*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ*

*НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ*

*“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”*

*Лабораторна робота №8*

*з предмету “Системне програмування”*

*Виконав:*

*Студент 2-го курсу ФІОТ*

*групи ІО-22*

*Бас Андрій*

*Київ 2014*

Лабораторна робота №8

Використання функцій API Win32 у програмах на асемблері

**Мета**: Навчитися використовувати функції Windows динамічного виділення пам’яті та запису файлів у програмах на асемблері.

*Роздруківка тексту програми*

.586

.model flat, stdcall

option casemap : none ;розрізнювати великі та маленькі букви

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

; а також include для інших заголовочних файлів

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

include \masm32\include\comdlg32.inc

includelib \masm32\lib\comdlg32.lib

include module.inc

include longop.inc

.const

;Len equ 267

;FVal equ 1000

Len equ 8

FVal equ 51

.data

captionHeaderFactorial db "Lab 5, factorial", 0

captionHeaderSquare db "Lab 5, square", 0

textBufFactorial dd Len dup(0)

textBufSquare dd Len \* 4 dup(0)

R dd Len dup(0)

pTemp dd ? ;Len dup(0) ; temporary pointer

RF dd Len \* 2 dup(0)

; count factorial correctly only for N <= 51, for bigger numbers needs bigger arrays (more than 8) and also counters updated)

N dd ?

;szFileName db 256 dup(0)

hFile dd 0

pResFile dd 0

szFileName db "ololo.txt", 0

szTextBuf db "Рядок тексту, записаний у файл", 10, 13, 10, 13, 0

newLine db 0ah, 0dh, 0ah, 0dh, 0

.code

MySaveFileName proc

LOCAL ofn : OPENFILENAME

invoke RtlZeroMemory, ADDR ofn, SIZEOF ofn ; спочатку усі поля обнулюємо

mov ofn.lStructSize, SIZEOF ofn

mov ofn.lpstrFile, OFFSET szFileName

mov ofn.nMaxFile, SIZEOF szFileName

invoke GetSaveFileName, ADDR ofn ; виклик вікна File Save As

ret

MySaveFileName endp

main:

;call MySaveFileName

;cmp eax, 0 ; перевірка: якщо у вікні було натиснуто кнопку Cancel, то EAX=0

;je @exit

invoke GlobalAlloc, GPTR, 32 \* Len

mov pTemp, eax

mov dword ptr [eax], 1

invoke CreateFile, ADDR szFileName,

GENERIC\_WRITE,

FILE\_SHARE\_WRITE,

0, CREATE\_ALWAYS,

FILE\_ATTRIBUTE\_NORMAL,

0

cmp eax, INVALID\_HANDLE\_VALUE

je @exit ;доступ до файлу неможливий

mov hFile, eax

;mov dword ptr [pTemp], 1

mov N, 1

@factorial:

; give Mul\_N32\_LONGOP all params

push Len

push pTemp

push N

push offset R

call Mul\_N32\_LONGOP

; copy content of R to pTemp

push Len

push offset dword ptr [R]

push pTemp

call Copy\_LONGOP

; get line

push offset dword ptr [textBufFactorial]

push offset dword ptr [R]

push Len \* 32

call StrHex\_MY

; write current number to file

invoke lstrlen, ADDR textBufFactorial

invoke WriteFile, hFile, ADDR textBufFactorial, eax, ADDR pResFile, 0

;write new line to file

invoke lstrlen, ADDR newLine

invoke WriteFile, hFile, ADDR newLine, eax, ADDR pResFile, 0

inc N

cmp N, FVal

jle @factorial

invoke CloseHandle, hFile

invoke GlobalFree, pTemp ; free memory

; output

push offset dword ptr [textBufFactorial]

push offset dword ptr [R]

push Len \* 32

call StrHex\_MY

invoke MessageBoxA, 0, ADDR textBufFactorial, ADDR captionHeaderFactorial, MB\_ICONINFORMATION

;push Len

;push offset dword ptr [R]

;push 7

;push offset dword ptr [RF]

;call Div\_N32\_LONGOP

mov ecx, Len \* 4

@c1:

mov dword ptr[textBufSquare + 4 \* ecx - 4], 0ffffffffh

dec ecx

jnz @c1

push offset dword ptr [textBufSquare]

push offset dword ptr [R]

push Len \* 32

call StrDec\_MY

invoke MessageBoxA, 0, ADDR textBufSquare, ADDR captionHeaderSquare, MB\_ICONINFORMATION

@exit:

invoke ExitProcess, 0

end main