**《学生成绩管理系统》**

**微机接口实验报告**

姓名：吴浩男

学号：2013211228

班级：2013211303

学院：计算机学院

2015年12月10日

目录

1. 设计内容及要求……………………………………………………3
2. 设计思路及方法……………………………………………………3
3. 数据结构、各功能模块具体实现…………………………4
4. 程序测试及运行效果…………………………………………….6
5. 实验心得和遇到的问题…………………………………………8
6. 源代码……………………………………………………………………10
7. **设计内容及要求**

设计一个学生成绩管理程序，一个学生包括姓名、班级、学号、成绩。试编写程序，实现以下功能：

1. 成绩需要精确到 1位小数
2. 可以录入学生成绩（ 10 进制形式）
3. 可以按照要求（如学号或成绩）进行排序显示
4. 可以统计平均成绩
5. 可以统计各个分数的人数
6. **题目分析和设计思路**
7. **对题目的分析;**
8. 由于8086中没有对浮点数的运算，需要我们对浮点数进行模拟。因为只需要一位小数，我们对原始成绩乘以10就可以多一位精度，在输出的时候，除以10，再输出就可以。
9. 因为8086的一个寄存器最长16位，即65536。在求平均成绩的时候，不能让总和溢出。所以，限制至多输入100个学生。
10. 为了能够比较学号的大小，我对学号的输入也作出了限制，即范围为0-65535
11. 同样，姓名和班级，我用字符串来储存，字符串的最大长度为16，因为要保存一个’$’符号，能输入的最大长度为15。
12. **程序设计思路：**

因为汇编提供的接口比较复杂，需要对接口进行封装。实现了以下基本过程：

|  |  |  |  |
| --- | --- | --- | --- |
| 过程名称 | 实现功能 | 参数 | 返回值 |
| getInput | 获得输入并消除长度、尾部加上’$’符号 | 无 | 无 |
| getEnd | 获得学生结构体占用内存块后的部分 | STU | di |
| getNumber | 将字符串转化为数字 | TMP\_STR | ax |
| getScore | 将字符串转为精确到小数点位一位的浮点数，浮点数乘十储存 | TME\_STR | ax |
| printDecimal | 将整数转化为字符串并打印 | ax | 无 |
| printScore | 将整数转化为字符串并打印 | ax | 无 |
| sort | 根据需求进行打印 | STU/dl | 无 |

**高层程序模块设计：**

|  |  |  |  |
| --- | --- | --- | --- |
| 过程名称 | 功能 | 参数 | 返回值 |
| getInfo | 输入新加学生信息 | ax/bx/dx | 无 |
| printInfo | 输出所有的学生信息 | STU | 无 |
| average | 求平均成绩 | STU | 无 |
| count | 求各个成绩段的人数 | STU | ah/al/bh/al |

1. **数据结构、各功能模块具体实现**
2. **数据结构**
3. student

这是一个100个学生构成的结构体。每个结构体有16个字节，其中名字占10 个字节，班级占2字节，学号占2个字节，成绩占2个字节.

**myname db 10 dup(0) 名称**

**class dw 0 班级**

**number dw 0 学号**

**score dw 0 分数**

1. STU

这是一个64大小限制的student结构体数组，用于存放学生数组

1. STU\_NAME

记录学生数组的大小

1. TMP\_STR

在字符串和数字转换中，用于保存临时的信息。

1. 一系列STR常量

用于保存各个菜单的输出

1. TEN常量

表示10，用于转化字符串和整数

1. **模块功能实现**
2. getInfo

错误处理

判断学生成绩

输入学生学号

输入学生班级

输入学生姓名

判断数组已满

结束

1. printInfo

这个模块和getInfo逻辑相似，则不详述。

1. SORT

我使用的是代码量最小的冒泡排序。

其伪代码如下：

for j = length - 1 to 1

for i = 1 to j

if compare(i,i+1)

swap(I,i+1)

实际的汇编语言，是上面伪代码的翻译结果

因为引入了一个ID数组，需要利用比较函数进行得到结果。

1. getNumber

将数字转化成10进制表示字符串，我们可以不断的除以10，将余数记录下来，此时字符串是低位在最前面，在将字符串反转就可以。

其中有一个特殊情况，是要处理数字为0的情况，对其进行特殊判断就可以了。

1. printDecimal

将字符串转化成数字，利用公式res = res \* 10 + num，从高位开始递推，就可以得到最终的结果。

1. getScore

将含有小数点的字符串转化为数字，逻辑与getNumber相同，注意处理小数点。

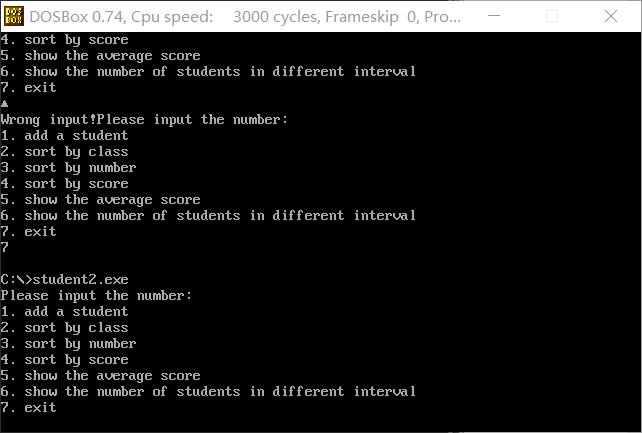
1. printScore

将整数转化为倒数第二位是小数点的字符串并输出，逻辑与printDecimal相同，注意处理小数点。

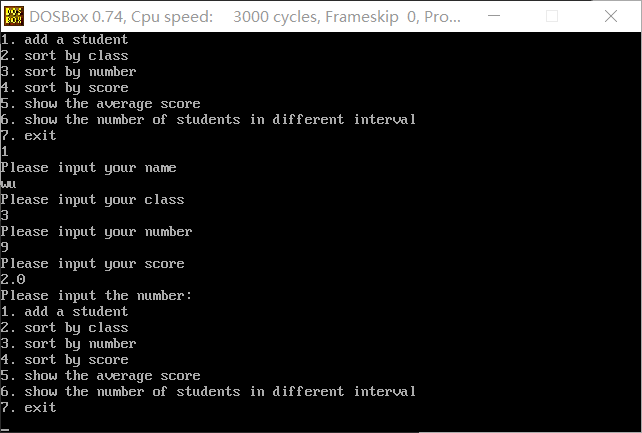
1. **错误处理**

程序中一共处理了以下情况：

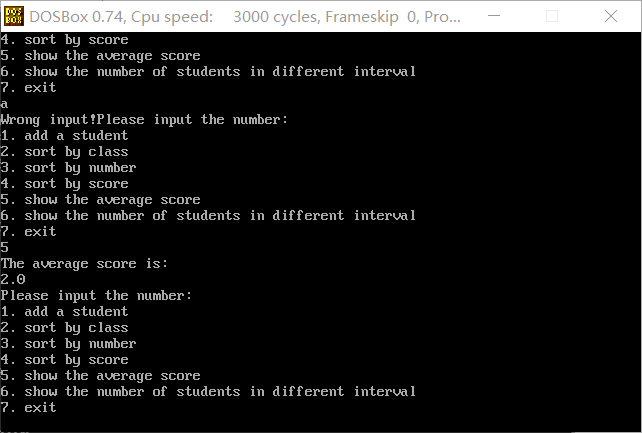
1. 在主程序中，对不合法的输入进行了处理，只有输入0-6才能进行相关的功能。
2. 在输入学生信息中，限制姓名和班级最长为15个字符。
3. 在输入学生信息中，成绩的输入可以为带一位精度的浮点数和不带浮点的整数
4. **程序测试及运行效果**
5. 程序主界面



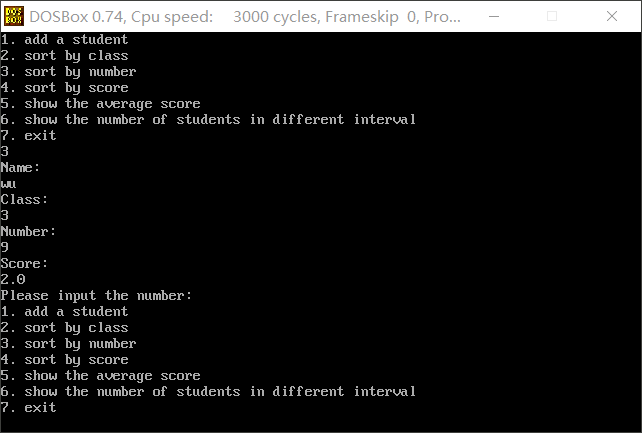
1. 输入学生信息



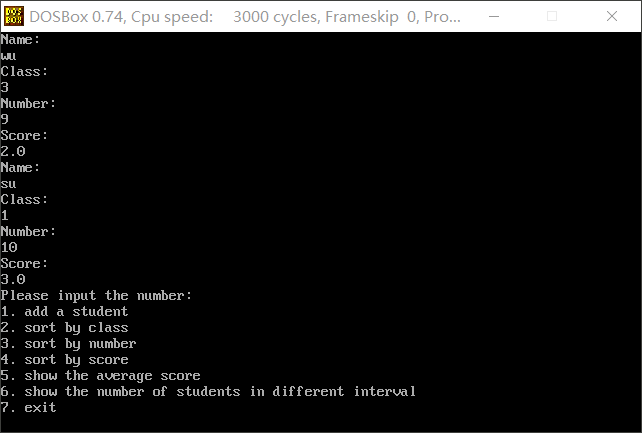
1. 求出平均成绩



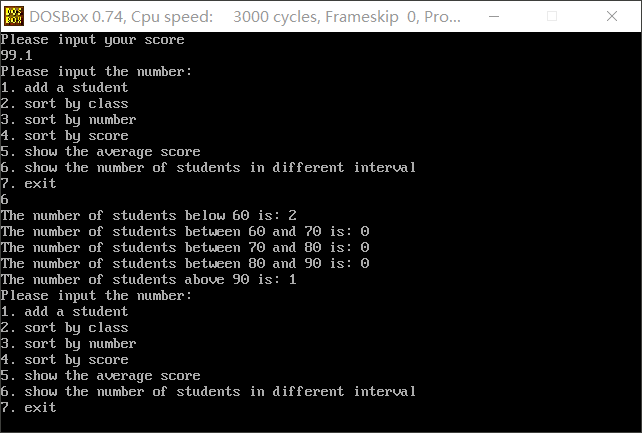
1. 根据学号排序



1. 根据成绩排序



1. 统计分段信息



1. **实验心得和遇到的问题**
2. **实验中遇到的问题和解决方法**

这是第一次用汇编语言写大型的程序。因为有以前的经验，在写代码前对整体的程序框架做了规划，但是还是遇到了很多问题。

首先遇到的是语法问题。在课堂上，老师已经讲了很多不合法的语法。但是有一些语法并没有涉及，只有在写代码的过程中才会使用。通过写汇编语言的代码，对汇编语言的代码有了更深层次的了解。

然后遇到的是参数传递的问题。最开始我是使用的堆栈传递参数的方法，但是后来发现很容易出现堆栈使用的错误。最后选择了使用寄存器传递参数。类似C语言中的全局变量来思考。

之后，我对于像switch语句一样的多重跳转，想简化结构。通过查阅数据，我找到了跳转表的使用方式。通过使用跳转表，大大简化了程序的跳转，让程序的结构更加清晰。

1. **实验中的不足**

因为时间不足的缘故，仅仅完成了代码的主体部分，对一些错误输入的判断和处理还没有完善。同时，只是简单设计了界面，并没有设计非常美观的界面。

而且，对于输入的数据，应该可以保存到文件，可以做数据持久化，这个功能没有完成。

1. **实验总结**

通过编写这次学生信息管理系统，首先对汇编语言的语法有了更深层次的了解。通过和高级语言进行对比，也了解到高级语言一些语言特性的原因。

和高级语言相比，汇编语言比较难以调试，通过这次编写代码，对DEBUG这样简单调试程序的使用有了一定的了解。

编写代码的过程中，我也查阅了相关的资料，了解了许多课本上没有的东西，开阔了视野，提高了自己的专业素质。