НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №3**

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

Виконала:

студентка 2 курсу

ФІОТ гр. ІВ-71

Молчанова В.С.

Перевірив:

ст. вик. Порєв В.М.

Київ – 2019

**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=10  
X = N+10 = 20; Y = 2X = 40

**Результати:**

|  |  |  |  |
| --- | --- | --- | --- |
| Тип даних | Значення | Результати виконання програми | |
| шістнадцятиковий код | двійковий код |
| Ціле 8-бітове | 20 | 14 | 0001 0100 |
| -20 | EC | 1110 1100 |
| Ціле 16-бітове | 20 | 0014 | 0000 0000 0001 0100 |
| -20 | FFEC | 1111 1111 1110 1100 |
| Ціле 32-бітове | 20 | 00000014 | 0000 0000 0000 0000 0000 0000 0001 0100 |
| -20 | FFFFFFEC | 1111 1111 1111 1111 1111 1111 1110 1100 |
| Ціле 64-бітове | 20 | 000000000 00000014 |  |
| -20 | FFFFFFFF FFFFFFEC |  |
| Число у 32-бітовому форматі | 20.0 | 41A00000 | 0100 0001 1010 0000 0000 0000 0000 0000 |
| -40.0 | C2200000 | 1100 0010 0010 0000 0000 0000 0000 0000 |
| 20.20 | 41A1999A | 0100 0001 1010 0001 1001 1001 1001 1010 |
| Число у 64-бітовому форматі  з плаваючою точкою | 20.0 | 40340000 00000000 | 0100 0000 0011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| -40.0 | C0440000 00000000 | 1100 0000 0100 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| 20.20 | 4034A3333 33333333 | 0100 0000 0011 0100 1010 0011 0011 0011 0011 0011 0011 0011 00110011 0011 0011 |
| Число у 80-бітовому форматі  з плаваючою точкою | 20.0 | 4003A00000000000000000 | 0100 0000 0000 0011 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| -40.0 | C004A0000000000000000 | 1100 0000 0000 0100 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| 20.20 | 4003A199 99999999 99999A | 0100 0000 0000 0011 1010 0001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1010 |

Аналіз результатів:

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

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

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

**Програмний код:**

**main3.asm**

.586

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include module.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

TextBuf db 64 dup(?)

Value1 db 20

Value2 db -20

Value3 dw 20

Value4 dw -20

Value5 dd 20

Value6 dd -20

Value7 dq 20

Value8 dq -20

Value9 dd 20.0

Value10 dd -40.0

Value11 dd 20.20

Value12 dq 20.0

Value13 dq -40.0

Value14 dq 20.20

Value15 dt 20.0

Value16 dt -40.0

Value17 dt 20.20

Caption db "Lab3",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

**module.asm**

.586

.model flat, c

.code

;процедура StrHex\_MY записує текст шістнадцятькового коду

;перший параметр - адреса буфера результату (рядка символів)

;другий параметр - адреса числа

;третій параметр - розрядність числа у бітах (має бути кратна 8)

StrHex\_MY proc

push ebp

mov ebp,esp

mov ecx, [ebp+8] ;кількість бітів числа

cmp ecx, 0

jle @exitp

shr ecx, 3 ;кількість байтів числа

mov esi, [ebp+12] ;адреса числа

mov ebx, [ebp+16] ;адреса буфера результату

@cycle:

mov dl, byte ptr[esi+ecx-1] ;байт числа - це дві hex-цифри

mov al, dl

shr al, 4 ;старша цифра

call HexSymbol\_MY

mov byte ptr[ebx], al

mov al, dl ;молодша цифра

call HexSymbol\_MY

mov byte ptr[ebx+1], al

mov eax, ecx

cmp eax, 4

jle @next

dec eax

and eax, 3 ;проміжок розділює групи по вісім цифр

cmp al, 0

jne @next

mov byte ptr[ebx+2], 32 ;код символа проміжку

inc ebx

@next:

add ebx, 2

dec ecx

jnz @cycle

mov byte ptr[ebx], 0 ;рядок закінчується нулем

@exitp:

pop ebp

ret 12

StrHex\_MY endp

;ця процедура обчислює код hex-цифри

;параметр - значення AL

;результат -> AL

HexSymbol\_MY proc

and al, 0Fh

add al, 48 ;так можна тільки для цифр 0-9

cmp al, 58

jl @exitp

add al, 7 ;для цифр A,B,C,D,E,F

@exitp:

ret

HexSymbol\_MY endp

end

**module.inc**

EXTERN StrHex\_MY : proc

**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.