

# 学习报告

唐浩

信息与计算科学 3200102118

2022 年 6 月 28 日

## 1 shell 文件运行

- shell 脚本文件内容：

```
#!/bin/sh

salutation = "Hello"
echo $salutation
echo "The program $0 is now running"
echo "The second parameter was $2"
echo "The first parameter was $1"
echo "The parameter list was $*"
echo "The user' s home directory is $HOME"
echo "Please enter a new greeting"
read salutation
echo $salutation
echo "The script is now complete"
exit 0
```

- 尝试运行：

```
sh try.sh
```

- 运行结果如下:

```
Hello
The program ./try_var is now running
The second parameter was bar
The first parameter was foo
The parameter list was foo bar baz
The user's home directory is /home/rick
Please enter a new greeting
Sire
Sire
The script is now complete
```

- 工作原理:

操纵参数和环境变量;

该脚本创建变量`salutation`, 显示其内容;

然后显示各种参数变量和环境变量`$HOME`是如何存在并具有适当值的。

## 2 个人理解

对于 `#!/bin/sh`, 虽然shell编程是以 `verb#` 为注释, 但是它却不是。  
`#!/bin/sh` 是对shell的声明;

(指此脚本使用 `/bin/sh`(shell 脚本路径) 来解释执行, `#!` 是特殊的表示符)

如果没有声明, 则脚本将在默认的 shell 中执行, 默认 shell 是由用户所在的系统定义为执行 shell 脚本的 shell. 如果脚本被编写为在 Kornshell ksh 中运行, 而默认运行 shell 脚本的为 C shell csh, 则脚本在执行过程中很可能失败, 所以建议大家把 `"#!/bin/sh"` 当成C 语言的main函数一样。

`echo` 用于显示字符串, 直接在 `echo` 后面加上想要显示的内容就好。在 `echo` 的后面, 不仅可以加字符串, 还可以加变量名:

```
#定义变量str
str = "Hello world"
```

#在echo后加上str变量，一样可以显示出来  
echr ``\$str, good morning"

Hello world, good morning

echo 后面接不同的引号也会有不同的效果：

单引号将所有字符都看成普通字符，双引号会解释 \$ , \和' 这三种特殊字符，不加引号的话则会解释所有特殊字符。效果如下：

| 输入命令               | 输出内容  | 解释                                      |
|--------------------|---|---|
| echo 'USER*(date)' | USER*(date)                                       | 单引号无视所有特殊字符<br>所有字符在它眼里都是普通字符<br>都是芸芸众生 |
| echo "USER*(date)" | qingyang*<br>2022 年 06 月 28 日<br>星期二 19:49:01 CST | 双引号会无视文件通配符<br>但是 "\$" "\" 会起作用         |