|  |
| --- |
| Министерство науки и высшего образования РФ |
| Федеральное государственное автономное |
| образовательное учреждение высшего образования |
| **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

|  |
| --- |
|  |
| **ОТЧЁТ О ПРАКТИЧЕСКОЙ РАБОТЕ №2** |
| Выявление и сбор требований |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Преподаватель | |  |  |  | Пересунько Е.О. |
|  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ23-17/1б, 032322546 |  |  |  | Гуртякин Е.А |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2024

## Цель

Изучить основы языка ассемблера.

## Задачи

## Разработать программу на Ассемблере, реализующую вычисление Z для заданных пользователем X и Y.

## Описание варианта задания

6. Z = ((X+1)/Y - 1)\*2X;

7. Z = Y\*( 2-(Y+1)/X );

8. Z = (XY - 1)/(X+Y);

9. Z = X^3 + Y -1;

10.Z = (XY + 1)/ X^2;

# Ход работы

Программа была написана на assembler с использованием компилятора Nasm и линкера golink для x86 архитектуры для win32.

Ниже приведён код программы.

global Start

extern printf, scanf, ExitProcess

;=================================================

section '.data' data readable writable

;=================================================

    prompt\_x db "Enter value for X: ", 0

    prompt\_y db "Enter value for Y: ", 0

    prompt\_eq db "Select equation (1-5):", 0

    format\_input db "%d", 0         ; Формат ввода

    format\_output db "%d", 0         ; Формат вывода

    result\_output db "Z = %d", 10, 0 ; Формат для вывода результата (с новой строкой)

    newline db 0xA, 0

    x dd 0                          ; Переменная для X

    y dd 0                          ; Переменная для Y

    z dd 0                          ; Переменная для Z

    eq dd 0

;=================================================

section '.code' code readable writable executable

;=================================================

    Start:

        ; Выводим сообщение

        push prompt\_x

        call printf ;

        add esp, 4                      ; Очистка стека

        ; Вводим x

        push x

        push format\_input

        call scanf

        add esp, 8                      ; Очистка стека

        ; Выводим сообщение

        push prompt\_y

        call printf

        add esp, 4                      ; Очистка стека

        ; Вводим y

        push y

        push format\_input

        call scanf

        add esp, 8                      ; Очистка стека

        ; Выводим сообщение

        push prompt\_eq

        call printf

        add esp, 4                      ; Очистка стека

        ; Выбор уравнения

        push eq

        push format\_input

        call scanf

        add esp, 8                      ; Очистка стека

        mov eax, [eq]

        cmp eax, 1

        je eq1

        cmp eax, 2

        je eq2

        cmp eax, 3

        je eq3

        cmp eax, 4

        je eq4

        cmp eax, 5

        je eq5

        jmp exit

        ; Завершение программы

        call ExitProcess

;=================================================

eq1: ; Вычисление Z = ((X + 1)/Y - 1) \* 2X

;=================================================

    ; 1. Загрузить X в EAX

    mov eax, [x]

    ; 2. Увеличить X на 1

    add eax, 1

    ; 3. Разделить на Y

    mov ebx, [y]                    ; Загрузить Y в EBX

    cdq                             ; Расширение EAX в EDX:EAX перед делением

    idiv ebx                        ; EAX = (X+1) / Y, результат в EAX

    ; 4. Вычесть 1

    sub eax, 1

    ; 5. Умножить результат на 2\*X

    mov ebx, [x]                    ; Загрузить X в EBX

    shl ebx, 1                      ; Умножить X на 2

    imul eax, ebx                   ; Умножить результат на 2\*X

    ; 6. Сохранить результат в переменную z

    mov [z], eax

    ; Вывод результата

    push dword [z]

    push result\_output

    call printf

    add esp, 8                      ; Очистка стека

    jmp exit

;=================================================

eq2: ; Вычисление  Z = Y \* ( 2 - (Y+1)/X )

;=================================================

    ; 1. Загрузить Y в EAX

    mov eax, [y]

    ; 2. Увеличить Y на 1

    add eax, 1

    ; 3. Разделить на X

    mov ebx, [x]                    ; Загрузить X в EBX

    cdq                             ; Расширение EAX в EDX:EAX

    idiv ebx                        ; EAX = (Y+1)/X, результат в EAX

    ; 4. Вычесть результат из 2

    mov ebx, 2                      ; EBX = 2

    sub ebx, eax                    ; EBX = 2 - (Y+1)/X

    ; 5. Умножить результат на Y

    mov eax, [y]                    ; Загрузить Y в EAX

    imul eax, ebx                   ; EAX = Y \* (2 - (Y+1)/X)

    ; 6. Сохранить результат в переменную z

    mov [z], eax

    ; Вывод результата

    push dword [z]

    push result\_output

    call printf

    add esp, 8                      ; Очистка стека

    jmp exit

;=================================================

eq3: ; Вычисление Z = (XY - 1)/(X+Y)

;=================================================

    ; 1. Загрузить X в EAX

    mov eax, [x]

    ; 2. Умножить на Y

    mov ebx, [y]

    imul eax, ebx                   ; EAX = X \* Y

    ; 3. Вычесть 1

    sub eax, 1                      ; EAX = XY - 1

    ; 4. Сложить X и Y

    mov ebx, [x]

    add ebx, [y]                    ; EBX = X + Y

    ; 5. Разделить (XY - 1) на (X + Y)

    cdq                             ; Расширение EAX перед делением

    idiv ebx                        ; EAX = (XY - 1)/(X + Y)

    ; 6. Сохранить результат в переменную z

    mov [z], eax

    ; Вывод результата

    push dword [z]

    push result\_output

    call printf

    add esp, 8                      ; Очистка стека

    jmp exit

;=================================================

eq4: ; Вычисление Z = X^3 + Y - 1

;=================================================

    ; 1. Загрузить X в EAX

    mov eax, [x]

    ; 2. Возвести X в куб

    imul eax, eax                   ; EAX = X \* X (X^2)

    mov ebx, [x]                    ; Загрузить X в EBX

    imul eax, ebx                   ; EAX = X^3

    ; 3. Добавить Y

    add eax, [y]                    ; EAX = X^3 + Y

    ; 4. Вычесть 1

    sub eax, 1                      ; EAX = X^3 + Y - 1

    ; 5. Сохранить результат в переменную z

    mov [z], eax

    ; Вывод результата

    push dword [z]

    push result\_output

    call printf

    add esp, 8                      ; Очистка стека

    jmp exit

;=================================================

eq5: ; Вычисление Z = (XY + 1) / X^2

;=================================================

    ; 1. Загрузить X в EAX

    mov eax, [x]

    ; 2. Умножить X на Y

    mov ebx, [y]

    imul eax, ebx                   ; EAX = X \* Y

    ; 3. Добавить 1

    add eax, 1                      ; EAX = XY + 1

    ; 4. Возвести X в квадрат

    mov ebx, [x]

    imul ebx, ebx                   ; EBX = X^2

    ; 5. Разделить (XY + 1) на X^2

    cdq                             ; Расширение EAX перед делением

    idiv ebx                        ; EAX = (XY + 1) / X^2

    ; 6. Сохранить результат в переменную z

    mov [z], eax

    ; Вывод результата

    push dword [z]

    push result\_output

    call printf

    add esp, 8                      ; Очистка стека

    jmp exit

;=================================================

exit: ; Выход

;=================================================

    call ExitProcess

# Выводы

Были получены навыки работы с ассемблером. Поставленная задача была выполнена.