
浙江大学实验报告

课程名称: Linux 程序设计 实验类型: 综合型

实验项目名称: 实验 2: shell 程序设计

学生姓名: 赵彬彬 专业: 计算机科学与技术 学号: 3140102395

电子邮件地址: 313220146@qq.com

实验日期: 2016 年 7 月 20 日

一、实验目的和要求

1. 学习 Bourne shell 的 shell 脚本的基本概念
2. 学会 shell 程序如何执行
3. 学习 shell 变量的概念及使用方法
4. 学会 Bourne shell 脚本的命令行参数是如何传给 shell 程序的
5. 学习 Bourne shell 脚本命令替换的概念
6. 学习编写 Bourne shell 脚本的一些基本原则
7. 通过写简短的脚本, 学会编写 Bourne shell 脚本程序的方法

二、主要仪器设备

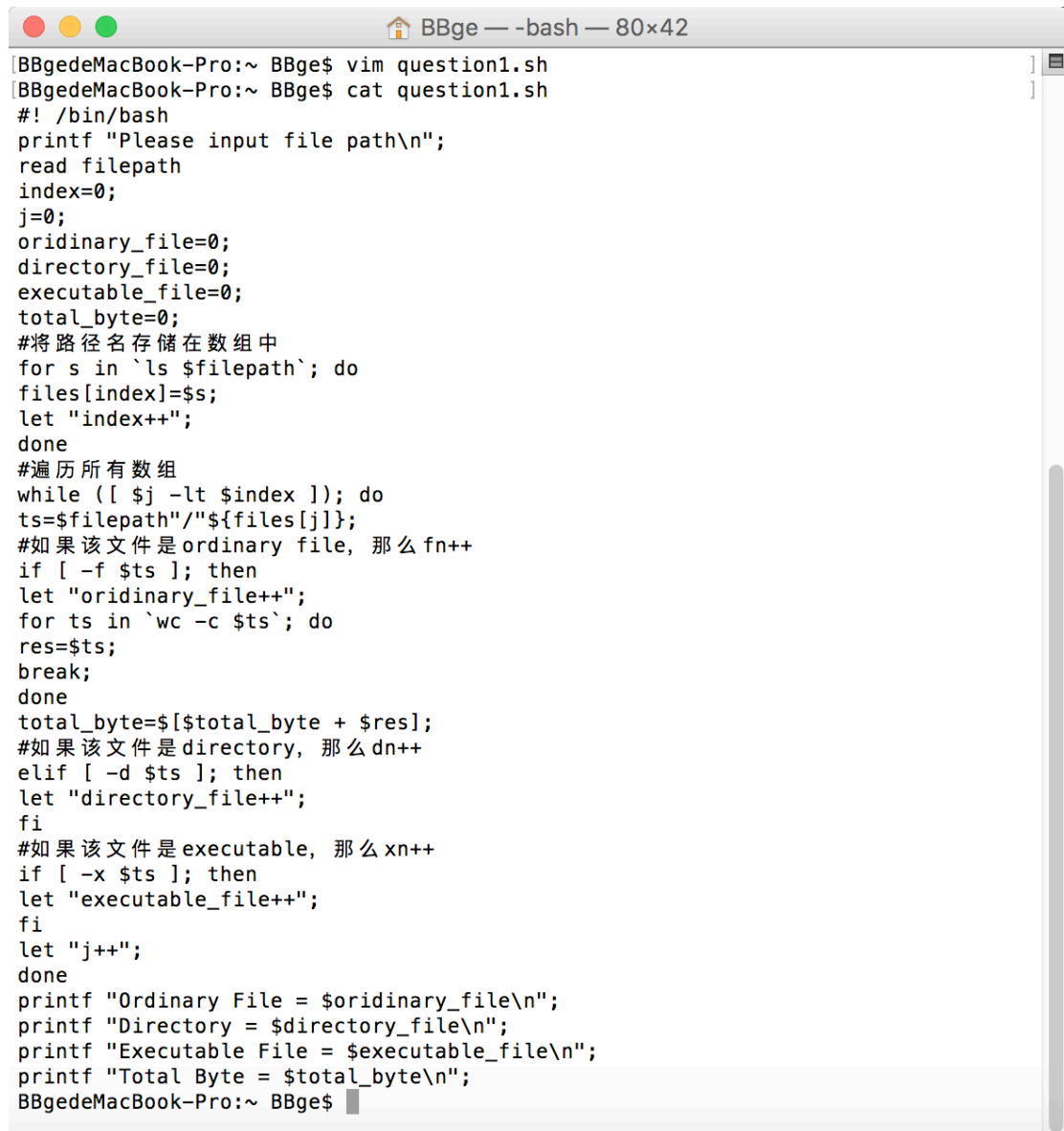
计算机配置: 处理器 2.7GHz Intel Core i5, 内存 8GB 1867 MHz DDR3

操作系统环境: OS X El Capitan 10.11.3 版本

LINUX 版本: Mac OS X El Capitan 10.11.3 (类 UNIX)

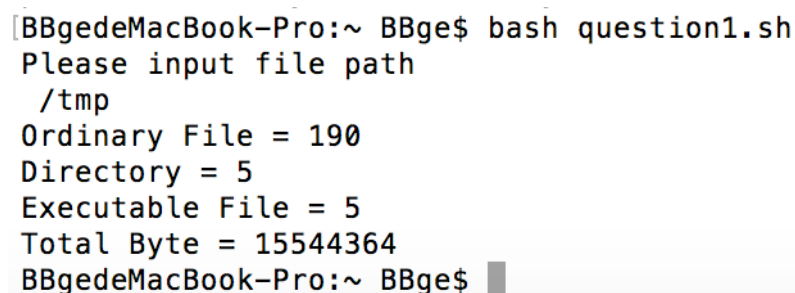
二、 实验内容与结果

1. 统计指定目录下的普通文件、子目录及可执行文件的数目，统计指定目录下所有文件字节数总和，目录的路径名字由参数传入。



```
BBgedeMacBook-Pro:~ BBge$ vim question1.sh
BBgedeMacBook-Pro:~ BBge$ cat question1.sh
#!/bin/bash
printf "Please input file path\n";
read filepath
index=0;
j=0;
oridinary_file=0;
directory_file=0;
executable_file=0;
total_byte=0;
#将路径名存储在数组中
for s in `ls $filepath`; do
files[index]=$s;
let "index++";
done
#遍历所有数组
while ([ $j -lt $index ]); do
ts=$filepath/"${files[j]}";
#如果该文件是 ordinary file, 那么 fn++
if [ -f $ts ]; then
let "oridinary_file++";
for ts in `wc -c $ts`; do
res=$ts;
break;
done
total_byte=$((total_byte + $res));
#如果该文件是 directory, 那么 dn++
elif [ -d $ts ]; then
let "directory_file++";
fi
#如果该文件是 executable, 那么 xn++
if [ -x $ts ]; then
let "executable_file++";
fi
let "j++";
done
printf "Ordinary File = $oridinary_file\n";
printf "Directory = $directory_file\n";
printf "Executable File = $executable_file\n";
printf "Total Byte = $total_byte\n";
BBgedeMacBook-Pro:~ BBge$
```

图 1 代码



```
BBgedeMacBook-Pro:~ BBge$ bash question1.sh
Please input file path
/tmp
Ordinary File = 190
Directory = 5
Executable File = 5
Total Byte = 15544364
BBgedeMacBook-Pro:~ BBge$
```

图 2 执行结果

2. 用 bash 编写程序，该程序从键盘输入的 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。

因为从键盘输入 100 个整数太过麻烦，我选择用生成 100 个随机数的方式来代替。

```
BBge — -bash — 87x48
[r-126-97-25-172:~ BBge$ vim question2.sh
[r-126-97-25-172:~ BBge$ cat question2.sh
#随机生成100个数存储到数组中，替代手工输入100个数
for((i=0;i<100;++i)); do
    #random 0 9999;
    array[$i]=$((RANDOM%10000));
    printf "%d " ${array[$i]};
done
printf "\n";

i=0;
#定义一个max和min用于之后寻找最大最小值
max=0;
min=999999999;
#定义sum用来求和，用average来获得平均值
sum=0;
average=0;
while([ $i -lt ${#array[@]} ]); #将数组长度赋值给i，也可以直接用100
do
    if [ $min -gt ${array[$i]} ]; #如果min比当前值大，那么将当前值赋值给min
    then
        min=${array[$i]};
    fi
    if [ $max -lt ${array[$i]} ]; #如果max比当前值小，那么将当前值赋值给max
    then
        max=${array[$i]};
    fi
    let "i++";
done

#排序
for((i=1;i< ${#array[@]};i++));
do
    k=${array[$i]};
    for((j=i-1;j>=0;j--));
    do
        if [ ${array[$j]} -gt $k ];
        then
            array[${j} + 1 ]=${array[$j]};
            array[$j]=$k;
        fi
    done
done

#求平均值
for((i=0;i<${#array[@]};i++));
do
    sum=$((sum+${array[$i]}));
done

#output the res:MAX,MIN,SORT
echo "最大值: $max";
echo "最小值: $min";
echo "总和: $sum";
awk 'BEGIN{printf "平均值: %.2f\n",('$sum'/'${#array[@]}')}' ;
echo "排序后的结果(从小到大) :";
i=0;
while [[ $i -lt ${#array[@]} ]];
do
    printf "%d " ${array[$i]};
    let "i++";
done
printf "\n";
r-126-97-25-172:~ BBge$
```

图 3 代码

```

r-126-97-25-172:~ BBge$ bash question2.sh
8229 6494 4380 5595 999 3897 4826 6294 5004 8576 3478 9037 1372 5991 3947 3370 3030 907
7 732 5553 2244 179 5746 1228 5567 8755 6723 2855 210 0 2518 555 6736 4650 2185 763 976
4 6798 2398 3920 9628 7863 7354 1317 5861 9511 2300 2769 549 7684 5917 7560 907 995 555
9697 6614 100 6165 9925 2520 1939 599 4477 6265 9274 3987 6824 7866 1143 9251 9214 206
4 625 3057 8376 1533 169 2210 652 2398 5811 3478 4345 598 2657 5841 4358 7773 1460 1015
3067 920 7394 9 5453 9863 4886 930 7802
最大值：9925
最小值：0
总和：439049
平均值：4390.49
排序后的结果(从小到大)：
0 9 100 169 179 210 549 555 555 598 599 625 652 732 763 907 920 930 995 999 1015 1143 1
228 1317 1372 1460 1533 1939 2064 2185 2210 2244 2300 2398 2398 2518 2520 2657 2769 285
5 3030 3057 3067 3370 3478 3478 3897 3920 3947 3987 4345 4358 4380 4477 4650 4826 4886
5004 5453 5553 5567 5595 5746 5811 5841 5861 5917 5991 6165 6265 6294 6494 6614 6723 67
36 6798 6824 7354 7394 7560 7684 7773 7802 7863 7866 8229 8376 8576 8755 9037 9077 9214
9251 9274 9511 9628 9697 9764 9863 9925
r-126-97-25-172:~ BBge$

```

图 4 执行结果

3. 编写一个 shell 脚本，输入一个字符串，忽略（删除）非字母后，检测该字符串是否为回文(palindrome)。对于一个字符串，如果从前向后读和从后向前读都是同一个字符串，则称之为回文串。例如，单词“mom”，“dad”和“noon”都是回文串。

```

r-126-97-25-172:~ BBge$ vim question3.sh
r-126-97-25-172:~ BBge$ cat question3.sh
echo 请输入一个字符串：
read str_1;
#除去字符串中除了'a' - 'z' 和 'A' - 'Z'之外的所有字符
str_2=`echo -n $str_1|tr -d -c 'a-zA-Z'`;
echo 经过转化后的字符串：$str_2;
length=`echo -n $str_2|wc -c`;
mid=$((length/2));
i=1;
#判断是否回文
while [ $i -le $mid ]; do
    #character_1代表从第一个字符开始
    character_1=`echo -n $str_2|cut -c $i`;
    #character_2代表从第二个字符开始
    character_2=`echo -n $str_2|cut -c $length`;
    if [ $character_1 != $character_2 ]; then
        echo "$str_2 不是一个回文串";
        break;
    fi
    i=$((i+1));
    length=$((length-1));
done
if [ $character_1 = $character_2 ];then
    echo "$str_2 是一个回文串"
fi

```

图 5 代码

```

r-126-97-25-172:~ BBge$ bash question3.sh
请输入一个字符串：
BBge666egBB
经过转化后的字符串：BBgeegBB
BBgeegBB 是一个回文串
r-126-97-25-172:~ BBge$ bash question3.sh
请输入一个字符串：
BBeg123BB
经过转化后的字符串：BBegBB
BBegBB 不是一个回文串

```

图 6 实验结果

4. 用 bash 编写程序，实现一个简单的**作业管理系统**。用文件形式实现数据的存储。系统至少具备以下的基本功能：

系统中根据不同的权限分为三类用户：管理员、教师、学生，简要说明如下：

- 管理员：

创建、修改、删除、显示（list）教师帐号；教师帐户包括教师工号、教师姓名，教师用户以教师工号登录。

创建、修改、删除课程；绑定（包括添加、删除）课程与教师用户。课程名称以简单的中文或英文命名。

- 教师：

对某门课程，创建或导入、修改、删除学生帐户，根据学号查找学生帐号；学生帐号的基本信息包括学号和姓名，学生使用学号登录。

发布课程信息。包括新建、编辑、删除、显示（list）课程信息等功能。

布置作业或实验。包括新建、编辑、删除、显示（list）作业或实验等功能。

查找、打印所有学生的完成作业情况。

- 学生：

在教师添加学生账户后，学生就可以登录系统，并完成作业和实验。

基本功能：新建、编辑作业或实验功能；查询作业或实验完成情况。

该实验在 mac 下的 bash 完成，因为没有找到 mac 下 dialog 的安装包，因此没有使用图形界面，只是命令行界面。

运行程序：

```
Assignment Manager — -bash — 80x24
Last login: Mon Aug  1 19:29:07 on ttys000
BBgedeMacBook-Pro:~ BBge$ cd Documents/linux/assignment_2/Assignment\ Manager/
BBgedeMacBook-Pro:Assignment Manager BBge$ bash main.sh
```

主界面如下：

```
Assignment Manager — bash main.sh — 80x24
#####
#      欢迎使用作业管理系统      #
#      作者:赵彬彬                #
#      学号:3140102395            #
#      version:1.2                #
#####
      请选择登录用户
      1 — 管理员
      2 — 教师
      3 — 学生
      0 — 退出
```

管理员登录界面：

```
Assignment Manager — bash ▾ bash main.sh — 80×24

#####
#      欢迎使用作业管理系统      #
#####
管理员你好，请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
□
```

创建教师账号：

```
Assignment Manager — sleep ▾ bash main.sh — 80×24

#####
管理员你好，请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
1
你将要创建一个新的教师账号
请输入教师编号：
003
请输入教师名字：
gao yunjun
Name: gaoyunjun
ID: 003
是否创建账号？(y/n)
y
OK
█
```

删除教师账号:

```
Assignment Manager — sleep ▸ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
2
你将要删除一个教师账号
请输入教师编号:
003
ID: 003
是否删除账号? (y/n)
y
OK
█
```

列出所有教师账号:

```
Assignment Manager — bash ▸ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
3
老师编号 老师名字
-----
001 chenyue
002 wengkai
-----
输入q退出
█
```

添加一门课程:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
4
你将要添加一门新的课程
请输入课程编号:
002
请输入课程名字:
ads
Name: ads
ID: 002
是否创建课程? (y/n)
y
OK
█
```

删除一门课程:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
5
你将要删除一门课程
请输入课程编号:
002
课程编号: 002
是否删除该课程? (y/n)
y
OK
█
```


列出所有课程:

```
Assignment Manager — bash ▾ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
6
课程编号 课程名字
-----
001 ds
-----
输入q退出
█
```

绑定老师和课程:

```
Assignment Manager — sleep ▾ bash main.sh — 80×24
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
7
你将要绑定老师和课程
请输入课程编号:
001
001
课程编号有效!
请输入教师编号:
002
002
教师编号有效!
课程编号: 001
教师编号: 002
是否绑定? (y/n)
y
OK
█
```

将老师和课程解绑:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
8
你将要解绑一门课程
请输入课程编号:
001
Input teacher ID:
001
课程编号: 001
教师编号: 001
解绑? (y/n)
y
OK
```

列出绑定的老师和课程:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
管理员你好, 请根据指示输入编号
1 — 创建教师账号
2 — 删除教师账号
3 — 列出所有教师账号
4 — 添加一门课程
5 — 删除一门课程
6 — 列出所有课程
7 — 绑定老师和课程
8 — 将老师和课程解绑
9 — 列出绑定的老师和课程
0 — 退出
9
课程编号 教师编号
-----
001 002
-----
输入 q 退出
```

教师界面:

```
Assignment Manager — bash ▸ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
█
```

修改课程信息:

```
Assignment Manager — bash ▸ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
1
当前课程信息:
教师编号  课程编号
002 002
按任意键继续

请输入新的课程信息(教师编号和课程编号):
002 001█
```

添加学生:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
2
你将要添加一个学生账号
请输入学生编号:
st3
请输入学生名字:
Zhao Binbin
学生姓名: ZhaoBinbin
学生编号: st3
是否创建账号? (y/n)
y
OK
█
```

删除学生:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
# 欢迎使用作业管理系统 #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
3
你将要删除一个学生账号
请输入学生编号:
st3
学生编号: st3
是否删除账号? (y/n)
y
OK
█
```

列出所有学生:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
4
学生编号 学生名字
-----
st1 bbge
st2 Refere
-----
输入q退出
█
```

添加作业:

```
Assignment Manager — sleep ◀ bash main.sh — 80×24
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
5
你将要添加一份新的作业
请输入作业编号:
homework2
请输入作业名字:
pta1111
作业名字: pta1111
作业编号: homework2
是否创建该作业? (y/n)
y
OK
█
```

列出作业:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
7
作业编号 作业名字
-----
homework1 pta1000
homework2 pta1111
-----
输入q退出
█
```

列出已上传的作业:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
8
作业编号_学生编号:
homework1_st1 homework2_st1
按任意键继续
█
```

查看学生作业:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
老师您好, 请根据指示输入编号
1 — 修改课程信息
2 — 添加学生
3 — 删除学生
4 — 列出所有学生
5 — 添加作业
6 — 删除作业
7 — 列出作业
8 — 列出已上传作业
9 — 查看学生的作业
0 — 退出
9
请输入作业编号_学生编号查看:
example : homework1_st1
homework1_st1
#include<stdio.h>
int main()
{
    printf("hello world");
    return 0;
}
按任意键继续
█
```

学生登录界面:

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
学生你好, 请根据指示输入编号
1 — 列出作业
2 — 编辑作业
0 — 退出
█
```

列出作业：

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
学生你好，请根据指示输入编号
1 — 列出作业
2 — 编辑作业
0 — 退出
1
作业编号 作业名字
-----
homework1 pta1000
homework2 pta1111
-----
输入 q退出
█
```

编辑作业：

```
Assignment Manager — bash ◀ bash main.sh — 80×24
#####
#      欢迎使用作业管理系统      #
#####
学生你好，请根据指示输入编号
1 — 列出作业
2 — 编辑作业
0 — 退出
2
请输入作业编号：
homework1
homework1
作业编号有效！
当前作业答案：
print "hello Python"
请输入新的作业答案：
print "hello Script"
█
```

四、讨论、心得

本次实验主要是 shell 编程，其中用的 shell 是 bash 版本。本来想在 Ubuntu 系统上完成，但因为自己这台电脑开虚拟机有点卡，并且 mac 本身也自带 bash，所以选择在 mac 上完成本次实验。一开始打算将作业管理系统做成图形界面的，百度了解到 dialog 可以将 shell 程序做成图形界面，但没找到 dialog for mac，因此只能做成命令行界面。

shell 编程对自己来说可以说是零基础，和 C 语言在许多方面有很大的不同。简单的一个空格都有可能造成程序结果的错误。本次实验最难的是 task4，做这个实验的时候百度了很多资料，也参考了一些代码，虽然仍有一些 bug，但基本功能都已经实现。task4 用到了大量的 bash 命令，能够非常方便的显示文件或文件夹中的内容，体会到了 bash 这种 shell 的便捷。

完成本次实验的过程中，自己遇到了许多困难，感觉到了自己能力的不足，但最后完成这次实验的时候，还是很有成就感的。通过这次实验，自己对 Linux 这个系统也有了更多的了解。