

Shell脚本学习

一. 小试牛刀

1.1 文件的权限以及文件权限的改变

chmod 命令详解

该命令的作用是修改用户对文件的权限，以及文件本身的权限

命令的语法：

```
chmod mode file
```

其中mode可以设定为字符串 `[ugoa][+ -=][rwx]`

其中 `[ugoa]` 代表的是：

- | | | | |
|---|--|-----------|---------------------|
| 1 | | u (owner) | 代表文件所有者，即创建文件的人 |
| 2 | | g (group) | 代表和文件所有者相同组的用户 |
| 3 | | o (other) | 表示非文件所有者和相同group的用户 |
| 4 | | a (all) | 表示所有用户 |

`[+ -=]` 代表：

- | | | | |
|---|--|---|-----------------|
| 1 | | + | 表示给指定的用户授权指定的权限 |
| 2 | | - | 表示撤销指定用户的某一个权限 |
| 3 | | = | 将指定用户的指定权限重新设置 |

`[rwx]` 代表：

- | | | | |
|---|--|---|-------|
| 1 | | r | 可读权限 |
| 2 | | w | 可写权限 |
| 3 | | x | 可执行权限 |

使用方式：

给文件所有者设置可读权限：

```
chmod u+r test.txt
```

给所有用户设置可执行权限

```
chmod a+x test.txt
```

将其他用户的可写权限撤销

```
chmod o-w test.txt
```

777表示什么：

chmod可以使用八进制数来指定权限，无需再使用指定的权限和用户的字母来进行标识，通过读写执行等3个权限的数字来进行设置

权限	rwx
7 读写执行	rwx
6 读写	rw
5 读执行	r-x
4 只读	r--
3 写执行	-wx
2 只写	-w-
1 只执行	--x
0 无权限	---

r的权限数字为4，w的权限数字为2，x的权限数字为1

777表示什么：

因为文件的权限分为3种用户，分别为：u文件所有者、g文件的组用户、o其他用户，所以777表示u、g、o都是7的权限

文件前面的权限具体怎么表示：

```
> ls -al
total 16
drwxr-xr-x 2 kirito users 4096 Oct 10 14:07 .
drwxr-xr-x 4 kirito users 4096 Oct 11 23:02 ..
-rw-r--r-- 1 kirito users 136 Oct 10 14:01 makefile
-rw-r--r-- 1 kirito users 350 Oct 10 14:04 process_0.cpp
```

文件的详细信息部分一共有7列：

第一列的第一位：文件类型；

- -普通文件
- d目录文件

```
> ls -al
total 16
drwxr-xr-x 2 kirito users 4096 Oct 10 14:07 .
drwxr-xr-x 4 kirito users 4096 Oct 11 23:02 ..
```

该文件是特殊的文件，一个是本目录文件，一个是上一级目录文件

- b块设备文件
- c字符设备
- l符号链接文件
- p管道文件
- s套接字文件

第一列：文件的具体权限；

分为3组，分别代表文件所有者、文件所有组、其他人的权限

第二列：表示该文件的连接数；

第三列：文件所有者的名字；

第四列：文件的所有组的名字；

第5列：文件的大小，大小是字节；

第6列：文件的更新时间；

第7列：文件的名字；

ps:

1)在echo中转移换行符

默认情况下，`echo` 会在输出文本的尾部追加一个换行符。可以使用选项-n来禁止这种行为。`echo` 命令如果使用包含转义序列的字符串的时候，需要在前面加上参数 `-e`。

```
echo -e "1\t2\t3"
```

2)打印彩色输出

脚本可以使用转义序列在终端中生成彩色文本

文本颜色是由相对应的彩色码来描述的。其中包括：重置=0，黑色=40，红色=41，绿色=42，黄色=43，蓝色=44，洋红=45，青色=46，白色=47

```
echo -e "\e[1;42m Green Background \e[0m"
```

`\e[42m` 将颜色设置为绿色，`\e[0m` 将颜色重新置回。

1.2 变量和环境变量

所有的编程语言都利用变量来存放数据，以备随后使用或者修改，但是脚本语言不是这样的，**脚本语言不要求在创建变量之前声明其类型**，后面的值是什么类型，这个变量就是什么类型，在变量名的前面加上一个美元符号就可以访问变量的值。

shell定义了一些变量，用来保存用到的配置信息，比如可用的打印机、搜索路径等，这些变量叫做环境变量。（就好比我们在Windows下安装一些程序之后，我们需要设置环境变量一样，我们设置过后。就可以在Windows下的命令行窗口直接去访问相关的内容）

1.3 基本操作

1.3.1 可以使用等号操作符为变量赋值

```
varName=value
```

ps:注意 `varName = value` 和 `varName=vaalue` 是不一样的，**两边没有空格的等号是赋值操作符，加上空格的等号表示的是等量关系测试。**

在变量名的前面加上美元符号（\$）就可以去访问变量的内容：

```
var="value"
```

```
echo $var
```

也可以这样写：

```
echo ${var}
```

输出结果如下：

```
value
```

因为shell使用空白字符来分隔单词，所以说当我们需要这样输出某一个字符串的时候：

`echo "we have $count ${fruit}(s)"` 如果我们不使用{}的话，shell就会将 `fruit(s)` 当作变量的名字，所以说我们需要加上一对花括号来告诉shell这里的变量名是`fruit`,而不是 `fruit(s)`。

获得变量值的长度：

`length=${#var}`

1.3.2 采集终端信息

`tput`和`stty`是两款终端处理工具