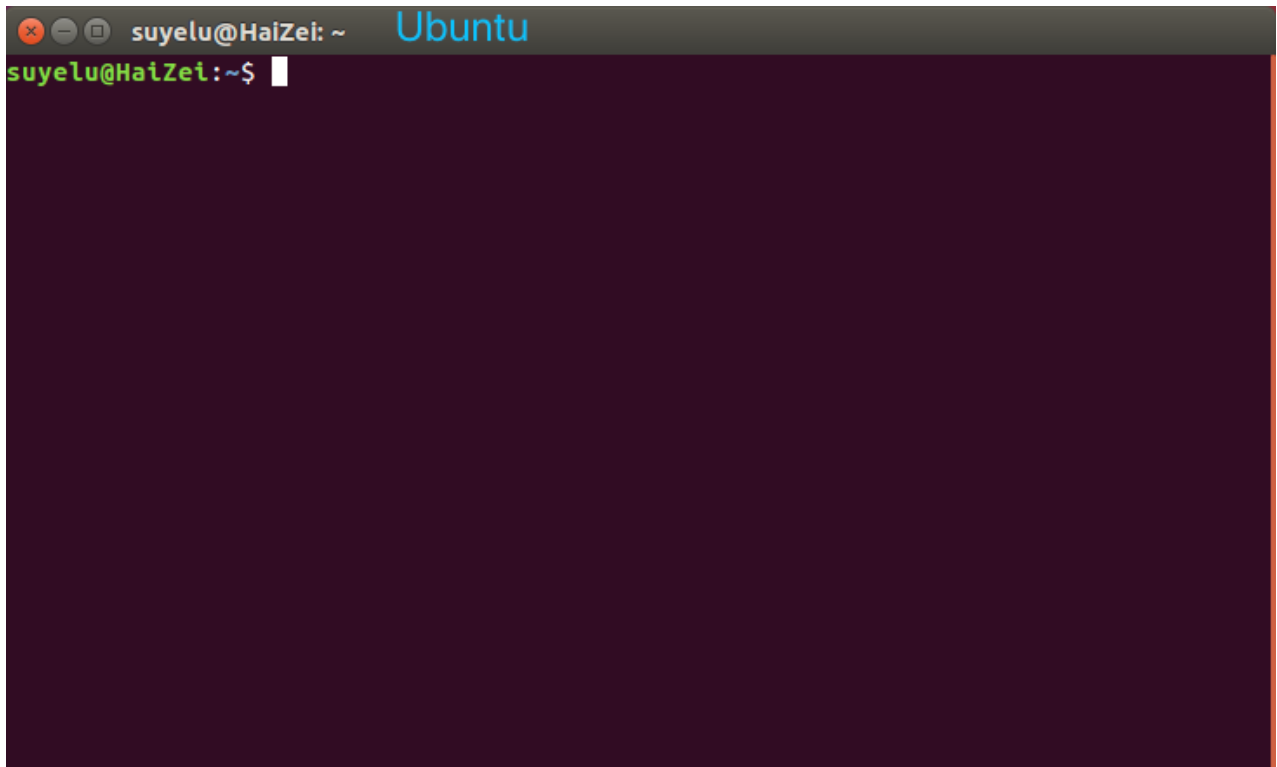
- 文件权限

 - 与文件权限有关的命令

- 用户

Terminal

终端（`terminal`）= `tty`（Teletypewriter，电传打印机），作用是提供一个命令的输入输出环境，在linux操作系统下使用组合键 `ctrl + alt + T` 打开的就是终端，可以认为 `terminal` 和 `tty` 是同义词。



Mac 推荐安装 [iterm2](#)

shell

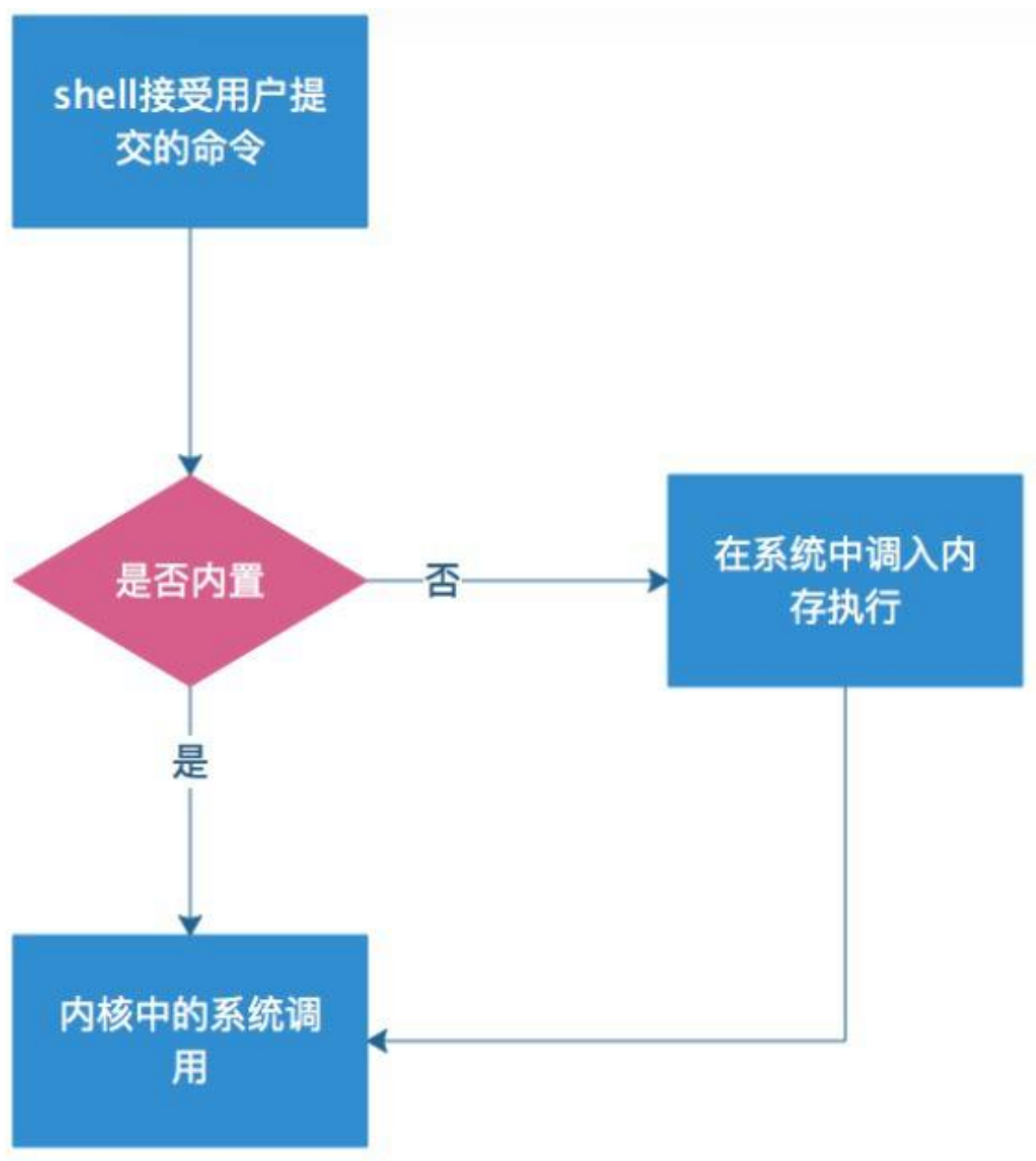
shell 是一种人机交互的接口。**shell** 有壳的意思，是指“*提供使用者使用界面*”的软件，是一种命令解析器，是Linux内核的一个壳，负责外界与Linux内核的交互。

windows 的cmd就是一种shell。所以 **shell** 并不只是指命令语言-Bash/Zsh。

其实GUI本身也是一种shell。

用户在 `shell` 中提交命令，`shell` 负责接收用户的命令，并扮演命令解析器的角色。

当你打开一个 `Terminal` 时，操作系统会将 `Terminal` 和 `Shell` 关联起来，当我们在 `Terminal` 中输入命令的时候，`Shell` 就负责解释命令。



命令

命令是人和计算机交互的基本单位，人使用 `命令` 将要做什么事传达给计算机，计算机做出解析，并做出回应。

命令也有自己的语法结构：

和人说话一样，命令的主要结构也可以看做为主谓宾的简单句和复杂句，在命令中，省略主语

举个例子

```
1 cp(谓语动词) fileA(宾语) fileB(宾语)
```

```
1 cp(谓语动词) -ar(方式状语) fileA(宾语) fileB(宾语)
```

一般情况下 `-` 后面加选项的缩写，`--` 后面加选项的全程。

命令细节

Linux命令由以下几个部分组成：命令名，分隔符，选项和操作对象组成。

命令名

命令名标识了命令的功能，如cp, rm, mv, rename, fdisk等等，都能很轻易的看出该命令的功能，还有一些命令，有二级子命令，如apt-get install | remove, install和remove就是apt-get的子命令。

分隔符

分隔符通常为空格，连续的多个空格会被视为一个空格。

一些特殊的符号也属于分隔符，例如管道|，重定向>,<<,<,>>,后台运行&，序列执行&&、;都是特殊的分隔符，在使用这些特殊符号的时候，不需要额外加空格，但为了使命令更易读，通常会加上空格。

选项

- 命令的选项通常用 `-` 连接，通常为一个字母，是选项的首字母。大多数命令都可以使用 `-h` 来查看该命令的帮助。
- 命令的选项也可以使用 `--` 来引导，接的是选项的全程，效果与`-`连接单个字母是一样的。

操作对象

操作对象为该命令动作的承受着。

格式约定

- 使用 `[]` 来标记可以选择的选项
- 使用 `|` 来表示不能同时使用的参数
- 选项通常紧跟命令名，当然，在很多时候我们可以省略命令的选项而使用默认参数

程序与进程

我们执行命令的时候，每一个命令，其实对应的就是系统中的一个程序。下图使用 `which` 命令查找 `cp` 命令的位置，并使用 `ll` 和 `file` 命令查看 `cp` 命令的具体信息：

```
suyelu@HaiZei-Tech:~$ which cp
/bin/cp
suyelu@HaiZei-Tech:~$ ll /bin/cp
-rwxr-xr-x 1 root root 126032 Apr  1 2012 /bin/cp*
suyelu@HaiZei-Tech:~$ file /bin/cp
/bin/cp: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=0x781ac534b5c465b8c04082f9be1c2ab68682a70e, stripped
suyelu@HaiZei-Tech:~$
```

通过 `ll` 和 `file` 命令可以看出文件 `cp` 是一个可执行的二进制文件，也就是我们平时所说的程序。

那么什么是程序呢？

计算机程序是指一组指示计算机执行动作或做出判断的指令，通常用某种程序设计语言编写，运行于某种目标体系结构上。

什么又是进程呢？

进程是程序在内存中的镜像。

路径

绝对路径

绝对路径的起始点为根目录 `/`，例如 `/usr/bin/cp` 就是一个绝对路径。

相对路径

相对路径的起始点为当前路径 `.`，假如用户当前目录为 `/home/suyelu/`，那么同样的文件 `cp`，其相对路径为 `../../usr/bin/cp`。

相对路径和绝对路径在使用上各有差异，在使用时需要考虑实际的情况选择。

远程路径

在很多时候，我们会需要访问本机之外的资源，这个时候远程路径就有了用武之地了。

远程路径的一般表示方法为：`协议://用户名:密码@位置/路径:端口` 比如：

`http://haizeix.com`

```
ftp://user:passwd@ftp.haizeix.com:21
```

远程路径根据应用的不同，具体表示方法和所需要的参数都不太一致，从以上两个远程路径就可以看出。

路径相关的命令

```
cd #change directory
```

```
ls #list
```

```
pwd #print working directory
```

特殊路径

- `~` 家目录
- `-` 上次工作目录

请自己研究一下这几个路径是什么含义。

软件

在Linux中，软件并没有像注册表这样的概念。

注册表：是Microsoft Windows中的一个重要的数据库，用于存储系统和应用程序的设置信息。安装软件。理论上只需要拷贝相关文件，并执行其主程序就可以。

通常，一个软件包含的内容会分别被拷贝到同级别的 `bin` `lib` `share` 和 `/etc` 目录下。

`bin` 存放程序的可执行文件。在系统环境变量中将该路径添加进去，就可以直接执行程序。

`lib` 库文件集中存放，方便共享。

`share` 存放程序需要的其他资源。

`/etc` 配置文件存放路径，大部分的程序的配置文件都可以在这个路径下找到。

配置方式

在Linux中，所有的配置操作都可以使用纯文本的配置文件来配置，为了方便使用，有很多程序都会提供命令接口供用户更加便捷来个性化的配置自己的系统。

在Linux中做网络配置，即可以通过修改配置文件的方式来配置，可以直接使用命令来配置：

```
suyelu@HaiZei-Tech:/etc/network$ pwd
/etc/network
suyelu@HaiZei-Tech:/etc/network$ cat interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 172.17.211.174
netmask 255.255.240.0
up route add -net 0.0.0.0 netmask 0.0.0.0 gw 172.17.223.253 dev eth0
```

`/etc/network/interface` 文件为Ubuntu的网络配置文件，通过修改该文件，并重启网络，就可以实现网络的配置工作。

当然也可以直接使用ifconfig命令来直接修改网络配置：

```
1 ifconfig eth0 172.17.211.175
```

这条命令将把设备 `eth0` 的ip地址更新为 `172.17.211.175`。

对于interfaces文件中的其他所有配置，都可以使用命令来修改，在这里不一一举例，在后期的学习中，我们会学到它们的用法。

隐藏文件

在Linux也有隐藏文件，与Windows不同的是，如果我们想查看隐藏文件不需要去文件夹选项中修改配置，让系统在显示文件的时候显示被隐藏的文件或者文件夹。

Linux的隐藏文件都是以 `.` 开头的，也就是说所有以 `.` 开头的文件都会被系统识别为隐藏文件。

```
suyelu@HaiZei-Tech:~$ ls
code
suyelu@HaiZei-Tech:~$ ls -a
.  .bash_history  .bashrc  code  .DS_Store  .ssh
.. .bash_logout  .cache  ._.DS_Store  .profile  .viminfo
suyelu@HaiZei-Tech:~$
```

如图，使用 `ls` 命令添加 `a` 选项，就可以显示隐藏文件。

这里有两个特殊的目录 `.` 和 `..`，分别为当前目录和父目录。

如何在显示隐藏文件的时候，不显示 `.` 和 `..` 呢？自己尝试一下吧！

文件类型

使用 `ll` 命令可以查看当前文件夹下所有文件的详细信息。

`ll` 等效于 `ls -al`

```
1  suyelu@HaiZei-Tech:~$ ll
2  total 52
3  drwxr-xr-x 5 suyelu suyelu 4096 Apr 27 11:40 ./
4  drwxr-xr-x 3 root    root   4096 Apr 23 13:40 ../
5  -rw----- 1 suyelu suyelu 2064 Apr 27 11:49 .bash_history
6  -rw-r--r-- 1 suyelu suyelu  220 Apr 23 13:40 .bash_logout
7  -rw-r--r-- 1 suyelu suyelu 3486 Apr 23 13:40 .bashrc
8  drwx----- 2 suyelu suyelu 4096 Apr 23 17:25 .cache/
9  drwxrwxr-x 3 suyelu suyelu 4096 Apr 23 18:07 code/
10 -rw-r--r-- 1 suyelu suyelu 4096 Apr 23 18:06 ../.DS_Store
11 -rw-r--r-- 1 suyelu suyelu 6148 Apr 23 18:43 .DS_Store
12 -rw-r--r-- 1 suyelu suyelu  675 Apr 23 13:40 .profile
13 drwxrwxr-x 2 suyelu suyelu 4096 Apr 26 21:32 .ssh/
14 -rw----- 1 suyelu suyelu 1876 Apr 27 11:40 .viminfo
```

该命令主要输出了七列内容，分别为：权限、文件数、所属用户、所属群组、文件大小、常见日期、文件名。

第一列权限，主要可以分为以下四个部分：文件类型，所属用户权限，所属组权限，其他用户权限。

以刚才执行`ll`的结果中的 `../` 目录的权限为例：

文件类型	所属用户权限	所属群组权限	其他用户权限
d	rwx	r-x	r-x

- 第一部分文件类型为 `d` ,代表这个文件是一个目录(directory)，目录是一种特殊的文件；
- 第二部分所属用户权限为 `rwx` ，代表该文件拥有者拥有可读(read),可写(write),可执行(execute)的权限；
- 第三部分所属群组权限为 `r-x` ，代表与该文件拥有者在一个群组的用户具有可读，可执行的权限，在这里 `-` 顶替了 `w` 的位置，代表没有写权限；
- 第四部分其他用户权限同样为可读，可执行。

需要注意的是 `rwx` 的顺序是固定的！

Linux中的文件类型除了目录 `d` 之外，总共有7种文件类型，在这里我们做一个简要的了解：

- `-` regular file 普通文件

- **d** directory 目录
- **l** link 链接
- **b** block 块设备 存储数据以供系统存取的接口设备，也就是硬盘
- **c** character 字符设备 串口设备，键盘，鼠标等
- **s** socket套接字
- **p** pipe 管道

对于普通文件又可以分为以下三种：

- 纯文本文件

纯文本文件使用ASCII编码，这是Linux系统中最常见的一种文件类型，之所以成为纯文本文件，是因为这种类型的文件是我们可以直接读取的内容，在Linux，几乎所有的配置文件都属于这种类型。

- 二进制文件

二进制文件是系统中的可执行文件(不包括脚本)，计算机只能认识并执行二进制文件。二进制文件不能使用 **cat** 等命令直接读出。

- 数据格式的文件

在一些程序运行过程中，需要读取特定格式的文件，这种文件被称为数据文件(data file)。这种文件通常也不能使用cat命令读出。例如：**/var/log/wtmp** 文件。

```
suyelu@HaiZei-Tech:~$ ll /var/log/wtmp
-rw-rw-r-- 1 root utmp 8832 Apr 27 11:50 /var/log/wtmp
suyelu@HaiZei-Tech:~$ file /var/log/wtmp
/var/log/wtmp: data
suyelu@HaiZei-Tech:~$ cat /var/log/wtmp
runlevel3.2.0-29-genericUjY)ty22UjYty11UjYpts/0UjYshutdown3.2.0-29-genericVjY,~reboot3.2
.0-29-genericgjY2~runlevel3.2.0-29-genericgjY+ty22LOGINgjYwtty11LOGINwpjYpts/0ts/
0suyelu1.189.203.231Y
suyelu@HaiZei-Tech:~$ last -f /var/log/wtmp
suyelu pts/0 1.189.203.231 Thu Apr 27 11:50 still logged in
reboot system boot 3.2.0-29-generic Thu Apr 27 11:49 - 14:40 (02:51)
suyelu pts/0 1.189.203.231 Thu Apr 27 11:27 - down (00:22)
suyelu pts/0 1.189.203.231 Thu Apr 27 10:02 - 10:52 (00:50)
suyelu pts/0 1.189.203.231 Wed Apr 26 20:56 - 22:00 (01:04)
suyelu pts/2 221.212.116.50 Mon Apr 24 23:12 - 23:34 (00:22)
suyelu pts/1 221.212.116.50 Mon Apr 24 22:35 - 23:36 (01:00)
suyelu pts/0 221.212.116.50 Mon Apr 24 22:35 - 23:35 (01:00)
suyelu pts/0 113.4.248.139 Mon Apr 24 10:38 - 11:17 (00:38)

wtmp begins Mon Apr 24 10:38:20 2017
suyelu@HaiZei-Tech:~$
```

使用ll命令查看文件，可以看到该文件为普通文件

```
1 suyelu@HaiZei-Tech:~$ ll /var/log/wtmp
2 -rw-rw-r-- 1 root utmp 8832 Apr 27 11:50 /var/log/wtmp
```

使用file命令查看文件详情，返回为 **data**

```
1  suyelu@HaiZei-Tech:~$ file /var/log/wtmp
2  /var/log/wtmp: data
```

使用cat命令读取该文件为乱码，但是使用last命令则可以读取该文件中的内容。

文件权限

在上节我们已经提到了文件的权限包含三组(**u** 用户, **g** 群组, **o** 其他用户), 每一组又都包含三组具体的权限(**r** 读, **w** 写, **x** 执行)。

对于文件权限, 我们也可以用下面的方式来表示:

r	w	x	r	w	x	r	w	x
(2 ²)4	(2 ¹)2	(2 ⁰)1	(2 ²)4	(2 ¹)2	(2 ⁰)1	(2 ²)4	(2 ¹)2	(2 ⁰)1

也就是权限的每一组都由三个十进制的数字表示, 该组的权限就可以简单的用着三个十进制的数字相加得到。

一个权限为 **rwxr-xr-x** 的文件, 则它的权限可以使用 **755** 来表示。

与文件权限有关的命令

```
1  chmod #更改文件权限
2  chown #更改文件所属用户
3  chgrp #更改文件所属组
```

- **chmod** 的用法

```
1  chmod a+x file #给file文件的ugo都赋予执行的权限
2  chmod o-x file #将file文件o减去执行权限
3  chmod 755 file #设置file文件的权限为rwxr-xr-x
4  chmod u=rwx,go=rx file #设置file文件的权限为rwxr-xr-x
```

- **chown** 的用法

```
1  chown suyelu:haizei file #修改file的所属用户是suyelu,所属组为haizei
2  chown -R suyelu:haizei directory #修改目录directory及目录下的所有文件的所属用户是suyelu,所属组为haizei
3  chown suyelu file #修改file的所属用户为suyelu
```

- **chgrp** 的用法

```
1  chgrp root file #修改file所属的组为root
```

更加详细的用法在后续继续学习！

用户

Linux有两类用户,分别是 `root` 和 `普通用户`。

- 超级管理员：`root`

`root` 拥有系统的完全控制权，所以在使用Linux系统的时候，需要慎重使用root用户，更多的自由与权限同样也意味着更大的风险。

实际上，有很多Linux系统在默认情况下是不开放root用户的，这是出于安全方面的考虑。

- 普通用户

普通用户拥有的权限就没有 `root` 用户那么大了，它只能做系统允许做的事。普通用户可以执行大部分的命令，但是root专有的命令却不能执行。

```
1  suyelu@HaiZei-Tech:~$ reboot
2  reboot: Need to be root
```

后续的学习中，我们也会学习如何让普通用户拥有root的权限。

在使用过程中，可以使用 `su` 命令来切换用户。

```
1  suyelu@HaiZei-Tech:~$ whoami
2  suyelu
3  suyelu@HaiZei-Tech:~$ su - root
4  Password:
5  root@HaiZei-Tech:~# whoami
6  root
7  root@HaiZei-Tech:~# su - suyelu
8  suyelu@HaiZei-Tech:~$ whoami
9  suyelu
```

`su - root` 这里的 `-` 代表什么意思？