

浙江大学实验报告

专业: 计算机科学与技术
姓名: 吴同
学号: 3170104848
日期: 2019年8月15日

课程名称: Linux 程序设计 指导老师: 季江民 电子邮件: wutongcs@zju.edu.cn
实验名称: shell 程序设计 实验类型: 设计型 联系电话: 18888922355

一、实验环境

1. 硬件环境

实验所用的计算机为 MacBook Pro (13-inch, 2017, Four Thunderbolt 3 Ports), CPU 为 Intel Core i5 双核 3.1 GHz, 内存为 8 GB 2133 MHz LPDDR3。

2. 操作系统

实验所用的操作系统为 macOS Mojave 10.14.5 Beta。macOS 基于 Darwin BSD 的 Unix 内核, 符合单一 Unix 规范, 在 macOS 的终端下可以正常完成本实验的全部内容。

二、实验内容、结果及分析

1. 统计文件数目

需求描述: 统计指定目录下的普通文件、子目录及可执行文件的数目, 统计该目录下所有普通文件字节数总和, 目录的路径名由参数传入。

源代码:

```
1 #!/bin/bash
2 #UTF-8
3 #*****
4 #名称:      lab2-1.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      1. 统计指定目录下的普通文件、子目录、可执行文件的数目。
8 #          2. 统计该目录下所有普通文件字节数总和。
9 #参数:      目录的路径 (可选)
10 #####
11
12 #检查运行程序时传入的参数个数是否符合要求, 程序最多传入一个参数, 即目录的路径。
13 #错误信息显示为红色。
14 if test $# -gt 1; then
15     echo -e "\033[31m参数个数过多! \033[0m"
16     exit 1
17 fi
18
```

```

19 #将传入的参数设为待查询目录，若参数为空，后续运行结果不受影响。
20 directory=$1
21
22 #输出待查询的目录，目录显示为深绿色。
23 #若参数不为空，则参数为待查询目录；若参数为空，则查询当前目录。
24 echo -en "查询目录: \033[36m"
25 if test $# -lt 1; then
26     pwd
27 else
28     echo $directory
29 fi
30
31 #统计该目录下不同种类文件的个数，包括隐藏文件。
32 #数字显示为紫色。
33 #利用长格式ls的输出结果进行判断，第一个字符为文件类型，'-'为普通文件，'d'为目录。
34 echo -en "\033[0m普通文件个数: \033[35m"
35 ls -Al $directory | grep -c "^-"
36 echo -en "\033[0m目录文件个数: \033[35m"
37 ls -Al $directory | grep -c "^d"
38 #若文件权限中有x或s，则文件为可执行。
39 echo -en "\033[0m可执行文件个数: \033[35m"
40 ls -Al $directory | grep -c "^-[rw-]*[sx]"
41
42 #统计该目录下所有普通文件的总字节数。
43 echo -en "\033[0m普通文件总字节数: \033[35m"
44 declare -i sum=0
45 #对该目录下的ls -Al输出结果进行两次正则匹配，
46 #第一次匹配取普通文件的前五个字段，第二次匹配取第五个字段。
47 for line in $(ls -Al $directory | grep -o
48     "[^-\\([[:blank:]]*[[:blank:]]*\)\\{4\\}[^[:blank:]]*" | grep -o
49     "[[:blank:]]*\$")
50 do
51     let sum+=line
52 done
53 echo -e "$sum\033[0m"
54
55 #退出程序
56 exit 0

```

实验结果:



```
[→ lab2-1 bash lab2-1.sh /
查询目录: /
普通文件个数:      6
目录文件个数:      27
可执行文件个数:    0
普通文件总字节数: 11148
```

图 1: lab2-1 测试结果 1 (查询根目录)

```
[→ lab2-1 bash lab2-1.sh /etc
查询目录: /etc
普通文件个数: 0
目录文件个数: 0
可执行文件个数: 0
普通文件总字节数: 0]
```

图 2: lab2-1 测试结果 2 (无查询结果)

```
[→ lab2-1 bash lab2-1.sh .
查询目录: .
普通文件个数: 1
目录文件个数: 0
可执行文件个数: 1
普通文件总字节数: 2082]
```

图 3: lab2-1 测试结果 3 (查询相对路径)

```
[→ lab2-1 bash lab2-1.sh
查询目录: /Users/wutong/OneDrive/Course/Linux/lab/lab2/src/lab2-1
普通文件个数: 1
目录文件个数: 0
可执行文件个数: 1
普通文件总字节数: 2082]
```

图 4: lab2-1 测试结果 4 (无参数传入)

2. 数据处理

需求描述: 从键盘输入 100 个整数, 计算其平均值、最大值和最小值并输出, 从小到大排序并输出。
源代码:

```
1 #!/bin/bash
2 #UTF-8
3 #*****
4 #名称:      lab2-2.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      1. 统计100个整数的平均值、最大值、最小值。
8 #          2. 将100个整数按从小到大的顺序输出。
9 #参数:      无
10 #*****
11
12 #检查运行程序时传入的参数个数是否符合要求, 该程序不传入参数。
13 #错误信息显示为红色。
14 if test $# -gt 0; then
15     echo -e "\033[31m格式错误! \033[0m"
16     exit 1
17 fi
18
19 #从键盘输入100个整数
20 #将100用变量表示, 方便数据规模的修改
21 hundred=100
```

```
22 echo "输入$hundred个整数: "
23 declare -i sum=0
24 declare -i i=0
25 #支持以多行输入, 读满100个整数为止
26 #外层循环每次读入一行
27 while test $i -lt $hundred
28 do
29     #将读入的文本行划分为数组
30     read -a line
31     #内层循环每次处理行中的一个数
32     for integer in ${line[@]}
33     do
34         if test $i -lt $hundred; then
35             array[i++]=$integer
36             sum+=$integer
37         fi
38     done
39 done
40
41 #输出结果, 排序结果显示为黄色
42 #先讲数组以列输出到管道, 再用sort命令排序
43 sorted=$(echo ${array[@]} | tr ' ' '\n' | sort -g)
44 echo -e "排序后的数组为: \033[33m"
45 echo $sorted
46
47 #平均值、最小值、最大值显示为紫色
48 echo -en "\033[0m平均值为: \033[35m"
49 #平均值直接计算
50 echo `expr $sum / $hundred` 
51 echo -en "\033[0m最小值为: \033[35m"
52 #最小值和最大值直接提取排序结果的第一个和最后一个
53 echo $sorted | grep -o "^[[:digit:]]*[[:blank:]]"
54 echo -en "\033[0m最大值为: \033[35m"
55 echo $sorted | grep -o "[[:blank:]][[:digit:]]*$"
56 echo -en "\033[0m"
57
58 #退出程序
59 exit 0
```

实验结果: 实验过程中, 通过重定向标准输入实现从文件读入输入数据。为了生成多组测试数据, 我用 Python 语言编写了随机生成测试数据的程序, 以进行多组检验。

```
[→ lab2-2 python3 lab2-2.py && bash lab2-2.sh < lab2-2.in
输入 100个整数:
排序后的数组为:
145 211 262 330 355 489 490 500 560 571 723 895 939 1025 1027 1109 1525 1573 1616 1806 1895 1971 203
8 2231 2255 2293 2335 2483 2497 2735 2805 2833 3188 3320 3324 3347 3409 3523 3733 3774 4025 4078 420
5 4369 4447 4528 4667 4707 4723 4777 4785 4806 4813 5012 5159 5174 5275 5357 5455 5559 5685 581
4 5939 6169 6379 6395 6418 6468 6522 6735 6832 6942 7008 7016 7095 7316 7435 7495 7515 7614 7759 780
7 7881 8039 8230 8274 8282 8377 8543 8648 8676 8786 9195 9340 9375 9508 9522 9699 9816
平均值为: 4738
最小值为: 145
最大值为: 9816
```

图 5: lab2-2 测试结果 1 (普通)

```
[→ lab2-2 bash lab2-2.sh < lab2-2.in
输入 100个整数:
排序后的数组为:
145 145 211 262 330 355 489 490 500 560 571 723 895 939 1025 1027 1109 1525 1573 1616 1806 1895 1971
2038 2231 2255 2293 2335 2483 2497 2735 2805 2833 3188 3320 3324 3347 3409 3523 3733 3774 4025 4078
4205 4369 4447 4528 4667 4707 4723 4777 4785 4806 4813 5012 5159 5174 5275 5357 5455 5559 5685 5814
5939 6169 6379 6395 6418 6468 6522 6735 6832 6942 7008 7016 7095 7316 7435 7495 7515 7614 7759 7807
7881 8039 8230 8274 8282 8377 8543 8648 8676 8786 9195 9340 9375 9508 9522 9699 9816
平均值为: 4687
最小值为: 145
最大值为: 9816
```

图 6: lab2-2 测试结果 2 (最小值有重复)

测试程序源代码:

```
1 # **** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
2 #名称:          lab2-2.py
3 #作者:          吴同
4 #学号:          3170104848
5 #功能:          随机生成100个10000以内的整数, 并随机换行, 以测试lab2-2.sh的正确性。
6 #           生成数据写入lab2-2.in文件中。
7 #参数:          无
8 # **** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
9
10 import random
11 #随机生成10000个整数
12 hundred=100
13 array = []
14 for i in range(hundred):
15     array.append(random.randint(0, 10000))
16 #生成结果写入lab2-2.in文件中
17 with open('lab2-2.in', 'w') as f:
18     for element in array:
19         f.write(str(element))
20         #写入一个数据后, 随机选择空格或换行
21         if random.randint(0, 1) is 1:
22             f.write(' ')
23         else:
24             f.write('\n')
```

3. 回文串

需求描述: 输入一个字符串, 忽略非字母后, 检测该字符串是否为回文串。

源代码:

```
1 #!/bin/bash
2 #UTF-8
3 #*****
4 #名称:      lab2-3.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      输入一个字符串, 判断是否是回文串。
8 #参数:      无
9 #####
10
11 #检查运行程序时传入的参数个数是否符合要求, 该程序不传入参数。
12 #错误信息显示为红色。
13 if test $# -gt 0; then
14     echo -e "\033[31m格式错误! \033[0m"
15     exit 1
16 fi
17
18 #读入字符串
19 echo "请输入字符串: "
20 read line
21
22 #删除字符串中的非字母并将其显示出来, 处理过的字符串显示为紫色
23 word=$(echo $line | tr -d -c [:alpha:])
24 echo -en "\033[35m$word\033[0m"
25
26 #判断处理后的字符串是否是回文串
27 #从字符串的两端向中间依次判断
28 for ((i=1; i<=${#word}/2; i++))
29 do
30     #比较第i个和第len(string)-(i-1)个字符是否相同
31     if test $(echo $word | cut -c $i) != $(echo $word | cut -c
32         ${#word}-$i); then
33         echo "不是回文串"
34         exit 0
35     fi
36 done
37 echo "是回文串"
38
39 #退出程序
40 exit 0
```

实验结果:

```
[→ lab2-3 bash lab2-3.sh
请输入字符串:
eshibeebihse
eshibeebihse是回文串]
```

图 7: lab2-3 测试结果 1 (回文串, 偶数个字符)

```
[→ lab2-3 bash lab2-3.sh
请输入字符串:
aggweaewgga
aggweaewgga是回文串]
```

图 8: lab2-3 测试结果 2 (回文串, 奇数个字符)

```
[→ lab2-3 bash lab2-3.sh
请输入字符串:
vasvbefa
vasvbefa不是回文串]
```

图 9: lab2-3 测试结果 3 (非回文串)

```
[→ lab2-3 bash lab2-3.sh
请输入字符串:
34_32g
g是回文串]
```

图 10: lab2-3 测试结果 4 (回文串, 有非字母)

```
[→ lab2-3 bash lab2-3.sh
请输入字符串:
*(@#ng2&* Cf/
ngf不是回文串]
```

图 11: lab2-3 测试结果 5 (非回文串, 有非字母)

4. 作业管理系统

需求描述:

- 数据需求

本作业管理系统的数据分为两大模块，分别是用户和课程。用户数据包括用户类型（学生、教师、管理员），用户账号（工号或学号），用户密码。课程数据包括课程号，课程信息，课程任务。两大模块之间的联系是教师的授课信息和学生的选课信息。

- 功能需求

本系统的所有操作均有菜单提示，并通过输入选项来选择相应的操作。启动系统时，用户选择登录模式（学生、教师、管理员）进行登录验证。

管理员模式下，用户可对教师信息和课程信息进行管理。管理教师信息的功能包括教师信息（工号和姓名）的增删改查，以及将教师与课程绑定或解绑。管理课程信息的功能包括课程信息的增删改查。此外，管理员可对所有用户的密码进行修改。

教师模式下，用户可选择自己所教的课程进行管理，包括对选课学生、课程信息、课程任务的管理。管理选课学生的功能包括选课学生的增删改查，如果学生尚未存在于系统，则添加进系统。教师可对系统中的所有学生信息进行修改，无论其是否选课。管理课程信息的功能包括课程信息的增删改查。管理课程任务的功能包括发布任务、删除任务、修改任务、查询任务和查看所有学生的完成情况。此外，教师可修改本人的密码。

学生模式下，用户可选择自己所修读的课程进行管理，包括查看课程信息和任务管理。作业管理的功能包括查看任务，上传作业，下载已上传作业，重交作业。此外，学生可修改本人的密码。

设计文档:

- 设计思想

本作业管理系统全部采用 shell 程序设计。shell 的特点是有较强的文本处理能力，而数值运算的能力较弱。并且，shell 相对于其他脚本语言更加接近操作系统内核，可直接使用管道和重定向功能。此外，shell 同其他脚本语言一样，逐行解释执行。基于 shell 的这些特点，本系统在设计的过程中遵循以下思想：

基于文件系统。系统中的所有数据全部以文本文件的形式存于磁盘中，以利于 shell 脚本进行处理。用户和课程的基本信息统一存于目录文件中，而教师绑定课程和学生选课这样的联系型信息，则每一对联系都对应一个普通文件或目录文件。这样做，既方便存在性的判断，又方便数据的存储，同时避免了数据的冗余。

自顶向下的架构。系统在运行的过程中，逐级选择操作的类型，函数调用关系清晰，呈树状结构。每个函数一般只有一个入口，和一个出口。在函数执行完毕后，直接调用菜单函数，避免了由于层层调用导致返回值混乱的问题。

实时交互。程序的主要运算对象是文本，通过调用命令并结合管道功能对文本进行处理，处理后的结果及时进行输出。这与其他语言的程序设计中提倡交互与核心计算分开的设计思想不同。因为在 shell 自身的主要功能就是用于人机交互的，并且由于重定向功能，与用户交互与处理文件并没有什么不同。如果强行按照其他语言那样将交互与计算分开，这显然不符合 shell 的特质。

- 文件结构

系统所用的所有数据文件存放于 data 目录下，如下图所示。passwd 文件用于存放用户密码。catalog/下有三个文件，分别存放教师、学生、课程的基本信息。teacher_course/下存放教师绑定课程的信息，每对课程绑定对应于一个文本文件，文件内存放课程信息。student_course_teacher/下存放学生的选课信息，每个选课记录对应于一个目录，每个目录下有若干子目录，文件名是该课程的任务，目录下的文件是学生提交的作业。

```
[→ lab2-4 tree data
data
├── catalog
│   ├── course
│   ├── student
│   └── teacher
└── passwd
    ├── student_course_teacher
    │   ├── 3170104848_0113N002_0095129
    │   │   └── HW1
    │   │       └── admin.sh
    │   ├── 3170105166_0113N002_0095129
    │   │   └── HW1
    │   └── 3170105762_0113N002_0095129
    │       └── HW1
    └── teacher_course
        ├── 0095129_01102081
        ├── 0095129_0113N002
        └── 0095129_01192101
```

图 12: 数据文件的目录树

- 功能模块

程序根据用户登录模式的不同，分为三大功能模块（学生、教师、管理员），并有一顶层模块用于调用这三大功能模块中的一个。

系统运行时，首先执行 main.sh 脚本，选择登录模式后，程序通过 source 命令(.) 将 module.sh 文件包含进来。模块的实现文件中只有该模块所用的若干函数。全部包含后，调用该模块的 main 函数。

管理员模块中，程序实现了管理员登录，教师用户的增删改查，课程的增删改查，教师与课程的绑定和解绑，以及修改任意用户的密码。

教师模块中，程序实现了教师登录，选课学生的增删改查，课程信息的增删改查，课程任务的增删改查，以及修改本人密码。

学生模块中，程序实现了学生登录，查看课程信息，上传和下载作业，以及修改本人密码。

- 实现方法

用户登录功能的实现方法是，用户输入密码后，对密码进行加密运算，再与密码文件中的密文进行字符串比对。加密规则是，管理员用户先进行一次 md5 加密，再在尾部加上“admin”字符，再进行一次 md5，再在尾部加上“nimda”，再进行一次 sha1 加密。教师用户进行一次 sha1 加密，学生用户进行一次 md5 加密。修改密码时，先运行登录函数，进行身份确认，再输入新密码，按对应加密规则加密后，删除密码文件中的旧行，加入新行。

教师用户和学生用户的增删改查，通过查询和改变 data/catalog/teacher 和 data/catalog/student 中的数据来实现用户的查询、添加、删除和修改用户名。在添加或删除用户的过程中，要同时添加或删除 data/passwd 中相应的行。课程的增删改查，通过查询和修改 data/catalog/course 中相应的行来实现。所有查询均支持正则表达式。教师是否已绑定课程和学生是否已选课的记录，通过检查 data/teacher_course 和 data/student_course_teacher 下是否有相应的文件存在来实现。

课程信息的增删改查，通过操作 data/teacher_course 相应文件的内容来实现。在文件内部，每一条信息之间通过“|”符号来分割。在发布信息的过程中，允许输入转义字符。

课程任务的布置和删除，通过操作 data/student_course_teacher 下的目录文件来实现。学生上传和下载的作业，全部储存在相应的目录中。

- 可改进点

增加日志功能。对数据所做的每一项修改都在日志中有所记录，一旦程序遇到故障，则可进行数据的恢复。另外，通过记录每个用户所做的操作，在出现非法操作时，管理员可以进行核查。

在数据文件的起始行增加标志位，防止多个程序同时对一个文件进行读写，造成数据的错乱。

在学生提交作业时对输入的路径进行检查，以防学生输入非法路径，将其无权限读取的文件拷贝到个人的目录下。在学生下载已交作业时也对输入的路径进行检查，以防学生将文件写入其无权限写入的目录下。

源代码:

```
1 #!/bin/bash
2 #UTF-8
3 #*****#
4 #名称:      main.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      选择登录模式，调用相关模块。
8 #*****#
9
10 #显示登录选项菜单
11 #序号显示为紫色
12 echo -e "您的身份: "
13 echo -e "\033[35m1 \033[0m学生"
14 echo -e "\033[35m2 \033[0m教师"
15 echo -e "\033[35m3 \033[0m管理员"
16 echo -e "\033[35mq \033[0m退出系统"
17
18 #输入登录模式，将该模块的文件包含进来，并调用模块的main函数
19 read mode
20 case $mode in
21     1) . ./student.sh
22         main_student;;
23     2) . ./teacher.sh
24         main_teacher;;
25     3) . ./admin.sh
26         main_admin;;
27     #正常退出程序
28     q) exit 0;;
29     *) echo -e "\033[31m输入非法! \033[0m"
30         #登录界面输入非法退出程序
31         exit 2;;
32 esac
33
34 #未知异常退出程序
35 exit 1
```

```
1 #!/bin/bash
2 #UTF-8
```

```
3 #*****  
4 #名称:      admin.sh  
5 #作者:      吴同  
6 #学号:      3170104848  
7 #功能:      管理员模块的函数实现  
8 #*****  
9  
10 #管理员主函数  
11 main_admin()  
12 {  
13     #管理员登录  
14     login_admin  
15     echo -e "\033[32m欢迎admin! \033[0m"  
16     #选择管理员操作  
17     mode_admin  
18     #未知异常退出程序  
19     exit 1  
20 }  
21  
22 #管理员登录认证  
23 login_admin()  
24 {  
25     #输入不回显  
26     stty -echo  
27     #加密运算所用的变量  
28     salt="admin"  
29     salt_reverse="nimda"  
30     #在密码文件中找到管理员密码  
31     #管理员密码所在行第一个字段为 "a"  
32     passwd=$(cat data/passwd | grep ^a | awk '{print $3}')  
33     #超过三次输入错误密码, 直接退出程序  
34     for ((i=0; i<3; i++))  
35     do  
36         #输入密码  
37         echo -n "请输入管理员登录密码: "  
38         read passwd_try  
39         echo ""  
40         #加密运算  
41         #先将明文进行md5加密, 再在尾部加上"admin"字符, 再进行一次md5,  
42         #再在尾部加上"nimda", 再进行一次sha1加密  
43         passwd_try_md5=$(echo $($passwd_try | md5)$salt | md5)  
44         passwd_try_sha=$(echo $($passwd_try_md5)$salt_reverse |  
45             shasum | grep -o "^[[:blank:]]*")  
46         #字符串比对  
47         if test $passwd = $passwd_try_sha; then  
48             #撤销输入不回显  
49             stty echo  
50             return 0  
51         else  
52             echo -e "\033[31m密码错误! \033[0m"
```

```
52         fi
53     done
54     #撤销输入不回显
55     stty echo
56     #超过三次输入密码错误退出程序
57     exit 3
58 }
59
60 #以下是与管理功能有关的函数
61
62 #管理员主菜单
63 mode_admin()
64 {
65     #序号显示为紫色
66     echo -e "\n请选择操作: "
67     echo -e "\033[35m1 \033[0m管理教师信息"
68     echo -e "\033[35m2 \033[0m管理课程信息"
69     echo -e "\033[35m3 \033[0m修改用户密码"
70     echo -e "\033[35mq \033[0m退出系统"
71     #输入所选择模式的序号
72     read mode
73     case $mode in
74         1) manage_teacher;;
75         2) manage_course;;
76         3) change_password;;
77         #正常退出程序
78         q) exit 0;;
79         *) echo -e "\033[31m输入非法! \033[0m"
80             mode_admin;;
81     esac
82     #未知异常退出程序
83     exit 1
84 }
85
86 #教师信息管理主菜单
87 manage_teacher()
88 {
89     echo -e "\n请选择操作: "
90     echo -e "\033[35m1 \033[0m查询教师信息"
91     echo -e "\033[35m2 \033[0m增加教师信息"
92     echo -e "\033[35m3 \033[0m删除教师信息"
93     echo -e "\033[35m4 \033[0m修改教师信息"
94     echo -e "\033[35m5 \033[0m绑定课程信息"
95     echo -e "\033[35m6 \033[0m取消课程绑定"
96     echo -e "\033[35mr \033[0m返回上级菜单"
97     echo -e "\033[35mq \033[0m退出系统"
98     #输入所选择模式的序号
99     read mode
100    case $mode in
101        1) query_teacher;;
```

```
102     2) add_teacher;;
103     3) delete_teacher;;
104     4) change_teacher;;
105     5) bind_course;;
106     6) cancel_bind_course;;
107     r) mode_admin;;
108     #正常退出程序
109     q) exit 0;;
110     *) echo -e "\033[31m输入非法! \033[0m"
111         manage_teacher;;
112     esac
113     #未知异常退出程序
114     exit 1
115 }
116
117 #课程信息管理主菜单
118 manage_course()
119 {
120     echo -e "\n请选择操作: "
121     echo -e "\033[35m1 \033[0m查询课程信息"
122     echo -e "\033[35m2 \033[0m增加课程信息"
123     echo -e "\033[35m3 \033[0m删除课程信息"
124     echo -e "\033[35m4 \033[0m修改课程信息"
125     echo -e "\033[35mr \033[0m返回上级菜单"
126     echo -e "\033[35mq \033[0m退出系统"
127     #输入所选择模式的序号
128     read mode
129     case $mode in
130         1) query_course;;
131         2) add_course;;
132         3) delete_course;;
133         4) change_course;;
134         r) mode_admin;;
135         #正常退出程序
136         q) exit 0;;
137         *) echo -e "\033[31m输入非法! \033[0m"
138             manage_course;;
139     esac
140     #未知异常退出程序
141     exit 1
142 }
143
144 #以下是与教师信息管理有关的函数
145
146 #增加教师用户
147 add_teacher()
148 {
149     #输入教师工号
150     echo -e "\n请输入教师工号 (输入cancel可取消): "
151     read t_id
```

```
152 #判断输入的字符串是否为cancel
153 if test "cancel" = $t_id; then
154     manage_teacher
155     #未知异常退出程序
156     exit 1
157 fi
158 #判断该工号是否已存在
159 if test 0 != $(cat data/catalog/teacher | awk '{print $1}' | grep -c
160     "$t_id"); then
161     echo -e "\033[31m该工号已存在! \033[0m"
162     manage_teacher
163     #未知异常退出程序
164     exit 1
165 fi
166 #若该工号不存在，则添加教师
167 #输入教师姓名
168 echo -e "\n请输入教师姓名（输入cancel可取消）："
169 read t_name
170 #判断输入的字符串是否为cancel
171 if test "cancel" = $t_name; then
172     manage_teacher
173     #未知异常退出程序
174     exit 1
175 fi
176 #将新添加的教师信息写入教师信息文件中
177 echo $t_id $t_name >> data/catalog/teacher
178 #将教师账号添加进密码文件中，初始密码为123456
179 password=123456
180 #将明文进行sha1加密后写入文件
181 password_sha=$(echo $password | shasum | grep -o "^[^[:blank:]]*")
182 echo "t" $t_id $password_sha >> data/passwd
183 #提示操作成功，显示为绿色
184 echo -e "\n\033[32m增加教师成功! \033[0m"
185 manage_teacher
186 #未知异常退出程序
187 exit 1
188 }
189 #删除教师用户
190 delete_teacher()
191 {
192     #输入教师工号
193     echo -e "\n请输入教师工号（输入cancel可取消）："
194     read t_id
195     #判断输入的字符串是否为cancel
196     if test "cancel" = $t_id; then
197         manage_teacher
198         #未知异常退出程序
199         exit 1
200 fi
```

```
201 #检查该用户是否存在
202 if test 0 != $(cat data/catalog/teacher | awk '{print $1}' | grep -c
203     "$t_id"); then
204     #显示该教师的工号和姓名, 显示为黄色
205     echo -e "\n所要删除的教师为: \033[33m"
206     cat data/catalog/teacher | grep "^$t_id"
207     echo -en "\033[31m确认删除吗? \033[0m"
208     #确认删除
209     read verify
210     case $verify in
211         y | Y | yes | Yes | YES)
212             #将教师信息文件和密码文件中对应的行删掉
213             sed -i "" '/'$t_id'/d' data/catalog/teacher
214             sed -i "" '/'$t_id'/d' data/passwd
215             echo -e "\n\033[32m删除成功! \033[0m";;
216         *)
217             echo "取消删除";;
218     esac
219     manage_teacher
220     #未知异常退出程序
221     exit 1
222 else
223     #错误信息显示为红色
224     echo -e "\033[31m该工号不存在! \033[0m"
225     manage_teacher
226     #未知异常退出程序
227     exit 1
228 fi
229 }
230 #修改教师信息
231 change_teacher()
232 {
233     #输入教师工号
234     echo -e "\n请输入教师工号 (输入cancel可取消) : "
235     read t_id
236     #判断输入的字符串是否为cancel
237     if test "cancel" = $t_id; then
238         manage_teacher
239         #未知异常退出程序
240         exit 1
241     fi
242     #检查该用户是否存在
243     if test 0 != $(cat data/catalog/teacher | awk '{print $1}' | grep -c
244         "$t_id"); then
245         #显示该教师的工号和姓名, 显示为黄色
246         echo -e "\n所要修改的教师为: \033[33m"
247         cat data/catalog/teacher | grep "^$t_id"
248         #输入修改的信息
249         echo -e "\n\033[0m请输入修改后的教师姓名 (输入cancel可取消) : "
```

```
249     read t_name
250     if test "cancel" = $t_name; then
251         manage_teacher
252         #未知异常退出程序
253         exit 1
254     fi
255     #删除文件中的旧行
256     sed -i "" '/'$t_id'/d' data/catalog/teacher
257     #向文件中写入新行
258     echo $t_id $t_name >> data/catalog/teacher
259     #修改成功提示信息显示为绿色
260     echo -e "\n\033[32m修改成功! \033[0m"
261     manage_teacher
262     #未知异常退出程序
263     exit 1
264 else
265     echo -e "\033[31m该工号不存在! \033[0m"
266     manage_teacher
267     #未知异常退出程序
268     exit 1
269 fi
270 }
271 #查询教师信息，选择查询模式
272 query_teacher()
273 {
274     echo -e "\n请选择查询方式："
275     echo -e "\033[35m1 \033[0m按教师工号查询"
276     echo -e "\033[35m2 \033[0m按教师姓名查询"
277     echo -e "\033[35m3 \033[0m显示所有教师"
278     echo -e "\033[35mr \033[0m返回上级菜单"
279     echo -e "\033[35mq \033[0m退出系统"
280     #输入所选择模式的序号
281     read mode
282     case $mode in
283         1) query_teacher_id;;
284         2) query_teacher_name;;
285         3) query_teacher_all;;
286         r) manage_teacher;;
287         #正常退出程序
288         q) exit 0;;
289         *) echo -e "\033[31m输入非法! \033[0m"
290             query_teacher;;
291     esac
292     #未知异常退出程序
293     exit 1
294 }
295 #以下是与教师信息查询有关的函数
296
297
298
```

```
299 #按工号查询教师信息
300 query_teacher_id()
301 {
302     #输入教师工号
303     echo -e "\n请输入教师工号: "
304     read t_id
305     if test 0 != $(cat data/catalog/teacher | grep -c "^$t_id"); then
306         #显示查询到的教师信息, 信息显示为黄色
307         echo -e "\n所查询的教师为: \033[33m"
308         cat data/catalog/teacher | grep "^$t_id"
309         echo -en "\033[0m"
310         query_teacher
311         #未知异常退出程序
312         exit 1
313     else
314         echo -e "\033[31m教师不存在! \033[0m"
315         query_teacher
316         #未知异常退出程序
317         exit 1
318     fi
319 }
320
321 #按姓名查询教师信息
322 query_teacher_name()
323 {
324     #输入教师工号
325     echo -e "\n请输入教师姓名: "
326     read t_name
327     if test 0 != $(cat data/catalog/teacher | grep -c "$t_name"); then
328         #显示查询到的教师信息, 信息显示为黄色
329         echo -e "\n所查询的教师为: \033[33m"
330         #如果出现同名教师, 输出结果按工号排序
331         cat data/catalog/teacher | grep "$t_name" | sort
332         echo -en "\033[0m"
333         query_teacher
334         #未知异常退出程序
335         exit 1
336     else
337         echo -e "\033[31m教师不存在! \033[0m"
338         query_teacher
339         #未知异常退出程序
340         exit 1
341     fi
342 }
343
344 #输出所有教师信息
345 query_teacher_all()
346 {
347     #检查教师文件中是否有记录存在
348     if test 0 != $(cat data/catalog/teacher | wc -l); then
```

```
349     echo -e "\n所查询的教师为: \033[33m"
350     #输出结果按工号排序, 显示为黄色
351     cat data/catalog/teacher | sort
352     echo -en "\033[0m"
353     query_teacher
354     #未知异常退出程序
355     exit 1
356 else
357     echo -e "\033[31m教师不存在! \033[0m"
358     query_teacher
359     #未知异常退出程序
360     exit 1
361 fi
362 }
363
364 #以下是与教师课程绑定有关的函数
365
366 #绑定课程
367 bind_course()
368 {
369     echo -e "\n请输入教师工号 (输入cancel可取消) : "
370     read t_id
371     #判断输入的字符串是否为cancel
372     if test "cancel" = $t_id; then
373         manage_teacher
374         #未知异常退出程序
375         exit 1
376     fi
377     #检查教师账户是否存在
378     if test 0 = $(cat data/catalog/teacher | grep -c "^$t_id"); then
379         echo -e "\033[31m教师不存在! \033[0m"
380         manage_teacher
381         #未知异常退出程序
382         exit 1
383     fi
384     echo -e "\n请输入课程代号 (输入cancel可取消) : "
385     read c_id
386     #判断输入的字符串是否为cancel
387     if test "cancel" = $c_id; then
388         manage_teacher
389         #未知异常退出程序
390         exit 1
391     fi
392     #检查课程信息是否存在
393     if test 0 = $(cat data/catalog/course | grep -c "^$c_id"); then
394         echo -e "\033[31m课程不存在! \033[0m"
395         manage_teacher
396         #未知异常退出程序
397         exit 1
398     fi
```

```
399     #检查教师与课程是否已经绑定
400     if test -f "data/teacher_course/$t_id""_$c_id"; then
401         echo -e "\033[31m该教师已绑定该课程! \033[0m"
402         manage_teacher
403         #未知异常退出程序
404         exit 1
405     fi
406     #所有检查均通过, 绑定教师与课程, 创建"教师工号_课程号"文件
407     touch "data/teacher_course/$t_id""_$c_id"
408     echo -e "\n\033[32m教师与课程成功绑定! \033[0m"
409     manage_teacher
410     #未知异常退出程序
411     exit 1
412 }
413
414 #取消绑定
415 cancel_bind_course()
416 {
417     #判断输入的字符串是否为cancel
418     echo -e "\n请输入教师工号 (输入cancel可取消) : "
419     read t_id
420     if test "cancel" = $t_id; then
421         manage_teacher
422         #未知异常退出程序
423         exit 1
424     fi
425     #检查教师账户是否存在
426     if test 0 = $(cat data/catalog/teacher | grep -c "^$t_id"); then
427         echo -e "\033[31m教师不存在! \033[0m"
428         manage_teacher
429         #未知异常退出程序
430         exit 1
431     fi
432     echo -e "\n请输入课程代号 (输入cancel可取消) : "
433     read c_id
434     #判断输入的字符串是否为cancel
435     if test "cancel" = $c_id; then
436         manage_teacher
437         #未知异常退出程序
438         exit 1
439     fi
440     #检查课程信息是否存在
441     if test 0 = $(cat data/catalog/course | grep -c "^$c_id"); then
442         echo -e "\033[31m课程不存在! \033[0m"
443         manage_teacher
444         #未知异常退出程序
445         exit 1
446     fi
447     #检查教师与课程是否已经绑定
448     #若已经绑定, 删除"教师工号_课程号"文件
```

```
449     if test -f "data/teacher_course/${t_id}"_"$c_id"; then
450         rm "data/teacher_course/${t_id}"_"$c_id"
451         echo -e "\n\033[32m教师与课程解除绑定! \033[0m"
452         manage_teacher
453         #未知异常退出程序
454         exit 1
455     else
456         echo -e "\033[31m该教师未与该课程绑定! \033[0m"
457         manage_teacher
458         #未知异常退出程序
459         exit 1
460     fi
461 }
462
463 #以下是与课程管理有关的函数
464
465 #增加新的课程信息
466 add_course()
467 {
468     echo -e "\n请输入课程代号 (输入cancel可取消) : "
469     read c_id
470     #判断输入的字符串是否为cancel
471     if test "cancel" = $c_id; then
472         manage_course
473         #未知异常退出程序
474         exit 1
475     fi
476     #检查课程信息是否存在
477     if test 0 != $(cat data/catalog/course | awk '{print $1}' | grep -c
478         "$c_id"); then
479         echo -e "\033[31m课程代号已存在! \033[0m"
480         manage_course
481         #未知异常退出程序
482         exit 1
483     fi
484     echo -e "\n请输入课程名称 (输入cancel可取消) : "
485     read c_name
486     #判断输入的字符串是否为cancel
487     if test "cancel" = $c_name; then
488         manage_course
489         #未知异常退出程序
490         exit 1
491     fi
492     #将增加的课程写入课程信息文件
493     echo $c_id $c_name >> data/catalog/course
494     echo -e "\n\033[32m增加课程成功! \033[0m"
495     manage_course
496     #未知异常退出程序
497     exit 1
498 }
```

```
498
499 #删除课程信息
500 delete_course()
501 {
502     echo -e "\n请输入课程代号 (输入cancel可取消) : "
503     read c_id
504     #判断输入的字符串是否为cancel
505     if test "cancel" = $c_id; then
506         manage_course
507         #未知异常退出程序
508         exit 1
509     fi
510     #检查课程信息是否存在
511     if test 0 != $(cat data/catalog/course | awk '{print $1}' | grep -c
512         "$c_id"); then
513         echo -e "\n所要删除的课程为: \033[33m"
514         #显示课程信息, 显示为黄色
515         cat data/catalog/course | grep "^$c_id"
516         #确认是否删除
517         echo -en "\033[31m确认删除吗? \033[0m"
518         read verify
519         case $verify in
520             y | Y | yes | Yes | YES)
521                 #删除课程信息文件中对应的行
522                 sed -i "" '/'$c_id'/d' data/catalog/course
523                 echo -e "\n\033[32m课程删除成功! \033[0m";
524             *)
525                 echo "取消删除";;
526         esac
527         manage_course
528         #未知异常退出程序
529         exit 1
530     else
531         echo -e "\033[31m该课程不存在! \033[0m"
532         manage_course
533         #未知异常退出程序
534         exit 1
535     fi
536 }
537
538 #修改课程信息
539 change_course()
540 {
541     echo -e "\n请输入课程代号 (输入cancel可取消) : "
542     read c_id
543     #判断输入的字符串是否为cancel
544     if test "cancel" = $c_id; then
545         manage_course
546         #未知异常退出程序
```

```
547     exit 1
548 fi
549 #检查课程信息是否存在
550 if test 0 != $(cat data/catalog/course | awk '{print $1}' | grep -c
551     "$c_id"); then
552     echo -e "\n所要修改的课程为: \033[33m"
553     #显示课程信息, 显示为黄色
554     cat data/catalog/course | grep "^$c_id"
555     #读入新的课程名称
556     echo -e "\n\033[0m请输入新的课程名称 (输入cancel可取消) : "
557     read c_name
558     #判断输入的字符串是否为cancel
559     if test "cancel" = $c_name; then
560         manage_course
561         #未知异常退出程序
562         exit 1
563     fi
564     #删除文件中的旧行
565     sed -i "" '/'$c_id'/d' data/catalog/course
566     #向文件中写入新行
567     echo $c_id $c_name >> data/catalog/course
568     echo -e "\n\033[32m修改课程成功! \033[0m"
569     manage_course
570     #未知异常退出程序
571     exit 1
572 else
573     echo -e "\033[31m该课程不存在! \033[0m"
574     manage_course
575     #未知异常退出程序
576     exit 1
577 fi
578 }
579 #查询课程信息, 选择查询方式
580 query_course()
581 {
582     echo -e "\n请选择查询方式: "
583     echo -e "\033[35m1 \033[0m按课程代号查询"
584     echo -e "\033[35m2 \033[0m按课程名称查询"
585     echo -e "\033[35m3 \033[0m显示所有课程"
586     echo -e "\033[35mr \033[0m返回上级菜单"
587     echo -e "\033[35mq \033[0m退出系统"
588     read mode
589     case $mode in
590         1) query_course_id;;
591         2) query_course_name;;
592         3) query_course_all;;
593         r) manage_course;;
594             #正常退出程序
595         q) exit 0;;
```

```
596 *) echo -e "\033[31m输入非法! \033[0m"
597     query_course;;
598 esac
599 #未知异常退出程序
600 exit 1
601 }
602
603 #以下是与课程信息查询有关的函数
604
605 #按课程号查询课程
606 query_course_id()
607 {
608     echo -e "\n请输入课程代号: "
609     read c_id
610     #判断该课程号是否存在
611     if test 0 != $(cat data/catalog/course | grep -c "^$c_id"); then
612         echo -e "\n所查询的课程为: \033[33m"
613         #显示课程信息, 显示为黄色
614         cat data/catalog/course | grep "^$c_id"
615         echo -en "\033[0m"
616         query_course
617         #未知异常退出程序
618         exit 1
619     else
620         echo -e "\033[31m课程不存在! \033[0m"
621         query_course
622         #未知异常退出程序
623         exit 1
624     fi
625 }
626
627 #按课程名查询课程信息
628 query_course_name()
629 {
630     echo -e "\n请输入课程名称: "
631     read c_name
632     #判断该课程名称是否存在
633     if test 0 != $(cat data/catalog/course | grep -c "$c_name$"); then
634         echo -e "\n所查询的课程为: \033[33m"
635         #显示课程信息, 显示为黄色
636         #若存在多个同名课程, 按课程号排序
637         #支持正则表达式
638         cat data/catalog/course | grep "$c_name$" | sort
639         echo -en "\033[0m"
640         query_course
641         #未知异常退出程序
642         exit 1
643     else
644         echo -e "\033[31m课程不存在! \033[0m"
645         query_course
```

```
646     #未知异常退出程序
647     exit 1
648 fi
649 }
650
651 #显示所有课程信息
652 query_course_all()
653 {
654     #判断是否有课程信息存在
655     if test 0 != $(cat data/catalog/course | wc -l); then
656         echo -e "\n所查询的课程为: \033[33m"
657         #显示所有课程信息, 按课程号排序
658         cat data/catalog/course | sort
659         echo -en "\033[0m"
660         query_course
661         #未知异常退出程序
662         exit 1
663     else
664         echo -e "\033[31m课程不存在! \033[0m"
665         query_course
666         #未知异常退出程序
667         exit 1
668     fi
669 }
670
671 #以下是与修改密码有关的函数
672
673 #修改密码, 判断所修改用户的类型并调用相关函数
674 change_password()
675 {
676     #管理员身份验证
677     login_admin
678     #输入要修改密码的用户名
679     echo -e "\n请输入用户名 (输入cancel可取消) : "
680     read user
681     #判断输入的字符串是否为cancel
682     if test "cancel" = $user; then
683         mode_admin
684         return 1
685     fi
686     #判断用户是否存在
687     if test 0 != $(cat data/passwd | awk '{print $2}' | grep -c "$user");
688         then
689             #判断用户类型
690             case $(cat data/passwd | grep -o "^[^[:blank:]]{1}[^[:blank:]]$user" |
691                 awk '{print $1}') in
692                 a) change_passwd_admin;;
693                 t) change_passwd_teacher $user;;
694                 s) change_passwd_student $user;;
695             esac
```

```
694     mode_admin
695     #未知异常退出程序
696     exit 1
697 else
698     echo -e "\033[31m该用户不存在! \033[0m"
699     mode_admin
700     #未知异常退出程序
701     exit 1
702 fi
703 }
704
705 #修改管理员密码
706 change_passwd_admin()
707 {
708     #输入不回显
709     stty -echo
710     #进行加密用的变量
711     salt="admin"
712     salt_reverse="nimda"
713     #输入两次新密码
714     echo -n "请输入新密码: "
715     read passwd
716     echo ""
717     echo -n "请再输入一遍: "
718     read passwd_again
719     echo ""
720     #检验两次输入是否一致
721     if test $passwd != $passwd_again; then
722         echo -e "\033[31m两次输入不一致! \033[0m"
723         mode_admin
724         exit 1
725 fi
726     #将明文转换为密文
727     passwd_md5=$(echo $($passwd" | md5)$salt | md5)
728     passwd_sha=$(echo $($passwd_md5)$salt_reverse | shasum | grep -o
729     "^[^[:blank:]]*")
730     #删除密码文件中的旧行
731     sed -i "" '/admin/d' data/passwd
732     #将新行写入密码文件
733     echo "a" "admin" $passwd_sha >> data/passwd
734     #撤销输入不回显
735     stty echo
736     #输出修改成功的提示, 字体为绿色
737     echo -e "\n\033[32m修改成功! \033[0m"
738     mode_admin
739     #未知异常退出程序
740     exit 1
741 }
742 #修改教师密码
```

```
743 change_passwd_teacher()
744 {
745     #输入不回显
746     stty -echo
747     #输入两次新密码
748     echo -n "请输入新密码: "
749     read passwd
750     echo ""
751     echo -n "请再输入一遍: "
752     read passwd_again
753     echo ""
754     #检验两次输入是否一致
755     if test $passwd != $passwd_again; then
756         echo -e "\033[31m两次输入不一致! \033[0m"
757         mode_admin
758         exit 1
759     fi
760     #将明文转换为密文，进行一次sha1加密
761     passwd_sha=$(echo $passwd | shasum | grep -o "^[^[:blank:]]*")
762     #删除密码文件中的旧行
763     sed -i "" '/'$1'/d' data/passwd
764     #将新行写入密码文件
765     echo "t" $1 $passwd_sha >> data/passwd
766     #撤销输入不回显
767     stty echo
768     #输出修改成功的提示，字体为绿色
769     echo -e "\n\033[32m修改成功! \033[0m"
770     mode_admin
771     #未知异常退出程序
772     exit 1
773 }
774 #修改学生密码
775 change_passwd_student()
776 {
777     #输入不回显
778     stty -echo
779     #输入两次新密码
780     echo -n "请输入新密码: "
781     read passwd
782     echo ""
783     echo -n "请再输入一遍: "
784     read passwd_again
785     echo ""
786     #检验两次输入是否一致
787     if test $passwd != $passwd_again; then
788         echo -e "\033[31m两次输入不一致! \033[0m"
789         mode_admin
790         exit 1
791     fi
```

```
793     #将明文转换为密文，进行一次md5加密
794     passwd_md5=$(echo $passwd | md5)
795     #删除密码文件中的旧行
796     sed -i "" '/'$1'/d' data/passwd
797     #将新行写入密码文件
798     echo "$t" $1 $passwd_md5 >> data/passwd
799     #撤销输入不回显
800     stty echo
801     #输出修改成功的提示，字体为绿色
802     echo -e "\n\033[32m修改成功! \033[0m"
803     mode_admin
804     #未知异常退出程序
805     exit 1
806 }
```

```
1 #!/bin/bash
2 #UTF-8
3 #***** * * * * *
4 #名称:      teacher.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      教师模块的函数实现
8 #***** * * * * *
9
10 #教师主函数
11 main_teacher()
12 {
13     echo -n "请输入教师工号: "
14     read t_id
15     #检查学生信息是否存在
16     if test 0 = $(cat data/catalog/teacher | grep $t_id | wc -l); then
17         echo -e "\033[31m用户不存在! \033[0m"
18         #未知异常退出程序
19         exit 1;
20     fi
21     #教师姓名
22     login_teacher
23     #查找教师姓名
24     t_name=$(cat data/catalog/teacher | grep $t_id | awk '{print $2}')
25     echo -e "\033[32m欢迎"$t_name"老师! \033[0m"
26     #选择操作
27     mode_teacher
28     #未知异常退出程序
29     exit 1
30 }
31
32 #教师登录认证
33 login_teacher()
34 {
```

```
35     #输入不回显
36     stty -echo
37     #在密码文件中找到该教师的密码
38     #教师用户所在的行以字母t开头
39     passwd=$(cat data/passwd | grep "^\t \$t_id" | awk '{print $3}')
40     #超过三次输入错误密码, 直接退出程序
41     for ((i=0; i<3; i++))
42     do
43         #输入密码
44         echo -n "请输入登录密码: "
45         read passwd_try
46         echo ""
47         #对密码明文进行sha1运算
48         passwd_try_sha=$(echo $passwd_try | shasum | grep -o
49             "[^[:blank:]]*")
50         #字符串比对
51         if test $passwd = $passwd_try_sha; then
52             #撤销输入不回显
53             stty echo
54             return 0
55         else
56             echo -e "\033[31m密码错误! \033[0m"
57         fi
58     done
59     #撤销输入不回显
60     stty echo
61     #超过三次输入密码错误退出程序
62     exit 3
63 }
64 #教师主菜单
65 mode_teacher()
66 {
67     echo -e "\n请选择操作: "
68     echo -e "\033[35m1 \033[0m管理课程信息"
69     echo -e "\033[35m2 \033[0m修改用户密码"
70     echo -e "\033[35mq \033[0m退出系统"
71     read mode
72     case $mode in
73         1) choose_course_t;;
74         2) change_passwd_t;;
75         #正常退出程序
76         q) exit 0;;
77         *) echo -e "\033[31m输入非法! \033[0m"
78             mode_teacher;;
79     esac
80     #未知异常退出程序
81     exit 1
82 }
```

```
84 #选择要操作的课程
85 choose_course_t()
86 {
87     echo -e "\n请选择课程: "
88     declare -i i=1
89     #第一行为提示输入0返回
90     echo -e "\033[35m0 \033[0m返回"
91     #列出该教师的所有课程, 按课程号排序
92     #按课程信息文件夹内的文件名进行匹配, 文件名中只保留课程号部分
93     for course in $(ls data/teacher_course/$t_id* | grep -o "[[:alnum:]]*\$"
94                 | sort)
95     do
96         #按“序号(紫色) 课程号 课程名”的格式输出
97         echo -e "\033[35m$i"\033[0m $course $(cat data/catalog/course |
98             grep "^$course" | awk '{print $2}'))
99         i=i+1
100    done
101    declare -i choose_course
102    read choose_course
103    #判断输入是否为0(返回)
104    if test $choose_course -eq 0; then
105        mode_teacher
106        #未知异常退出程序
107        exit 1
108    #判断输入是否合法
109    else if test $choose_course -le $(ls data/teacher_course/$t_id* | grep
110        -c "[[:alnum:]]*\$"); then
111        #将课程列表中的所选项赋值给c_id
112        c_id=$(ls data/teacher_course/$t_id* | grep -o "[[:alnum:]]*\$" |
113            sort | sed -n $choose_course"p")
114        #根据课程号搜索课程名
115        c_name=$(cat data/catalog/course | grep "^$c_id" | awk '{print $2}')
116        manage_course_t
117        #未知异常退出程序
118        exit 1
119    fi
120    fi
121    #未知异常退出程序
122    exit 1
123 }
124 }
125
126 #教师课程管理菜单
127 manage_course_t()
128 {
129     echo -e "\n请选择操作: "
```

```
130 echo -e "\033[35m1 \033[0m选课名单管理"
131 echo -e "\033[35m2 \033[0m课程信息管理"
132 echo -e "\033[35m3 \033[0m作业实验管理"
133 echo -e "\033[35mr \033[0m返回上级目录"
134 echo -e "\033[35mq \033[0m退出系统"
135 read mode
136 case $mode in
137     1) manage_student_t;;
138     2) manage_info;;
139     3) manage_work;;
140     r) mode_teacher;;
141     #正常退出程序
142     q) exit 0;;
143     *) echo -e "\033[31m输入非法! \033[0m"
144         manage_course_t
145         #未知异常退出程序
146         exit 1;;
147     esac
148     #未知异常退出程序
149     exit 1
150 }
151
152 #以下为学生管理有关操作的函数
153
154 #教师管理学生菜单
155 manage_student_t()
156 {
157     echo -e "\n请选择操作: "
158     echo -e "\033[35m1 \033[0m查看选课学生"
159     echo -e "\033[35m2 \033[0m添加选课学生"
160     echo -e "\033[35m3 \033[0m删除选课学生"
161     echo -e "\033[35m4 \033[0m修改学生信息"
162     echo -e "\033[35mr \033[0m返回上级目录"
163     echo -e "\033[35mq \033[0m退出系统"
164     read mode
165     case $mode in
166         1) query_student_t;;
167         2) add_student_t;;
168         3) delete_student_t;;
169         4) change_student_t;;
170         r) manage_course_t;;
171         q) exit 0;;
172         *) echo -e "\033[31m输入非法! \033[0m"
173             manage_student_t;;
174     esac
175     exit 0
176 }
177
178 #添加选课学生
179 add_student_t()
```

```
180  {
181      echo -e "\n请输入学生学号 (输入cancel可取消) : "
182      read s_id
183      #判断输入是否为cancel
184      if test "cancel" = $s_id; then
185          manage_student_t
186          #未知异常退出程序
187          exit 1
188      fi
189      #判断学生信息是否存在
190      if test 0 != $(cat data/catalog/student | awk '{print $1}' | grep -c
191                  "$s_id"); then
192          #判断学生是否选课
193          if test -d data/student_course_teacher/$s_id"_"$c_id*; then
194              echo -e "\033[31m该学生已选课! \033[0m"
195              manage_student_t
196              #未知异常退出程序
197              exit 1
198          fi
199          #学生信息存在且未选课, 建立选课信息的目录
200          mkdir data/student_course_teacher/$s_id"_"$c_id"_"$t_id
201          echo -e "\n\033[32m添加成功! \033[0m"
202          #函数执行完毕
203          manage_student_t
204          #未知异常退出程序
205          exit 1
206      fi
207      #学生信息不存在, 导入学生信息, 再添加选课记录
208      echo -e "\n学生信息不存在, 请输入学生姓名 (输入cancel可取消) : "
209      read s_name
210      if test "cancel" = $s_name; then
211          manage_student_t
212          #未知异常退出程序
213          exit 1
214      fi
215      #将学生信息写入文件
216      echo $s_id $s_name >> data/catalog/student
217      #初始密码为123456
218      password=123456
219      password_md=$(echo $password | md5)
220      #将加密过的密码写入密码文件
221      echo "s" $s_id $password_md >> data/passwd
222      #添加选课记录
223      mkdir data/student_course_teacher/$s_id"_"$c_id"_"$t_id
224      echo -e "\n\033[32m增加学生成功! \033[0m"
225      manage_student_t
226      #未知异常退出程序
227      exit 1
228 }
```

```
229 #删除学生选课信息
230 delete_student_t()
231 {
232     echo -e "\n请输入学生学号 (输入cancel可取消) : "
233     read s_id
234     #判断输入是否为cancel
235     if test "cancel" = $s_id; then
236         manage_student_t
237         #未知异常退出程序
238         exit 1
239     fi
240     #判断选课记录的目录是否存在
241     if test -d data/student_course_teacher/$s_id"_"$c_id"_"$t_id; then
242         #删除选课记录的目录
243         rm -rf data/student_course_teacher/$s_id"_"$c_id"_"$t_id
244         echo -e "\n\033[32m删除成功! \033[0m"
245         manage_student_t
246         #未知异常退出程序
247         exit 1
248     else
249         echo -e "\033[31m无选课信息! \033[0m"
250         manage_student_t
251         #未知异常退出程序
252         exit 1
253     fi
254 }
255
256 #修改学生信息
257 change_student_t()
258 {
259     echo -e "\n请输入学生学号 (输入cancel可取消) : "
260     read s_id
261     #判断输入是否为cancel
262     if test "cancel" = $s_id; then
263         manage_student_t
264         #未知异常退出程序
265         exit 1
266     fi
267     #检查该学生信息是否存在
268     if test 0 != $(cat data/catalog/student | awk '{print $1}' | grep -c
269         "$s_id"); then
270         echo -e "\n所要修改的学生为: \033[33m"
271         #显示该学生的信息, 显示为黄色
272         cat data/catalog/student | grep "^$s_id"
273         echo -e "\n\033[0m请输入修改后的学生姓名 (输入cancel可取消) : "
274         read s_name
275         #判断输入是否为cancel
276         if test "cancel" = $s_name; then
277             manage_student_t
278             #未知异常退出程序
```

```
278         exit 1
279     fi
280     #删除旧行
281     sed -i "" '/'$s_id'/d' data/catalog/student
282     #将新信息写入文件
283     echo $s_id $s_name >> data/catalog/student
284     echo -e "\n\033[32m修改成功! \033[0m"
285     manage_student_t
286     #未知异常退出程序
287     exit 1
288     #学生信息不存在
289 else
290     echo -e "\033[31m该学号不存在! \033[0m"
291     manage_student_t
292     #未知异常退出程序
293     exit 1
294 fi
295 }
296
297 #查询学生信息，选择查询方式
298 query_student_t()
299 {
300     echo -e "\n请选择查询方式："
301     echo -e "\033[35m1 \033[0m按学号查询"
302     echo -e "\033[35m2 \033[0m按姓名查询"
303     echo -e "\033[35m3 \033[0m查询选课名单"
304     #该功能不对教师开放
305     #echo -e "\033[35m4 \033[0m查询所有学生"
306     echo -e "\033[35mr \033[0m返回上级目录"
307     echo -e "\033[35mq \033[0m退出系统"
308     read mode
309     case $mode in
310         1) query_student_id;;
311         2) query_student_name;;
312         3) query_student_course;;
313         #4) query_student_all;;
314         r) manage_student_t;;
315         #正常退出程序
316         q) exit 0;;
317         *) echo -e "\033[31m输入非法! \033[0m"
318             query_student_t;;
319     esac
320     #未知异常退出程序
321     exit 1
322 }
323
324 #按学号查询学生（核心部分），被调用前s_id变量已被正确设置
325 query_student()
326 {
327     #检查该学生信息是否存在
```

```
328 if test 0 = $(cat data/catalog/student | grep -c "^$s_id"); then
329   echo -e "\033[31m该学生不存在! \033[0m"
330   return 1
331 #检查该学生是否选课
332 else if test -d data/student_course_teacher/$s_id_"$c_id_"$t_id; then
333   echo -e "\n所查询的学生为: \033[33m"
334   cat data/catalog/student | grep "^$s_id"
335   echo -en "\033[0m"
336   return 0
337 #未选课的学生教师不可查询
338 else
339   echo -e "\033[31m该学生未选课! \033[0m"
340   return 1
341   fi
342   fi
343 }
344 #按学号查询学生
345 query_student_id()
346 {
347   echo -e "\n请输入学生学号: (输入cancel可取消)"
348   read s_id
349   #判断输入是否为cancel
350   if test "cancel" = $s_id; then
351     query_student_t
352     #未知异常退出程序
353     exit 1
354   fi
355   #调用查询函数
356   query_student
357   query_student_t
358   #未知异常退出程序
359   exit 1
360 }
361 }
362 #按姓名查找学生
363 query_student_name()
364 {
365   echo -e "\n请输入学生姓名: (输入cancel可取消)"
366   read s_name
367   #判断输入是否为cancel
368   if test "cancel" = $s_name; then
369     query_student_t
370     #未知异常退出程序
371     exit 1
372   fi
373   #查找姓名对应的学号, 若有重名按学号排序
374   for s_id in $(cat data/catalog/student | grep "$s_name$" | sort | awk
375     '{print $1}')
376   do
```

```
377     #调用查询函数
378     query_student
379 done
380 query_student_t
381 #未知异常退出程序
382 exit 1
383 }
384
385 #查看课程的所有选课学生
386 query_student_course()
387 {
388     #判断是否有学生选课
389     if test 0 != $(ls -1d data/student_course_teacher/*_"$c_id"_"$t_id |
390         wc -l); then
391         echo -e "\n所查询的学生为: \033[33m"
392         #通过选课信息的文件名查找学号
393         for s_id in $(ls -1d data/student_course_teacher/*_"$c_id"_"$t_id |
394             cut -d "/" -f 3 | grep -o "^[[:digit:]]*")
395             do
396                 cat data/catalog/student | grep "^$s_id"
397             done
398             echo -en "\033[0m"
399             query_student_t
400             #未知异常退出程序
401             exit 1
402         else
403             echo -e "\033[31m无学生选课! \033[0m"
404             query_student_t
405             #未知异常退出程序
406             exit 1
407         fi
408     }
409
410     #查看所有学生
411     query_student_all()
412     {
413         #判断学生信息是否为空
414         if test 0 != $(cat data/catalog/student | wc -l); then
415             echo -e "\n所查询的学生为: \033[33m"
416             #输出按学号排序
417             cat data/catalog/student | sort
418             echo -en "\033[0m"
419             query_student_t
420             #未知异常退出程序
421             exit 1
422         else
423             echo -e "\033[31m学生不存在! \033[0m"
424             query_student_t
425             #未知异常退出程序
426             exit 1
427 }
```

```
425     fi
426 }
427
428 #以下为管理课程信息的函数
429 #管理课程信息的菜单
430 manage_info()
431 {
432     echo -e "\n请选择操作: "
433     echo -e "\033[35m1 \033[0m发布信息"
434     echo -e "\033[35m2 \033[0m编辑信息"
435     echo -e "\033[35m3 \033[0m删除信息"
436     echo -e "\033[35m4 \033[0m显示信息"
437     echo -e "\033[35mr \033[0m返回上级目录"
438     echo -e "\033[35mq \033[0m退出系统"
439     read mode
440     case $mode in
441         1) add_info;;
442         2) edit_info;;
443         3) delete_info;;
444         4) list_info
445             manage_info;;
446         r) manage_course_t;;
447         #正常退出程序
448         q) exit 0;;
449         *) echo -e "\033[31m输入非法! \033[0m"
450             manage_info
451             #未知异常退出程序
452             exit 1;;
453     esac
454     #未知异常退出程序
455     exit 1
456 }
457
458 #添加课程信息
459 add_info()
460 {
461     echo -e "请输入课程信息: "
462     read line
463     #向文件写入新信息
464     #每条信息用' | '分割
465     echo -n "$line| " >> data/teacher_course/$t_id_"$c_id"
466     echo -e "\n\033[32m信息发布成功! \033[0m"
467     manage_info
468     #未知异常退出程序
469     exit 1
470 }
471
472 #删除课程信息
473 delete_info()
474 {
```

```
475     #列出所有课程信息
476     list_info
477     echo -e "\n请选择要删除的信息: "
478     declare -i choice
479     read choice
480     #判断输入的选项序号是否合法
481     #每条信息之间用' | '分割, 获取总信息条数
482     if test $choice -le $(awk -v RS='\0' -F ' | ' '{print NF-1}'
483         data/teacher_course/$t_id"_"$c_id); then
484         #将该条信息以外的所有信息重新写入文件, 覆盖原文件
485         #awk内两个for循环先后执行, 分别复制该条信息之前的和之后的
486         echo $($awk -v RS='\0' -v col_del=$choice -v ORS="" -F ' | ' '{for(i=1;
487             i<col_del; i++){print $i" | "}} for(i=col_del+1; i<NF; i++){print
488             $i" | "})' data/teacher_course/$t_id"_"$c_id) >
489             data/teacher_course/$t_id"_"$c_id
490     else
491         echo -e "\033[31m非法输入! \033[0m"
492     fi
493     manage_info
494     #未知异常退出程序
495     exit 1
496 }
497 #编辑课程信息
498 edit_info()
499 {
500     #显示所有信息
501     list_info
502     echo -e "\n请选择要编辑的信息: "
503     declare -i choice
504     read choice
505     #判断输入的选项序号是否合法
506     #每条信息之间用' | '分割, 获取总信息条数
507     if test $choice -le $(awk -v RS='\0' -F ' | ' '{print NF-1}'
508         data/teacher_course/$t_id"_"$c_id); then
509         echo "请输入修改后的信息: "
510         read newLine
511         #覆盖原文件, 该条信息前后的原文复制
512         #awk内两个for循环, 先复制该条信息之前的, 再写入新信息, 再复制该条信息之后的
513         echo $($awk -v RS='\0' -v col_edit=$choice -v ORS="" -v line=$newLine
514             -F ' | ' '{for(i=1; i<col_edit; i++){print $i" | "}} {print line" | "}
515             for(i=col_edit+1; i<NF; i++){print $i" | "})'
516             data/teacher_course/$t_id"_"$c_id) >
517             data/teacher_course/$t_id"_"$c_id
518     else
519         echo -e "\033[31m非法输入! \033[0m"
520     fi
521     manage_info
522     #未知异常退出程序
523     exit 1
```

```
516 }
517
518 #列出所有信息
519 list_info()
520 {
521     echo -e "\n所有课程信息如下: \033[32m"
522     #以' | '为分割符, 逐条信息列出
523     for ((i=1; i<=$(awk -v RS='`' -F ' | ' '{print NF-1}' \
524           data/teacher_course/$t_id"_"$c_id); i++))
525     do
526         echo -e $i $(cat data/teacher_course/$t_id"_"$c_id | cut -d " | " -f
527             $i)
528     done
529     echo -en "\033[0m"
530     return 0
531 }
532
533 #以下为管理课程任务的函数
534 #管理任务信息的菜单
535 manage_work()
536 {
537     echo -e "\n请选择操作: "
538     echo -e "\033[35m1 \033[0m布置任务"
539     echo -e "\033[35m2 \033[0m编辑任务"
540     echo -e "\033[35m3 \033[0m删除任务"
541     echo -e "\033[35m4 \033[0m显示任务"
542     echo -e "\033[35m5 \033[0m完成情况"
543     echo -e "\033[35mr \033[0m返回上级目录"
544     echo -e "\033[35mq \033[0m退出系统"
545     read mode
546     case $mode in
547         1) add_work;;
548         2) edit_work;;
549         3) delete_work;;
550         4) list_work
551             manage_work;;
552         5) check_work;;
553         r) manage_course_t;;
554         q) exit 0;;
555         *) echo -e "\033[31m输入非法! \033[0m"
556             manage_work
557             #未知异常退出程序
558             exit 1;;
559     esac
560     #未知异常退出程序
561     exit 1
562 }
563
564 #添加新任务
565 add_work()
```

```
564 {
565     echo -e "\n请输入任务名称 (输入cancel可取消) : "
566     read work_name
567     #判断输入是否为cancel
568     if test "cancel" = $work_name; then
569         manage_work
570         #未知异常退出程序
571         exit 1
572     fi
573     #在所有该课程的学生选课目录下创建任务目录, 用于存放学生提交的文件
574     for s_dir in $(ls -d data/student_course_teacher/*$c_id"_"$t_id)
575     do
576         mkdir $s_dir/$work_name
577     done
578     echo -e "\n\033[32m任务创建成功! \033[0m"
579     manage_work
580     #未知异常退出程序
581     exit 1
582 }
583
584 #删除任务
585 delete_work()
586 {
587     #列出所有任务
588     list_work
589     echo -e "\n请选择要删除的任务: "
590     declare -i choice
591     read choice
592     #判断输入是否合法
593     if test $choice -le $(ls $s_dir | wc -l); then
594         #根据输入的序号确定任务名称
595         work_del=$(ls $s_dir | head -n $choice)
596         #删除所有该课程的学生选课目录下的该作业目录
597         rm -rf data/student_course_teacher/*$c_id"_"$t_id/$work_del
598     else
599         echo -e "\033[31m非法输入! \033[0m"
600     fi
601     manage_work
602     #未知异常退出程序
603     exit 1
604 }
605
606 #编辑任务名称
607 edit_work()
608 {
609     #列出所有任务
610     list_work
611     echo -e "\n请选择要编辑的任务: "
612     declare -i choice
613     read choice
```

```
614 #判断输入是否合法
615 if test $choice -le $(ls $s_dir | wc -l); then
616     #根据输入的序号确定任务名称
617     work_old=$(ls $s_dir | head -n $choice)
618     echo -e "\n请输入新的任务名称: "
619     read work_new
620     for s_dir in $(ls -d
621         data/student_course_teacher/*$c_id"_"$t_id/$work_old)
622     do
623         #重命名所有该课程的学生选课目录下的该作业目录
624         mv $s_dir $(echo $s_dir | sed 's/'$work_old'/'$work_new'/')
625     done
626 else
627     echo -e "\033[31m非法输入! \033[0m"
628 fi
629 manage_work
630 #未知异常退出程序
631 exit 1
632 }
633 #列出所有任务
634 list_work()
635 {
636     #任选一个选课学生的选课目录
637     s_dir=$(ls -d data/student_course_teacher/*$c_id"_"$t_id | head -1)
638     echo -e "\n所布置的任务为: \033[32m"
639     declare -i i=1
640     #列出选课目录下的所有目录名, 按字典序排列
641     for work in $(ls $s_dir | sort)
642     do
643         echo $i $work
644         i=i+1
645     done
646     echo -en "\033[0m"
647     return 0
648 }
649
650 #检查任务提交情况
651 check_work()
652 {
653     #列出所有任务
654     list_work
655     echo -e "\n请选择要检查的任务: "
656     declare -i choice
657     read choice
658     #判断输入是否合法
659     if test $choice -le $(ls $s_dir | wc -l); then
660         #根据输入的序号确定任务名称
661         work_check=$(ls $s_dir | head -n $choice)
662         echo -e "\n完成情况如下: "
```

```
663     #检查每个选课学生的该作业目录
664     for s_dir in data/student_course_teacher/*$c_id"_"$t_id/$work_check
665     do
666         #从作业目录的路径名中提取学号
667         s_id=$(echo $s_dir | cut -d '/' -f 3 | cut -d '_' -f 1)
668         #若作业目录下有文件, 视为已提交, 否则视为未提交
669         if test 0 != $(ls $s_dir | wc -l); then
670             echo -e $s_id "\033[32m已提交\033[0m"
671         else
672             echo -e $s_id "\033[34m未提交\033[0m"
673         fi
674     done
675     echo -en "\033[0m"
676 else
677     echo -e "\033[31m非法输入! \033[0m"
678 fi
679 manage_work
680 #未知异常退出程序
681 exit 1
682 }
683
684 #修改教师密码
685 change_passwd_t()
686 {
687     #身份验证
688     login_teacher
689     #输入不回显
690     stty -echo
691     echo -n "请输入新密码: "
692     read passwd
693     echo ""
694     echo -n "请再输入一遍: "
695     read passwd_again
696     echo ""
697     #检验两次输入是否一致
698     if test $passwd != $passwd_again; then
699         echo -e "\033[31m两次输入不一致! \033[0m"
700         mode_admin
701         #未知异常退出程序
702         exit 1
703     fi
704     #对密码明文进行加密
705     passwd_sha=$(echo $passwd | shasum | grep -o "^[^[:blank:]]*")
706     #删除密码文件中的旧行
707     sed -i "" '/'$t_id'/d' data/passwd
708     #向密码文件中写入新行
709     echo "t" $t_id $passwd_sha >> data/passwd
710     #撤销输入不回显
711     stty echo
712     #修改成功提示, 显示为绿色
```

```
713     echo -e "\n\033[32m修改成功! \033[0m"
714     mode_teacher
715     #未知异常退出程序
716     exit 1
717 }

1 #!/bin/bash
2 #UTF-8
3 #*****学生模块*****
4 #名称:      student.sh
5 #作者:      吴同
6 #学号:      3170104848
7 #功能:      学生模块的函数实现
8 #*****学生模块*****
9
10 #学生主函数
11 main_student()
12 {
13     echo -n "请输入学生学号: "
14     read s_id
15     #检查学生信息是否存在
16     if test 0 = $(cat data/catalog/student | grep $s_id | wc -l); then
17         echo -e "\033[31m用户不存在! \033[0m"
18         #未知异常退出程序
19         exit 1;
20     fi
21     #学生登录
22     login_student
23     #查找学生姓名
24     t_name=$(cat data/catalog/student | grep $s_id | awk '{print $2}')
25     echo -e "\033[32m欢迎"$t_name"同学! \033[0m"
26     #选择操作
27     mode_student
28     #未知异常退出程序
29     exit 1
30 }
31
32 #学生登录认证
33 login_student()
34 {
35     #输入不回显
36     stty -echo
37     #在密码文件中找到该学生的密码
38     #学生用户所在的行以字母s开头
39     passwd=$(cat data/passwd | grep "^s $s_id" | awk '{print $3}')
40     #超过三次输入错误密码, 直接退出程序
41     for ((i=0; i<3; i++))
42     do
43         #输入密码
```

```
44     echo -n "请输入登录密码: "
45     read passwd_try
46     echo ""
47     #对密码明文进行md5运算
48     passwd_try_md5=$(echo $passwd_try | md5)
49     #字符串比对
50     if test $passwd = $passwd_try_md5; then
51         #撤销输入不回显
52         stty echo
53         return 0
54     else
55         echo -e "\033[31m密码错误! \033[0m"
56     fi
57 done
58 #撤销输入不回显
59 stty echo
60 #超过三次输入密码错误退出程序
61 exit 3
62 }
63
64 #学生主菜单
65 mode_student()
66 {
67     echo -e "\n请选择操作: "
68     echo -e "\033[35m1 \033[0m查看课程信息"
69     echo -e "\033[35m2 \033[0m修改用户密码"
70     echo -e "\033[35mq \033[0m退出系统"
71     read mode
72     case $mode in
73         1) choose_course_s;;
74         2) change_passwd_s;;
75         #正常退出程序
76         q) exit 0;;
77         *) echo -e "\033[31m输入非法! \033[0m"
78             mode_student;;
79     esac
80     #未知异常退出程序
81     exit 1
82 }
83
84 #选择要操作的课程
85 choose_course_s()
86 {
87     echo -e "\n请选择课程: "
88     declare -i i=1
89     #第一行为提示输入0返回
90     echo -e "\033[35m0 \033[0m返回"
91     #列出该学生的所有课程, 按课程号排序
92     #按选课信息文件夹内的文件名进行匹配
93     #第一次分割去掉上级目录名, 第二次分割取课程号
```

```
94     for course in $(ls -d data/student_course_teacher/$s_id* | cut -d "/" -f 3 | cut -d "_" -f 2 | sort)
95   do
96     #按“序号（紫色） 课程号 课程名”的格式输出
97     echo -e "\033[35m$i"\033[0m $course $(cat data/catalog/course |
98       grep "^$course" | awk '{print $2}'))
99     i=i+1
100   done
101   declare -i choose_course
102   read choose_course
103   #判断输入是否为0（返回）
104   if test $choose_course -eq 0; then
105     mode_student
106     #未知异常退出程序
107     exit 1
108   #判断输入是否合法
109   else if test $choose_course -le $(ls -d
110     data/student_course_teacher/$s_id* | wc -l); then
111     #将课程列表中的所选项赋值给c_id
112     c_id=$(ls -d data/student_course_teacher/$s_id* | cut -d "/" -f 3 |
113       cut -d "_" -f 2 | sort | sed -n $choose_course"p")
114     #搜索该学生的该课程对应的文件名，确定教师
115     t_id=$(ls -d data/student_course_teacher/$s_id*$c_id* | cut -d "/" -f 3 | cut -d "_" -f 3)
116     #根据课程号搜索课程名
117     c_name=$(cat data/catalog/course | grep "^$c_id" | awk '{print $2}')
```

```
118   manage_course_s
119   #未知异常退出程序
120   exit 1
121 fi
122 fi
123 fi
124 #未知异常退出程序
125 exit 1
126 }
127 }

128 #学生课程管理菜单
129 manage_course_s()
130 {
131   echo -e "\n请选择操作: "
132   echo -e "\033[35m1 \033[0m查看通知"
133   echo -e "\033[35m2 \033[0m管理作业"
134   echo -e "\033[35mr \033[0m返回上级"
135   echo -e "\033[35mq \033[0m退出系统"
136   read mode
137   case $mode in
```

```
139     1) show_info;;
140     2) manage_work_s;;
141     r) mode_student;;
142     #正常退出程序
143     q) exit 0;;
144     *) echo -e "\033[31m输入非法! \033[0m"
145         manage_course_s;;
146     esac
147     #未知异常退出程序
148     exit 1
149 }
150
151 #显示课程信息
152 show_info()
153 {
154     echo -e "\n所有课程信息如下: \033[32m"
155     #读取课程信息文件, 课程信息以' '|分割, 利用awk得出信息总条数
156     for ((i=1; i<=${awk -v RS='\\0' -F '|' '{print NF-1}' \
157             data/teacher_course/$t_id"_"$c_id); i++))
158     do
159         #显示第i条信息, 信息显示为绿色
160         echo -e $i $(cat data/teacher_course/$t_id"_"$c_id | cut -d "|" -f
161             $i)
162     done
163     echo -en "\033[0m"
164     manage_course_s
165     #未知异常退出程序
166     exit 1
167 }
168
169 #以下为学生作业有关操作的函数
170 #学生作业操作的菜单
171 manage_work_s()
172 {
173     echo -e "\n请选择操作: "
174     echo -e "\033[35m1 \033[0m查看作业"
175     echo -e "\033[35m2 \033[0m提交作业"
176     echo -e "\033[35m3 \033[0m重交作业"
177     echo -e "\033[35m4 \033[0m检查作业"
178     echo -e "\033[35mr \033[0m返回上级"
179     echo -e "\033[35mq \033[0m退出系统"
180     read mode
181     case $mode in
182         1) show_work
183             manage_work_s;;
184         2) add_work;;
185         3) edit_work;;
186         4) check_work;;
187         r) mode_student;;
188         #正常退出程序
```

```
187     q) exit 0;;
188     *) echo -e "\033[31m输入非法! \033[0m"
189         manage_work_s;;
190     esac
191     #未知异常退出程序
192     exit 1
193 }
194
195 #显示该课程的所有任务的提交情况
196 show_work()
197 {
198     #任务文件所在的目录
199     s_dir=$(ls -d data/student_course_teacher/*$c_id"_"$t_id | head -1)
200     #显示该课程的所有任务
201     echo -e "\n该课程的任务为: \033[32m"
202     declare -i i=1
203     #按任务名称的字母序排列
204     for work in $(ls $s_dir | sort)
205     do
206         #显示任务名称
207         echo -en $i $work" "
208         #如果该目录下存在文件，则视为已提交，反之视为未提交
209         if test 0 = $(ls
210             data/student_course_teacher/$s_id"_"$c_id"_"$t_id/$work | wc
211             -l); then
212             echo "未提交"
213         else
214             echo "已提交"
215         fi
216         i=i+1
217     done
218     echo -en "\033[0m"
219     #函数执行完毕后返回
220     return 0
221 }
222
223 #学生提交作业
224 add_work()
225 {
226     #显示所有任务完成情况
227     show_work
228     echo -e "\n请选择要提交的任务（输入0返回）: "
229     declare -i work_choose
230     read work_choose
231     #判断输入是否为0（返回）
232     if test $work_choose -eq 0; then
233         manage_work_s
234         #未知异常退出程序
235         exit 1
236     #判断输入是否合法
```

```
235     else if test $work_choose -le $(ls $s_dir | wc -l); then
236         #任务序号对应的任务名
237         work_name=$(ls $s_dir | sort | sed -n $work_choose"p")
238         echo -e "\n请输入要提交的文件 (输入cancel可取消) : "
239         read fileName
240         #判断输入是否为cancel
241         if test "cancel" = $fileName; then
242             manage_course_s
243             #未知异常退出程序
244             exit 1
245         fi
246         #将选定的文件复制到任务目录下
247         cp $fileName $s_dir/$work_name
248         echo -e "\n\033[32m提交成功! \033[0m"
249         #未知异常退出程序
250         manage_course_s
251         exit 1
252     else
253         echo -e "\033[31m输入非法! \033[0m"
254         choose_course_s
255         #未知异常退出程序
256         exit 1
257     fi
258 fi
259 }
260
261 #学生编辑已提交作业
262 edit_work()
263 {
264     #显示所有任务完成情况
265     show_work
266     echo -e "\n请选择要重交的任务 (输入0返回) : "
267     declare -i work_choose
268     read work_choose
269     #判断输入是否为0 (返回)
270     if test $work_choose -eq 0; then
271         manage_work_s
272         #未知异常退出程序
273         exit 1
274     #判断输入是否合法
275     else if test $work_choose -le $(ls $s_dir | wc -l); then
276         #任务序号对应的任务名
277         work_name=$(ls $s_dir | sort | sed -n $work_choose"p")
278         echo -e "\n请输入要提交的文件 (输入cancel可取消) : "
279         read fileName
280         #判断输入是否为cancel
281         if test "cancel" = $fileName; then
282             manage_course_s
283             #未知异常退出程序
284             exit 1
```

```
285         fi
286         #删除该任务已提交的文件
287         rm $s_dir/$work_name/*
288         #将选定的文件复制到任务目录下
289         cp $fileName $s_dir/$work_name
290         echo -e "\n\033[32m重交成功! \033[0m"
291         manage_course_s
292         #未知异常退出程序
293         exit 1
294     else
295         echo -e "\033[31m输入非法! \033[0m"
296         choose_course_s
297         #未知异常退出程序
298         exit 1
299         fi
300     fi
301 }
302
303 #查看已提交作业
304 check_work()
305 {
306     #显示所有任务完成情况
307     show_work
308     echo -e "\n请选择要检查的任务 (输入0返回) : "
309     declare -i work_choose
310     read work_choose
311     if test $work_choose -eq 0; then
312         manage_work_s
313         #未知异常退出程序
314         exit 1
315     #判断输入是否合法
316     else if test $work_choose -le $(ls $s_dir | wc -l); then
317         #任务序号对应的任务名
318         work_name=$(ls $s_dir | sort | sed -n $work_choose"p")
319         echo -e "\n请输入下载地址 (输入cancel可取消) : "
320         read downloadDir
321         #判断输入是否为cancel
322         if test "cancel" = $downloadDir; then
323             manage_course_s
324             #未知异常退出程序
325             exit 1
326         fi
327         #将作业目录复制到所输入的目录下
328         cp -r $s_dir/$work_name $downloadDir
329         echo -e "\n\033[32m下载成功! \033[0m"
330         manage_course_s
331         #未知异常退出程序
332         exit 1
333     else
334         echo -e "\033[31m输入非法! \033[0m"
```

```
335     choose_course_s
336     #未知异常退出程序
337     exit 1
338     fi
339   fi
340 }
341
342 #修改学生密码
343 change_passwd_s()
344 {
345   #身份验证
346   login_student
347   #输入不回显
348   stty -echo
349   echo -n "请输入新密码: "
350   read passwd
351   echo ""
352   echo -n "请再输入一遍: "
353   read passwd_again
354   echo ""
355   #检验两次输入是否一致
356   if test $passwd != $passwd_again; then
357     echo -e "\033[31m两次输入不一致! \033[0m"
358     mode_admin
359     #未知异常退出程序
360     exit 1
361   fi
362   #对密码明文进行加密
363   passwd_md5=$(echo $passwd | md5)
364   #删除密码文件中的旧行
365   sed -i "" '/'$s_id'/d' data/passwd
366   #向密码文件中写入新行
367   echo "$s" $s_id $passwd_md5 >> data/passwd
368   #撤销输入不回显
369   stty echo
370   #修改成功提示, 显示为绿色
371   echo -e "\n\033[32m修改成功! \033[0m"
372   mode_student
373   #未知异常退出程序
374   exit 1
375 }
```

实验结果:

```
→ lab2-4 bash main.sh
您的身份：
1 学生
2 教师
3 管理员
q 退出系统
3
请输入管理员登录密码: ?
```

图 13: 输入密码时屏幕不回显

```
→ lab2-4 bash main.sh
您的身份：
1 学生
2 教师
3 管理员
q 退出系统
3
请输入管理员登录密码:
密码错误!
请输入管理员登录密码:
欢迎admin!
```

图 14: 管理员登录成功

```
请选择操作：
1 管理教师信息
2 管理课程信息
3 修改用户密码
q 退出系统
1

请选择操作：
1 查询教师信息
2 增加教师信息
3 删除教师信息
4 修改教师信息
5 绑定课程信息
6 取消课程绑定
r 返回上级菜单
q 退出系统
```

图 15: 教师信息管理菜单

```
请选择查询方式：
1 按教师工号查询
2 按教师姓名查询
3 显示所有教师
r 返回上级菜单
q 退出系统
1

请输入教师工号：
0095129

所查询的教师为：
0095129 周黎明
```

图 16: 按工号查询教师

```
请输入教师姓名：  
沈.*  
  
所查询的教师为：  
0000091 沈莉萍  
0007295 沈静  
0015109 沈林维  
0087354 沈国平  
0091384 沈晓红  
0091385 沈莉萍  
0094432 沈岳良  
0097075 沈宏  
0098444 沈筱筠
```

图 17: 按姓名查询教师 (正则表达式)

```
3  
  
所查询的教师为：  
0000042 郑芳  
0000057 李宁  
0000091 沈莉萍  
0000123 丁世平  
0000139 陈学群  
0000143 陈志坚  
0000157 鲁定辉  
0000232 曹倩  
0001012 金洪英  
0001242 高志华  
0002034 董萍
```

图 18: 显示系统内所有教师部分截图

```
请输入教师工号 (输入 cancel 可取消)：  
0095129  
该工号已存在！
```

图 19: 增加教师 (工号冲突)

```
请输入教师工号 (输入 cancel 可取消)：  
0095124  
  
请输入教师姓名 (输入 cancel 可取消)：  
翁恺  
  
增加教师成功！
```

图 20: 增加教师

```
请输入教师工号 (输入 cancel 可取消)：  
0003320  
  
所要删除的教师为：  
0003320 刘志军  
确认删除吗？ y  
  
删除成功！
```

图 21: 删除教师

```
请输入教师工号 (输入 cancel 可取消) :  
0600099  
  
所要修改的教师为:  
0600099 助教师  
  
请输入修改后的教师姓名 (输入 cancel 可取消) :  
助教  
  
修改成功!
```

图 22: 修改教师

```
请输入教师工号 (输入 cancel 可取消) :  
0095129  
  
请输入课程代号 (输入 cancel 可取消) :  
01192101  
该教师已绑定该课程!
```

图 23: 绑定课程 (冲突)

```
请输入教师工号 (输入 cancel 可取消) :  
0095129  
  
请输入课程代号 (输入 cancel 可取消) :  
01102081  
  
教师与课程成功绑定!
```

图 24: 绑定课程

```
请输入教师工号 (输入 cancel 可取消) :  
0095129  
  
请输入课程代号 (输入 cancel 可取消) :  
01102081  
  
教师与课程解除绑定!
```

图 25: 解除绑定

```
请输入课程代号:  
211.*  
  
所查询的课程为:  
21121330 操作系统  
21121320 图像信息处理  
21121340 计算机网络  
21188142 课程综合实践 II  
21191062 计算机体系结构
```

图 26: 按课程号查询课程 (正则表达式)

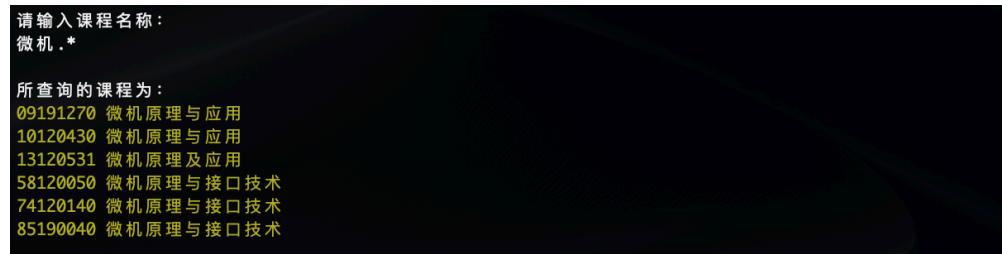


图 27: 按课程名称查询课程 (正则表达式)

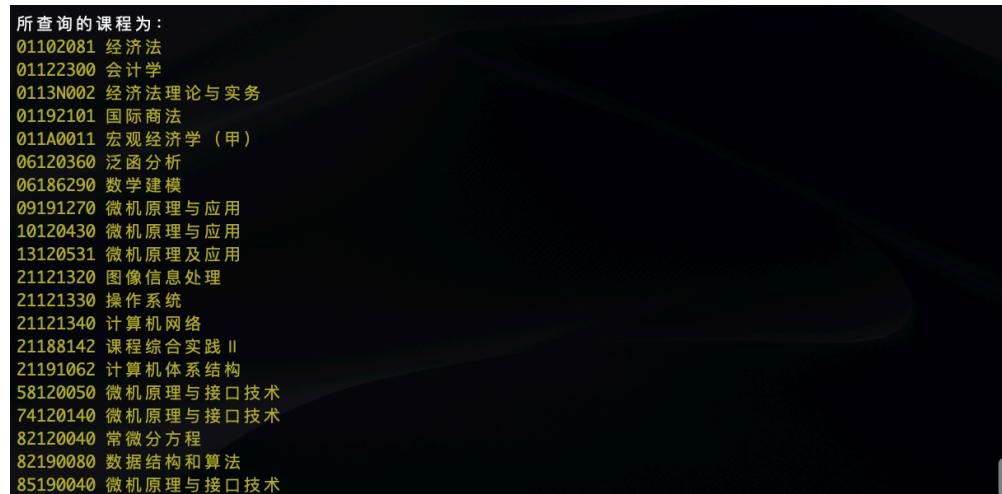


图 28: 显示所有课程

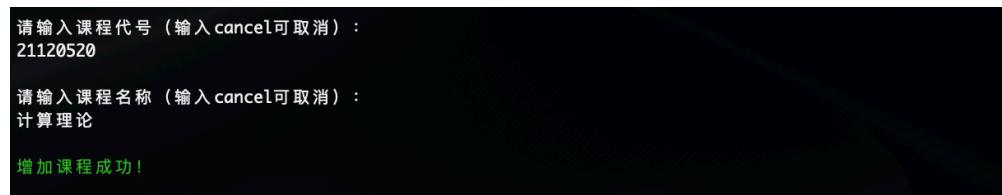


图 29: 增加课程

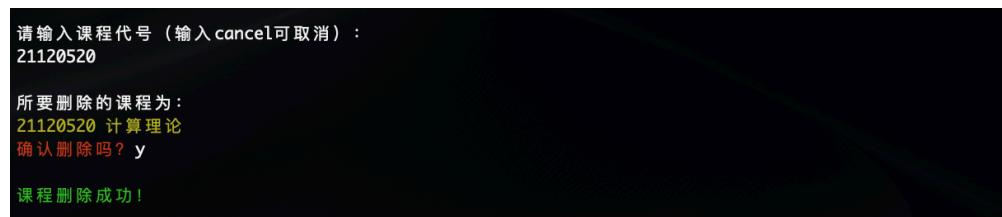


图 30: 删除课程

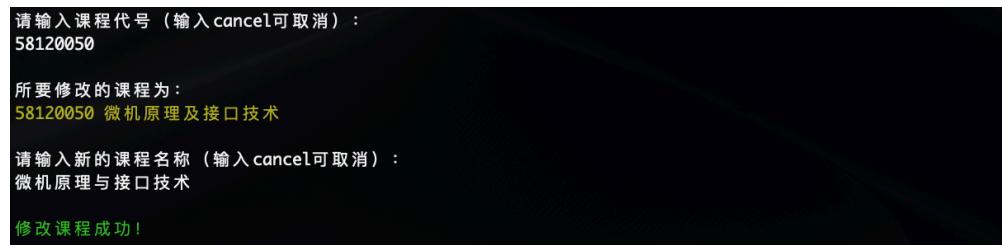


图 31: 修改课程

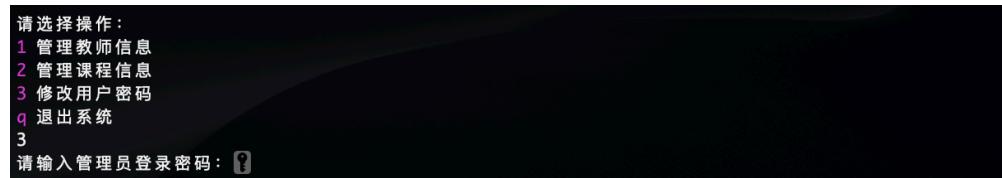


图 32: 修改密码（验证身份）



图 33: 修改密码



图 34: 教师身份登录



图 35: 选择管理的课程



图 36: 管理选课学生菜单



图 37: 查看选课名单



图 38: 查看选课名单 (未选课)



图 39: 添加选课学生



图 40: 删选课学生

```
请输入学生学号 (输入cancel可取消) :  
3170101234  
  
所要修改的学生为:  
3170101234 杨森  
  
请输入修改后的学生姓名 (输入cancel可取消) :  
cancel
```

图 41: 修改学生信息 (已取消)

```
请选择操作:  
1 发布信息  
2 编辑信息  
3 删除信息  
4 显示信息  
r 返回上级目录  
q 退出系统  
1  
请输入课程信息:  
请于10月1日前完成作业1!  
  
信息发布成功!
```

图 42: 发布课程信息

```
所有课程信息如下:  
1 欢迎选课!  
2 请于10月1日前完成作业1!  
  
请选择要编辑的信息:  
2  
请输入修改后的信息:  
请于10月7日前完成作业1!
```

图 43: 编辑课程信息

```
所有课程信息如下:  
1 欢迎选课!  
2 请于10月7日前完成作业1!  
  
请选择要删除的信息:  
2
```

图 44: 删 除课程信息

```
4  
所有课程信息如下:  
1 欢迎选课!
```

图 45: 显示课程信息

```
请选择操作：  
1 布置任务  
2 编辑任务  
3 删除任务  
4 显示任务  
5 完成情况  
r 返回上级目录  
q 退出系统  
1  
  
请输入任务名称（输入cancel可取消）：  
HW2  
  
任务创建成功！
```

图 46: 布置任务

```
请选择要编辑的任务：  
1  
  
请输入新的任务名称：  
lab1
```

图 47: 编辑任务

```
所布置的任务为：  
1 HW2  
2 lab1  
  
请选择要删除的任务：  
2
```

图 48: 删除任务

```
4  
  
所布置的任务为：  
1 lab1
```

图 49: 列出任务

```
所布置的任务为：  
1 lab1  
  
请选择要检查的任务：  
1  
  
完成情况如下：  
3170104848 已提交  
3170105166 未提交  
3170105762 未提交
```

图 50: 查看任务完成情况

```
[→ lab2-4 bash main.sh
您的身份：
1 学生
2 教师
3 管理员
q 退出系统
1
请输入学生学号: 3170104848
[请输入登录密码:
欢迎吴同同学!]
```

图 51: 学生登录

```
请选择操作:
1 查看课程信息
2 修改用户密码
q 退出系统
1

请选择课程:
0 返回
1 0113N002 经济法理论与实务
```

图 52: 选择要管理的课程

```
请选择操作:
1 查看通知
2 管理作业
r 返回上级
q 退出系统
1

所有课程信息如下:
1 欢迎选课!
```

图 53: 查看课程通知

```
请选择操作:
1 查看作业
2 提交作业
3 重交作业
4 检查作业
r 返回上级
q 退出系统
1

该课程的任务为:
1 lab1 已提交
```

图 54: 查看作业

```
该课程的任务为：  
1 lab1 已提交  
  
请选择要提交的任务（输入0返回）：  
1  
  
请输入要提交的文件（输入cancel可取消）：  
student.sh  
  
提交成功！
```

图 55: 提交作业

```
该课程的任务为：  
1 lab1 已提交  
  
请选择要重交的任务（输入0返回）：  
1  
  
请输入要提交的文件（输入cancel可取消）：  
teacher.sh  
  
重交成功！
```

图 56: 重交作业

```
该课程的任务为：  
1 lab1 已提交  
  
请选择要检查的任务（输入0返回）：  
1  
  
请输入下载地址（输入cancel可取消）：  
/Users/wutong/Desktop  
  
下载成功！
```

图 57: 检查作业

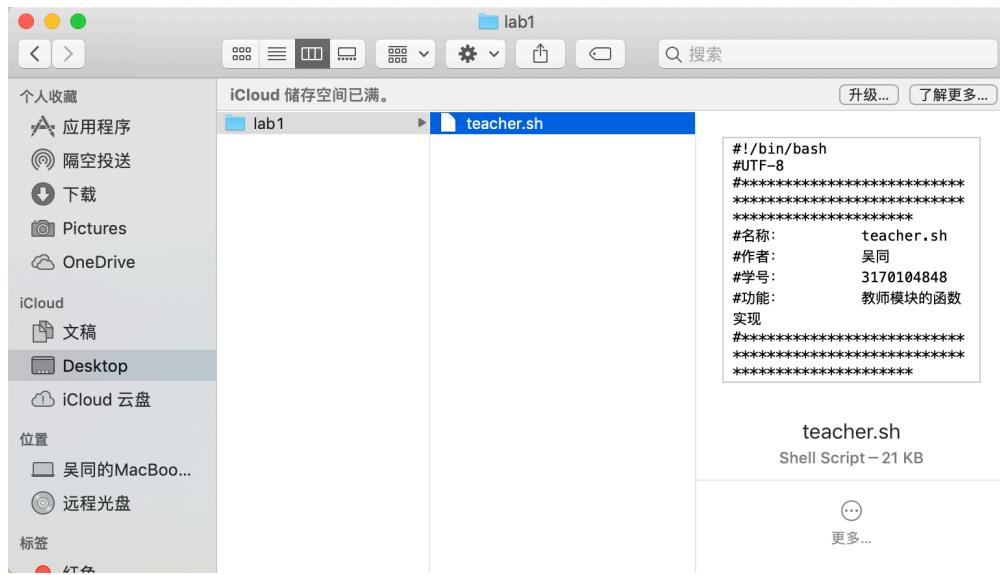


图 58: 已下载的作业

三、讨论与心得

通过本次实验，我初步掌握了 shell 程序设计的一般方法，对 shell 的程序设计思想有所体会。shell 虽然是一门脚本语言，但其接近操作系统底层，有很多与其他脚本语言不同之处。shell 由于有管道和重定向功能，其对文本的处理能力非常强大。但 shell 对数值的处理能力较弱，只支持整数的计算，且编写复杂，效率较低，在编程的过程中应减少使用。

在编写四个程序的过程中，我注重交互的友好，尽可能使输出信息充分详细，并对输出文本中的不同内容进行了颜色的区分。在第四个程序输入密码时，遵循 Unix 的惯例，进行了不回显键盘输入的设置。

在第一个小程序编写的过程中，出现了 /etc 目录下无法显示总字节数的问题，原因是该目录下没有普通文件。经过思考后，我在循环外部声明 sum 变量并赋初值为 0，问题得到解决。另外又出现了 /bin 目录下可执行文件个数显示错误的问题，经调试发现是没有将 s 权限算在内。这些都是微小的 bug，但却需要测试时找出足够的特殊样例并仔细观察才能发现。

在第二个小程序编写的过程中，我开始仿照其他编程语言的方式，将逐个数据按整数进行处理。但这样做运算速度非常慢，后来我将所有字符均以文本形式处理，进行行列间变换，对文本进行排序，运算效率显著提升。

在第三个小程序编写的过程中，我在循环的每次过程中，将字符串进行切割，取需要的部分。之所以没有使用数组，是为了使 shell 程序尽可能由多个小工具组合而成，减少使用变量的原则。由于处理的没有过于庞大的数据，这样做可以在对性能无明显影响的前提下，使程序更加简洁，增加可移植性。

在作业管理系统编写的过程中，我真正体会到了 shell 编程较其他语言编程的一些与众不同之处。shell 的最主要功能是完成用户与操作系统内核的交互。虽然 shell 是脚本语言，但如果强行遵守很多高级语言程序设计的规范，反倒是弄巧成拙。shell 程序设计中，应尽量使程序的结构简单，多运用命令的组合，减少变量的使用。我在测试过程中使用超过一万条的学生数据，查询效率较高，体现了 shell 对文本强大的处理能力。

本次实验是我第一次编写 shell 程序，程序中的很多写法还有待改进，shell 的程序设计思想还需要更深入的体会。还是要多加练习，多阅读优秀的开源代码，提高程序设计的水平。