Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА № 3**

з дисципліни «Системне програмування-1» на тему

«Створення модульних проектів на асемблері у середовищі Visual Studio та вивчення форматів представлення чисел»

ВИКОНАЛА:

студентка ІІ курсу ФІОТ

групи ІО-64

Бровченко Анастасія

Залікова - 6403

ПЕРЕВІРИВ:

ст.вик. Порєв В. М.

Київ – 2018

**ЛАБОРАТОРНА РОБОТА №3**

*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.

**I. 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.

***Варіант***: 3.

**Х =** 3 + 10 = **13**

**Y =** X \* 2 = 13 \* 2 = **26**

**ІІ. Код програми**

**main3.asm**

.586

.model flat, stdcall

option casemap :none ;розрізнювати великі та маленькі букви

include \masm32\include\windows.inc

include D:\masm32\include\kernel32.inc

include D:\masm32\include\user32.inc

include module.inc

includelib D:\masm32\lib\kernel32.lib

includelib D:\masm32\lib\user32.lib

.data

TextBuf db 64 dup(?)

Caption db "Лабораторна робота №3",0

Text db "Добрий ранок!", 10, 13, "Автор: Бровченко Анастасія Вікторівна",0

value1 db 13

value2 db -13

value3 dw 13

value4 dw -13

value5 dd 13

value6 dd -13

value7 dq 13

value8 dq -13

value9 dd 13.0

value10 dd -13.0

value11 dd 13.13

value12 dq 13.0

value13 dq -26.0

value14 dq 13.13

value15 dt 13.0

value16 dt -26.0

value17 dt 13.13

.code

main:

invoke MessageBoxA, 0, ADDR Text, ADDR Caption, 0

push offset TextBuf

push offset value1

push 8

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value2

push 8

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value3

push 16

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value4

push 16

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value5

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value6

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value7

push 64

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value8

push 64

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value9

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value10

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value11

push 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value12

push 64

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value13

push 64

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value14

push 64

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value15

push 80

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value16

push 80

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

push offset TextBuf

push offset value17

push 80

call StrHex\_MY

invoke MessageBoxA, 0, ADDR TextBuf, ADDR Caption, 0

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

;ця процедура записує 8 символів HEX коду числа

;перший параметр - 32-бітове число

;другий параметр - адреса буфера тексту

DwordToStrHex proc

push ebp

mov ebp,esp

mov ebx,[ebp+8] ;другий параметр

mov edx,[ebp+12] ;перший параметр

xor eax,eax

mov edi,7

@next:

mov al,dl

and al,0Fh ;виділяємо одну шістнадцяткову цифру

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

cmp ax,58

jl @store

add ax,7 ;для цифр A,B,C,D,E,F

@store:

mov [ebx+edi],al

shr edx,4

dec edi

cmp edi,0

jge @next

pop ebp

ret 8

DwordToStrHex endp

End

**module.inc**

EXTERN StrHex\_MY : proc

EXTERN DwordToStrHex : proc

**III. Результат**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Типи даних*** | ***Зн-я*** | ***Шістнадцятковий код*** | ***Двійковий код*** |
| Ціле 8-бітове | 13 | 0D |  |
| -13 | F3 |  |
| Ціле 16-бітове | 13 | 000D | 0000 0000 0000 1101 |
| -13 | FFF3 | 1111 1111 1111 0011 |
| Ціле 32-бітове | 13 | 0000 000D | 0000 0000 0000 0000 0000 0000 0000 1101 |
| -13 | FFFF FFF3 | 1111 1111 1111 1111 1111 1111 1111 0011 |
| Ціле 64-бітове | 13 | 0000 0000 0000 000D | 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1101 |
| -13 | FFFF FFFF FFFF FFF3 | 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 0011 |
| Число у 32-бітовому форматі з плаваючою точкою | 13.0 | 4150 0000 |  |
| -26.0 | C1D0 0000 |  |
| 13.13 | 4152 147B |  |
| Число у 64-бітовому форматі з плаваючою точкою | 13.0 | 402А 0000 0000 0000 |  |
| -26.0 | С03А 0000 0000 0000 |  |
| 13.13 | 402А 428F 5C28 F5C3 |  |
| Число у 80-бітовому форматі з плаваючою точкою | 13.0 | 4002 D000 0000 0000 0000 | 1000 0000 0000 0101 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| -26.0 | C003 D000 0000 0000 0000 | 1100 0000 0000 0011 1101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 |
| 13.13 | 4002 D214 7AE1 47AE 147B |  |

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

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

***80 бітів****: 1 біт - знак, 2-16 біти - експонента, 17 біт- ціла частина, 18-80 біти - мантиса*

**ІV. Висновок**

У ході виконання лабораторної роботи було закріплено на практиці навички створення модульних проектів у середовищі Microsoft Visual Studio 2017 та застосовано знання про представлення чисел у комп’ютері.