МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Технічна Кібернетика»

**Лабораторна робота №1**

**з дисципліни «Основи программування»**

**на тему: «**Розробка та виконання програми лінійної структури на мові С++. Середовище розробки та виконання програм Microsoft Visual Studio.**»**

Виконав:

студент гр.ПЗ1911

Сафонов Д.Є.

Прийняла:

Нежуміра О.И.

Дніпро, 2019

**Тема.** Команди пересилання та обміну даними в мові Ассемблера.

**Мета.** Вивчити принципи організації пам’яті комп’ютера, адресації даних. Ознайомитись з поняттям ефективної, логічної і фізичної адреси. Вивчити регістри центрального процесора, їх призначення і взаємозв’язок. Вивчити команди пересилання і обміну даними, набути практичних навичок їх використання.

***1.Задача:***

Розробити програму для обробки інформації, що міститься в сегменті даних:

1) змінити текст згідно завданню обробки тексту (колонка «Обробка тексту» в таблиці);

2) записати в пам’ять логічну адресу (сегментну адресу і зміщення) другого числа (колонка «Змінні»);

3) переставити місцями старшу і молодшу частини чисел (колонка «Змінні») у форматі слова (двома способами) і подвійного слова (одним способом). Всі три способи обміну частин чисел повинні бути різними (команди пересилання, команда обміну, стек). Результати записати в нові змінні.

Таблиця 1

|  |  |  |
| --- | --- | --- |
| № | Змінні | Обробка тексту |
| 18 | text        db ‘Don’t bite off more than you can chew.’  numbers dd -193758                db 222                dw 45005 | Поміняти місцями слова «more» і «than». |

***2. Лістинг програми;***

Turbo Assembler Version 3.0 10/07/19 11:37:31 Page 1

2v4.asm

1 ﻿;NAME CODE OPERAND

2

3 ;сегмент стека

4 0000 sseg segment stack 'stack'

5 0000 80\*(??) db 128 dup(?)

6 0080 sseg ends

7

8 ; DATA WRITING TO SEGMENT

9 0000 dseg segment 'data'

10

11 ; текст

12

13 0000 44 6F 6E 27 74 20 62+ text db "Don't bite off more than you can chew.";38bytes(0-37)

14 69 74 65 20 6F 66 66+

15 20 6D 6F 72 65 20 74+

16 68 61 6E 20 79 6F 75+

17 20 63 61 6E 20 63 68+

18 65 77 2E

19

20 0026 FFFD0B22 numbers dd -193758;0 offset

21 002A DE db 222;4 bytes offset

22 002B AFCD dw 45005;5 bytes offset

23

24 002D ???????? ef\_ad222 dd ?;effective address of 222 in numbers

25 0031 ???????? A dd ?;new number1

26 0035 ???? B dw ?;new number3

27 0037 ???? C dw ?;new number3v2

28

29 0039 dseg ends ;DATA WRITING END

30

31 ;CODE SEGMENT

32 0000 cseg segment 'code'

33 assume cs:cseg, ds:dseg, ss:sseg

34

35 ;мітка початку програми з ім’ям start

36 0000 B8 0000s start: mov ax, dseg

37 0003 8E D8 mov ds, ax

38

39 ;body

40 0005 FF 36 0011r push word ptr [text+17];"re" to stack

41 0009 FF 36 000Fr push word ptr [text+15];"mo" to stack

42 000D FF 36 0016r push word ptr [text+22];"an" to stack

43 0011 FF 36 0014r push word ptr [text+20];"th" to stack

44

45 0015 8F 06 000Fr pop word ptr [text+15];"th" to A+15

46 0019 8F 06 0011r pop word ptr [text+17];"an" to A+17

47 001D 8F 06 0014r pop word ptr [text+20];"mo" to A+20

48 0021 8F 06 0016r pop word ptr [text+22];"re" to A+22 ~end of task1

49

50 0025 B8 0028r lea ax, byte ptr [numbers+2];/effective address of 222 in numbers/ to ax

51 0028 BA 2000 mov dx, 2000h;

52 002B 89 16 002Dr mov word ptr [ef\_ad222], dx;

53 002F A3 002Fr mov word ptr [ef\_ad222+2], ax;/effective address of 222 in numbers/ to /ef\_ad222/ ~end of task2

54

55 0032 A1 0028r mov ax, word ptr [numbers+2]

56 0035 A3 0031r mov word ptr [A], ax

57 0038 A1 0026r mov ax, word ptr [numbers]

Turbo Assembler Version 3.0 10/07/19 11:37:31 Page 2

2v4.asm

58 003B A3 0033r mov word ptr [A+2], ax;exchanging lower and upper byte ~end of task 3.1

59

60 003E FF 36 002Br push word ptr [numbers+5];/45005/ to stack

61 0042 8F 06 0035r pop B;/45005/ to B

62 0046 A1 0035r mov ax, B;/45005/ to ax

63 0049 86 E0 xchg ah, al;exchanging upper and lower byte

64 004B A3 0035r mov B, ax;/45005/ with swapped upper and lower bytes to B ~end of task3.2

65

66 004E FF 36 002Br push word ptr [numbers+5];/45005/ to stack

67 0052 58 pop ax;/45005/ to ax

68 0053 86 C4 xchg al, ah;exchanging upper and lower byte

69 0055 A3 0037r mov C, ax;~end of task 3.3

70

71 ;повернення керування ОС

72 0058 B4 4C mov ah, 4Ch

73 005A CD 21 int 21h

74 005C cseg ends

75 end start ;END

Turbo Assembler Version 3.0 10/07/19 11:37:31 Page 3

Symbol Table

Symbol Name Type Value

??DATE Text "10/07/19"

??FILENAME Text "2v4 "

??TIME Text "11:37:31"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CSEG

@FILENAME Text 2V4

@WORDSIZE Text 2

A Dword DSEG:0031

B Word DSEG:0035

C Word DSEG:0037

EF\_AD222 Dword DSEG:002D

NUMBERS Dword DSEG:0026

START Near CSEG:0000

TEXT Byte DSEG:0000

Groups & Segments Bit Size Align Combine Class

CSEG 16 005C Para none CODE

DSEG 16 0039 Para none DATA

SSEG 16 0080 Para Stack STACK

***3. Аналіз результатів виконання програми на основі скрінів екрану;***

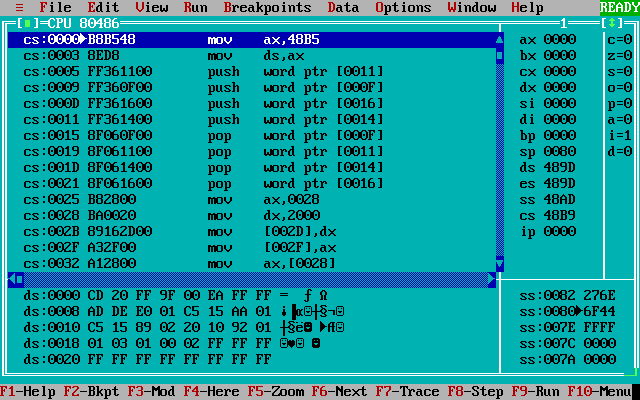


Рисунок 1

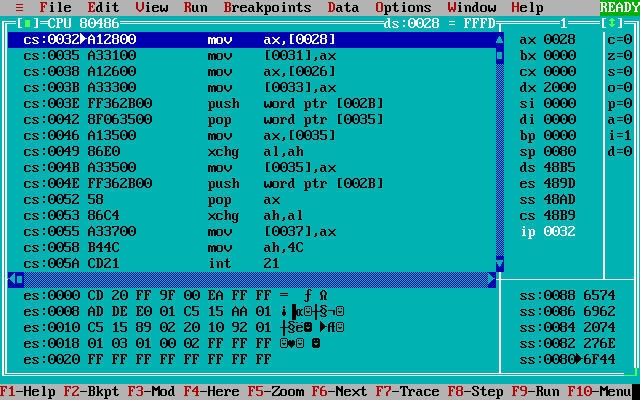


Рисунок 2

Програма працює корректно.

***4. Висновки щодо призначення регістрів, адресації даних, призначення команд пересилання й обміну та їх операндів.***

До стеку можна записувати лише 16бітні дані, зчитувати також тільки 16 біт.

Зчитуються тільки останні записані 16біт, тобто якщо ми запишемо слово1 слово2 слово3, то коли мі и будем їх зчитувати, ми зчитаемо слово3 слово2 слово1, тобто у зворотному порядку.

Якщо потрібно записати у змінну декілька директив, наприклад слово, байт та подвійне слово, то треба записувати їх за допомогою команд x ptr [a], де x – назва директиви, яку треба записати, а – адреса зміщення відносно початку змінної, куди треба записати директиву.