МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Лабораторна робота №9

з предмету “Системне програмування”

Виконав:

студент 2-го курсу ФІОТ

групи ІО-31

Долинний О.В.

Київ 2015

Лабораторна робота №9

Використання функцій API Win32 у програмах на асемблері

Мета: Навчитися використовувати у програмах на асемблері функції Windows

динамічного виділення пам’яті та запису файлів.

**Завдання:**

1. Створити у середовищі MS Visual Studio проект з ім’ям Lab9.

2. Написати вихідний текст програми згідно варіанту завдання. У проекті мають бути головний файл main9.asm та модулі module (за необхідності) та модуль longop попередніх робіт.

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 – файл числових значень згідно варіанту завдання.

7. Проаналізувати та прокоментувати результати, вихідний текст та дизасембльований машинний код програми.

**Варіант завдання:** 11

Запрограмувати на асемблері запис у файл масиву значень факторіалу *n*! (*n* від

1 до *nmax*). Вказування імені файлу у стандартному діалоговому вікні (функція

GetSaveFileName).

*nmax* = 30 + 2×*H* = 30 + 2×*11 = 52*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вар | Варіант циклу факторіалу | Код результату | Масиви для даних підвищеної розрядності | Масив-буфер для  імені файлу |
| 11 | 1 | шістнадцятковий | динамічні | статичний |

Роздруківка тексту програми

.586

.model flat, stdcall

OPTION CaseMap:None

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

include \masm32\include\comdlg32.inc

includelib \masm32\lib\comdlg32.lib

include \masm32\include\windows.inc

include \work\Lab4\Lab4\module.inc

include \work\Lab4\Lab4\longop.inc

.const

max dd 52

space db 13, 10

.data

n dd 1

pRes dd ?

pBuf dd ?

sFileName db 256 dup(0)

pFileName dd ?

hFile dd ?

fact\_text db 45 dup (0)

is\_correct dd ?

.code

MySaveFileName proc

LOCAL ofn : OPENFILENAME

invoke RtlZeroMemory, ADDR ofn, SIZEOF ofn

mov ofn.lStructSize, SIZEOF ofn

mov ofn.lpstrFile, OFFSET sFileName

mov ofn.nMaxFile, 256

invoke GetSaveFileName,ADDR ofn

ret

MySaveFileName endp

Factorial proc

push ebp

mov ebp, esp

mov edx, 1 ;лічильник

mov edi, pRes

inc edi

mov ecx, 32

xor eax, eax

rep stosb

mov ebx, pRes

mov byte ptr[ebx], 1 ; buf = 1

@cycle:

inc edx

cmp edx, n

jg @endfactorial

mov esi, pRes

mov edi, pBuf

mov ecx, 32

rep movsb

mov edi, pRes

mov ecx, 32

xor eax, eax

rep stosb

push edx

push pBuf

push edx

push pRes

push 32

call Mul\_N\_32\_LONGOP

pop edx

jmp @cycle

@endfactorial:

pop ebp

ret

Factorial endp

main:

call MySaveFileName

cmp eax, 0

je @exit

invoke CreateFile, ADDR sFileName,

GENERIC\_WRITE,

FILE\_SHARE\_WRITE,

0, CREATE\_ALWAYS,

FILE\_ATTRIBUTE\_NORMAL,

0

cmp eax, INVALID\_HANDLE\_VALUE

je @exit

mov hFile, eax

invoke GlobalAlloc, GPTR, 52

mov pRes, eax

add eax, 16

mov pBuf, eax

@cycle:

mov ecx, n

cmp ecx, max

jg @end

call Factorial

push offset fact\_text

push pRes

push 256

call StrHex\_MY

invoke WriteFile, hFile, ADDR fact\_text, 45, ADDR is\_correct, 0

invoke WriteFile, hFile, ADDR space, 2, ADDR is\_correct, 0

inc n

jmp @cycle

@end:

invoke GlobalFree, pRes

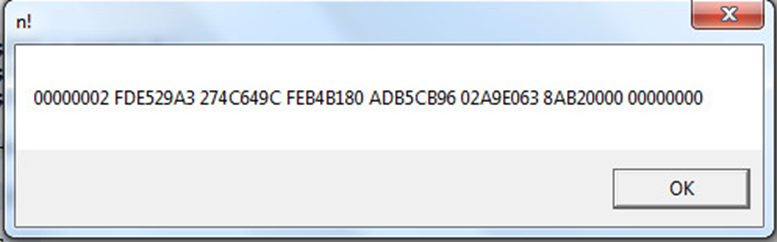
@exit:

invoke CloseHandle, hFile

invoke ExitProcess, 0

end main

**Результати: (52!)**

 (перевірка з 5 лаби)

**Аналіз результатів:**

Дана програма виконує операцію обчислення факторіалу, виводить його в файл в шістнадцятковому форматі. Результати співпадають з прогнозованими.

**Висновок:**

Під час виконання лабораторної роботи було покращено навички написання власних модулів, роботи з циклами, а також були закріплено основні навички в використанні функцій API Win32 у програмах на асемблері.