

计算机组成原理 实验报告
PB14000556 陈晓彤

实验题目：基于 16 位 x86 汇编的简单冒泡排序

实验时间：2016.3.30

实验环境：

windows xp 虚拟机；
文本编辑器 notepad++ 写汇编源代码；
MASM 5.0 进行汇编和链接；
cmd-debug 调试运行

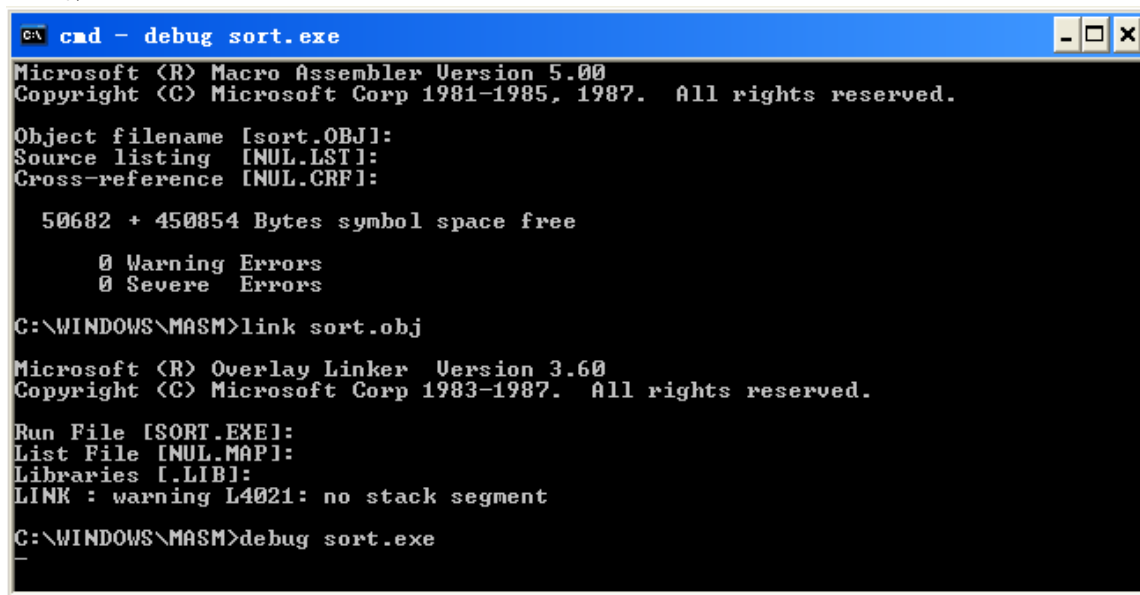
实验准备：

实验内容很简单，两层循环实现冒泡排序，循环次数，当前内存地址的数值各自选择寄存器来保存，每次比较相邻两地址的数值，注意不能直接在两个内存地址间交换；搭建环境用老师推荐的 debug，写 16 位的代码；查阅了一些书籍和网上的资料，基本掌握了 8086 体系架构特点和汇编语言语法规则之后，开始正式实验；

实验过程：

编写程序：数据段第一个数记录排序数目，之后存放目标数值；
代码段由 CX 记录外层循环数，BX 记录内层循环数，AX 存当前的数值；
比较后若不符合升序则交换相邻的两个元素，这里使用 XCHG 指令能减少一条指令；
根据要求，排序的数字体现个人信息，如截图所示，排序前，十个数字分别为“PB14000556”的 ASCII 码，排序后，变为“00014556BP”，文件命名为 sort.asm；
编译链接：将 sort.asm 放入 masm 文件夹中，运行 masm sort.asm 和 link sort.obj 命令，生成可执行文件 sort.exe；
调试阶段：进入 debug 调试，先-u 检查代码是否与源文件一致，发现吻合后，看到机器为 data 段分配地址为 0B42，输入-d 0B42:0 检查数据，发现与预想吻合，-g 执行程序，再打-d 0B42:0 检查数据，发现已经排好顺序，实验结果正确
具体情况见截图

过程截图：



```
C:\>cmd - debug sort.exe

Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [sort.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50682 + 450854 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\WINDOWS\MASM>link sort.obj

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [SORT.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\WINDOWS\MASM>debug sort.exe
```

```
C:\cmd - debug sort.exe
C:\WINDOWS\MASM>link sort.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [SORT.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment

C:\WINDOWS\MASM>debug sort.exe
-u
0B44:0000 B8420B      MOV     AX,0B42
0B44:0003 8ED8             MOV     DS,AX
0B44:0005 BB0000      MOV     BX,0000
0B44:0008 8B8F0000      MOV     CX,[BX+0000]
0B44:000C 49              DEC     CX
0B44:000D 03C9           ADD     CX,CX
0B44:000F 8BD9           MOV     BX,CX
0B44:0011 8B870000      MOV     AX,[BX+0000]
0B44:0015 3B870200      CMP     AX,[BX+0002]
0B44:0019 7608           JBE     0023
0B44:001B 87870200      XCHG    AX,[BX+0002]
0B44:001F 89870000      MOV     [BX+0000],AX
-
C:\cmd - debug sort.exe
0B44:0005 BB0000      MOV     BX,0000
0B44:0008 8B8F0000      MOV     CX,[BX+0000]
0B44:000C 49              DEC     CX
0B44:000D 03C9           ADD     CX,CX
0B44:000F 8BD9           MOV     BX,CX
0B44:0011 8B870000      MOV     AX,[BX+0000]
0B44:0015 3B870200      CMP     AX,[BX+0002]
0B44:0019 7608           JBE     0023
0B44:001B 87870200      XCHG    AX,[BX+0002]
0B44:001F 89870000      MOV     [BX+0000],AX
-u
0B44:0023 83EB02      SUB     BX,+02
0B44:0026 83FB02      CMP     BX,+02
0B44:0029 73E6       JNB     0011
0B44:002B 83E902      SUB     CX,+02
0B44:002E 83F902      CMP     CX,+02
0B44:0031 73DC       JNB     000F
0B44:0033 B44C       MOV     AH,4C
0B44:0035 CD21      INT     21
0B44:0037 FF508D      CALL    [BX+SI-73]
0B44:003A 8600       XCHG    AL,[BX+SI]
0B44:003C FF508D      CALL    [BX+SI-73]
0B44:003F 46         INC     SI
0B44:0040 8050E84D   ADC     BYTE PTR [BX+SI-18],4D
-
```

```

C:\>cmd - debug sort.exe
0B44:003D 50          PUSH    AX
0B44:003E 8D4680         LEA     AX,[BP-80]
0B44:0041 50          PUSH    AX
-d 0b42:0
0B42:0000 0A 00 50 00 42 00 31 00-34 00 30 00 30 00 00 00  ..P.B.1.4.0.0.0.
0B42:0010 35 00 35 00 36 00 00 00-00 00 00 00 00 00 00 00  5.5.6.....
0B42:0020 B8 42 0B 8E D8 BB 00 00-8B 8F 00 00 49 03 C9 BA  .B.....I...
0B42:0030 02 00 8B D9 8B 87 00 00-3B 87 02 00 76 08 87 87  .....;...v...
0B42:0040 02 00 89 87 00 00 83 EB-02 3B DA 73 E7 83 C2 02  .....;...s...
0B42:0050 3B D1 76 DE B4 4C CD 21-50 8D 86 00 FF 50 8D 46  ;...L.?P....P.F
0B42:0060 80 50 E8 4D FA 83 C4 06-0A C0 75 03 E9 7B FF 5E  .P.M.....u.<.^
0B42:0070 8B E5 5D C3 83 3E 56 07-20 72 0A B8 1C 04 50 E8  ..l..>U. r....P.
-g
Program terminated normally
-d 0b42:0
0B42:0000 0A 00 30 00 30 00 30 00-31 00 34 00 35 00 35 00  ..0.0.0.1.4.5.5.
0B42:0010 36 00 42 00 50 00 00 00-00 00 00 00 00 00 00 00  6.B.P.....
0B42:0020 B8 42 0B 8E D8 BB 00 00-8B 8F 00 00 49 03 C9 BA  .B.....I...
0B42:0030 02 00 8B D9 8B 87 00 00-3B 87 02 00 76 08 87 87  .....;...v...
0B42:0040 02 00 89 87 00 00 83 EB-02 3B DA 73 E7 83 C2 02  .....;...s...
0B42:0050 3B D1 76 DE B4 4C CD 21-50 8D 86 00 FF 50 8D 46  ;...L.?P....P.F
0B42:0060 80 50 E8 4D FA 83 C4 06-0A C0 75 03 E9 7B FF 5E  .P.M.....u.<.^
0B42:0070 8B E5 5D C3 83 3E 56 07-20 72 0A B8 1C 04 50 E8  ..l..>U. r....P.

```

运行时间：0.001s

源代码：

```

DSEG SEGMENT
    NUMBERS    DW    10,50H,42H,31H,34H,30H,30H,30H,35H,35H,36H
DSEG ENDS

CSEG SEGMENT
    ASSUME     DS:DSEG,CS:CSEG

START:
    MOV     AX,DSEG
    MOV     DS,AX
    MOV     BX,0
    MOV     CX,WORD PTR    NUMBERS[BX]
    DEC     CX
    ADD     CX,CX
    MOV     DX,2

OUTLOOP:  MOV     BX,CX
INLOOP:   MOV     AX,NUMBERS[BX]
          CMP     AX,NUMBERS[BX+2]
          JBE     NOT_SWAP
          XCHG    AX,NUMBERS[BX+2]
          MOV     NUMBERS[BX],AX
NOT_SWAP: SUB     BX,2
          CMP     BX,DX
          JAE     INLOOP
          ADD     DX,2
          CMP     DX,CX
          JBE     OUTLOOP

```

```
MOV  AH,4CH  
INT 21H
```

```
CSEG  ENDS
```

```
END START
```