МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №6

з предмету “Системне програмування”

Виконала:  
Студентка 2 курсу  
ФІОТ гр. ІО-43  
Даніленко Наталія

Київ – 2016

**Лабораторна робота №6**

**Програмування побітових операцій**

**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:**

Починаючи з N-го розряду виконується побітове OR з M-бітовою маскою, яка вказується відповідним параметром процедури.

Розрядність: 352 біти.

**Текст програми:**

.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

includelib D:\masm32\lib\kernel32.lib

includelib D:\masm32\lib\user32.lib

include moduleE.inc

include longopP.inc

.data

N0 db 104 dup(170)

n dd 128

m0 db 22 dup(85)

m dd 146

TextBuf0 db 256 dup(?)

TextBuf1 db 256 dup(?)

Caption0 db "Вхідні дані",0

Caption1 db "Отриманий результат",0

.code

start:

push offset TextBuf0

push offset N0

push 352

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf0, ADDR Caption0, MB\_ICONINFORMATION

push offset N0

push offset m0

push n

push m

call Or\_Longop

push offset TextBuf1

push offset N0

push 352

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf1, ADDR Caption1, MB\_ICONINFORMATION

invoke ExitProcess, 0

end start

.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

.586

.model flat, c

.code

Or\_Longop proc

push ebp

mov ebp, esp

mov ecx, [ebp + 8] ;розрядність маски

mov edx, [ebp + 12] ;номер біта

mov esi, [ebp + 16] ;адреса маски

mov edi, [ebp + 20] ;адреса вихідного рядка

shr ecx, 3

shr edx, 3

cycleOR:

add edx, ecx

mov al, byte ptr [edi + edx - 1]

or al, byte ptr [esi + ecx - 1]

mov byte ptr [edi + edx - 1], al

sub edx, ecx

dec ecx

jnz cycleOR

pop ebp

ret 16

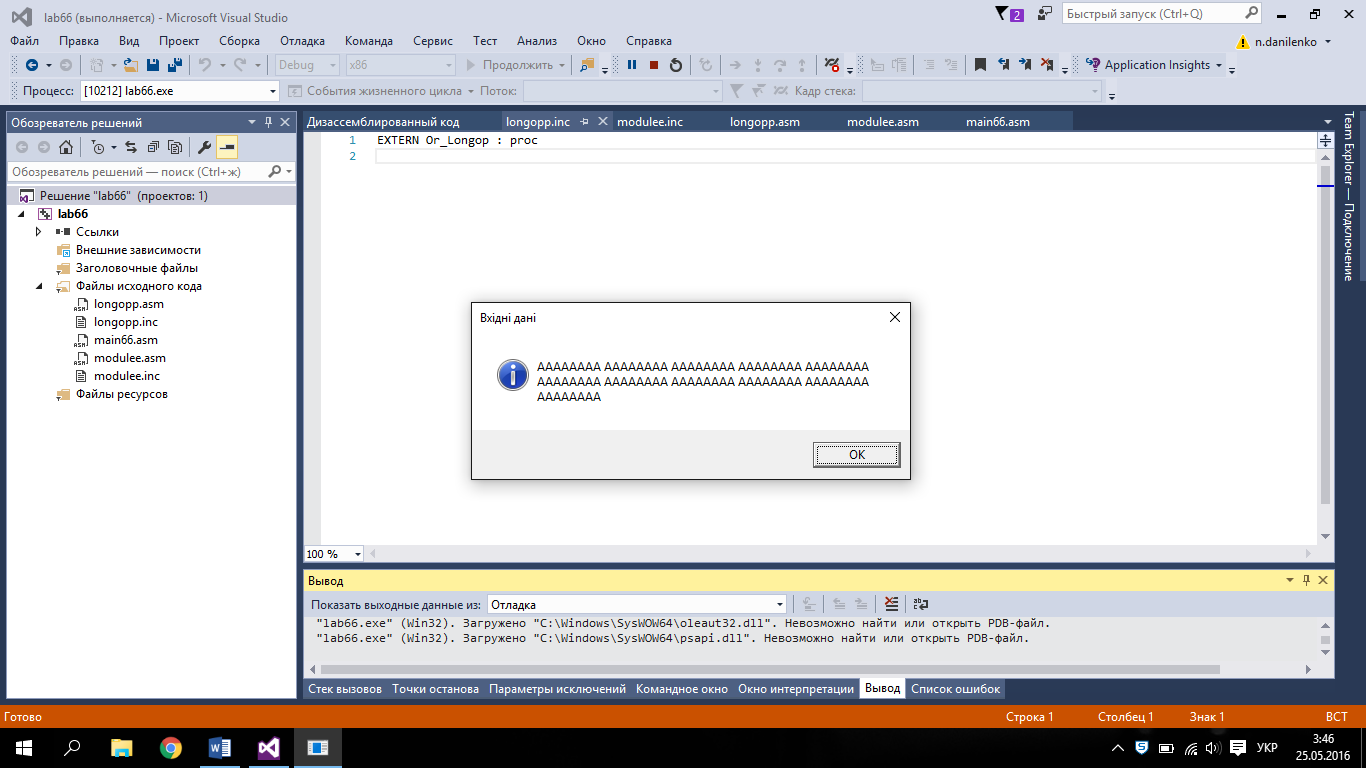
Or\_Longop endp

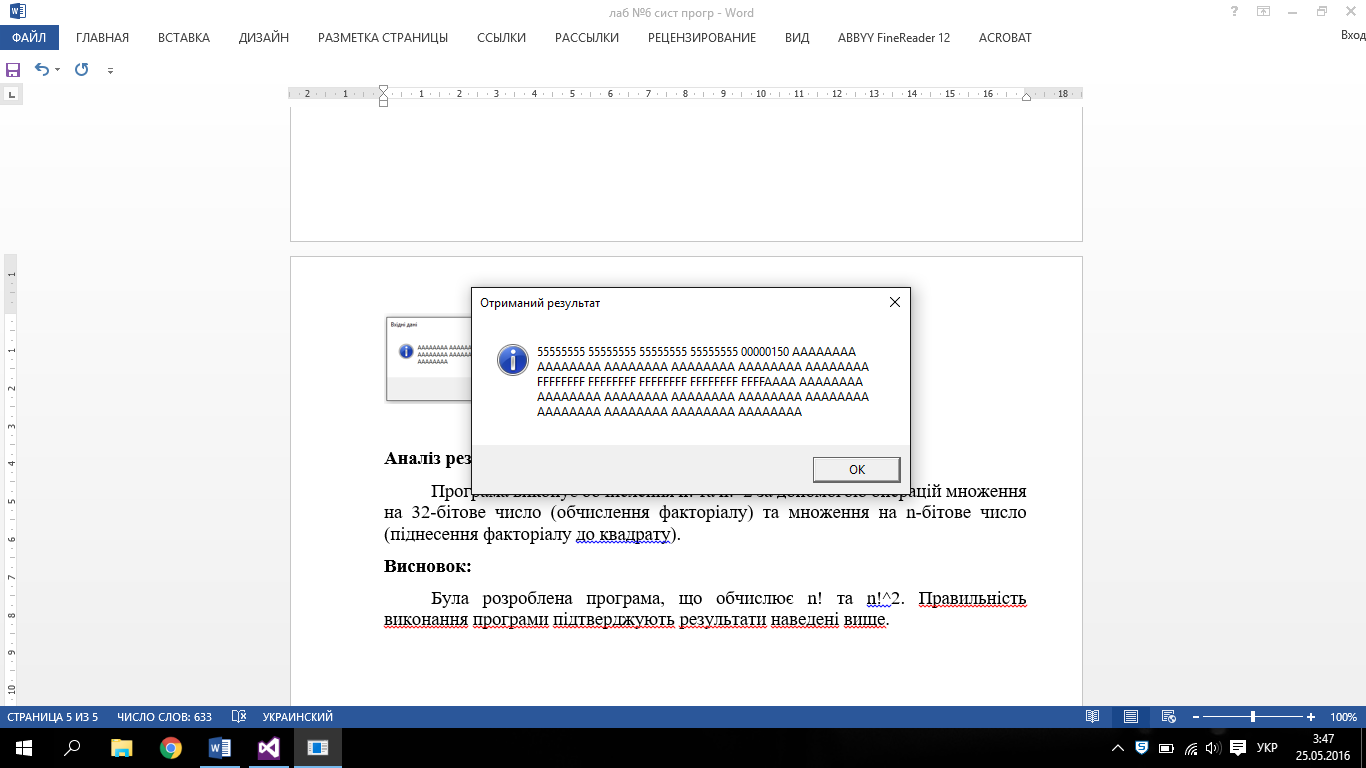
End

EXTERN StrHex\_MY : proc

EXTERN Or\_Longop : proc

**Результати виконання програми:**





**Аналіз результатів:**

Програма виконує починаючи з N-го розряду побітове OR з M-бітовою маскою, яка вказується відповідним параметром процедури.

**Висновок:**

Була розроблена програма, що виконує побітове OR з M-бітовою маскою (початок з n-ого розряду). Параметри m та n були обрані самостійно. Правильність виконання програми підтверджують результати наведені вище.