НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №6

з дісципліни **«**Системне програмування**»**

Виконав:

студент 2 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Порєв В.М.

Київ 2014 р.

.586

.model flat, stdcall

option casemap :none

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

option casemap :none

include \masm32\include\windows.inc

include module.inc

include longop.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.const

.data

TextBuf db 900 dup(?)

Caption db 'Result', 0

ValueA dd 80030001h, 80040001h, 80050002h

ValueB dd 1,2,3,4,5,6,7,8,9,10h,11h,12h,13h,14h,15h,16h,17h,18h,19h,20h,21h,22h,23h

.code

start:

push offset ValueB

push 96

call Shift

push offset ValueB

push offset ValueA

push 96

call Wr

;

push offset TextBuf

push offset ValueB

push 736

call StrHex\_MY

invoke MessageBox, 0, ADDR TextBuf, ADDR Caption, MB\_ICONINFORMATION

invoke ExitProcess,0

end start

.586

.model flat, c

.data

counter dd ?

counter2 dd ?

m dd 31

Nbit dd ?

N dd ?

tmp dd ?

p db ?

p1 db ?

.code

Write proc

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

cmp ah, 0

jz @set0

or byte ptr [esi+ebx], al

jmp @goon

@set0:

not al

and byte ptr [esi+ebx], al

@goon:

mov eax,eax

Write endp

Read proc

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

mov ah, byte ptr [edx+ebx]

and ah, al

Read endp

Wr proc

push ebp

mov ebp,esp

mov esi, [ebp+16] ;число

mov edx, [ebp+12] ;біти

mov edi, [ebp+8] ;кількість

mov Nbit,0

cycle:

xor eax,eax

mov ebx,Nbit

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

mov ah, byte ptr [edx+ebx]

and ah, al

mov ebx,Nbit

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

cmp ah, 0

jz @set0

or byte ptr [esi+ebx], al

jmp @goon

@set0:

not al

and byte ptr [esi+ebx], al

@goon:

inc Nbit

dec edi

jnz cycle

pop ebp

ret 12

Wr endp

Shift proc

push ebp

mov ebp,esp

mov edx, [ebp+12] ;число

mov edi, [ebp+8] ;кількість

mov Nbit,736

mov counter,736

sub counter,edi

inc counter

cycle:

mov ebx,Nbit

sub ebx,edi

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

mov ah, byte ptr [edx+ebx]

and ah, al

mov ebx,Nbit

mov ecx, ebx

shr ebx, 3

and ecx, 07h

mov al, 1

shl al, cl

cmp ah, 0

jz @set0

or byte ptr [edx+ebx], al

jmp @goon

@set0:

not al

and byte ptr [edx+ebx], al

@goon:

dec Nbit

dec counter

jnz cycle

pop ebp

ret 8

Shift endp

End

.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