

**程序设计实践实验报告**

**实 验（四）**

题 目 基于文件的信息管理系统

专 业 计算机科学与技术

学　　 号 2021113679

班　　 级 2103101

学 生 吴嘉阳

指 导 教 师 吴锐

实 验 地 点 管理楼G709

实 验 日 期 2022.8.31

**计算学部**

**目 录**

[第1章 实验基本信息 - 2 -](#_Toc112495298)

[1.1 实验目的 - 2 -](#_Toc112495299)

[1.2 实验环境与工具 - 2 -](#_Toc112495300)

[1.2.1 硬件环境 - 2 -](#_Toc112495301)

[1.2.2 软件环境 - 2 -](#_Toc112495302)

[1.2.3 开发工具 - 2 -](#_Toc112495303)

[第2章 小型成绩管理系统 - 3 -](#_Toc112495304)

[2.1 小型成绩管理系统总体设计（4分） - 3 -](#_Toc112495305)

[2.2 主要功能模块设计（25分） - 3 -](#_Toc112495306)

[2.2.1成绩录入（3分） - 3 -](#_Toc112495307)

[2.2.2成绩查询（3分） - 3 -](#_Toc112495308)

[2.2.3成绩修改（3分） - 3 -](#_Toc112495309)

[2.2.4成绩删除（4分） - 4 -](#_Toc112495310)

[2.2.5成绩计算（4分） - 4 -](#_Toc112495311)

[2.2.6成绩排序（4分） - 4 -](#_Toc112495312)

[2.2.7成绩统计分析（4分） - 5 -](#_Toc112495313)

[第3章 跨平台调试 - 6 -](#_Toc112495314)

[3.1 简述建立树莓派平台的过程并给出成功访问的截图（4分） - 6 -](#_Toc112495315)

[3.2 将成绩管理系统移植到ARM架构的树莓派平台（6分） - 6 -](#_Toc112495316)

# 第1章 实验基本信息

**(本章满分1分)**

## 实验目的

* 掌握结构化数据排序方法及文件相关操作
* 掌握共享库的建立与使用
* 掌握模块化程序设计和增量测试方法
* （选）掌握跨平台程序编写、调试方法

## 1.2 实验环境与工具

### 1.2.1 硬件环境

Intel 10750H x86\_64 树莓派 arm

### 1.2.2 软件环境

WSL2 Ubuntu20.04 64位

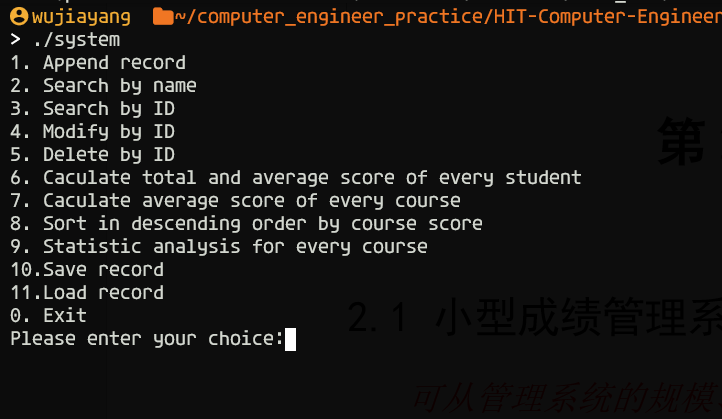
### 1.2.3 开发工具

Gcc vim

# 第2章 小型成绩管理系统

**(本章满分29分)**

## 2.1 小型成绩管理系统总体设计（4分）



**主要功能：**

（1）通过键盘录入每个学生的学号、姓名和各科成绩；

（2）按姓名查询学生成绩，并显示；

（3）按学号查询学生成绩，并显示；

（4）修改某个学生成绩；

（5）删除某个学生成绩；

（6）计算每个学生总成绩和平均成绩，并显示；

（7）计算班级各科平均成绩，并显示；

（8）按某一科成绩由高到低排序，并显示（包括其他科目成绩）；

（9）按优秀（90~100）、良好（80~89）、中等（70~79）、及格（60~69）、不及格（0~59）五个类别，对每科成绩分别统计每个类别的人数及所占的百分比，并显示结果；

（10）导出，以文本或二进制文件形式输出1、4、5的结果，文本文件每行一个记录；

（11）导入，从文本或二进制文件读入所有学生成绩信息；

（12）退出。

## 2.2 主要功能模块设计（25分）

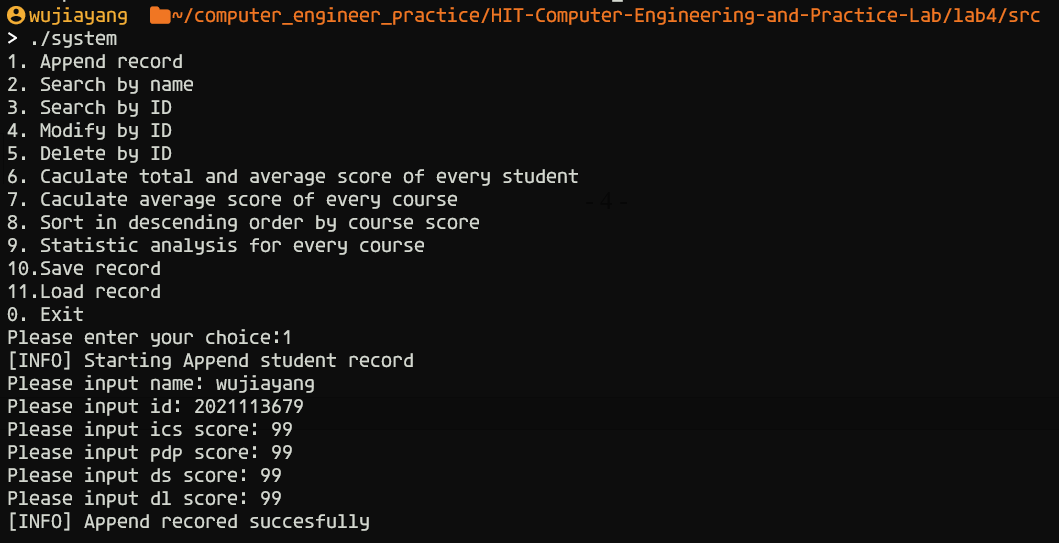
### 2.2.1成绩录入（3分）

从键盘输入至少5名学生的成绩，必须包括本人的信息。

相关函数代码与说明：

1. **void** AppendRecord(STU\* student\_record, **int**\* num){
2. printf("[INFO] Starting Append student record\n");
3. **if** (\*num >= 44){
4. printf("[ERROR] The number of students has reached maximum\n");
5. **return**;
6. }
7. printf("Please input name: ");
8. scanf("%s", (student\_record + \*num)-> stu\_name );
9. printf("Please input id: ");
10. scanf("%lu", &(student\_record + \*num) -> stu\_id);
11. printf("Please input ics score: ");
12. scanf("%f", &(student\_record + \*num)->score.ics);
13. printf("Please input pdp score: ");
14. scanf("%f", &(student\_record + \*num)->score.pdp);
15. printf("Please input ds score: ");
16. scanf("%f", &(student\_record + \*num)->score.ds);
17. printf("Please input dl score: ");
18. scanf("%f", &(student\_record + \*num)->score.dl);
20. (\*num) ++;
21. printf("[INFO] Append recored succesfully\n");
22. **return** ;
23. }

测试结果截图：



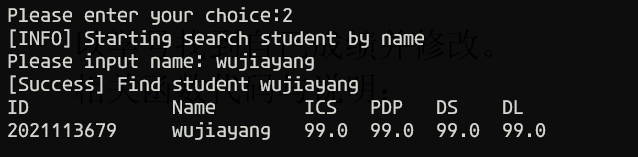
### 2.2.2成绩查询（3分）

以姓名查询本人的成绩。

相关函数代码与说明：

1. **void** SearchByName(STU\* student\_record, **int** num){
2. printf("[INFO] Starting search student by name\n");
3. **char** input\_name[32];
4. **int** i = 0;
5. printf("Please input name: ");
6. scanf("%s", input\_name);
7. **for** (i = 0; i < num; i++) {
8. **if**(strcmp(input\_name, (student\_record + i)->stu\_name) == 0) {
9. printf("[Success] Find student %s\n", input\_name);
10. printf("ID             Name        ICS   PDP   DS    DL\n");
11. printf("%-14lu %-10s  %-.1f  %-.1f  %-.1f  %-.1f\n", (student\_record + i)->stu\_id,\
12. (student\_record + i)->stu\_name,\
13. (student\_record + i)->score.ics,\
14. (student\_record + i)->score.pdp,\
15. (student\_record + i)->score.ds,\
16. (student\_record + i)->score.dl);
17. **return**;
18. }
19. }
20. printf("[INFO] Can't find student %s", input\_name);
21. **return**;
22. }

测试结果截图：



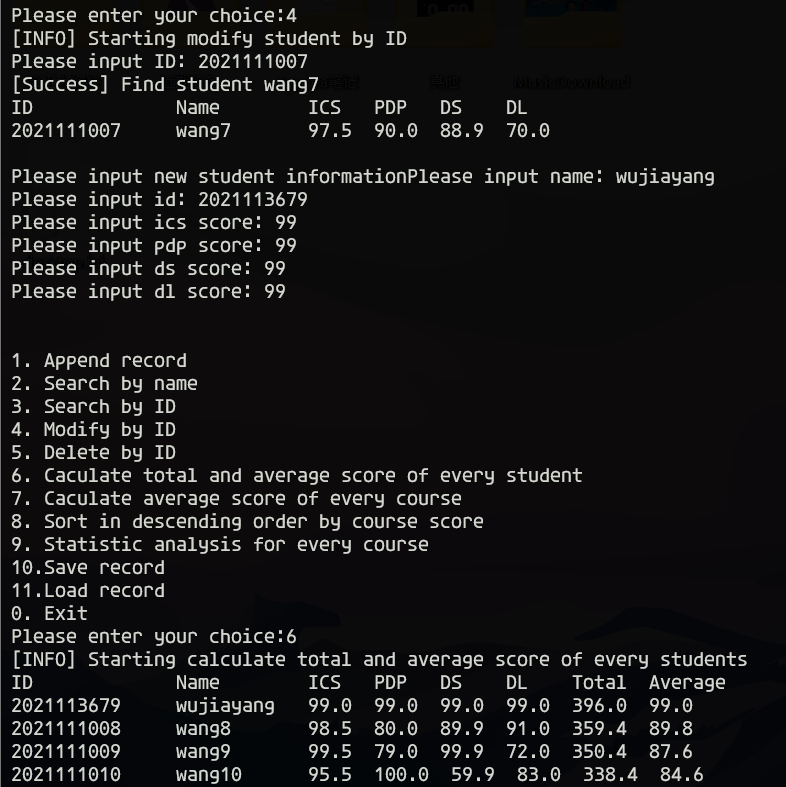
### 2.2.3成绩修改（3分）

以学号找到自己成绩并修改。

相关函数代码与说明：

1. **void** ModifyByID(STU\* student\_record, **int** num){
2. printf("[INFO] Starting modify student by ID\n");
3. unsigned **long** input\_id = 0;
4. **int** i = 0;
5. printf("Please input ID: ");
6. scanf("%lu", &input\_id);
7. **for** (i = 0; i < num; i++) {
8. **if**((student\_record + i)->stu\_id == input\_id) {
9. printf("[Success] Find student %s\n", (student\_record + i) -> stu\_name);
10. printf("ID             Name        ICS   PDP   DS    DL\n");
11. printf("%-14lu %-10s  %-.1f  %-.1f  %-.1f  %-.1f\n", (student\_record + i)->stu\_id,\
12. (student\_record + i)->stu\_name,\
13. (student\_record + i)->score.ics,\
14. (student\_record + i)->score.pdp,\
15. (student\_record + i)->score.ds,\
16. (student\_record + i)->score.dl);
17. printf("\nPlease input new student information");
19. printf("Please input name: ");
20. scanf("%s", (student\_record + i)-> stu\_name );
21. printf("Please input id: ");
22. scanf("%lu", &(student\_record + i) -> stu\_id);
23. printf("Please input ics score: ");
24. scanf("%f", &(student\_record + i)->score.ics);
25. printf("Please input pdp score: ");
26. scanf("%f", &(student\_record + i)->score.pdp);
27. printf("Please input ds score: ");
28. scanf("%f", &(student\_record + i)->score.ds);
29. printf("Please input dl score: ");
30. scanf("%f", &(student\_record + i)->score.dl);
31. **return**;
32. }
33. }
34. printf("[INFO] Can't find student %s", (student\_record + i)-> stu\_name);
35. **return**;
37. }

测试结果截图：



### 2.2.4成绩删除（4分）

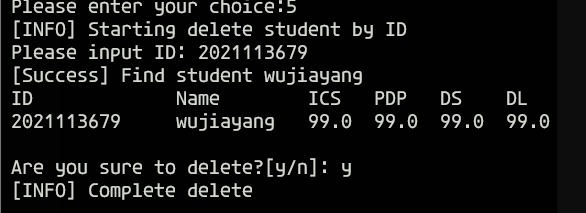
以学号为索引删掉本人成绩记录。

相关函数代码与说明：

1. **void** DeleteByID(STU\* student\_record, **int**\* num){
2. printf("[INFO] Starting delete student by ID\n");
3. unsigned **long** input\_id = 0;
4. **int** i = 0;
5. printf("Please input ID: ");
6. scanf("%lu", &input\_id);
7. **for** (i = 0; i < \*num; i++) {
8. **if**((student\_record + i)->stu\_id == input\_id) {
9. printf("[Success] Find student %s\n", (student\_record + i) -> stu\_name);
10. printf("ID             Name        ICS   PDP   DS    DL\n");
11. printf("%-14lu %-10s  %-.1f  %-.1f  %-.1f  %-.1f\n", (student\_record + i)->stu\_id,\
12. (student\_record + i)->stu\_name,\
13. (student\_record + i)->score.ics,\
14. (student\_record + i)->score.pdp,\
15. (student\_record + i)->score.ds,\
16. (student\_record + i)->score.dl);
17. printf("\nAre you sure to delete?[y/n]: ");
18. **char** judge;
19. judge = getchar();
20. judge = getchar();
21. **if** (judge == 'y') {
22. **for**(**int** j = i; j < (\*num)-1; j++) {
23. strcpy((student\_record + j)->stu\_name, (student\_record + j + 1)->stu\_name);
24. (student\_record + j)->stu\_id = (student\_record + j + 1) -> stu\_id;
25. (student\_record + j)->score = (student\_record + j + 1) -> score;
26. }
27. (\*num) --;
28. printf("[INFO] Complete delete");
29. } **else**{
30. printf("[INFO] Stop delete");
31. }
32. **return**;
33. }
34. }
35. printf("[INFO] Can't find student %s", (student\_record + i)-> stu\_name);
36. **return**;


40. }

测试结果截图：



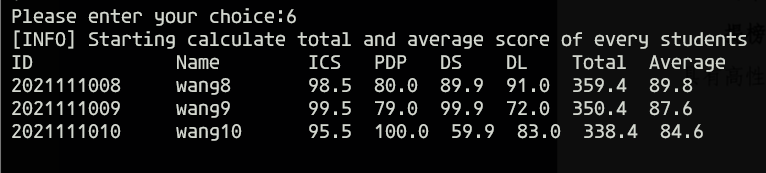
### 2.2.5成绩计算（4分）

计算各科平均成绩并显示。

相关函数代码与说明：

1. **void** CalcStudents(STU\* student\_record, **int** num){
2. printf("[INFO] Starting calculate total and average score of every students\n");
3. printf("ID             Name        ICS   PDP   DS    DL    Total  Average\n");
4. **for**(**int** i = 0; i < num; i++){
5. (student\_record + i)->total = (student\_record + i)->score.ics + \
6. (student\_record + i)->score.pdp + \
7. (student\_record + i)->score.dl + \
8. (student\_record + i)->score.ds;
9. (student\_record + i)->average = (student\_record + i)->total /4;
10. printf("%-14lu %-10s  %-.1f  %-.1f  %-.1f  %-.1f  %-.1f  %-.1f\n", (student\_record + i)->stu\_id,\
11. (student\_record + i)->stu\_name,\
12. (student\_record + i)->score.ics,\
13. (student\_record + i)->score.pdp,\
14. (student\_record + i)->score.ds,\
15. (student\_record + i)->score.dl,\
16. (student\_record + i)->total,\
17. (student\_record + i)->average);
18. }
19. **return**;
20. }

测试结果截图：



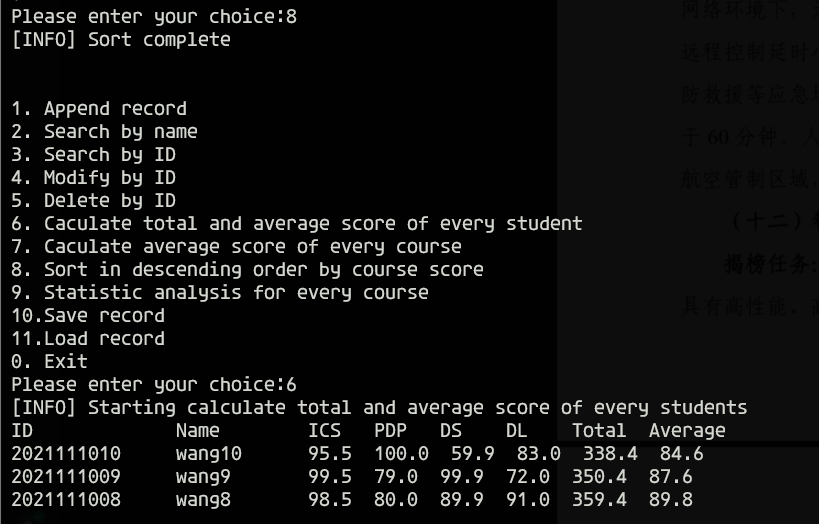
### 2.2.6成绩排序（4分）

任意一科成绩为准进行降序排序。

相关函数代码与说明：

1. **void** SortCourseScore(STU\* student\_record, **int** num){
2. **if**(num == 0 && student\_record->total == 0){
3. printf("[ERROR] Please run function 6 to compute total and average score of each person\n");
4. **return**;
5. }
6. **int** (\*compare)(ElemType\* a, ElemType\* b);
7. compare = &cmp;
8. /\* HeapSort(student\_record, num, compare); \*/
9. MergeSort(student\_record, 0, num - 1, compare);
10. printf("[INFO] Sort complete\n");
12. }

测试结果截图：



### 2.2.7成绩统计分析（4分）

按五个类别，对每科成绩分别统计每个类别的人数及所占的百分比。

相关函数代码与说明：

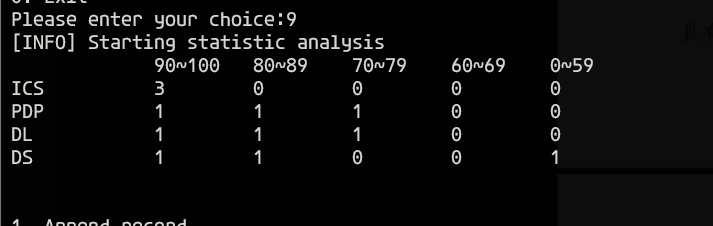
1. **void** StatisticAnalysis(STU\* student\_record, **int** num){
2. printf("[INFO] Starting statistic analysis\n");
4. **int** data[4][5];
5. memset(data, 0, **sizeof**(**int**) \* 5 \*4);
7. printf("             90~100   80~89    70~79    60~69    0~59\n");
8. **for**(**int** i = 0; i < num; i++) {
9. **if** ((student\_record + i)->score.ics >= 90 &&
10. (student\_record + i)->score.ics <= 100){
11. data[0][0] ++;
12. }
13. **else** **if** ((student\_record + i)->score.ics >=80 &&
14. (student\_record + i)->score.ics < 90){
15. data[0][1] ++;
16. }
17. **else** **if** ((student\_record + i)->score.ics >= 70 &&
18. (student\_record + i)->score.ics < 80){
19. data[0][2] ++;
20. }
21. **else** **if** ((student\_record + i)->score.ics >= 60 &&
22. (student\_record + i)->score.ics < 70){
23. data[0][3] ++;
24. }
25. **if** ((student\_record + i)->score.ics >= 0 &&
26. (student\_record + i)->score.ics < 60){
27. data[0][4] ++;
28. }

31. **if** ((student\_record + i)->score.pdp >= 90 &&
32. (student\_record + i)->score.pdp <= 100){
33. data[1][0] ++;
34. }
35. **else** **if** ((student\_record + i)->score.pdp >=80 &&
36. (student\_record + i)->score.pdp < 90){
37. data[1][1] ++;
38. }
39. **else** **if** ((student\_record + i)->score.pdp >= 70 &&
40. (student\_record + i)->score.pdp < 80){
41. data[1][2] ++;
42. }
43. **else** **if** ((student\_record + i)->score.pdp >= 60 &&
44. (student\_record + i)->score.pdp < 70){
45. data[1][3] ++;
46. }
47. **if** ((student\_record + i)->score.pdp >= 0 &&
48. (student\_record + i)->score.pdp < 60){
49. data[1][4] ++;
50. }

53. **if** ((student\_record + i)->score.dl >= 90 &&
54. (student\_record + i)->score.dl <= 100){
55. data[2][0] ++;
56. }
57. **else** **if** ((student\_record + i)->score.dl >=80 &&
58. (student\_record + i)->score.dl < 90){
59. data[2][1] ++;
60. }
61. **else** **if** ((student\_record + i)->score.dl >= 70 &&
62. (student\_record + i)->score.dl < 80){
63. data[2][2] ++;
64. }
65. **else** **if** ((student\_record + i)->score.dl >= 60 &&
66. (student\_record + i)->score.dl < 70){
67. data[2][3] ++;
68. }
69. **if** ((student\_record + i)->score.dl >= 0 &&
70. (student\_record + i)->score.dl < 60){
71. data[2][4] ++;
72. }

75. **if** ((student\_record + i)->score.ds >= 90 &&
76. (student\_record + i)->score.ds <= 100){
77. data[3][0] ++;
78. }
79. **else** **if** ((student\_record + i)->score.ds >=80 &&
80. (student\_record + i)->score.ds < 90){
81. data[3][1] ++;
82. }
83. **else** **if** ((student\_record + i)->score.ds >= 70 &&
84. (student\_record + i)->score.ds < 80){
85. data[3][2] ++;
86. }
87. **else** **if** ((student\_record + i)->score.ds >= 60 &&
88. (student\_record + i)->score.ds < 70){
89. data[3][3] ++;
90. }
91. **if** ((student\_record + i)->score.ds >= 0 &&
92. (student\_record + i)->score.ds < 60){
93. data[3][4] ++;
94. }
95. }
96. printf("%-12s %-8d %-8d %-8d %-8d %-8d\n", "ICS", data[0][0], data[0][1], data[0][2], data[0][3], data[0][4]);
97. printf("%-12s %-8d %-8d %-8d %-8d %-8d\n", "PDP", data[1][0], data[1][1], data[1][2], data[1][3], data[1][4]);
98. printf("%-12s %-8d %-8d %-8d %-8d %-8d\n", "DL", data[2][0], data[2][1], data[2][2], data[2][3], data[2][4]);
99. printf("%-12s %-8d %-8d %-8d %-8d %-8d\n", "DS", data[3][0], data[3][1], data[3][2], data[2][3], data[3][4]);
101. }

测试结果截图：

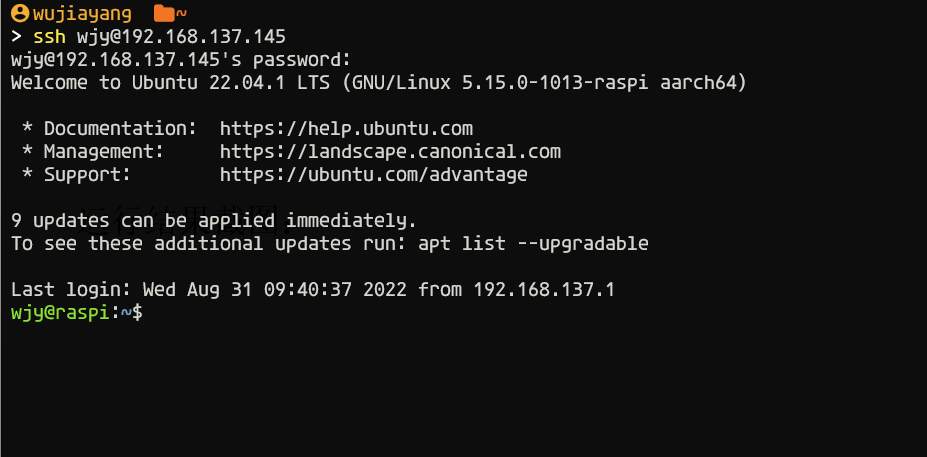


# 第3章 跨平台调试

**(选做内容，满分10分，加到本次实验总成绩满分100为止)**

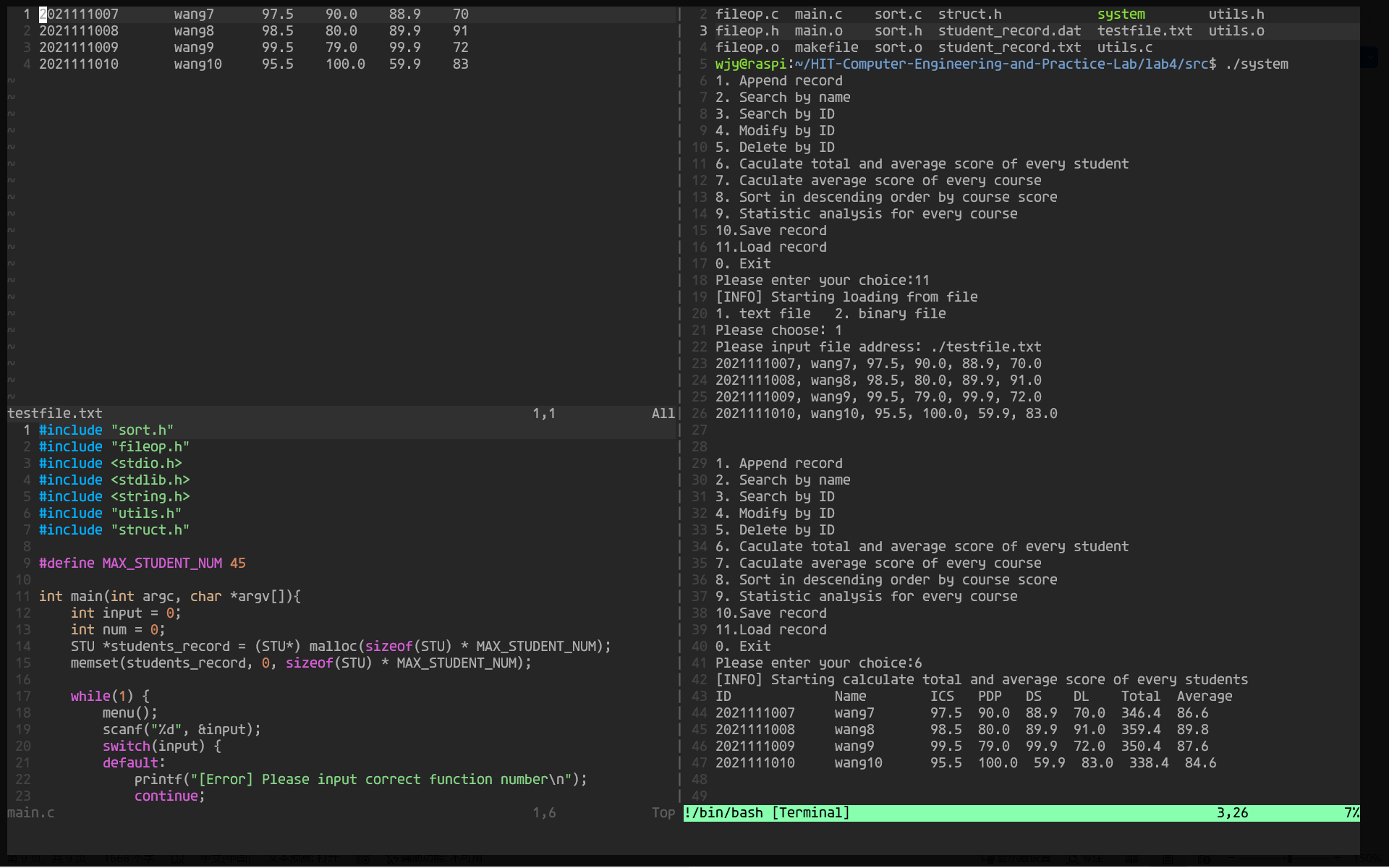
## 3.1 简述建立树莓派平台的过程并给出成功访问的截图（4分）

通过Win32DiskManager或者RaspiImager将下载的系统镜像烧录到tf卡中，因为没有显示器，所以在boot目录下创建ssh文件，然后打开树莓派，将网线连接至电脑，通过arp -a查看树莓派的网络ipv4端口。通过ssh [wjy@192.168.137.145](mailto:wjy@192.168.137.145)连接树莓派，通过WINSCP往树莓派上传输文件并执行。我采用另一套方案，通过在wsl2端通过git将所有代码上传至github，然后在树莓派端git clone。



## 3.2 将成绩管理系统移植到ARM架构的树莓派平台（6分）

在树莓派下生成成绩管理系统。



运行结果截图：