­­Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Спеціалізованих комп’ютерних систем”



**Звіт**

до лабораторної роботи № 3

з дисципліни

# *​*Системне програмування

# Програмування задач лінійної структури. Обчислення виразів.**Варіант: 19**

**Виконав:**

ст. гр. КІ-207

Шаповал Віталій

**Перевірив:**

Асистент катедри ЕОМ

Максимів М. Р.

**Мета:** вивчити способи задання констант та змінних в Асемблері та набути навиків використання арифметичних команд над даними різного розміру.

**Завдання:**

1. Створити \*.exe програму, яка реалізовує обчислення, заданого варіантом виразу і зберігає результат в пам’яті. Вхідні операнди А, В, С, D, E, F вважати знаковими і довжиною в байтах, згідно з індексу; К – константа, довжина якої визначається значенням(згідно варіанту). Для її опису слід використати директиву EQU.

2. За допомогою Debug, відслідкувати правильність виконання програми (продемонструвати результати проміжних та кінцевих обчислень).

3. Скласти звіт про виконану роботу з приведенням тексту програми та коментарів до неї.

4. Дати відповідь на контрольні запитання

**Варіант: 19**

****

**Виконання:**

**Код:**

.686

.model flat, stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

    K EQU  4019h

    A DW -12d

    B DW 12d

    Cc DB 1

    D DW 200d

    E DB 10d

    TEMP1 DD ?

    TEMP2 DW ?

    TEMP3 DW ?

    X DD ?

    Message db 'X = A \* B + A \* C - D / E + K =             ', 13, 10

    NumberOfCharsToWrite dd $-Message

    format db '%d', 0

    hConsoleOutput dd 0

    NumberOfCharsWritten dd 0

.code

start:

; TEMP1 = A \* B

mov ax, A; A -> eax

CWDE

IMUL ax, B; A(eax) \* B -> eax

mov TEMP1, eax ; (A \* B)(eax) -> TEMP1

; TEMP2 = A \* C

mov al, Cc ; Cc -> al

CBW

IMUL ax, A ; C(al) \* A -> ax

mov TEMP2, ax ; (A \* C)(ax) -> TEMP2

; TEMP3 = D / E

mov ax, D ; D -> ax

IDIV E ; D(ax) / E -> ax

mov TEMP3, ax ; (D / E)(ax) -> TEMP3

; X = TEMP1 + TEMP2 - TEMP3 + K

; X = TEMP2 - TEMP3 + TEMP1 + K

mov ax, TEMP2 ; TEMP2 -> ax

sub ax, TEMP3 ; TEMP2 (ax) - TEMP3 -> ax

CWDE ; ax -> eax

add eax, TEMP1 ; (TEMP2 - TEMP3)(eax) + TEMP1 -> eax

add eax, K ; (TEMP2 - TEMP3 + TEMP1)(eax) + K -> eax

mov X, eax ; (TEMP2 - TEMP3 + TEMP1 + K)(eax) -> X

push X

push offset format

push offset [Message+32]

call wsprintfA

push -11

call GetStdHandle

mov hConsoleOutput, eax

push 0

push offset NumberOfCharsWritten

push NumberOfCharsToWrite

push offset Message

push hConsoleOutput

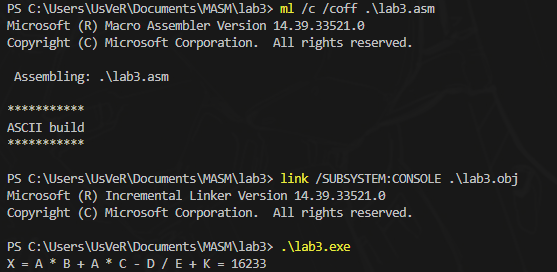
call WriteConsoleA

push 0

call ExitProcess

end start

**Результат виконання:**

****

Скрин 1. Результат виконання програми

**Дослідження правильности виконання програми**

Перегляд вмісту регістрів:  
EAX = 00000001 EBX = 0047A000 ECX = 0000002E EDX = 007B404D ESI = 007B1000 EDI = 007B1000 EIP = 007B1099 ESP = 0078F9D4 EBP = 0078F9F0 EFL = 00000202

Перегляд вмісту пам’яті:

0x007B3FFD 00 00 00 f4 ff 0c 00 01 c8 00 0a 70 ff ff ff f4 ...фя...И..pяяяф

0x007B400D ff 14 00 69 3f 00 00 58 20 3d 20 41 20 2a 20 42 я..i?..X = A \* B

0x007B401D 20 2b 20 41 20 2a 20 43 20 2d 20 44 20 2f 20 45 + A \* C - D / E

0x007B402D 20 2b 20 4b 20 3d 20 31 36 32 33 33 00 20 20 20 + K = 16233.

0x007B403D 20 20 20 0d 0a 2e 00 00 00 25 64 00 0c 00 00 00 ......%d.....

K EQU 4019h

A DW -12d

B DW 12d

Cc DB 1

D DW 200d

E DB 10d

TEMP1 DD ?

TEMP2 DW ?

TEMP3 DW ?

X DD ?

Message db 'X = A \* B + A \* C - D / E + K = ', 13, 10

NumberOfCharsToWrite dd $-Message

format db '%d', 0

A: -12 = 1000 0000 0000 11002 = 1111 1111 1111 01002 = ff f416 = f4 ff16

B: 12 = 0000 0000 0000 11002 = 00 0c16 = 0c 0016

Cc: 1 = 0000 00012 = 0116

D: 200 = 0000 0000 1100 10002 = 00 c816 = c8 0016

E: 10 = 0000 10102 = 0a16

TEMP1 = A\*B = -144 = 1000 0000 0000 0000 0000 0000 1001 00002 = 1111 1111 1111 1111 1111 1111 0110 11112 = 1111 1111 1111 1111 1111 1111 0111 00002 = ff ff ff 7016 = 70 ff ff ff16

TEMP2 = A\*С = -12 = 1000 0000 0000 11002 = 1111 1111 1111 01002 = ff f416 = f4 ff16

TEMP3 = D/E = 20 = 0000 0000 0001 01002 = 00 1416 = 14 0016

X = 16233 = 0000 0000 0000 0000 0011 1111 0110 10012 = 00 00 3f 6916 = 69 3f 00 0016

**Висновок:** виконавши цю лабораторну роботу я вивчив способи задання констант та змінних в Асемблері та набути навиків використання арифметичних команд над даними різного розміру.