Національний технічний університет України «Київський політехнічний інститут»

Кафедра обчислювальної техніки

Лабораторна робота №3

з дисципліни **«Системне програмування»**

**Створення модульних проектів на асемблері у середовищі Visual Studio**

**та вивчення форматів представлення чисел**

**Мета:** Навчитися створювати модульні проекти на асемблері, а також

закріпити знання основних форматів представлення чисел у комп’ютері.

Виконала:

студентка 2 курсу

ФІОТ гр. ІО-21

Любарська Лілія

Київ 2014 р.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям **Lab3**.

2. Написати вихідний текст програми згідно варіанту завдання. Вихідний текст

повинен бути у вигляді двох модулів на асемблері:

- головний модуль, у якому описується загальний хід виконання програми

від початку і до завершення. Цей модуль містить точку входу у програму,

впродовж роботи викликає процедури з інших модулів. Вихідний текст

головного модуля записати у файл **main3.asm**;

- другий модуль, який містить процедуру, яка викликається з головного

модуля. Цей модуль записати у файл **module.asm**.

3. Додати файли модулів у проект. У цьому проекті кожний модуль може

окремо компілюватися.

4. Скомпілювати вихідний текст і отримати виконуємий файл програми.

5. Перевірити роботу програми. Налагодити програму.

6. Отримати результати – кодовані значення чисел згідно варіанту завдання.

7. Проаналізувати та прокоментувати результати та вихідний текст.

.586

.model flat, stdcall

option casemap :none

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

option casemap :none

include \masm32\include\windows.inc

include module.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.const

.data

TextBuf db 64 dup(?)

Value1 db 00011010b

Value2 db 11100110b

Value3 dw 0000000000011010b

Value4 dw 1111111111100110b

Value5 dd 00000000000000000000000000011010b

Value6 dd 11111111111111111111111111100110b

Value7 dq 26

Value8 dq -26

Value9 dd 26.0

Value10 dd -52.0

Value11 dd 26.26

Value12 dq 26.0

Value13 dq -52.0

Value14 dq 26.26

Value15 dt 26.0

Value16 dt -52.0

Value17 dt 26.26

Caption db 'Lab 3', 0

.code

start:

push offset TextBuf

push offset Value1

push 8

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value2

push 8

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value3

push 16

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value4

push 16

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value5

push 32

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value6

push 32

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value7

push 64

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value8

push 64

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value9

push 32

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value10

push 32

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value11

push 32

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value12

push 64

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value13

push 64

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value14

push 64

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value15

push 80

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value16

push 80

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

push offset TextBuf

push offset Value17

push 80

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

invoke ExitProcess, 0

end start

X=26

Y=52

Ціле 8-бітове

X 1A 00011010

-X E6 11100110

Ціле 16-бітове

X 001A 0000000000011010

-X FFE6 1111111111100110

Ціле 32-бітове

X 0000001A 00000000000000000000000000011010

-X FFFFFFFE6 11111111111111111111111111100110

Ціле 64-бітове

X 00000000 0000001A

-X FFFFFFFF FFFFFFFE6

Число у 32-бітовому форматі

з плаваючою точкою

X.0 41D00000 0100 0001 1101 0000 0000 0000 0000 0000

-Y.0 C2500000 1100 0010 0101 0000 0000 0000 0000 0000

X.X 41D2147B

Число у 64-бітовому форматі

з плаваючою точкою

X.0 403A0000 00000000 0100 0000 0011 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

-Y.0 C04A000 00000000 1100 0000 0100 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

X.X 403A428F 5C28F5C3

Число у 80-бітовому форматі

з плаваючою точкою

X.0 4003D0000000 00000000 0100 0000 0000 0011 1101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

-Y.0 C004D0000000 00000000 1100 0000 0000 0100 1101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

X.X 4003D2147AE1 47AE147B

32 біти 1 біт-знак, 2-9 експонента, 10-32 мантисса

64 біти 1 біт-знак, 2-12 експонента, 13-64 мантисса

80 біт 1 біт-знак, 2-16 експонента,17-ціла част., 18-80 мантисса