# 汇编语言与逆向技术课程实验报告

## 实验三：bubble_sort 程序


学院：密码与网络空间安全学院

专业：信息安全

学号：2412950

姓名：路浩斌

班级：信安一班

# 一、实验目的

1、熟悉汇编语言的数据传送、寻址方式和算术运算

2、熟悉汇编语言过程的定义和使用

3、学习如何实现冒泡排序

# 二、实验原理

- 使用 StdIn 函数获得用户输入的十个小于 10000 的十进制整数
- 用户输入的十进制数对应的 ASCII 编码字符串存储在内存中,编写过程 *my_proc*1, 将 ASCII 字符串转换成 DWORD 数据, 并写入 *number* 数组中。
- 编写过程 *my_sort*，使用冒泡排序将 *number* 数组中的 10 个数进行排序。
- 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。
- 编写过程 *my_proc*2 将数组转成字符串并进行输出。
- 使用 ml 将 bubble_sort.asm 文件汇编到 bubble_sort.obj 目标文件，使用 link 将目标文件 bubble_sort.obj 链接成 bubble_sort.exe 可执行文件。

# 三、实验过程

## 源程序

```
.386
.model flat,stdcall
option casemap:none

include C:\masm32\include\windows.inc
include C:\masm32\include\kernel32.inc
include C:\masm32\include\masm32.inc
includelib C:\masm32\lib\masm32.lib
includelib C:\masm32\lib\kernel32.lib

.data
var1 db "Please␣input␣10␣numbers(0-10000):",0Dh,0Ah,0
```

```
13 var2 db "The␣result␣is:",0Dh,0Ah,0
14 buf db 100 DUP(0);输入缓冲区
15 number dd 10 DUP(0);数组
16 outbuf db 100 DUP(0);输出字符串
17 tempbuf db 12 DUP(0);单个整数转成字符串的缓冲区
18 crlf db 0Dh,0Ah,0
19
20 .code
21 ;my_proc1过程将输入的字符转成十进制整数数组（数组大小为10）
22 my_proc1 PROC
23 lea esi,buf
24 lea edi,number
25 xor eax,eax
26 ;一个一个字符的处理一个整数字符串
27 Nextchar:
28 mov bl,[esi]
29 ;检查该字符是否为结束字符或换行
30 cmp bl,0
31 je EndConv
32 cmp bl,13
33 je EndConv
34 ;检查该字符是否为空格
35 cmp bl,"␣"
36 je Nextnum
37 ;转成十进制整数
38 imul eax,eax,10
39 movzx ebx,bl
40 sub ebx,'0'
41 add eax,ebx
42 inc esi
43 jmp Nextchar
44 ;处理下一个整数字符串
45 Nextnum:
46 mov [edi],eax
47 add edi,4
48 xor eax,eax
49 inc esi
50 jmp Nextchar
51 ;全部转成了十进制整数，结束
52 EndConv:
```

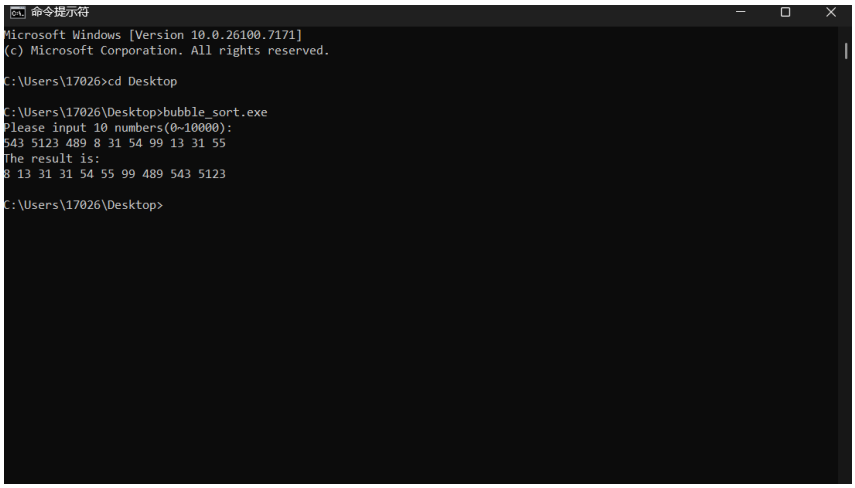
```
53 mov [edi],eax
54 ret
55 my_proc1 ENDP
56
57 ;将转化好的数进行排序
58 my_sort PROC
59 mov ecx,10
60 ;外层循环
61 outer_loop:
62 dec ecx
63 cmp ecx,0
64 jl EndSort
65 xor esi,esi
66 ;内层循环
67 inner_loop:
68 mov eax,[number+esi*4]
69 mov ebx,[number+esi*4+4]
70 cmp eax,ebx
71 jbe Noswap
72 mov [number+esi*4],ebx
73 mov [number+esi*4+4],eax
74 ;不需要交换的情况
75 Noswap:
76 inc esi
77 cmp esi,ecx
78 jl inner_loop
79 jmp outer_loop
80 ;排序完成
81 EndSort:
82 ret
83 my_sort ENDP
84
85 ;将排序好的数组转成字符串输出
86 my_proc2 PROC
87     push ebx
88     push esi
89     push edi
90     push ebp
91
92     lea esi, number
```

```
    lea edi, outbuf
    mov ecx, 10
;采用除十取余的方法将整数数组转化成字符串
convert_next_num:
    mov eax, [esi]
    add esi, 4

    lea ebx, tempbuf+11
    mov byte ptr [ebx], 0
    dec ebx

convert_digit:
    xor edx, edx
    mov ebp, 10
    div ebp
    add dl, '0'
    mov [ebx], dl
    dec ebx
    cmp eax, 0
    jne convert_digit
    inc ebx

copy_to_outbuf:
    mov al, [ebx]
    mov [edi], al
    inc ebx
    inc edi
    cmp al, 0
    jne copy_to_outbuf
    dec edi
    dec ecx
    cmp ecx, 0
    je finish

    mov byte ptr [edi], '␣'
    inc edi
    jmp convert_next_num

finish:
    mov byte ptr [edi], 0
```

```
133
134     pop ebp
135     pop edi
136     pop esi
137     pop ebx
138     ret
139 my_proc2 ENDP
140
141 ;主函数
142 start:
143 invoke StdOut,addr var1
144 invoke StdIn,addr buf,100
145 call my_proc1
146 call my_sort
147 invoke StdOut,addr var2
148 call my_proc2
149 invoke StdOut, addr outbuf
150 invoke StdOut,addr crlf
151 invoke ExitProcess,0
152 END start
```

## 实验步骤

1. **编辑**：使用文本编辑软件（如 Notepad）编写源程序文件（.asm）。
    - bubble_sort.asm
2. **编译**：使用 MASM 汇编程序 `ml.exe` 对源程序进行汇编，生成目标文件（.obj）。
    - C: \masm32 \bin \ml /c /Zd /coff bubble_sort.asm
3. **链接**：使用链接程序 `link.exe` 对目标文件进行链接，生成可执行文件（.exe）。
    - C:\masm32 \bin \Link /SUBSYSTEM:CONSOLE bubble_sort.obj
4. **执行**：在命令提示符（/cmd) 上执行可执行文件，观察屏幕显示结果：




**测试说明：**运行 bubble_sort.exe 后，在命令提示符中输入十个小于 10000 的十进制数程序能够正确排序并进行输出，验证了 $my\_proc1$ ,$my\_sort$ 和 $my\_proc2$ 三个过程的功能正确，程序逻辑符合预期。

## 汇编语言数组操作

数组的基本操作有定义、访问、遍历、排序和转字符串等，主要知识点总结如下：

- **数组定义与内存分配**：使用 db、dw、dd 等定义数组，并使用 DUP 初始化。例如：

```
number dd 10 DUP(0)   ; 定义10个DWORD元素的数组
buf    db 100 DUP(0) ; 输入缓冲区
outbuf db 100 DUP(0) ; 输出缓冲区
```

数组在内存中连续存储，低地址对应首元素。

- **数组指针访问**：使用 LEA 指令初始化指针，结合偏移量访问数组元素：

```
lea esi, number
mov eax, [esi + index*4] ; 读取第 index 个DWORD元素
mov [edi + index*4], eax ; 写入第 index 个元素
```

- **数组遍历**：
  - 使用循环计数寄存器（如 ECX）控制循环次数。
  - 使用数组元素值判断循环结束，如结束字符 0，回车换行字符等。
- **数组元素的读写**：
  - 从数组读：mov 寄存器，[数组地址 + 偏移]。
  - 向数组写：mov [数组地址 + 偏移]，寄存器。

- **数组交换**：在排序中，先把元素读到寄存器，再根据大小判断是否交换，最后写回数组：

```
mov eax, [number + esi*4]
mov ebx, [number + esi*4 + 4]
cmp eax, ebx
jbe Noswap;不交换的情况跳转
mov [number + esi*4], ebx
mov [number + esi*4 + 4], eax
```

- **数组转字符串**：整数数组转字符串，需使用临时缓冲区逐位转换：

  1. 取整数元素到寄存器。
  2. 使用除法取余法拆分每一位数字。
  3. 将数字字符写入临时缓冲区，再复制到输出缓冲区。

- **总结**：

  - 汇编语言数组操作可以理解为“寄存器 + 偏移量”访问内存。
  - 要擅用临时寄存器和开辟临时缓冲区
  - 需要手动控制边界，防止出界

# 四、实验结论及心得体会

**实验结论：**

- 编写的控制台程序能够正确将用户输入的十个十进制数进行排序并输出，程序功能实现正确。

**心得体会：**

- 熟悉了 MASM32 提供的基本函数调用、基本变址寻址方式、相对基址变址寻址方式，为后续汇编实验打下了基础。