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 № 6**

з дисципліни «Системне програмування-1» на тему

«Програмування побітових операцій»

ВИКОНАЛА:

студентка ІІ курсу ФІОТ

групи ІВ-71

Молчанова В.С.

Залікова - 7110

ПЕРЕВІРИВ:

ст.вик. Порєв В. М.

Київ – 2019

**Мета:** Навчитися програмувати на асемблері побітові операції, вивчити основні команди обробки бітів.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям **Lab6**.

2. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути три модуля на асемблері:

- головний модуль: файл **main6.asm**. Цей модуль створити та написати заново;

- другий модуль: використати **module** попередніх робіт;

- третій модуль: модуль **longop** попередньої роботи №5 доповнити новим кодом відповідно завданню.

3. 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, вихідний текст та дизасембльований машинний код програми.

**Варіант завдання:**

Запис M нулів починаючи з N-го біту. Решту бітів зробити 1

**Роздруківка тексту програми:**

**longop.asm**

.586

.model flat, c

include module.inc

.data

counter dd ?

start0 dd ?

.code

Write1and0 proc

push ebp

mov ebp, esp

mov ebx, [ebp+20] ;result

mov ecx, [ebp+16] ;capacity

mov esi, [ebp+12] ;N

mov edi, [ebp+8] ;M

mov counter, 0

add edi, esi ;початок одиниць зправа

xor eax, eax

@write1Cycle:

push ecx

mov edx, counter

mov ecx, edx

shr edx, 3

and ecx, 07h

mov al, 1

shl al, cl

pop ecx

or byte ptr [ebx+edx], al

inc counter

cmp counter, ecx ;якщо заповнили число до кінця - виходимо

jnz @write1Cycle

mov counter, esi

@write0Cycle:

push ecx

mov edx, counter

mov ecx, edx

shr edx, 3

and ecx, 07h

mov al, 1

shl al, cl

pop ecx

not al

and byte ptr [ebx+edx], al

inc counter

cmp counter, edi

jz @exit

jmp @write0Cycle

@exit:

mov esp, ebp

pop ebp

ret 16

Write1and0 endp

MyTest proc

push ebp

mov ebp, esp

mov ebx, [ebp+8] ;result

and byte ptr [ebx], 1

mov esp, ebp

pop ebp

ret 4

MyTest endp

end

**longop.inc**

EXTERN Write1and0 : proc

**main.asm**

.586

.model flat, stdcall

option casemap :none

include D:\masm32\include\kernel32.inc

include D:\masm32\include\user32.inc

include D:\masm32\include\windows.inc

include module.inc

include longop.inc

includelib D:\masm32\lib\kernel32.lib

includelib D:\masm32\lib\user32.lib

.data

Caption db "Лабораторна 6", 0

Text db "Виконала студентка ІВ-71", 13, 10, "Молчанова Варвара", 13, 10, 0

ResultCaption db "Результат", 0

ResultText db 68 dup (0)

empty2 dd 80 dup(0)

ResultValue db 68 dup (0)

.code

main:

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

push offset ResultValue

push 544

push 0

push 12

call Write1and0

push offset ResultText

push offset ResultValue

push 544

call StrHex\_MY

invoke MessageBoxA, 0, ADDR ResultText, ADDR ResultCaption, 0

invoke ExitProcess, 0

end main

**Висновок:**

Під час виконання лабораторної роботи були покращені навички написання власних модулів, роботи з циклами. Наввчились програмувати побітові операції на асемблері, вивчили основні команди обробки бітів.