|  |
| --- |
| 哈尔滨工业大学（深圳） |
| **汇编语言与接口技术 实验报告** |
| **实验三：子程序设计实例** |
| 学号：220111012  姓名：王靳  日期：2024.11.6 |

目录

[必做题 1](#_Toc119706802)

[1问题描述 1](#_Toc119706803)

[2解决方案和代码 1](#_Toc119706804)

[3调试过程与结果 1](#_Toc119706805)

[选做题 1](#_Toc119706806)

[1问题描述 1](#_Toc119706807)

[2解决方案及代码 1](#_Toc119706808)

[3调试过程与结果 1](#_Toc119706809)

[总结 1](#_Toc119706810)

# 必做题

## 1问题描述

给定一个字节类型的无序数组，请将该数组按从小到大的顺序排序，并计算数组之和。要求：

1.排序用子程序实现；

2.求和计算用宏实现。

数据段定义如下：

data segment

a db 10,6,24,39,33,77,15,100,36,4 ;数组

n db 10 ；数字个数

result dw ? ;数组和

data ends

单步调试，查看结果，或者用dos功能调用输出排序结果。

## 2解决方案和代码

冒泡排序，好在之前ppt上有示例。个人感觉比较重要的是：一定要设置栈，这样才有可能有函数的调用。然后这里的调用惯例采用了类似于cdecl的调用惯例，也就是参数全部采用栈来传递。我这里的cx作为outer\_loop的迭代器，dx作为inner\_loop的迭代器。Si指向数组，si在inner\_loop开始的时候进行初始化，指向数组首地址，inner\_loop每次结束的时候，step，si++。

DATA SEGMENT

a DB 10,6,24,39,33,77,15,100,36,4

n DB 10

result DW ?

DATA ENDS

STACK SEGMENT

DB 256 DUP (0)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

SUM\_ARRAY MACRO

PUSH AX

PUSH BX

PUSH CX

PUSH SI

MOV CX, n

LEA SI, a

XOR AX, AX

SUM\_LOOP:

MOV BL, [SI]

MOV BH, 0

ADD AX, BX

INC SI

LOOP SUM\_LOOP

MOV result, AX

POP SI

POP CX

POP BX

POP AX

ENDM

SORT PROC

PUSH BP

MOV BP, SP

PUSH SI

PUSH AX

PUSH CX

PUSH DX

MOV CX, n

DEC CX

OUTER\_LOOP:

MOV SI, OFFSET a

MOV DX, CX

INNER\_LOOP:

MOV AL, [SI]

MOV AH, [SI+1]

CMP AL, AH

JLE NO\_SWAP

MOV [SI], AH

MOV [SI+1], AL

NO\_SWAP:

INC SI

DEC DX

JNZ INNER\_LOOP

DEC CX

JNZ OUTER\_LOOP

POP DX

POP CX

POP AX

POP SI

POP BP

RET

SORT ENDP

START:

MOV AX, DATA

MOV DS, AX

MOV AX, STACK

MOV SS, AX

MOV SP, 256

CALL SORT

SUM\_ARRAY

MOV AH, 4CH

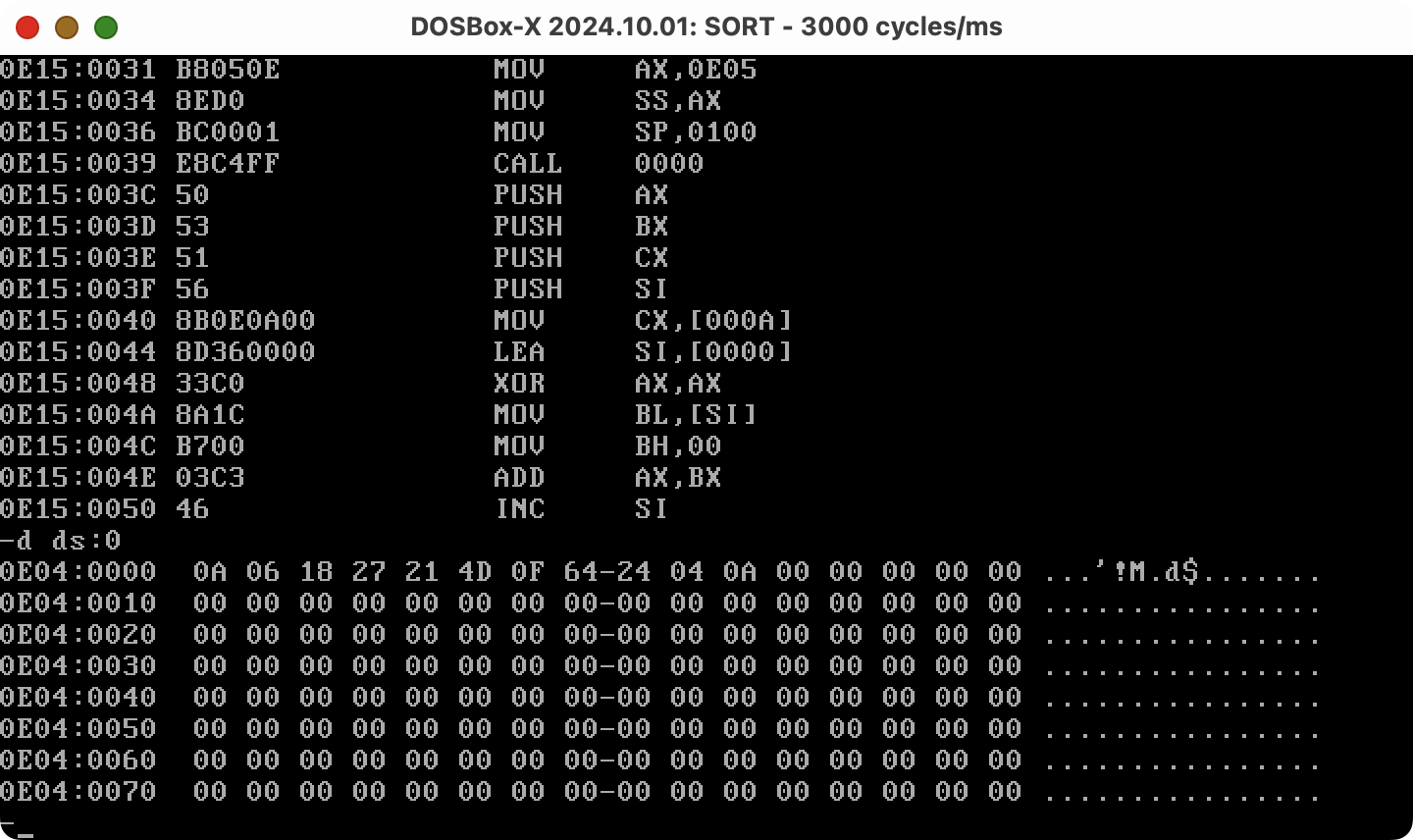
INT 21H

CODE ENDS

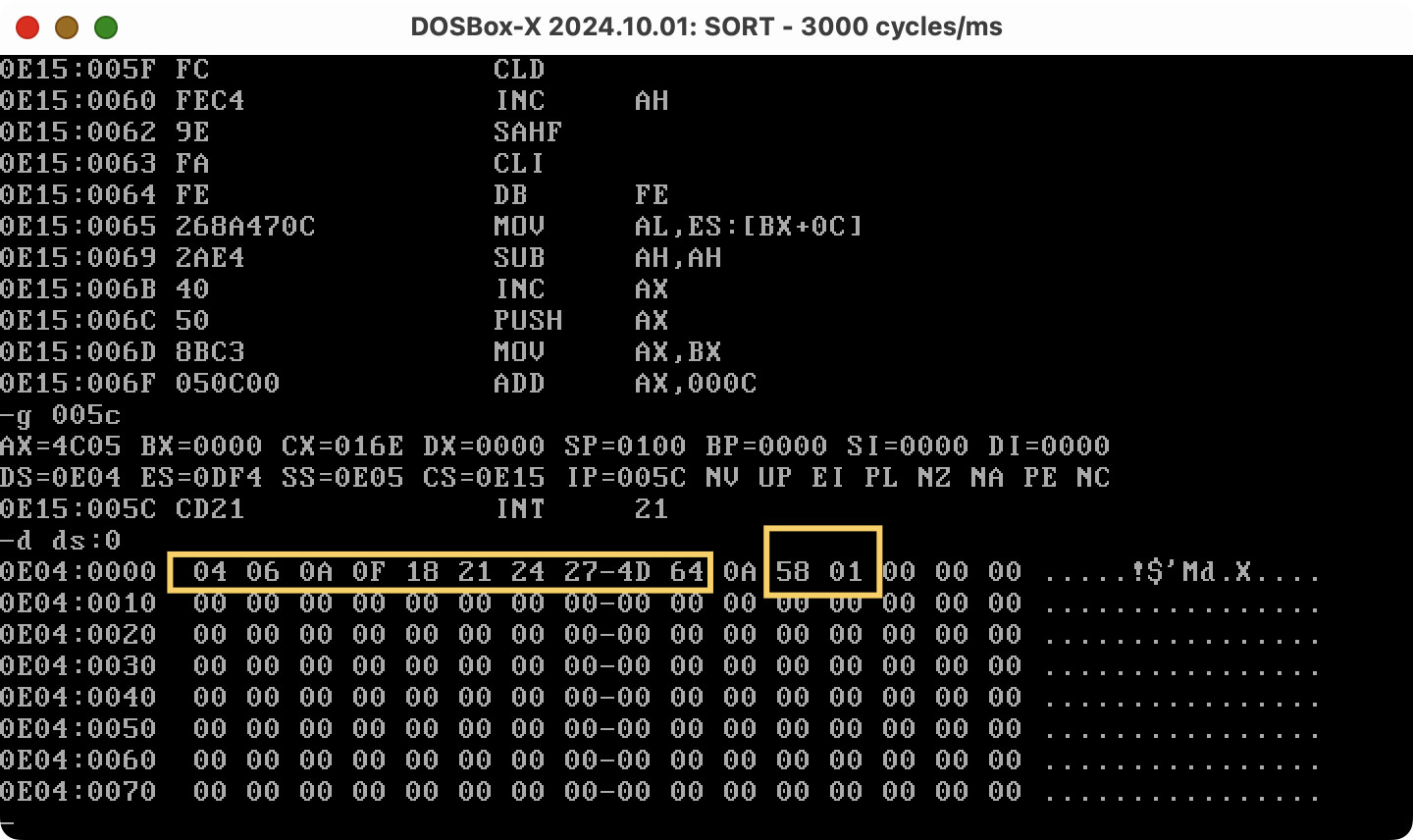
END START

## 3调试过程与结果

初始时，可以看到是：乱序的，并且最后并没有计算出来



执行完成后，可以看到，前面10个是有序的，并且可以看到计算结果



# 总结

一个建议，感觉x86汇编与8086汇编还是有一些区别的。然后dos未免过于老了，我认为，或许教授system v abi调用惯例或许会更好，衔接上后面的csapp课程。