**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине «Архитектура ЭВМ»

на тему: «Программирование целочисленных вычислений

»

Выполнил: студент гр. ИТП-21

Клевцов А.О.

Принял: преподаватель

Савельев В. А.

Гомель 2022

**Цель работы.**

* знакомство с работой центрального процессора и памяти;
* изучение арифметических команд Ассемблера;
* получение навыков работы с отладчиком.

**Задание:**

1. Написать программу, вычисляющую заданное в соответствии с номером студента в журнале выражение:

11. b =(a2-b2)/2 +a\*(k+1)

Код программы представлен ниже:

.386

.model flat, stdcall

option casemap: none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

includelib C:\masm32\lib\kernel32.lib

.data

a dw 5

b dw 2

k dw 3

.code ; b = (a^2 - b^2)/2 +a\*(k +1)

start:

mov ax, a ; AX=a

imul a ; AX=AX\*a

mov bx, ax ; BX=AX =(a^2)

mov ax, b ; AX=b

imul b ; AX=AX\*b =(b^2)

sub bx, ax ; BX=BX-AX =(a^2-b^2)

mov ax, bx ; AX=BX =(a^2-b^2)

mov cx, 2 ; CX=2

idiv cx ; AX=AX/CX =(a^2-b^2)/2

mov bx,ax ; BX=AX =(a^2-b^2)/2

mov ax, k ; AX=k

add ax, 1 ; AX=k+1

imul a ; AX =a\*(k+1)

add ax,bx ; AX=AX+BX =(a^2-b^2)/2 + a\*(k+1)

mov b, ax ; b=(a^2-b^2)/2 + a\*(k+1) , там где деление на 2, будет округление 10.5=>10

; b = (5\*5 - 2\*2)/2 + 5\*(3+1) = 10 + 20=30

; В результате выполнения программы в отладчике B = E1 00, E1(16ричн) =30(десятич)

invoke ExitProcess, NULL

end start

2. Открыть программу в отладчике. Указать адреса расположения переменных в памяти, заполнив таблицу:

|  |  |  |
| --- | --- | --- |
| Имя переменной | Адрес | Порядок байт в памяти |
| a | 403000 | 0500 |
| b | 403002 | 0200 |
| k | 403004 | 0300 |

3. Выполнить программу в пошаговом режиме. После выполнения каждого шага заносить данные в таблицу:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | EAX | EBX | ECX | b | FC | FP | FZ | FS | FA |
| 1 | 0019FFCC | 002C9000 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 2 | 00190005 | 002C9000 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 3 | 00190019 | 002C9000 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 4 | 00190019 | 00230019 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 5 | 00190002 | 00230019 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 6 | 00190004 | 00230019 | 0023B000 | 0200 | 0 | 1 | 1 | 0 | 1 |
| 7 | 00190004 | 00230015 | 0023B000 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 8 | 00190015 | 00230015 | 0023B000 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 9 | 00190015 | 00230015 | 00400002 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 10 | 0019000A | 00230015 | 00400002 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 11 | 0019000A | 0023000A | 00400002 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 12 | 00190003 | 0023000A | 00400002 | 0200 | 0 | 0 | 0 | 0 | 1 |
| 13 | 00190004 | 0023000A | 00400002 | 0200 | 0 | 0 | 0 | 0 | 0 |
| 14 | 00190014 | 0023000A | 00400002 | 0200 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0019001E | 0023000A | 00400002 | 0200 | 0 | 1 | 0 | 0 | 0 |
| 16 | 0019001E | 0023000A | 00400002 | 1E00 | 0 | 1 | 0 | 0 | 0 |

**Вывод:** в ходе работы были изучены типы переменных ассемблера и то как они размещаются в памяти, изучены команды пересылки данных ассемблера, получены навыков работы с отладчиком.