



《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）：17 计教学 2 班

学生姓名：王程钥

学号：17341146

时间：2018 年 10 月 16 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

1、设计思路

首先写一份C语言的冒泡排序代码，如下（头文件略去）。

```
int s[10];

int main(){
    for(int i=0;i<10;i++)scanf("%d",&s[i]);
    for(int i=0;i<10;i++)
        for(int j=0;j<10-i-1;j++)
            if(s[j]<s[j+1])swap(s[j],s[j+1]);
    for(int i=0;i<10;i++)printf("%d ",s[i]);
    printf("\n");
    return 0;
}
```

本实验根据以上这份C语言代码进行设计，主要思路是将这份C语言代码进行改编，从相应的汇编语句替换C语言语句。我们可以将冒泡排序的过程分为输入，输出，循环三个部分。主要需要解决以下几个问题：

(1) 如何输入输出数据。

Mips有专门的输入输出对应的指令。调用指令 `li $v0 5`，并调用`syscall`就可以读入一个int型变量。对于其他类型的变量，将5改成其他对应的数字即可。对于输出，先将要输出的数值存入`$a1`中，后调用指令 `li $v0 1` 即可完成int类型变量的输出。其他类型的输出同理。

(2) 如何存储数据。

数据存储在内存中。我们需要将读入后的变量立即存储到内存中。我们需要为数组分配一段内存，这个操作可以通过`.space`指令实现。数组名用`$t0`表示。对于本题，我们开一个大小为40字节的数组来存储这10个变量，每次执行 `sw $v0,0($t1)` 指令即可（`$t1`将在下文解释）。

(3) 如何进行循环。

C语言中有for语句可以直接进行循环操作。在mips语言中，我们则需要自己完成循环操作。我们定义一个循环变量`$t2`表示数组当前访问到的位置（类似C语言for循环中的i变量），并定义一个循环上界，存在`$s0`中（类似C语言for循环中的n）。每次完成循环后，我们先将`$t2`加一（类似C语言for循环中的i++），让其进入到下一位，后将`$t2`与`$s0`进行比较。若`$t2 < $s0`则跳转到循环的上部继续循环，否则自动跳出循环。

至于大小的比较，C语言中使用if语句，mips中则有专门的条件跳转指令，比如`beq`，`bnq`，`blt`之类的，根据情况调用就好了。本题调用了`blt`指令。

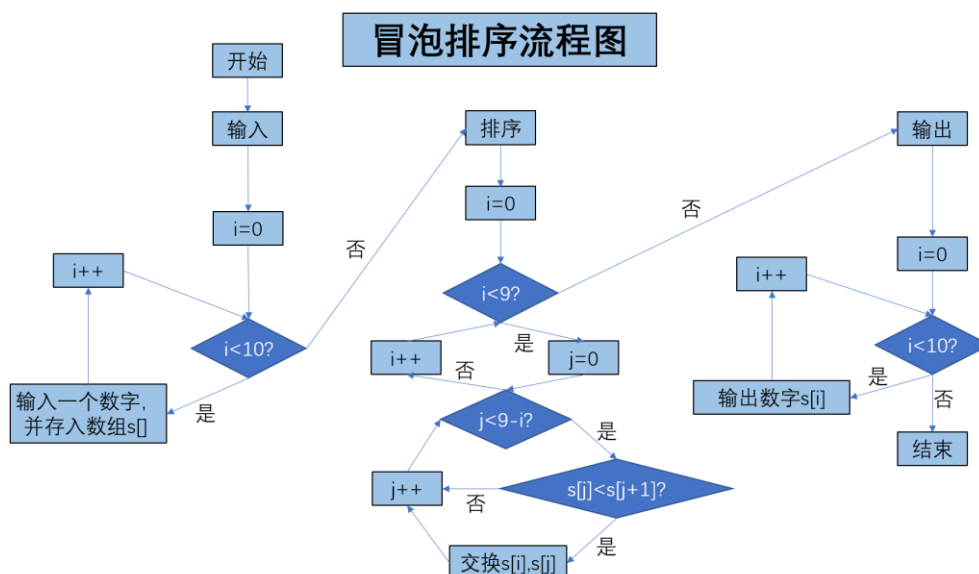
(4) 如何访问数组。

出于方便，我们开辟了一个变量专门存储当前访问的地址，并将其存在`$t1`中。每次执行一次循环，我们让`$t1=$t1+4`，即让其跳到下一个位置对应的地址。这样就可以快速访问数组内容了。

2、实验步骤

- (1) 输入待排序的数组
- (2) 将数组进行排序
- (3) 输出排序后的数组

3、流程图



如图为本实验的流程图。

4、实验结果与分析

```

Console
input the 10 integers(each num end of [enter]):
998244353
23333
1000000007
2147483647
0
100
19260817
233333
1
100
the sequence after sorting:
2147483647 1000000007 998244353 19260817 233333 23333 100 100 1 0
  
```

如图所示，输入了十个数字，从大到小排序后输出，顺序正确。

五. 实验心得

本次实验需要利用汇编语言完成冒泡排序，这个任务看起来简单，其实有一定的难度。

冒泡排序是一个很简单的算法，使用C语言去除输入输出后只需要三行代码，一分钟就可以写完。但是使用汇编语言，我整整写了差不多70行代码，写了两三个小时。

之前都是使用该机语言编程，从来没有接触过汇编语言，所以为了完成本次实验我需要学习一门全新的语言。使用一门全新的语言编程最大的难点就是对于局的不熟悉。因此为了完成本次实验，我查阅了很多资料，对于每句话尽量都找到了比较简短且合适的汇编语句。

为了完成本次实验，我先写了一份C语言的代码，并根据这份代码理清了汇编实现的主要框架和思路，将这份C语言代码一块一块翻译成汇编语言。

汇编语言相较于C语言等高级语言最大的不同就是，它直接针对内存进行操作，比C语言繁琐得多。如上文，C语言3行的代码在汇编李写了差不多70行。但汇编语言作为一项底层的语言，学习他还是很有必要的。学习汇编语言可以让我们对计算机的底层架构有更层次的了解，对计算机底层的原理有更深入地了解和认识，让我提升对计算机的理解。

【程序代码】

见附件 slowsort.asm