НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №5

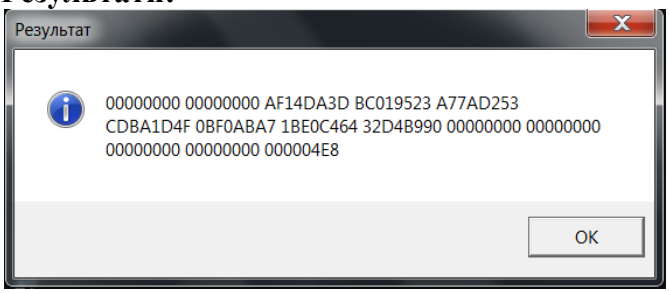
з дисципліни «Системне програмування»

Виконав:
студент 2 курсу
ФІОТ гр. ІО-33
Шуркіна Анастасія

Перевірив:
Порєв В.М.

Київ 2015 р.

Тема: Програмування множення чисел підвищеної розрядності.
Мета: Навчитися програмувати на асемблері множення чисел підвищеної розрядності, а також закріпити навички програмування власних процедур у модульному проекті.
Варіант завдання:N=28
n = N*2+30 = 86.
Результати:



Аналіз результатів: програмою викликаються два вікна-повдіомлення, які містять результати обчислення факторіалу числа та обчислення квадрату цього факторіалу.

Програмний код:

```
.586
.model flat, stdcall
option casemap : none

include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
include \masm32\include\windows.inc
includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\user32.lib
include \masm32\modules\module.inc
include \masm32\modules\longpop.inc

.data

ValueA dd 00000001h, 00000000h, 00000000h,
00000000h, 00000000h, 00000000h, 00000000h
factor dd 00000001h
temp dd 7 dup(0)
ValueB dd 14 dup(0)
temp2 dd 14 dup(0)

Caption db "Результат", 0
n32TextBuf db 224 dup(? )
nnTextBuf db 448 dup(? )
counter dd 0

.code

main :

@cycle1 :
push offset ValueA
push  factor
push offset temp
call Mul_N32_LONGOP
inc factor
inc counter
cmp counter, 86

mov ecx, 7
@swap:
mov ebx, dword ptr[temp + 4 * ecx - 4]
mov dword ptr[ValueA + 4 * ecx - 4], ebx
mov dword ptr[temp + 4 * ecx - 4], 0
dec ecx
jnz @swap

jb @cycle1

push offset n32TextBuf
push offset ValueA
push 224
call StrHex_MY
invoke MessageBoxA, 0, ADDR n32TextBuf, ADDR
Caption, MB_ICONINFORMATION

push offset ValueA
push offset ValueA
push offset ValueB
call Mul_NN_LONGOP

push offset nnTextBuf
push offset ValueB
push 448
call StrHex_MY
invoke MessageBoxA, 0, ADDR nnTextBuf, ADDR
Caption, MB_ICONINFORMATION


invoke ExitProcess, 0
end main
```

Longpop.asm

```
.586
.model flat, c
.data
counter dd 0
counterin dd 0
.code

Add_LONGOP proc
push ebp
mov ebp, esp

mov esi, [ebp + 16]; ESI = адреса A
mov ebx, [ebp + 12]; EBX = адреса B
mov edi, [ebp + 8]; EDI = адреса результату

mov ecx, 20; ECX = потрібна кількість повторень
mov edx, 0
clc; обнулює біт CF регістру EFLAGS
cycle :
mov eax, dword ptr[esi + 4 * edx]
adc eax, dword ptr[ebx + 4 * edx]; додавання групи
з 32 бітів
mov dword ptr[edi + 4 * edx], eax

inc edx
dec ecx; лічильник зменшуємо на 1
jnz cycle
```

```
pop ebp
ret 12

Add_LONGOP endp

Sub_LONGOP proc
push ebp
mov ebp, esp

mov esi, [ebp + 16]; ESI = адреса A
mov ebx, [ebp + 12]; EBX = адреса B
mov edi, [ebp + 8]; EDI = адреса результату

mov ecx, 120; ECX = потрібна кількість повторень
mov edx, 0
clc; обнулює біт CF регістру EFLAGS
cycle :
mov eax, dword ptr[esi + 4 * edx]
sbb eax, dword ptr[ebx + 4 * edx]; віднімання
групи з 32 бітів
mov dword ptr[edi + 4 * edx], eax

inc edx
dec ecx; лічильник зменшуємо на 1
jnz cycle
pop ebp
ret 12
Sub_LONGOP endp


Mul_N32_LONGOP proc
push ebp
mov ebp, esp
mov esi, [ebp + 16]
mov ebx, [ebp + 12]
mov edi, [ebp + 8]


mov ecx, 0
@cycle:
mov eax, dword ptr[esi + 4 * ecx]
mul ebx
add dword ptr[edi + 4 * ecx], eax
add dword ptr[edi + 4 * ecx + 4], edx
inc ecx
cmp ecx, 7
jb @cycle

pop ebp; відновлення стеку
ret 12
Mul_N32_LONGOP endp

Mul_NN_LONGOP proc
push ebp
mov ebp, esp
mov esi, [ebp + 16]; A
mov edi, [ebp + 8]; dest
mov ebp, [ebp + 12]; B

mov counter, 0
@cycle:
mov counterin, -7
@innerCycle :
mov ecx, counter
mov eax, dword ptr[esi + 4 * ecx]
mov ecx, counterin
mov ebx, dword ptr[ebp + 20 + 4 * ecx]
mul ebx
mov ebx, counter
add ebx, counter
add ebx, counter
add ebx, counter
add ebx, edi

adc dword ptr[20 + 4 * ecx + ebx], eax
adc dword ptr[24 + 4 * ecx + ebx], edx
inc counterin

jnz @innerCycle
inc counter

cmp counter, 7
jb @cycle

pop ebp; відновлення стеку
ret 12
Mul_NN_LONGOP endp

end
```

Висновки: під час виконання даної лабораторної роботи було створено програму, що розв’язує факторіал числа та квадрат цього факторіалу з підвищеною точністю. Було покращено навичкм створення модульних програм на асемблері в середовищі Microsoft Visual Studio.