Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №9

з дисципліни «Використання функцій API Win32 у програмах на асемблері»

на тему:

**«Програмування операцій ділення чисел»**

Виконала:

студентка 2-го курсу

групи ІО-32

Руденко Т.А.

Номер залікової книжки: 3224

Київ - 2015 р.

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

Лістинг:

.586

.model flat, stdcall

option casemap :none

include d:\masm32\include\kernel32.inc

include d:\masm32\include\user32.inc

include longop.inc

include module.inc

includelib d:\masm32\lib\kernel32.lib

includelib d:\masm32\lib\user32.lib

include d:\masm32\include\windows.inc

include d:\masm32\include\comdlg32.inc

includelib d:\masm32\lib\comdlg32.lib

.data

x dd 1

hFile dd 0

myFileName db 256 dup(0), 0 ;буфер для імені файлу

pResult dd 0

pBuf dd 0

hexCode db 1024 dup(0) , 13,10, 0

line db " ",13,10, 0

n dd 0

nm dd 78

pRes dd 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 myFileName

mov ofn.nMaxFile, SIZEOF myFileName

invoke GetSaveFileName,ADDR ofn ;виклик вікна File Save As

ret

MySaveFileName endp

main:

call MySaveFileName

cmp eax, 0

je @exit

invoke CreateFile, ADDR myFileName,

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

invoke GlobalAlloc, GPTR, 1024

mov pResult, eax

add eax, 512

mov pBuf, eax

mov dword ptr[eax], 1

@cycle:

inc dword ptr[n]

mov eax, dword ptr[n]

cmp eax, nm

jg @endf

push pResult

push pBuf

push x

call Mul\_N\_x\_32\_LONGOP

push offset hexCode

push pResult

push 420

call StrHex\_MY

invoke lstrlen, ADDR hexCode

invoke WriteFile, hFile, ADDR hexCode, eax, ADDR pRes, 0

invoke lstrlen, ADDR line

invoke WriteFile, hFile, ADDR line, eax, ADDR pRes, 0

inc x

push pResult

push pBuf

push 16

call COPY\_LONGOP

jmp @cycle

@endf:

invoke GlobalFree, pResult

invoke CloseHandle, hFile

@exit:

invoke ExitProcess, 0

end main