**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机系统1**  **项目名称 LC-3汇编语言求成绩等级**  **学 院 计算机与软件学院**  **专 业 计算机类**  **指导教师 陈飞**  **报 告 人 杨皓翔 学号 2023150139**  **实验时间 2024年5月9日**  **提交时间 2024年5月15日** |

**教务处制**

# 一、实验目的与要求

（1）掌握处理器仿真工具LC-3软件的安装和使用方法。

（2）学会在LC-3仿真环境下编辑程序和转换成可执行目标程序的方法 。

（2）学会在LC-3仿真环境下运行和调试程序的方法 。

# 二、实验内容与方法

利用提供的安装软件包和软件使用说明文档，完成以下试验内容：

（1）安装LC-3仿真器

（2）利用LC3EDIT输入机器代码程序（0/1模式）并创建可执行目标程序。

（3）利用LC3EDIT输入机器代码程序（hex模式）并创建可执行目标程序。

（4）利用LC3EDIT输入汇编代码程序并创建可执行目标程序。

（5）利用仿真器运用对应目标程序。

（6）学习和掌握断点，单步执行等调试方法和手段。

# 三、实验步骤与过程

（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

1. **题目**：使用LC-3汇编语言，编写程序实现16名学生成绩排序，及统计分析，成绩分类规则如下：A：全班排名前25%，且成绩在85分及以上；B：非A成绩，全班排名前50%，且成绩在75分及以上；C：非A、B成绩
2. **具体要求**：输入：16名学生成绩，存储于x3200至x320F，每个成绩为0至100之间，由16比特无符号整数表示；

输出：成绩降序排序，并存储于x4000至x400F内存位置，x4000位置成绩为最高成绩得A、B成绩的学生总人数，分别存储于x4100，及x4101位置

1. **实验思路**：首先排序找出所有学生成绩中的最大值，将其放在一个新的数组中，并从原数组中删除（或标记为已处理），重复此过程直到所有成绩都被排序；再进行分类统计，统计不同类成绩的数量
2. **排序：**
3. 寻找最大值：程序首先初始化各种寄存器，用R4作为指向成绩数组的指针。在第一个循环中，程序遍历成绩，使用寄存器R1来存储目前找到的最大成绩。如果找到一个更大的成绩，就更新R1的值。
4. 存储最大值：在找到最大值后，将其存储到另一个数组并从原数组中清除该值（或将其标记为已处理）。
5. 重复：重复上述步骤直到所有成绩都被排序并转移到新的数组中。
6. **分类统计：**
7. 统计A类成绩：在SORTED数组中检查前25%（前4个成绩）是否满足85分以上的条件，统计满足条件的成绩数量。
8. 统计B类成绩：继续检查接下来的25%（第5到第8个成绩）是否满足75分以上的条件，统计满足条件的成绩数量。
9. 输出结果：程序将通过ASCII码输出结果，分别输出A和B类成绩的数量。
10. **结束：**程序在输出了所有结果后通过HALT指令停止
11. 具体代码实现如下：

; 初始化部分

.ORIG x3000 ; 设置程序的起始地址为x3000

AND R0, R0, #0 ; 将寄存器R0清零

AND R7, R7, #0 ; 将寄存器R7清零，用作后面的索引

; 主要逻辑部分

READ LD R4, BEGIN ; 将BEGIN标签所指向的地址加载到R4，R4用作指针

AND R1, R1, #0 ; 将寄存器R1清零，用于存储当前最大值

AND R3, R3, #0 ; 将寄存器R3清零

ADD R3, R3, #10

ADD R3, R3, #6 ; R3设置为16，代表处理16个数字

LOOP1 LDR R5, R4, #0 ; 从R4指向的地址加载数据到R5

NOT R5, R5 ; 对R5进行按位取反

ADD R5, R5, #1 ; 将取反后的结果加1，实现二进制补码的负数转换

ADD R5, R5, R1 ; 将R5和当前最大值R1相加

BRp OK ; 如果R1>R5（即R5较小），则跳过更新最大值的步骤

LDR R1, R4, #0 ; 更新当前最大值R1

OK ADD R4, R4, #1 ; 将指针R4加1，指向下一个数据

ADD R3, R3, #-1 ; 递减计数器R3

BRp LOOP1 ; 如果R3大于0，继续循环

; 重新初始化指针和计数器，准备第二次循环

LD R4, BEGIN

AND R3, R3, #0 ; 重置计数器R3

ADD R3, R3, #10

ADD R3, R3, #6 ; 设置R3为16，代表处理16个数字

LOOP2 LDR R5, R4, #0 ; 从R4指向的地址加载数据到R5

NOT R5, R5

ADD R5, R5, #1

ADD R5, R5, R1

BRz YES ; 如果R5等于R1（即找到了最大值），跳到YES标签

ADD R4, R4, #1 ; 指针加1

ADD R3, R3, #-1 ; 递减计数器R3

BRp LOOP2 ; 如果R3大于0，继续循环

YES LDR R5, R4, #0 ; 从R4指向的地址加载数据到R5

LD R6, SORTED

ADD R6, R6, R7 ; 计算存储位置

STR R5, R6, #0 ; 将当前最大值存储到排序数组

STR R0, R4, #0 ; 将原位置的数据置0

ADD R7, R7, #1 ; 更新索引R7

ADD R6, R7, #-16

BRz GET ; 如果已处理16个数字，跳到GET

BRnzp READ ; 否则，回到READ开始新一轮读取和处理

GET ; 计算满足特定条件的数字数量

AND R1, R1, #0 ; 清零R1，用作结果A

AND R2, R2, #0 ; 清零R2，用作结果B

LD R4, SORTED

AND R3, R3, #0

ADD R3, R3, #4 ; 前25%的数字数量设为4

LOOP3 LD R6, EIGHT ; 85

LDR R5, R4, #0

NOT R5, R5

ADD R5, R5, #1

ADD R6, R5, R6

BRp CHECKB ; 如果R5<85，跳到CHECKB

ADD R1, R1, #1 ; 否则，结果A增加

NOB ADD R4, R4, #1

ADD R3, R3, #-1

BRp LOOP3 ; 继续LOOP3直到处理完前25%

; 输出结果A

LD R0, ASCII

ADD R0, R0, R1

OUT

LD R0, SPACE

OUT

; 处理第二部分数据

LD R4, SORTED

ADD R4, R4, #4

AND R3, R3, #0

ADD R3, R3, #4 ; 第二部分数据处理

LOOP4 LD R6, SEVEN

LDR R5, R4, #0

NOT R5, R5

ADD R5, R5, #1

ADD R6, R5, R6

BRp NOBB

ADD R2, R2, #1 ; 结果B增加

NOBB ADD R4, R4, #1

ADD R3, R3, #-1

BRp LOOP4 ; 继续LOOP4直到处理完第二部分数据

; 输出结果B

LD R0, ASCII

ADD R0, R0, R2

OUT

BRnzp DONE

CHECKB LD R6, SEVEN

ADD R6, R5, R6

BRp NOB

ADD R2, R2, #1

BRnzp NOB

DONE HALT ; 结束程序

; 数据和地址标签

BEGIN .FILL x3200 ; 开始的数据地址

SORTED .FILL x4000 ; 排序后数据存储地址

EIGHT .FILL #85

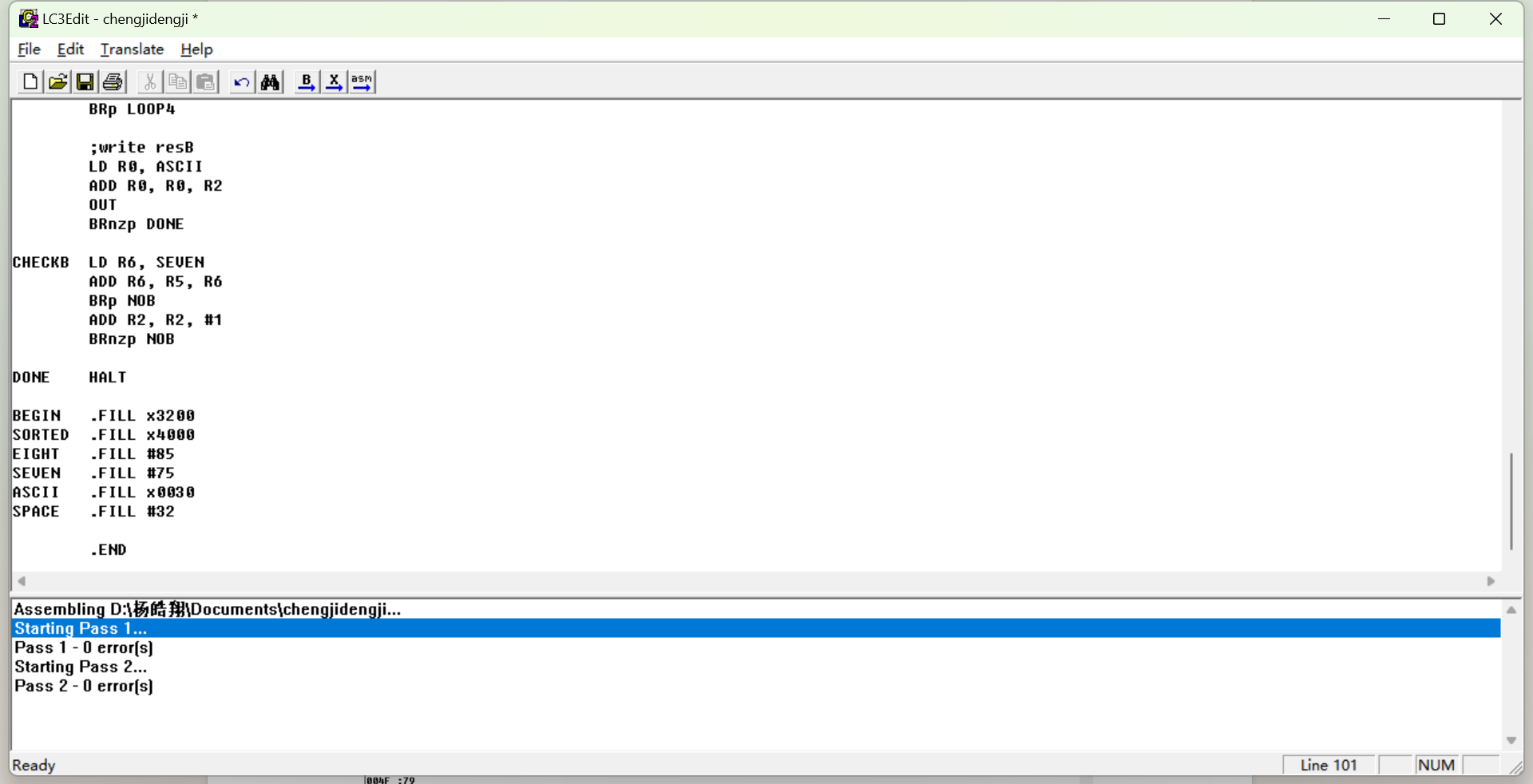
SEVEN .FILL #75

ASCII .FILL x0030 ; ASCII码基准

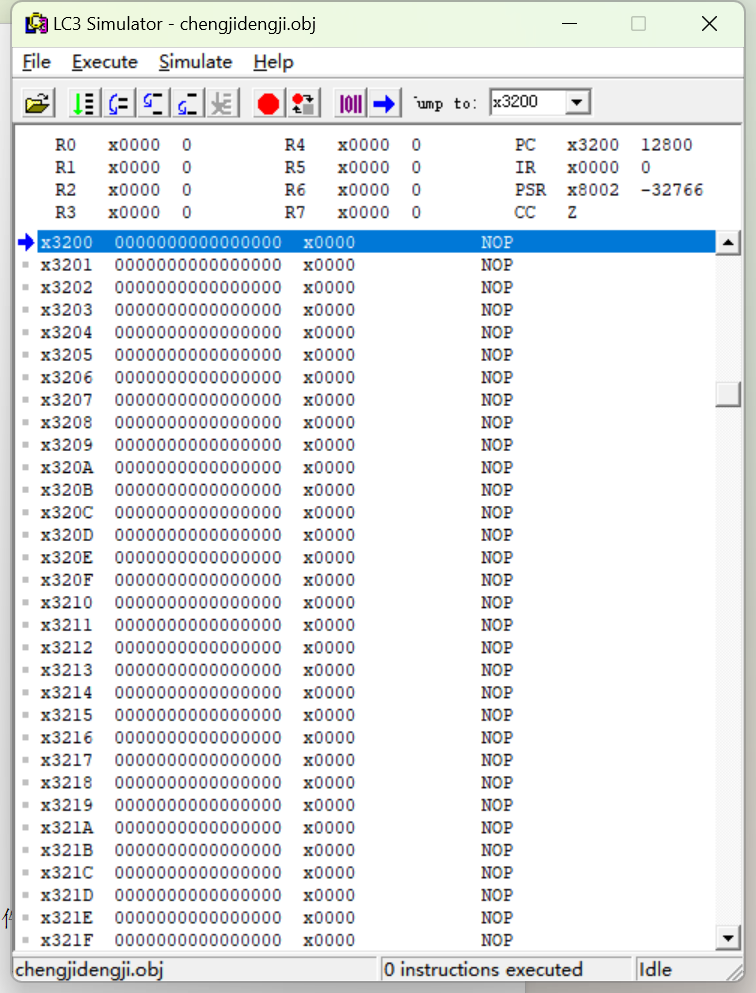
SPACE .FILL #32 ; 空格字符的ASCII码

.END ; 程序结束标记

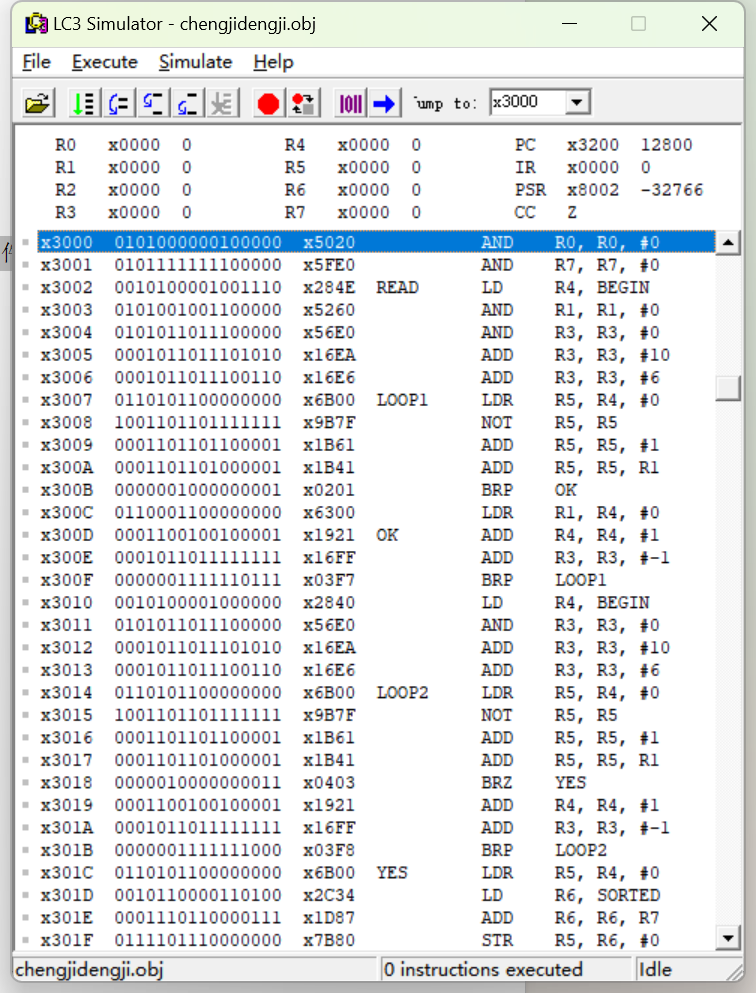
将代码输入到LC3Edit中，保存为.asm，然后点击，编译保存如图所示。



打开Simulate后点击File->Lode Program，选择数据对应的.obj文件，装载后在程序的x3200到x320F可以看到上面编译保存的数据，如图所示。



然后再次点击File->Lode Program，选择代码对应的.obj文件，装载后在程序的x3000开始可以看到所编写的代码

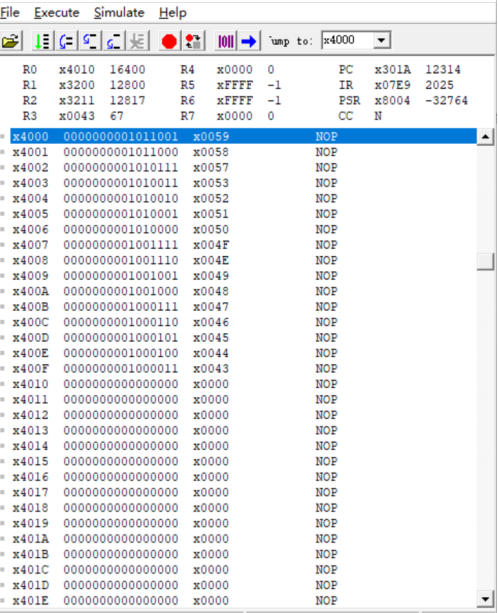


然后双击x301A前的灰色正方形图标，设置一个断点，该断点将程序分为两部分，第一部分为排序的程序，第二部分为记录A，B等级的人数的程序。点击图标，程序弹出了如图所示的窗口。

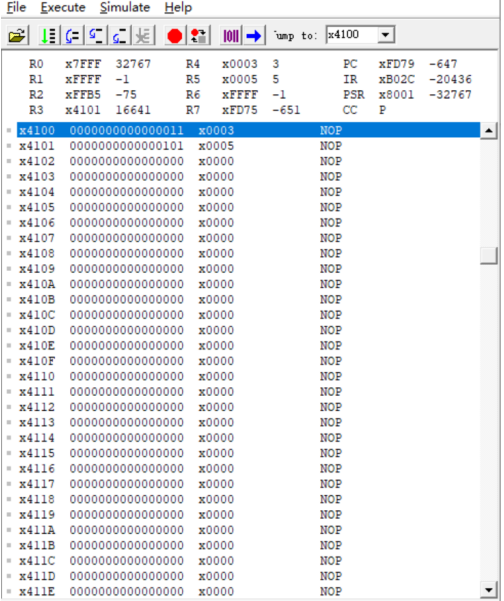
图形用户界面, 应用程序

描述已自动生成

然后在这个地方输入x4000，将程序跳转到x4000位置，可以看到，从x4000到x400F是排序好的数据



可以看到，成绩大于85分（即x0055），并且位于前25%的人数有3个，成绩大于75分（即x004B），不为等级A并且位于前50%的人数有5个，所以继续运行后，x4100中的值应该为x0003，x4101中的值应该为x0005,。现在将程序返回到断点位置，点击图标。程序运行结束后，将程序跳转到x4100处，可以看到，x4100中的值为x0003，x4101中的值为x0005



实验结果与预期一样，完结撒花

# 四、实验结论或体会

（撰写实验收获及思考）

1. 对存放在连续空间中的数据,可通过用指针指向其首地址,再指针不断 来实现类似于遍历数组的操作.
2. 与一个固定值比较大小时,可先在一个地址用.FILL填充该比较数,使用时用LD将其读入到寄存器中,再将当前值与
3. 比较数作差(转化为补码做加法),根据结果的正负或零用BR指令跳转到相应的地方.
4. 适当分段、使用LABLE和添加COMMENTS可让代码思路更清晰.
5. 输出数字时,应将其先存到寄存器 中, 加上'0'的ASCII码x0030后再用OUT输出,因为OUT只能输出 中的值.
6. 使用寄存器前注意清空.
7. 程序出现bug时添加断点方便调试.

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。