**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

*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»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-62

*Гомонець І.І.*

**Перевірив:**

Старший викладач

*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***: 4.

**Х =** 4 + 10 = **14**

**Y =** X \* 2 = 14 \* 2 = **28**

**ІІ. Код програми**

.586

.model flat, stdcall

include module.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \lib\kernel32.lib

includelib \lib\user32.lib

.data

TextBuf db 64 dup(?)

Caption db "Lab3",0

Text1 db "Автор: ",10, 13, "Гомонець Іван Іванович ",10, 13, "ІО-62", 0

value1 db 14

value2 db -14

value3 dw 14

value4 dw -14

value5 dd 14

value6 dd -14

value7 dq 14

value8 dq -14

value9 dd 14.0

value10 dd -28.0

value11 dd 14.14

value12 dq 14.0

value13 dq -28.0

value14 dq 14.14

value15 dt 14.0

value16 dt -28.0

value17 dt 14.14

.code

main:

invoke MessageBoxA, 0 , ADDR Text1 , 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

**III. Результат**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Типи даних*** | ***Зн-я*** | ***Шістнадцятковий код*** | ***Двійковий код*** |
| Ціле 8-бітове | 14 | 0E |  |
| -14 | F2 |  |
| Ціле 16-бітове | 14 | 000E | 0000 0000 0000 1110 |
| -14 | FFF2 | 1111 1111 1111 0010 |
| Ціле 32-бітове | 14 | 0000 000Е | 0000 0000 0000 0000 0000 0000 0000 1110 |
| -14 | FFFF FFF2 | 1111 1111 1111 1111 1111 1111 1111 0010 |
| Ціле 64-бітове | 14 | 0000 0000 0000 000Е | 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1110 |
| -14 | FFFF FFFF FFFF FFF2 | 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 0010 |
| Число у 32-бітовому форматі з плаваючою точкою | 14.0 | 4160 0000 |  |
| -28.0 | C1Е0 0000 |  |
| 14.14 | 4162 3D71 |  |
| Число у 64-бітовому форматі з плаваючою точкою | 14.0 | 402C 0000 0000 0000 |  |
| -28.0 | С03C 0000 0000 0000 |  |
| 14.14 | 402C 47AE 147A E148 |  |
| Число у 80-бітовому форматі з плаваючою точкою | 14.0 | 4002 E000 0000 0000 0000 | 1000000000000101110000000000000000000000000000000000000000000000000000000000000 |
| -28.0 | C003 E000 0000 0000 0000 | 11000000000000111110000000000000000000000000000000000000000000000000000000000000 |
| 14.14 | 4002 E23D 70A3 D70A 3D71 |  |

***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. Контрольні питання:**

**1. Як створюється 8-бітова перемінна?**

temp db ?

**2. Як створюється 16-бітова перемінна?**

temp dw ?

**3. Як створюється 32-бітова перемінна?**

temp dd ?

**4. Як створюється 64-бітова перемінна?**

temp dq ?

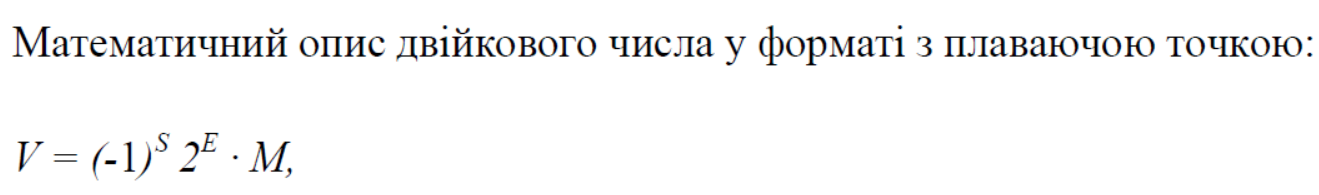
**5. Як створюється 80-бітова перемінна?**

temp dt ?

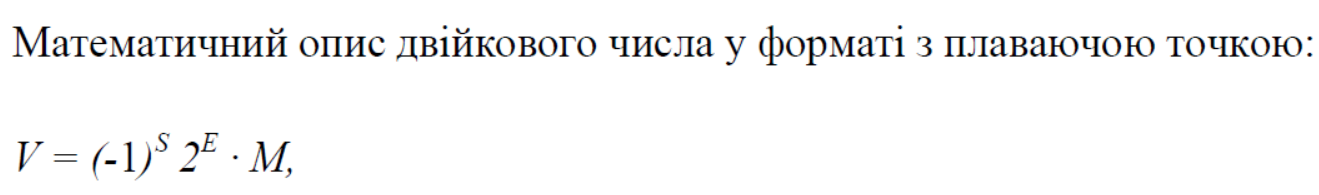
**6. Що таке додатковий код?**

**Найпоширеніший спосіб представлення від'ємних чисел у комп'ютерах. Дозволяє замість команди віднімання використовувати команду додавання, для знакових і беззнакових чисел, що зменшує вимоги до архітектури комп'ютера. Додатковий код від'ємного числа можна отримати так: інвертувати модуль числа у двійковому вигляді («перше доповнення») і додати одиницю («друге доповнення») або відняти число від нуля. Математично додатковий код Xдоп = 2N+1 — X, де X — число, яке треба представити у додатковому коді, N — к-сть розрядів числа.**

**7. Що таке експонента числа з плаваючою точкою?**

**** Е — експонента.

**8. Що таке мантиса?**

**** М — мантиса.

**9. Від чого залежіть діапазон представлення чисел у форматах з плаваючою**

**точкою?**

Від значення експоненти.

**10. Від чого залежіть точність представлення чисел у форматах з плаваючою точкою?**

Точністьпредставлення обумовлюється розрядністю мантиси.

**11. Що означає push offset?**

Записує в стек адресу перемінної.

**12. Що означає call?**

Виклик функції.

**V. 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.