

# 编写 shell 脚本

顾格非

3210103528

2022 年 6 月 28 日

## 1 前言

### 1.1 动机

**摘自 MIT 6.null 导言:** 作为计算机科学家, 我们都知道计算机最擅长帮助我们完成重复性的工作。但是我们却常常忘记这一点也适用于我们使用计算机的方式, 而不仅仅是利用计算机程序去帮我们求解问题。在从事与计算机相关的工作时, 我们有很多触手可及的工具可以帮助我们更高效的解决问题。但是我们中的大多数人实际上只利用了这些工具中的很少一部分, 我们常常只是死记硬背一些如咒语般的命令, 或是当我们卡住的时候, 盲目地从网上复制粘贴一些命令。

### 1.2 结缘

在准备六级考试的时候, 为了提高我的听力, 我选择了 MIT-Missing-Semester, 这门课旨在介绍一些上课不会教的工具, 比如 vim, Docker, Git ... 而 shell 编程就是其中之一。

在王何宇老师的《数学软件》, 我再一次接触了 shell。十分感谢这门课程, 下面是我的工作。

### 1.3 我的工作

之前接触了 Python 和 C 等高级语言, 这里我尝试去理解和运用 shell 语言。

1. 我学习了 shell 语法, 简单掌握了循环, 判断等结构, 下面我会在用 **shell 语法实现冒泡排序**。
2. shell 语法也有让一般编程费解的地方, **我复现了书上从函数中返回值的实验**, 并展示了结果。

## 2 正文

### 2.1 冒泡排序

冒泡排序的思想即每次排好一个数，限于篇幅我也不多介绍了吧。

### 2.2 脚本代码

```
#!/bin/bash
arr=(78 21 34 11 213 44)
echo "你的数组是：${arr[*]}"
for ((i=0;i<${#arr[*]};i++))
do
    for ((a=i+1;a<${#arr[*]};a++))
    do
        #如果前一个元素大于后一个元素，就交换位置，否则下一次循环
        if [ ${arr[$i]} -gt ${arr[$a]} ];then
            temp=${arr[$i]}
            arr[$i]=${arr[$a]}
            arr[$a]=$temp
        fi
    done
done
echo "你的数组升序排列为：${arr[*]}"
```

### 2.3 结果展示

将脚本命名为1.sh，采用bash 1.sh编译，结果如下：

你的数组是：78 21 34 11 213 44

你的数组升序排列为：11 21 34 44 78 213

## 3 正文 2

### 3.1 简单介绍

这是书上从函数中返回一个值的实验。在 shell 头文件之后，我们定义了一个函数`yes_or_no()`，在脚本开始执行时，函数被定义而不立即执行，而后面传入参数给这个函数，最终得到想要的结果。

### 3.2 脚本代码

这里换了一种好看的代码块！

```
1 #!/bin/bash
2 yes_or_no() {
3     echo "Is your name $* ?"
4     while true
5     do
6         echo -n "Enter yes or no: "
7         read x
8         case "$x" in
9             y | yes ) return 0;;
10            n | no ) return 1;;
11            * )      echo "Answer yes or no"
12        esac
13    done
14 }
15
16 echo "Original parameters are $*"
17 if yes_or_no "$1"
18 then
19     echo "Hi $1, nice name"
20 else
21     echo "Never mind"
22 fi
23 exit 0
```

### 3.3 结果展示

这里我在终端中输入`bash 2.sh frank`, (其中 `frank` 是像函数传入的参数作为名字)

```
Original parameters are frank
```

```
Is your name frank ?
```

```
Enter yes or no: yes
```

```
Hi frank, nice name
```