

实验【选做】 整数数组的冒泡排序

一、 实验目的

- 1、熟悉汇编语言的整数数组；
- 2、熟悉基址变址操作数、相对基址变址操作数；
- 3、掌握排序算法的底层实现细节

二、 实验环境

Windows 记事本的汇编语言编写环境

MASM32 编译环境

Windows 命令行窗口

三、 实验原理

同学们已经使用 C++ 高级语言实现了搜索和排序算法。汇编语言能够清晰地看到底层的实现细节，使得汇编语言为研究算法提供了另一个视角。20 世纪最著名的算法学家之一 Donald Knuth 在其出版的《The Art of Computer Programming》中也是使用汇编语言来编写例子的。

排序算法中，汇编语言的基址变址寻址方式和相对基址变址寻址方式起到了重要的作用。

基址变址（base-index）操作数把两个寄存器的值相加，得到一个偏移地址。两个寄存器分别称为基址寄存器（base）和变址寄存器（index）。格式为[base + index]，例如 `mov eax, [ebx + esi]`。在例子中，`ebx` 是基址寄存器，`esi` 是变址寄存器。基址寄存器和变址寄存器可以使用任意的 32 位通用寄存器。

相对基址变址（based-indexed with displacement）操作数把偏移、基址、变址以及可选的比例因子组合起来，产生一个偏移地址。常见的两种格式为：[base + index + displacement]和 displacement[base + index]，例子如下：

```
table dword 10h, 20h, 30h, 40h
```

```
row_size = ($ - table)
```

```
    dword 50h, 60h, 70h, 80h
```

```
    dword 90h, 0a0h, 0b0h, 0c0h
```

```
mov ebx, row_size  
  
mov esi, 2  
  
mov eax, table[ebx + esi * 4]
```

table 是一个二维数组，共 3 行 4 列。ebx 是基址寄存器，相当于二维数组的行索引，esi 是变址寄存器，相当于二维数组的列索引。

冒泡排序算法(Bubble Sort)的过程是从位置 0 和 1 开始比较每对数据的值，如果两个数据的顺序不对，就进行交换。如果一遍处理完之后，数组没有排好序，就开始下一次循环。在最多完成 n-1 次循环后，数组排序完成。

四、 实验内容

本次实验要求编写汇编程序 bubble_sort.asm，功能是将 Windows 命令行输入的 10 个 1 万以内的十进制无符号整数，进行排序，然后输出在 Windows 命令行中。10 个无符号整数之间用逗号","或者空格" "分割。

使用 StdIn 函数获得用户输入的十进制整数序列。StdIn 函数的定义在 \masm32\include\masm32.inc，库文件是 \masm32\lib\masm32.lib。StdIn 函数的定义“StdIn PROTO :DWORD,:DWORD”，有两个参数，第一个是内存存储空间的起始地址，第二个是内存存储空间的大小。

使用 StdOut 函数在 Windows 命令中输出排好序的十进制整数序列。StdOut 函数的定义在 \masm32\include\masm32.inc，库文件是 \masm32\lib\masm32.lib。StdOut 函数的定义“StdOut PROTO :DWORD”，只有一个参数，是内存存储空间的起始地址。

使用 ml 和 link 程序将源代码编译、链接成可执行文件 bubble_sort.exe。

五、 实验报告

1. 实验设计报告。
2. bubble_sort.asm 源代码和注释。
3. bubble_sort.exe 的测试过程和截图。
4. 汇编语言数组操作知识点的总结。