



《计算机组成原理与接口技术实验》 实验报告

(实验一)

学 院 名 称 : 数据科学与计算机学院

学 生 姓 名 : 张吉祺

学 号 : 16340286

专业 (班级) : 16 软件工程四 (7) 班

时 间 : 2018 年 4 月 10 日

成 绩 :

实验一：MIPS汇编语言程序设计

一. 实验目的

1. 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

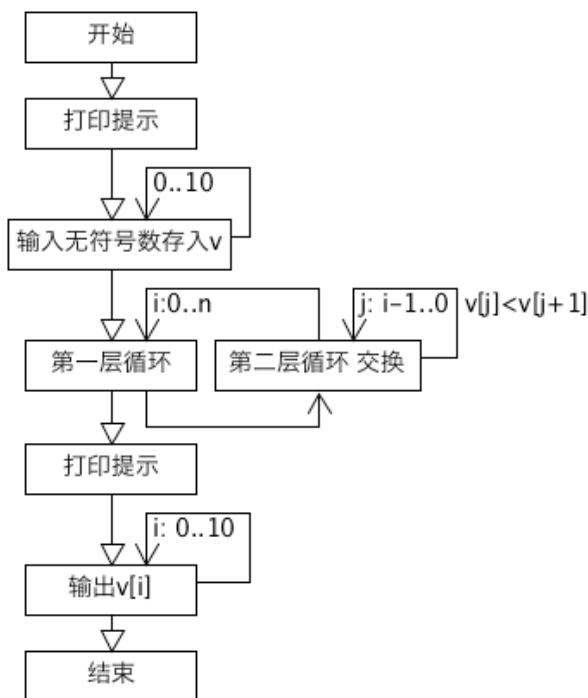
电脑一台、PCSpim模拟器软件一套。

四. 实验分析与设计

1. 先写出C风格的过程代码：

```
void swap (int v[], int n) {  
    int t0, t1;  
    t0 = v[n];  
    t1 = v[n + 1];  
    v[n] = t1;  
    v[n + 1] = t0;  
}  
  
void sort (int v[], int n) {  
    int i, j;  
    for (i = 0; i < n; i += 1) {  
        for (j = i - 1; j >= 0 && v[j] < v[j + 1]; j -= 1) {  
            swap(v, j);  
        }  
    }  
}
```

2. 画程序流程图



3. 设计思路和方法

程序的核心部分在于排序。对于排序，选择了较容易实现的冒泡排序。冒泡排序主要行为是数组相邻元素的交换。交换的实现是根据提供的数组地址和要交换的第一个元素的序号。由于int占四个字节，所以偏移量是序号*4，即序号左移2位，偏移量再加上数组地址就是要交换的第一个元素的地址。然后读取该元素的值存入临时变量，再用该地址+4获取相邻元素的值存入新的临时变量，再将两个临时变量的值分别存入第二个和第一个地址所指的内存中。冒泡排序还要实现两层循环，对值的读取类似上述方法。第一层循环从左到右遍历了数组元素的值。第二层循环在第一层循环从元素所在位置从右到左，当右元素大于左元素就进行交换，否则退出第二层循环（因为此时前面的元素已经在之前循环时排完序了）。

4. 实验结果及分析

Console

请输入10个无符号数字，每个数后加回车：

1

-1

请输入无符号数

3

4

3

8

90

666

123

321

987

排序结果：

987 666 321 123 90 8 4 3 3 1

能处理输入为负数的情况，要求用户重新输入，能实现从大到小排序。

五. 实验心得

1. 函数的传参问题

在整个程序中调用了多个函数，而函数的传入参数一般先依次存入从\$a0开始的寄存器，为了避免冲突和保持上下文一致应该在必要时将参数存入内存栈中，如果进行递归调用还要存入当前\$ra的值，并在调用返回时将参数从栈中恢复，并退回栈指针。

2. 函数的执行流程

与高级语言不同，汇编语言的jal跳转指令跳转后并不会自动跳回来，而是将当前pc存入\$ra，需要手动读取\$ra的值跳转回来。

3. 地址对齐问题

当在.data中最先分配的并不是数组v时，用指令

```
sw      $v0,    0($t0)
```

存入一个数会报Exception 5 [Address error in store] occurred and ignored的警告，由于sw是存入一个字，运行中查看了内存中对应地址的情况，确实没有在该地址的字从头对齐，这是因为.data中处于数组v之前分配的内存占用了该字的一部分导致并不是一个完整的空白可用的字。解决方法可以是

v:

```
.align 4
```

```
.space 40
```

加上align进行对齐，或者直接在.data中优先分配数组v，就不会遇到对齐问题了。

4. 区分好la和lw

la可通过.data中变量名取地址，lw用于通过内存地址取值。

【程序代码】

打包