苏州大学实验报告

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机科学与技术 | | 年级专业 | 计科 | | 姓名 |  | | 学号 | |  |
| 课程名称 | | 微型计算机原理及应用实践 | | | | | | 同组实验者 | | | 无 |
| 指导教师 | | 王宜怀 | 实验日期 | | 2025.4.10 | | 成绩 | | |  | |

|  |  |
| --- | --- |
| 实 验 名 称 | 基于构件方法的汇编程序设计 |

一、实验目的

（1）对构件基本应用方法有更进一步的认识，初步掌握基于构件的汇编设计与运行。

（2）理解汇编语言中顺序结构、分支结构和循环结构的程序设计方法。

（3）理解和掌握汇编跳转指令的使用方法和场合。

（4）掌握硬件系统的软件测试方法，初步理解printf输出调试的基本方法。

二、实验准备

（1）硬件部分。PC或笔记本计算机一台、开发套件一套。

（2）软件部分。根据电子资源中的“…\02-Document”文件夹下的电子版快速指南，下载合适的电子资源。

（3）软件环境。按照电子版快速指南中“安装软件开发环境”一节，进行有关软件工具的安装。

三、实验过程或要求

（1）验证性实验

① 下载开发环境AHL-GEC-IDE。

根据电子资源“…\05-Tool\AHL-GEC-IDE下载地址.txt”文件的指引，下载由SD-Arm开发的金葫芦集成开发环境到“…\05-Tool”文件夹中。该集成开发环境兼容一些常规的开发环境工程格式。

② 建立自己的工作文件夹。

按照“分门别类、各有归处”的原则建立自己的工作文件夹，并考虑随后内容的安排，建立其下级子文件夹。

③ 拷贝模板工程并重命名。

所有工程可通过拷贝模板工程建立。例如，可以拷贝“\04-Soft ware\ Exam4\_1”工程到自己的工作文件夹，并将其改为自己确定的工程名。建议尾端增加日期字样，以避免混乱。

④ 导入工程。

假设已经下载了 AHL-GEC-IDE，并放入“…\05-Tool”文件夹，且按安装电子档快速指南正确安装了有关工具，则可以开始运行“…\05-Tool\AHL-GEC-IDE\AHL-GEC-IDE.exe”文件，这一步打开了集成开发环境 AHL-GEC-IDE。接着单击“文件”→“导入工程”，导入拷贝到自己文件夹并重新命名的工程。导入工程后，左侧为工程树形目录，右边为文件内容编辑区，初始显示main.s文件的内容。

⑤ 编译工程。

在打开工程并显示文件内容的前提下，可编译工程。单击“编译”→“编译工程（01）”，则开始编译。

⑥ 下载并运行。

步骤一，硬件连接。用TTL-USB线（Micro 口）连接GEC底板上的“MicroUSB”串口与计算机的USB口。

步骤二，软件连接。单击“下载”→“串口更新”，将 进入 界面更新界面。点击“连接GEC”查找到目标GEC，则提示“成功连接……”。

步骤三，下载机器码。点击“选择文件”按钮导入被编译工程目录下的 Debug 中的.hex文件（看准生成时间，确认其是现在编译的程序），然后单击“一键自动更新”按钮，等待程序完成自动更新。

此时程序自动运行了。若遇到问题可参阅开发套件纸质版导引“常见错误及解决方法”一节，也可参阅电子资源“…\02-Document”文件夹中的快速指南。

⑦ 观察运行结果与程序的对应情况。

第一个程序运行结果（PC界面显示情况）。为了表明程序已经开始运行，在每个样例程序进入主循环之前，使用printf语句输出一段话，程序写入后立即执行，就会显示在开发环境下载界面的右下角文本框中，提示程序的基本功能。

利用printf 语句将程序运行的结果直接输出到PC屏幕上，使嵌入式软件开发的输出调试变得十分便利。调试嵌入式软件与调试PC软件几乎一样方便，其改变了传统交叉调试模式。

（2）设计性实验

在验证性实验的基础上，自行编程实现计算100以内的奇数相加所得的和，最后通过串口输出该结果。

（3）进阶实验★

自定义一组数据，采用一种排序算法对这组数据进行从小到大排序。

四、实验结果

（1）验证性实验

完成了将一组数进行冒泡递减排序，并将结果输出。

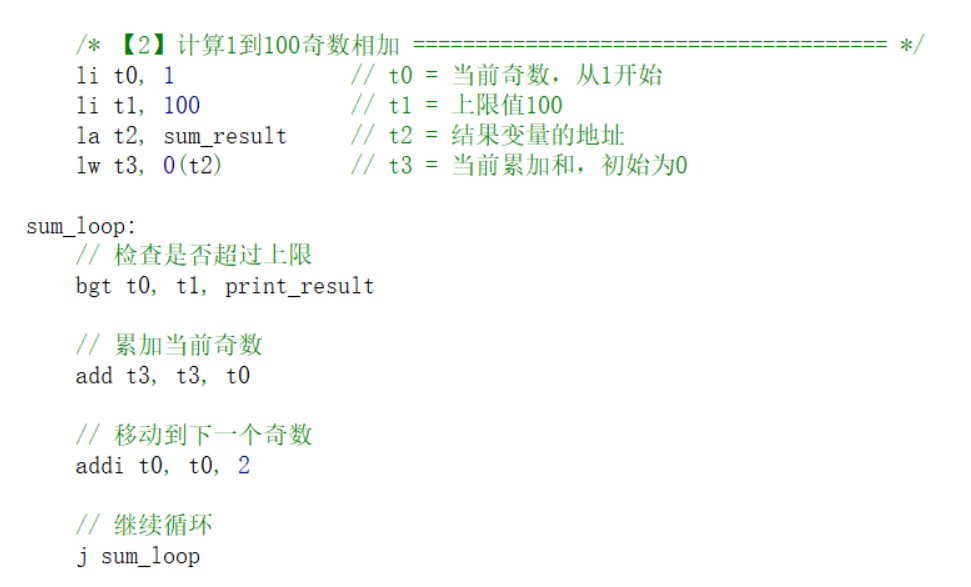


（2）设计性实验

完成了计算100以内的奇数相加所得的和，最后通过串口输出该结果。



设计性实验关键代码：



（3）进阶实验



进阶实验关键代码：

1. *//函数名称：selectionSort\_down*
2. *//函数返回：无*
3. *//参数说明：a0:用于存储数据的首地址，a1:数组字节数量*
4. *//功能概要：将一数组采用选择降序方式进行排列，并返回排序后的数组*
5. *//======================================================================\*/*
6. selectionSort\_down:
7. *// 1. 调整堆栈指针，分配空间*
8. addi  sp, sp, -32          *//分配32字节堆栈空间*
9. *// 2. 保存寄存器现场*
10. sw    ra, 28(sp)
11. sw    s0, 24(sp)    *// 保存数组地址*
12. sw    s1, 20(sp)    *// 保存数组长度*
13. sw    s2, 16(sp)    *// 外层循环计数器i*
14. sw    s3, 12(sp)    *// 内层循环计数器j*
15. sw    s4, 8(sp)     *// 最大元素索引max\_idx*
16. sw    s5, 4(sp)     *// 临时变量*
18. mv    s0, a0        *// s0 = 数组地址*
19. mv    s1, a1        *// s1 = 数组长度*
20. li    s2, 0         *// i = 0*
22. outer\_loop:
23. addi  t0, s1, -1
24. bge   s2, t0, sort\_done  *// 如果i >= n-1，排序完成*
26. mv    s4, s2        *// max\_idx = i*
27. addi  s3, s2, 1     *// j = i + 1*
29. inner\_loop:
30. bge   s3, s1, end\_inner\_loop  *// 如果j >= n，内层循环结束*
32. *// 获取array[j]*
33. add   t0, s0, s3
34. lb    t1, 0(t0)     *// t1 = array[j]*
36. *// 获取array[max\_idx]*
37. add   t0, s0, s4
38. lb    t2, 0(t0)     *// t2 = array[max\_idx]*
40. *// 比较array[j]和array[max\_idx]*
41. bleu  t1, t2, no\_new\_max  *// 如果array[j] <= array[max\_idx]，不更新max\_idx*
43. *// 更新max\_idx*
44. mv    s4, s3        *// max\_idx = j*
46. no\_new\_max:
47. addi  s3, s3, 1     *// j++*
48. j     inner\_loop
50. end\_inner\_loop:
51. *// 交换array[i]和array[max\_idx]*
52. beq   s4, s2, no\_swap\_needed  *// 如果max\_idx == i，不需要交换*
54. *// 获取array[i]的值*
55. add   t0, s0, s2
56. lb    t1, 0(t0)     *// t1 = array[i]*
58. *// 获取array[max\_idx]的值*
59. add   t0, s0, s4
60. lb    t2, 0(t0)     *// t2 = array[max\_idx]*
62. *// 执行交换*
63. sb    t1, 0(t0)     *// array[max\_idx] = array[i]*
64. add   t0, s0, s2
65. sb    t2, 0(t0)     *// array[i] = array[max\_idx]*
67. no\_swap\_needed:
68. addi  s2, s2, 1     *// i++*
69. j     outer\_loop
71. sort\_done:
72. *// 3. 恢复寄存器现场*
73. lw    ra, 28(sp)
74. lw    s0, 24(sp)
75. lw    s1, 20(sp)
76. lw    s2, 16(sp)
77. lw    s3, 12(sp)
78. lw    s4, 8(sp)
79. lw    s5, 4(sp)
80. *// 4. 恢复堆栈指针*
81. addi  sp, sp, 32
82. ret                        *//返回*

五、实验体会及问答

问答：

（1）若要通过串口输出冒泡排序结果，该如何实现？

编辑串口输出结果的信息.string后printf。

（2）有哪些排序算法，分别使用RISC-V汇编语言实现要点

冒泡排序:

1. bubbleSort\_down:
2. li    t6, count-1          *# 外层循环 n-1 次*
3. outer\_loop:
4. mv    t2, a0               *# 数组指针复位*
5. li    t5, 0                *# 内层循环计数器*
6. inner\_loop:
7. lb    t0, 0(t2)            *# 加载当前元素*
8. lb    t1, 1(t2)            *# 加载下一个元素*
9. blt   t0, t1, swap         *# 比较并决定是否交换*
10. addi  t2, t2, 1            *# 指针移动*
11. addi  t5, t5, 1
12. blt   t5, t6, inner\_loop    *# 内层循环继续*
13. addi  t6, t6, -1           *# 外层循环减一*
14. bnez  t6, outer\_loop

选择排序:

1. selectionSort\_down:
2. li    s2, 0               *# i = 0*
3. outer\_loop:
4. mv    s4, s2              *# max\_idx = i*
5. addi  s3, s2, 1           *# j = i+1*
6. inner\_loop:
7. lb    t1, array(s3)       *# 加载array[j]*
8. lb    t2, array(s4)       *# 加载array[max\_idx]*
9. bleu  t1, t2, no\_update   *# 比较*
10. mv    s4, s3              *# 更新max\_idx*
11. no\_update:
12. addi  s3, s3, 1           *# j++*
13. blt   s3, s1, inner\_loop  *# 继续内层循环*
14. *# 交换array[i]和array[max\_idx]*
15. lb    t1, array(s2)
16. lb    t2, array(s4)
17. sb    t1, array(s4)
18. sb    t2, array(s2)
19. addi  s2, s2, 1           *# i++*
20. blt   s2, t0, outer\_loop  *# 继续外层循环*

插入排序:

1. insertion\_sort:
2. li    t0, 1               *# i = 1*
3. outer:
4. lb    t1, array(t0)       *# key = array[i]*
5. mv    t2, t0              *# j = i*
6. inner:
7. blez  t2, insert          *# j <= 0时退出*
8. lb    t3, array(t2-1)     *# array[j-1]*
9. ble   t3, t1, insert      *# 若array[j-1] <= key则退出*
10. sb    t3, array(t2)       *# array[j] = array[j-1]*
11. addi  t2, t2, -1          *# j--*
12. j     inner
13. insert:
14. sb    t1, array(t2)       *# 插入key*
15. addi  t0, t0, 1           *# i++*
16. blt   t0, a1, outer       *# 继续循环*

体会：

通过本次实验，我对构件基本应用方法有了更进一步的认识，初步掌握了基于构件的汇编设计与运行。同时了解了汇编语言中顺序结构、分支结构和循环结构的程序设计方法，掌握了汇编跳转指令的使用方法和场合、硬件系统的软件测试方法，初步理解了printf输出调试的基本方法。汇编语言中的跳转指令是控制程序流程的关键，使用printf语句进行调试，能够直观地看到程序的运行结果，并根据输出调整程序逻辑，提高了解决问题的效率。

实验中，我在验证性实验的基础上，自行编程实现计算100以内的奇数相加所得的和，最后通过串口输出。并自定义一组数据，采用选择排序算法对这组数据进行从小到大排序。

通过这次实验，我不仅学习了汇编语言的基础知识，还掌握了硬件编程的基本技能。将理论知识应用到实践中，加深了我对基于构建方法的汇编程序设计的理解。