НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №3**

з дисципліни **«**Системне програмування**»**

Виконав:

студент 2 курсу

ФІОТ гр. ІО-33

Шуркіна Анастасія

Перевірив:

Порєв В.М.

Київ 2015 р.

**Тема:**Створення модульних проектів на асемблері у середовищі VisualStudio

та вивчення форматів представлення чисел.

**Мета:**Навчитися створювати модульні проекти на асемблері, а також закріпити знання основних форматів представлення чисел у комп’ютері.

**Варіант завдання:**N=28  
X = N+10 = 38; Y = 2X = 76

**Результати:**

|  |  |  |  |
| --- | --- | --- | --- |
| Тип даних | Значення | Результати виконання програми | |
| шістнадцятиковий код | двійковий код |
| Ціле 8-бітове | 38 | 26 | 0010 0110 |
| -38 | DA | 1101 1010 |
| Ціле 16-бітове | 38 | 0026 | 0000 0000 0010 0110 |
| -38 | FFDA | 1111 1111 1101 1010 |
| Ціле 32-бітове | 38 | 00000026 | 10 0110 |
| -38 | FFFFFFDA | 1111 1111 1111 1111 1111 1111 1101 1010 |
| Ціле 64-бітове | 38 | 00000000 |  |
| -38 | FFFFFFFF |  |
| Число у 32-бітовому форматі | 38.0 | 42180000 | 0100 0010 0001 1000 0000 0000 0000 0000 |
| -76.0 | C2980000 | 1100 0010 1001 1000 0000 0000 0000 0000 |
| 38.38 | 4219851F |  |
| Число у 64-бітовому форматі  з плаваючою точкою | 38.0 | 40430000 | 0100 0000 0100 0011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| -76.0 | C0530000 | 1100 0000 0101 0011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| 38.38 | 404330A3 |  |
| Число у 80-бітовому форматі  з плаваючою точкою | 38.0 | 4004398000000300000000 |  |
| -76.0 | C005398000000300000000 |  |
| 38.38 | 4004399851EB8351EB851F |  |

Аналіз результатів:

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

**Програмний код:**

.586

.model flat, stdcall

option casemap :none

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

include \masm32\include\windows.inc

includelib \masm32\\lib\kernel32.lib

includelib \masm32\lib\user32.lib

include module.inc

.data

TextBuf db 64 dup (?)

Value1 db 00100110b

Value2 db 11011010b

Value3 dw 0000000000100110b

Value4 dw 1111111111011010b

Value5 dd 00000000000000000000000000100110b

Value6 dd 11111111111111111111111111011010b

Value7 dq 38

Value8 dq -38

Value9 dd 38.0

Value10 dd -76.0

Value11 dd 38.38

Value12 dq 38.0

Value13 dq -76.0

Value14 dq 38.38

Value15 dt 38.0

Value16 dt -76.0

Value17 dt 38.38

Caption db 'Lab 3', 0

.code

main:

push offset TextBuf

push offset Value1

push 8

call StrHex\_MY

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

push offset TextBuf

push offset Value2

push 8

call StrHex\_MY

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

push offset TextBuf

push offset Value3

push 16

call StrHex\_MY

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

push offset TextBuf

push offset Value4

push 16

call StrHex\_MY

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

push offset TextBuf

push offset Value5

push 32

call StrHex\_MY

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

push offset TextBuf

push offset Value6

push 32

call StrHex\_MY

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

push offset TextBuf

push offset Value7

push 64

call StrHex\_MY

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

push offset TextBuf

push offset Value8

push 64

call StrHex\_MY

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

push offset TextBuf

push offset Value9

push 32

call StrHex\_MY

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

push offset TextBuf

push offset Value10

push 32

call StrHex\_MY

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

push offset TextBuf

push offset Value11

push 32

call StrHex\_MY

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

push offset TextBuf

push offset Value12

push 64

call StrHex\_MY

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

push offset TextBuf

push offset Value13

push 64

call StrHex\_MY

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

push offset TextBuf

push offset Value14

push 64

call StrHex\_MY

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

push offset TextBuf

push offset Value15

push 80

call StrHex\_MY

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

push offset TextBuf

push offset Value16

push 80

call StrHex\_MY

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

push offset TextBuf

push offset Value17

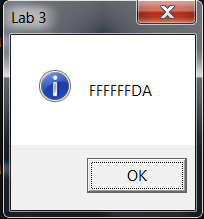
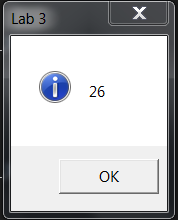
push 80

call StrHex\_MY

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

invoke ExitProcess, 0

end main



**Висновки:** під час виконання даної лабораторної роботи було створено програму, що перетворює деякі десяткові числа з заданою розрядністю в різні формати шіснадцяткової системи числення. Було вивчено основи створення модульних програм на асемблері в середовищі Microsoft Visual Studio.