

实验报告

实验要求：

- 基于 MIPS 汇编，设计一个冒泡排序程序，并用 Debug 工具调试执行。
- 测量冒泡排序程序的执行时间。

设计思路：

1. 先用系统调用读取一个 int 型数据,作为待排序线性表的表长
根据表长,设计循环完成待排序数据的读入
2. 完成核心算法“冒泡排序”,并将排序后的数据输出至屏幕
3. 利用 MIPS 汇编指令提供的伪操作来优化输入输出,以及为存放数据的线性表开辟空间
4. 利用 syscall 获取系统时间,进而计算排序程序执行时间

算法实现：

根据冒泡排序算法思路，设计两层循环 loop_1 与 loop_2

```
loop_1:
    addi $t2,$zero,0
loop_2:
    #计算 a[i] 地址,并取出 a[i],a[i+1]
    addi $s0,$t2,0
    mul $s0,$s0,4
    addu $s1,$s0,$t4
    lw $s2,0($s1)
    lw $t3,4($s1)

    #判断是否需要交换
    bge $s2,$t3,target
    sw $t3,0($s1)
    sw $s2,4($s1)

target:
```

```
#判断是否进行内部循环
```

```
addi $t2,$t2,1
```

```
addi $s0,$t2,1
```

```
sub $s1,$t1,$t0
```

```
blt $s0,$s1,loop_2
```

```
#判断是否进行外部循环
```

```
addi $t0,$t0,1
```

```
sub $s2,$t1,1
```

```
blt $t0,$s2,loop_1
```

之前已经用伪操作开辟了长度为 1024 字节的空间, 并且将地址赋给了 t4. 所以每次通过 t4 寄存器中的值来计算待比较的两个数据的地址. (注意 MIPS 指令集采用字节编址, 故实际计算地址时, 需要将循环变量乘以 $32/8 = 4$ 后再进行计算). 具体见代码中的注释;

之后进行比较. 如果需要交换两个数据, 则直接将两个寄存器中的值存入对方的原地址中即可, 而不需要设置中间变量(与高级语言的区别)

由于 MIPS 指令集中没有设置条件码, 而 beq 指令使用起来并不方便, 故查询到了其它的一些控制流分支指令:

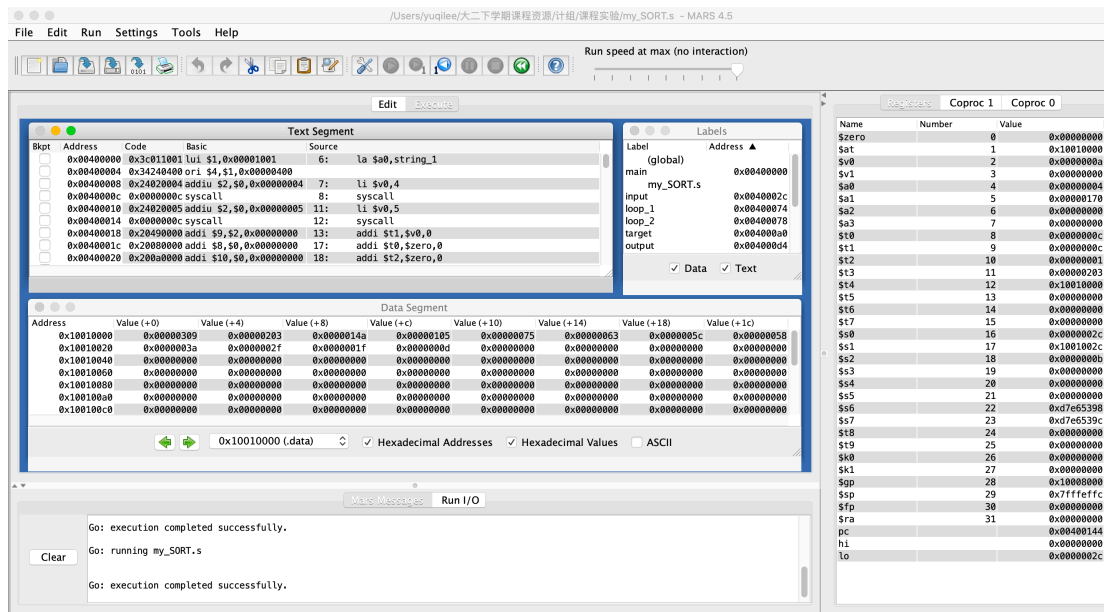
```
blt $t0,$t1,target # 若 $t0 < $t1, 则跳转至 target
```

```
bge $t0,$t1,target # 若 $t0 >= $t1, 则跳转至 target
```

以上两条指令可以根据两个寄存器中数据之间的**不等关系**直接完成跳转

调试执行:

使用 MARS 模拟器运行汇编代码.利用 MARS 自带的调试工具进行调试,界面如下:

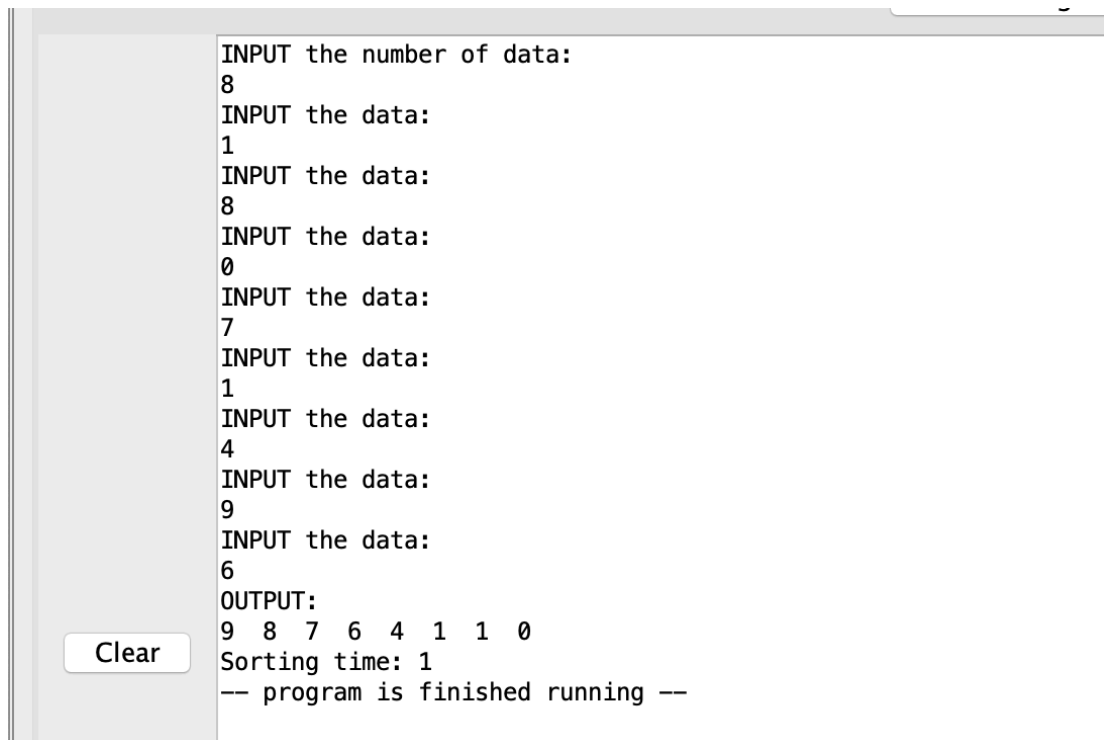


MARS 软件自带的 Debug 工具十分方便:

可以单步执行, 进入, 看到执行至每一步时的寄存器, 内存情况, 汇编时伪操作和 label 与内存地址之间的映射表等:

之后输入数据进行测试:

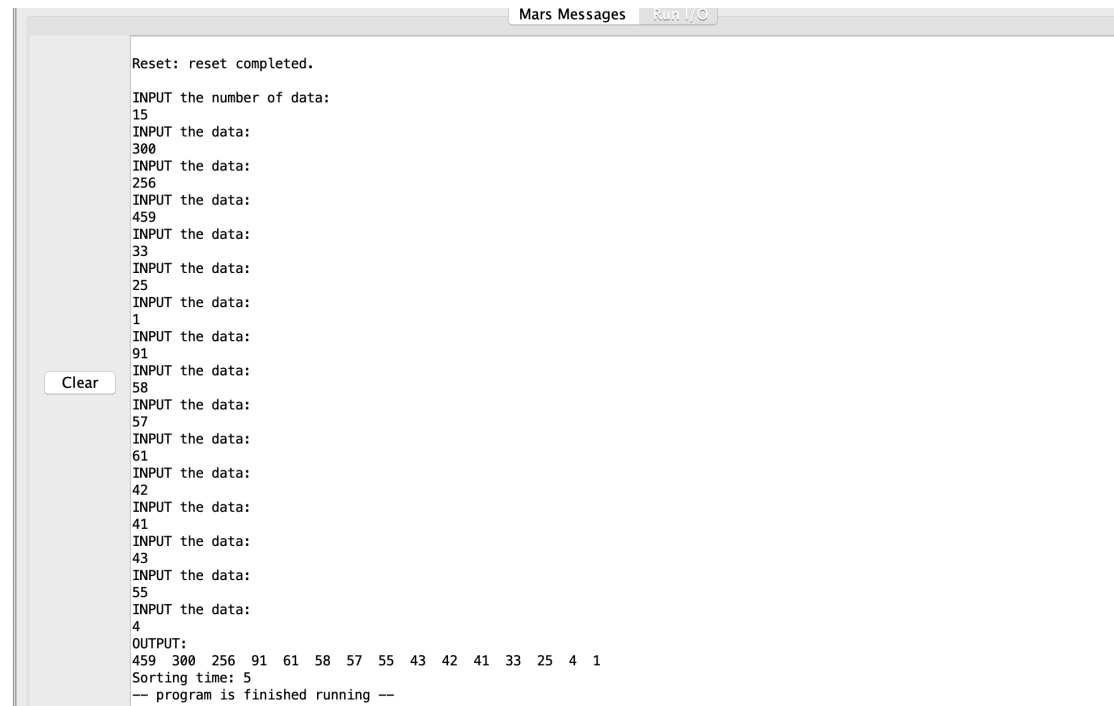
首先输入自己的学号: “18071496”



成功将学号中的数字按降序输出,排序程序用时 1ms

之后尝试输入更多数据:

300,256,459,33,25,1,91,58,57,61,42,41,43,55,4 共 15 个 int 型数据



The screenshot shows a window titled "Mars Messages" with a "Run I/O" button. The text inside the window is as follows:

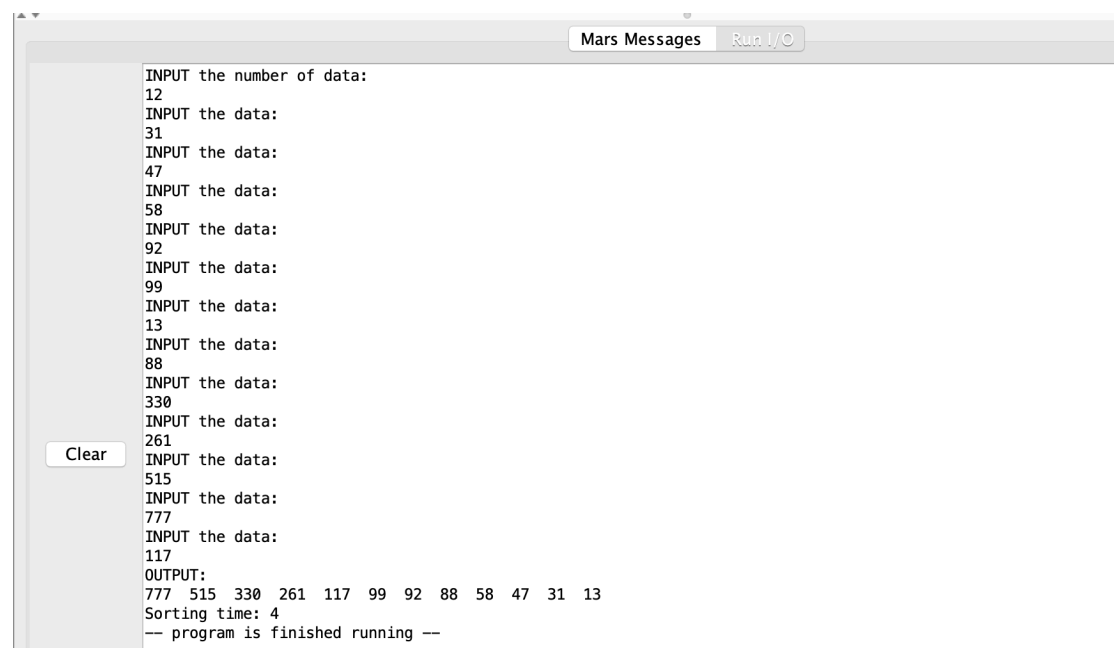
```
Reset: reset completed.  
  
INPUT the number of data:  
15  
INPUT the data:  
300  
INPUT the data:  
256  
INPUT the data:  
459  
INPUT the data:  
33  
INPUT the data:  
25  
INPUT the data:  
1  
INPUT the data:  
91  
INPUT the data:  
58  
INPUT the data:  
57  
INPUT the data:  
61  
INPUT the data:  
42  
INPUT the data:  
41  
INPUT the data:  
43  
INPUT the data:  
55  
INPUT the data:  
4  
  
OUTPUT:  
459 300 256 91 61 58 57 55 43 42 41 33 25 4 1  
Sorting time: 5  
-- program is finished running --
```

There is a "Clear" button on the left side of the window.

也成功将数据排序, 耗时 5ms

再进行一例测试:

输入:31,47,58,92,99,13,88,330,261,515,777,117 共 12 个 int 型数据



The screenshot shows a window titled "Mars Messages" with a "Run I/O" button. The text inside the window is as follows:

```
INPUT the number of data:  
12  
INPUT the data:  
31  
INPUT the data:  
47  
INPUT the data:  
58  
INPUT the data:  
92  
INPUT the data:  
99  
INPUT the data:  
13  
INPUT the data:  
88  
INPUT the data:  
330  
INPUT the data:  
261  
INPUT the data:  
515  
INPUT the data:  
777  
INPUT the data:  
117  
  
OUTPUT:  
777 515 330 261 117 99 92 88 58 47 31 13  
Sorting time: 4  
-- program is finished running --
```

There is a "Clear" button on the left side of the window.

排序完成,总用时 4ms

实验总结:

1. 第一次使用 MIPS 汇编进行编程, 体会到了该套指令集它与上学期所学 LC-3 指令集之间的不同:
 - a) 没有条件码
 - b) 更丰富的指令功能, 许多在 LC-3 中需要多条指令组合实现的功能(如相乘、条件判断等), 在 MIPS 中可以由已有指令实现
 - c) 在 MAR 模拟器的 java 环境支持下, MIPS 指令集可以调用 JVM 提供的许多系统调用接口, 如获取 systemtime, 创建线程/线程同步, 产生随机数等. 功能较 LC-3 更为强大
2. 熟悉了 MARS 模拟器的使用, 如载入/关闭代码文件、调试等
3. 加深了对 MIPS 指令集结构的掌握, 如各个寄存器的作用、一些常用的算术、数据转移、分支控制指令的具体应用等

附录:

1. “MIPS 冒泡排序实验-实验报告-PB18071496-李昱祁.pdf”
2. “my_SORT.s” (源码)