Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра ОТ

**ЗВІТ**

про виконання лабораторної роботи № 10

з дисципліни

“Системне програмування”

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Павлов Валерій  Георгійович |  | студент 2-го курсу  гр. ІП-53 ФІОТ  Голуб Роман Олександрович |

Київ – 2016

**Завдання до лабораторної роботи:**

1. Вивчити методи звернення до процедур і передачі в них параметрів [1].

2. Для свого варіанту індивідуального завдання до лабораторної роботи 9 розробити програму на мові Асемблер, в якій використовувати три процедури з різними способами передачі параметрів:

* через регістри;
* через стек;
* через загальну область пам'яті за допомогою директив EXTRN та PUBLIC.

3. Для цього чисельник дробу зі свого варіанту індивідуального завдання до лабораторної роботи 9 розділити на два доданка, з яких для першого застосувати передачу параметрів і результату через регістри, а для другого – через стек. Для знаменника використовувати метод оголошення загальних змінних директивами **public** і **extern**. Виведення результату\* виконати в основній програмі.

4. Розрахунки (п. 3) повторити в програмі для 5 значень змінних\*\*, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі..

**Лістинг Golub-10.asm:**

.386

.model flat,stdcall

option casemap: none

include \masm32\include\windows.inc

include \masm32\macros\macros.asm

include \masm32\include\masm32.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\msvcrt.inc

includelib \masm32\lib\masm32.lib

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\msvcrt.lib

.data

a1 dd 3.14, -8.5, 5.86, 12.9, 8.22

b1 dd 0.5, 9.77, 4.47, 15.23, 10.14

c1 dd 24.5, 31.28, 48.13, 100.13, 9.35

d1 dd 2.1, 3.33, 2.99, 7.94, 9.01

firhalfchislbuff dd ?

sechalfchislbuff dd ?

znambuff dq ?

a1buff dd ?

b1buff dd ?

rez1 dq ?, ?, ?, ?, ?

rezbuff db 15 dup(?)

rezhead db "Результат:",0

con\_53 dd 53

con\_2 dd 2

con\_4 dd 4

buffdd dd ?

buffdq dq ?

public a1buff, b1buff, con\_4

ZNAM proto

;(53 - sin(a/d) - 2c)/(a/4 - b)

.const

NULL equ 0

count MACRO a1, b1, c1, d1, firhalfchislchisl, sechalfchislchisl, znam, rez1, numb

; counting first half of chisl:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mov eax, con\_53

mov ebx, con\_2

mov ecx, c1[4\*numb]

|  |
| --- |
| Адрес возврата |

finit

call firHalfCHISL

mov firhalfchislbuff, edx

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; counting second half of chisl:$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

|  |
| --- |
| Адрес возврата |
| D1 |
| A1 |

push a1[4\*numb]

push d1[4\*numb]

call secHalfCHISL

mov sechalfchislbuff, edx

;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

;counting znam:##################################################

mov eax, a1[4\*numb]

mov ebx, b1[4\*numb]

mov a1buff, eax

mov b1buff, ebx

call ZNAM

fstp znambuff

;#################################################################

;counting result:!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

fld sechalfchislbuff; st(0) = sechalfchisl[8\*numb]

fld firhalfchislbuff; st(0) = firhalfchisl[8\*numb], st(1) = sechalfchisl[8\*numb]

fsub st(0), st(1); st(0) = firhalfchisl[8\*numb] - sechalfchisl[8\*numb], st(1) = sechalfchisl[8\*numb]

fld znambuff; st(0) = znam, st(1) = firhalfchisl[8\*numb] - sechalfchisl[8\*numb], st(2) = sechalfchisl[8\*numb]

fdiv st(1), st(0); st(0) = znam, st(1) = result, st(2) = sechalfchisl[8\*numb]

fstp buffdd; st(0) = result, st(1) = sechalfchisl[8\*numb]

;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

INVOKE FpuFLtoA,NULL, 13, offset rezbuff, SRC1\_FPU or SRC2\_DIMM

INVOKE MessageBox, NULL, offset rezbuff, offset rezhead, NULL

ENDM

zikl MACRO

counting:

count a1, b1, c1, d1, firhalfchislchisl, sechalfchislchisl, znam, rez1, edi

inc edi

cmp edi, 5

jne counting

ENDM

.code

main:

mov edi, 0

zikl

invoke ExitProcess, 0

firHalfCHISL proc

local con53, con2, c\_1

mov con53, eax

mov con2, ebx

mov c\_1, ecx

fild con53; st(0)=53

fild con2; st(0)=2 ; st(1)=53

fld c\_1; st(0)=c1 ; st(1)=2 ; st(2)=53

fmul st(1), st(0); st(0)=c1 ; st(1)=2\*c1 ; st(2)=2 ; st(3)=53

fstp buffdd; st(0)=2\*c1 ; st(2)=53

fsub st(1), st(0); st(0)=2\*c1 ; st(1)=53-2\*c1

fstp buffdd; st(0)=53-2\*c1

fstp c\_1

|  |
| --- |
| Адрес возврата |

mov edx, c\_1

ret

firHalfCHISL endp

secHalfCHISL proc

push ebp

mov ebp, esp

mov eax, [ebp+12]

mov ebx, [ebp+8]

pop ebp

mov buffdd, eax

fld buffdd; st(0)=a1

mov buffdd, ebx

fld buffdd; st(0)=d1 ; st(1)=a1

|  |
| --- |
| Адрес возврата |
| D1 |
| A1 |

fdiv st(1), st(0); st(0)=d1 ; st(1)=a1/d1

fstp buffdd; st(0)=a1/d1

fsin; st(0)=sin(a1/d1)

fstp buffdd

mov edx, buffdd

ret 8

secHalfCHISL endp

end main

**Module2.asm:**

.386

.model flat, stdcall

public ZNAM

extern a1buff:DWORD, b1buff:DWORD, con\_4:DWORD

.code

ZNAM proc

fld a1buff; st(0)=a1

fild con\_4; st(0)=con\_4 ; st(1)=a1

fdiv st(1), st(0); st(0)=con\_4 ; st(1)=a1/4

fstp a1buff; st(0)=a1/4

fld b1buff; st(0)=b1 ; st(1)=a1/4

fsub st(1), st(0); st(0)=b1 ; st(1)=a1/4 - b1

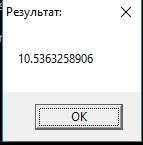
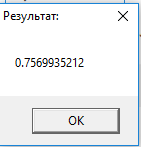
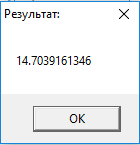
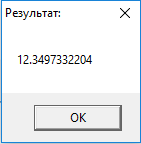
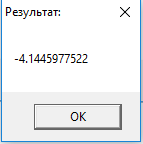
fstp a1buff; st(0)=a1/4 - b1

ret

ZNAM endp

End

**Скріншот роботи:**

**Аналіз:**

Контрольні розрахунки:

1. Sin(a/d)= 0.9971468346

53 -2c= 4

2\*с= 49

a/4= 0,785

53-sin(a/d) -2c= 3,0028531654

a/4 – b= 0,285

***(53-sin(a/d) -2c) / (a/4 – b) = 10,536326***

1. Sin(a/d)= -0.5555631476

53 -2c= -9,560001

2\*с= 65,42

a/4= -2,125

53-sin(a/d) -2c= -9,004438124555812

a/4 – b= -11,895

***(53-sin(a/d) -2c) / (a/4 – b) = 0,7569935***

1. Sin(a/d)= 0.9252623774283

53 -2c= -43,26

2\*с= 96,26

a/4= 1,465

53-sin(a/d) -2c= -44,1852522369514

a/4 – b= -3,005

***(53-sin(a/d) -2c) / (a/4 – b) = 14,70392***

1. Sin(a/d)= 0.998548423992

53 -2c=-147,26

2\*с= 200,26

a/4= 0,785

53-sin(a/d) -2c= -148,2587235823915

a/4 – b= -12,005

***(53-sin(a/d) -2c) / (a/4 – b) = 12,34973***

1. Sin(a/d)= 0.790925347281

53 -2c=34,3

2\*с= 18,7

a/4= 0,785

53-sin(a/d) -2c= 33,509071234793

a/4 – b= -8,085

***(53-sin(a/d) -2c) / (a/4 – b) = -4,144598***

**Висновок:**

Виконуючи дану лабораторну роботу, ми навчилися працювати з макросами, стандартними функціями для арифметичних операцій cопроцесору та форматованного виводу, використанням масивів даних, а головне з передачею параметрів у процедури, такі як: через регістри, через стек процессора, через директиви extern та public. Можна побачити, що розрахунки програми сходяться з роботою програми, що підтверджує її правильну структуровану роботу.