



《计算机组成原理实验》 实验报告

(实验一)

学院名称：数据科学与计算机学院

专业（班级）：16 计算机类 2 班

学生姓名：黄梓林

学号：16337102

时间：2017 年 10 月 15 日

成绩：

实验一：MIPS汇编语言程序设计实验

一. 实验目的

1. 初步认识和掌握MIPS汇编语言程序设计的基本方法
2. 熟悉PCSpim模拟器的使用。

二. 实验内容

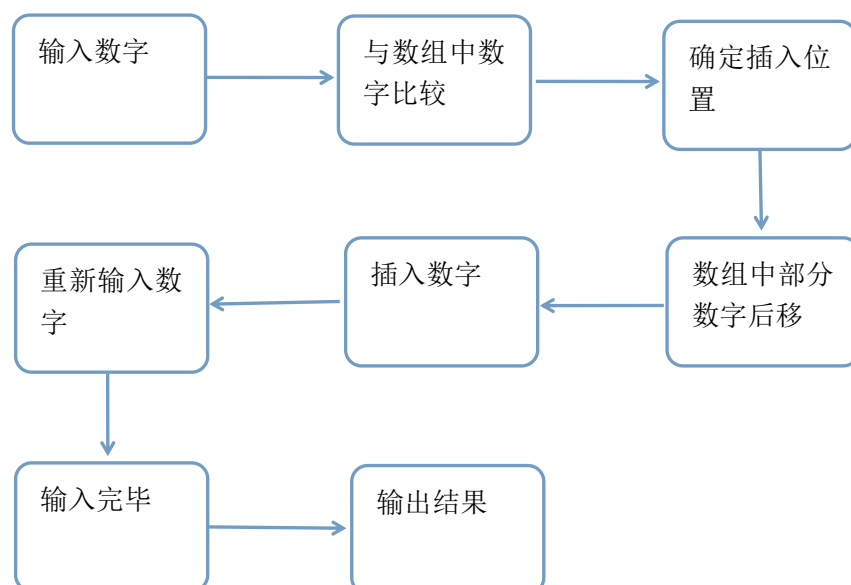
从键盘输入10个无符号字数或从内存中读取10个无符号字数并从大到小进行排序，排序结果在屏幕上显示出来

三. 实验器材

电脑一台，PCSpim 仿真器软件一套。

四. 实验过程与结果

程序流程图：



设计的思想与方法：

- 1、将实验内容分为打印提示、输入、排序、输出四部分
- 2、采用插入排序的方法，实现将输入按从大到小的顺序排序
- 3、输入、排序、输出依靠循环实现

分析：

- 1、输入的整数有10个，需设计计数器，将程序循环运行10次，当运行到第10次时，输出排序结果，退出程序
- 2、每次输入一个数字都需将其与数组中原有数字比较，需设计一个循环，将新数字与原有数字逐个比较，找出应插入的位置
- 3、在找到应插入的位置后，需将该位置后的所有数字后移一位，此步也通过循环实现；再将新数插入，返回到输入环节
- 4、在输入10个数字后，使用循环，打印数组成员，最后退出系统

实验步骤：

- 1、阅读相关资料，熟悉伪指令以及MIPS指令
- 2、构思系统实现方法，将系统功能分块
- 3、在草稿上写出各功能的实现与连接
- 4、在电脑上编写代码
- 5、检查、排除语法错误
- 6、运行系统，调试
- 7、调试通过，实验完成

实验结果：

```

Please enter an integer: 44
Please enter an integer: 56
Please enter an integer: 45
Please enter an integer: 76
Please enter an integer: 34
Please enter an integer: 23
Please enter an integer: 54
Please enter an integer: 65
Please enter an integer: 23
Please enter an integer: 54

76
65
56
54
54
45
44
34
23
23

```

输入10个数字，数字以降序输出

分析：

基本实现了排序功能，但由于在实现程序时为每个数字预留的空间为4个字节，所以，当数字过大时，会发生意想不到的错误。

五. 实验心得

1、在实验过程中，由于对伪指令以及 MIPS 指令不够熟悉，常常在编码编到一半时，又要打开课件查找一番，耗费了不少的时间，下次编程时应将使用频率较高的指令记住或摘抄下来。

2、在编程时，由于某些寄存器的需要重复利用的次数多，而某些寄存器储存了重要信息，若将两者弄混，将产生大量错误，须将当前已利用的寄存器做好归类。

3、在实验时，最初使用\$sp 开辟空间储存数字，但由于忽视了栈的调用与回收需要依据一定的次序，造成了错误，修改后选择使用.space 开辟空间

4、在调试时进入了死循环，检查代码后发现，问题是没有在 exit 中退出系统

5、在系统正常运行后，发现没有实现排序，检查代码后发现，错误发生原因是在比较数字时，将新输入的数字与原有数字的地址进行了比较，排除错误后，输出正确。

【程序代码】

```
.text
```

```
.globl main
```

```
main:
```

```

    li    $t0, 0           #初始化$t0=0,计算已经输入的数字的个数
    li    $t1, 10          #初始化$t1=10,记录所要输入数字的数量
    la    $t9, num         #将num的地址存到$t9

loop:
    beq    $t0, $t1, exit  #若$t0=10,跳转到exit

    li    $v0, 4           #打印字符串，输出
    la    $a0, str1
    syscall

    li    $v0, 5           #从键盘接收一整数，储存在$v0中
    syscall

    li    $t2, 1          #每次大循环开始时，都将$t2初始化为1
    bne    $t0, $zero, inloop #若$t0不等于0，跳转到inloop
    sw     $v0, 0($t9)     #若$t0=0，将数字储存在num中，$t9加4，$t0
加1
    addi   $t9, $t9, 4
    addi   $t0, $t0, 1
    j      loop

inloop:
    addi   $t7, $t0, 1
    beq    $t2, $t7, insert #若新加入的数为目前最大的数，跳转到insert
    sll    $t3, $t2, 2      #算出偏移量
    add    $t5, $t9, $zero  #将当前$t9储存的地址复制给$t5
    sub    $t5, $t5, $t3    # $t5储存的地址为将要同新输入的数比较的数的
地址

```

```

lw    $t8,  0($t5)
slt   $t6,   $v0,  $t8
li    $t4,   0           #初始化$t4=0,作为计数变量
bnez  $t6,   insert      #新输入的数较小, 进行插入
addi  $t2,   $t2,  1      #否则将$t2加1, 重新进入inloop循环
j     inloop

insert:                  #将新数插入, 部分原数后移
    addi $t3,  $t2,  -1    # $t3为比较次数
    slt  $t5,  $t4,  $t3   # $t4=$t3时, 执行insertend
    beqz $t5,  insertend
    sll  $t5,  $t4,  2      #计算偏移量
    sub  $t5,  $t9,  $t5    # $t9-偏移量
    addi $t6,  $t5,  -4     # $t9-偏移量-4
    lw   $t7,  0($t6)
    sw   $t7,  0($t5)
    addi $t4,  $t4,  1
    j    insert

insertend:
    sll  $t4,  $t4,  2      #计算偏移量
    sub  $t5,  $t9,  $t4    #将$v0存进适当位置
    sw   $v0,  0($t5)
    addi $t0,  $t0,  1      #表明数组元素加一
    addi $t9,  $t9,  4
    j    loop

exit:                  #打印结果
    li   $v0,  4           #打印字符串, 输出

```

```
    la    $a0, str2
    syscall

    bnez $t1, print    # $t1不等于0时，跳转到print

    li    $v0, 10      # 退出系统调用
    syscall

print:
    sll   $t3, $t1, 2    # 从数组的第一个元素开始打印
    sub   $t4, $t9, $t3
    lw    $a0, 0($t4)
    addi  $t1, $t1, -1
    li    $v0, 1
    syscall
    j     exit

.data
num:
    .space 40
str1:
    .asciiz "Please enter an integer: "
str2:
    .asciiz "\n"
```