

# MIPS 实验：排序算法

姓名：宋渝杰 学号：18340146

## 一、实验目的

熟悉 MIPS 汇编语言程序设计。

## 二、实验要求

输入 10 个整数，按从大到小或从小到大的顺序输出。

## 三、实验内容

### 思路：

先申请 10 个单位大小的数组内存，用于存储输入的 10 个整数，之后通过循环输入的方式，将输入的数据通过数组首地址加索引存储到相应位置。之后建立双重循环，以冒泡排序的思想，从数组中提出相邻的数据并进行对比或交换，排序结束之后通过循环方式输出数组元素。此处实现从大到小排序。

### 代码：

```
.data
array: .space 40          # 申请 10 个数据内存
str1: .asciiz "Enter 10 integers : "
nline: .asciiz "\n"
msg1: .asciiz "Output: \n"

.text

main:
    la $t0, array          # 数组首地址
    li $v0, 4              # 输出字符串
    la $a0, str1            # 输出信息 str1
    syscall
    la $a0, nline          # 换行符, "\n"
    li $v0, 4              # 换新一行输出
    syscall
```

```

    add $s0, $zero, $zero          # 相当于 int i;

get:
    li $v0, 5                      # 循环输入数据
    syscall
    sll $t1, $s0, 2
    add $t1, $t1, $t0
    sw $v0, 0($t1)
    addi $s0, $s0, 1
    slti $t3, $s0, 10
    bnez $t3, get                  # 若输入数据小于 10 个，则跳转 get，继续输入
    li $v0, 4
    syscall
    add $s0, $zero, $zero          # i=0

exloop:                             # 外循环
    add $s1, $zero, $zero          # j=0

inloop:                             # 内循环
    add $t1, $s1, $zero
    sll $t1, $t1, 2
    add $t1, $t1, $t0
    lw $s7, 0($t1)                 # 相当于 array[j]
    addi $t2, $s1, 1
    sll $t2, $t2, 2
    add $t2, $t2, $t0
    lw $s6, 0($t2)                 # 相当于 array[j+1]
    slt $t3, $s7, $s6              # 判断是否 array[j] < array[j+1] 若
array[j]<array[j+1], t3 = 1, 交换, 若>=, 则跳过交换阶段;
    beqz $t3, exit
    sw $s7, 0($t2)
    sw $s6, 0($t1)                 # 交换阶段

exit:
    addi $s1, $s1, 1               # j++
    addi $t4, $zero, 9
    sub $t5, $t4, $s0              # 9-i
    slt $t6, $s1, $t5             # 判断 j 是否小于 9-i，如果是的话，跳转 inloop
    bnez $t6, inloop
    addi $s0, $s0, 1               # i++
    slti $t6, $s0, 9              # 判断是否 i 是否小于 9，是的话跳转 exloop:
    bnez $t6, exloop

    la $a0, msg1                   # 输出信息 msg1
    li $v0, 4

```

```

    syscall
    add $s0, $zero, $zero
loop:
    sll $s1, $s0, 2                # 循环输出排序后的信息
    add $s1, $s1, $t0
    lw  $a0, 0($s1)
    li  $v0, 1
    syscall
    la  $a0, nline
    li  $v0, 4                    # 换新一行输出
    syscall
    addi $s0, $s0, 1
    slti $t3, $s0, 10
    bnez $t3, loop
    li  $v0, 10                  # 结束程序
    syscall

```

#### 四、实验总结

这个程序的主要难点在于循环输入，双重循环的实现，主要工程在于将 C 的冒泡排序方法翻译成汇编语言。实现难度不易，需要分清每个寄存器在 C 中代表什么内容。