МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «архітектура комп’ютера»**

**на тему: «Системи числення. Представлення даних в пам'яті комп'ютера»**

Виконав: студент гр. ПЗ181х

П І Б

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

Лабораторна робота №2

**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.

**Постановка задачі**

|  |  |  |
| --- | --- | --- |
| № | Зміні | Обробка текста |
| 19 | text        db "Honesty is the best policy."  numbers db -57                dw 2527                dd -777666 | Зсунути циклічно вліво на один символ перше слово |

**Лістинг**

Turbo Assembler Version 3.0 10/25/19 16:43:45 Page 1

lab2.asm

1 ; сегмент даних

2 0000 data segment para public 'data'

3 0000 48 6F 6E 65 73 74 79 txt db 'Honesty'

4 0007 C7 numbers db -57

5 0008 09DF dw 2527

6 000A FFF4223E dd -777666

7 000E ???? A dw ?

8 0010 ???? B dw ?

9 0012 ???????? C dd ?

10 0016 ???????? adr dd ? ;

11 001A data ends

12 ; сегмент стека

13 0000 stk segment stack

14 0000 0100\*(??) db 256 dup (?) ;поле для стека розміром 256 байт

15 0100 stk ends

16

17 ; сегмент команд

18 0000 code segment para public 'code'

19 assume cs:code, ds:data, ss:stk

20 0000 start:

21 0000 B8 0000s mov ax, data ;ініціалізація сегментного регістра ds

22 0003 8E D8 mov ds, ax ; на програмний сегмент даних data

23 0005 B8 0000 mov ax, 0

24

25 0008 A0 0000r mov al, txt

26

27 000B 8A 26 0001r mov ah, txt+1

28 000F 88 26 0000r mov txt, ah

29

30 0013 8A 26 0002r mov ah, txt+2

31 0017 88 26 0001r mov txt+1, ah

32

33 001B 8A 26 0003r mov ah, txt+3

34 001F 88 26 0002r mov txt+2, ah

35

36 0023 8A 26 0004r mov ah, txt+4

37 0027 88 26 0003r mov txt+3, ah

38

39 002B 8A 26 0005r mov ah, txt+5

40 002F 88 26 0004r mov txt+4, ah

41

42 0033 8A 26 0006r mov ah, txt+6

43 0037 88 26 0005r mov txt+5, ah

44

45 003B A2 0006r mov txt+6, al

46

47 ;-57 адрес

48 003E 8C 1E 0016r mov word ptr adr, ds

49 0042 B8 0007r lea ax, numbers

50 0045 A3 0018r mov word ptr adr+2, ax

51

52 ; 2527

53

54 0048 A1 0008r mov ax, word ptr numbers+1

55 004B 86 C4 xchg al, ah

56 004D A3 000Er mov A, ax

57

58

59 0050 A1 0008r mov ax, word ptr numbers+1

60 0053 88 26 0010r mov byte ptr B, ah

61 0057 A2 0011r mov byte ptr B+1, al

62

63

64 ;-777666

65

66 005A FF 36 000Ar push word ptr numbers+3

67 005E FF 36 000Cr push word ptr numbers+5

68 0062 8F 06 0012r pop word ptr C

69 0066 8F 06 0014r pop word ptr C+2

70

71 006A CD 21 int 21h

72 006C code ends

73 end start

Turbo Assembler Version 3.0 10/25/19 16:43:45 Page 2

Symbol Table

Symbol Name Type Value

??DATE Text "10/25/19"

??FILENAME Text "lab2 "

??TIME Text "16:43:45"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text LAB2

@WORDSIZE Text 2

A Word DATA:000E

ADR Dword DATA:0016

B Word DATA:0010

C Dword DATA:0012

NUMBERS Byte DATA:0007

START Near CODE:0000

TXT Byte DATA:0000

Groups & Segments Bit Size Align Combine Class

CODE 16 006C Para Public CODE

DATA 16 001A Para Public DATA

STK 16 0100 Para Stack

**Висновок**: Регістр процесора - комірка швидкодійної внутрішньої [пам'яті](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BF%D0%B0%D0%BC%27%D1%8F%D1%82%D1%8C) [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80), яка використовується для тимчасового збереження [операндів](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D0%BD%D0%B4), з якими безпосередньо проводяться обчислення, а також часто використовуваних даних з метою швидкого доступу до них. Вони мають свої адреса. Команди пересилання потрібні для того, щоб поміщати інформацію в регістри в регістр, які ті в свою чергу зберігають поки її не задіють.