|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«*Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по лабораторной работе № 2**

**Дисциплина: Машинно-зависимые языки программирования и основы компиляции**

**Название лабораторной работы: Программирование целочисленных вычислений**

Студент гр. ИУ6-41Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Тарасова**

(Подпись, дата) (И.О. Фамилия)

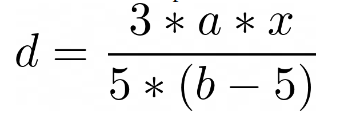
Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.С. Данилюк**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель работы:** изучение форматов машинных команд, команд целочисленной арифметики ассемблера и программирование целочисленных вычислений.

**Задание:** вычислить целочисленное выражение:

****

**Код программы:**

%include "../lib.asm"

section .data

ExitMsg db "Press Enter to Exit",10

lenExit equ $-ExitMsg

EnterAMsg db "Enter the A: "

lenEnterA equ $-EnterAMsg

EnterBMsg db "Enter the B which isn't 5: "

lenEnterB equ $-EnterBMsg

EnterXMsg db "Enter the X: "

lenEnterX equ $-EnterXMsg

ResMsg db "The result is: "

lenRes equ $-ResMsg

section .bss

a resd 1

b resd 1

d resd 1

x resd 1

OutBuf resb 10

InBuf resb 10

lenIn equ $-InBuf

section .text

global \_start

\_start:

; ; write EnterAMsg

mov rax, 1

mov rdi, 1

mov rsi, EnterAMsg

mov rdx, lenEnterA

syscall

; read A

mov rax, 0

mov rdi, 0

mov rsi, InBuf

mov rdx, lenIn

syscall

mov esi, InBuf

call StrToInt64

cmp ebx, 0

mov [a], eax

; write EnterBMsg

mov rax, 1

mov rdi, 1

mov rsi, EnterBMsg

mov rdx, lenEnterB

syscall

; read B

mov rax, 0

mov rdi, 0

mov rsi, InBuf

mov rdx, lenIn

syscall

mov esi, InBuf

call StrToInt64

cmp ebx, 0

mov [b], eax

; write EnterXMsg

mov rax, 1

mov rdi, 1

mov rsi, EnterXMsg

mov rdx, lenEnterX

syscall

; read X

mov rax, 0

mov rdi, 0

mov rsi, InBuf

mov rdx, lenIn

syscall

mov esi, InBuf

call StrToInt64

cmp ebx, 0

mov [x], eax

;знаменатель

mov ebx, [b] ; EBX:= b

mov edx, 5 ; EDX := 5

sub ebx, 5 ; EBX - 5

mov eax, ebx

imul edx ; EBX \* EDX

mov ecx, eax ; b:= EAX (произведение и частное всегда в EAX)

;числитель

mov eax, [a] ; AX:= a

mov ebx, 3 ; BX:= 3

imul ebx ; AX \* BX, резульат в EAX

mov ebx, [x] ; EBX:= x

imul ebx ; EAX \* EBX, резульат в EDX:EAX

; частное

mov ebx, ecx ; EBX:= ECX

idiv ebx ; EDX:EAX : EBX, результат в EAX, остаток в EDX

mov ebx, eax ; переносим данные из EAX в EBX

; write ResMsg

mov rax, 1

mov rdi, 1

mov rsi, ResMsg

mov rdx, lenRes

syscall

mov eax, ebx ; переносим число обратно в EAX

; конвертируем частное из целого в строку

mov esi, OutBuf

cwd

call IntToStr64

; write

mov edx, eax

mov eax, 1

mov edi, 1

syscall

; write ExitMsg

mov rax, 1

mov rdi, 1

mov rsi, ExitMsg

mov rdx, lenExit

syscall

; read Enter

mov rax, 0

mov rdi, 0

mov rsi, InBuf

mov rdx, lenIn

syscall

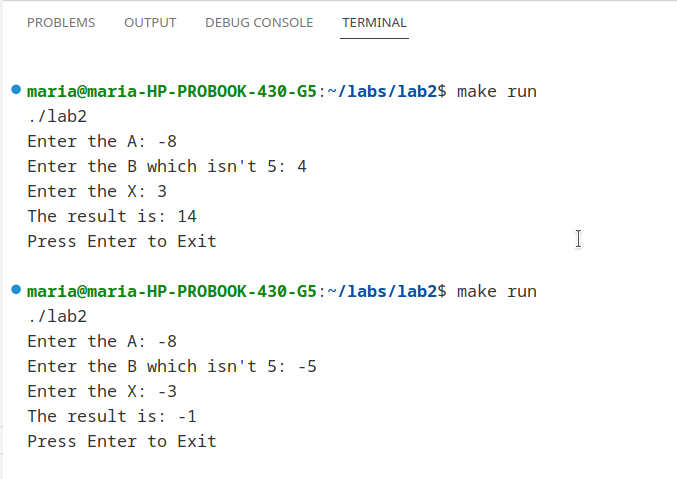
;exit

mov rax, 60

xor rdi, rdi

syscall

**Вывод программы:**

Рисунок 1: результат программы

**Вывод:** в данной лабораторной работе мы изучали форматы машинных команд, программировали целочисленные вычисления, научились распределять данные по регистрам, научились понимать, куда и в каком формате записываются данные при делении и умножении.

**Контрольные вопросы:**

**1.** Что такое машинная команда? Какие форматы имеют машинные команды процессора IA32? Чем различаются эти форматы?

Машинная команда представляет собой код, определяющий операцию вычислительной машины и данные, участвующие в операции.

Различают два вида команд, обрабатывающих операнд в памяти:

 - команды без байта sib;

 - команды, содержащие байт sib.

**2.** Назовите мнемоники основных команд целочисленной арифметики. Какие форматы для них можно использовать?

**1. Команда пересылки данных** – пересылает число размером 1, 2 или 4 байта из источника в приемник: *mov Приемник, Источник*

**2. Команды сложения** – складывает операнды, а результат помещает по адресу первого операнда. В отличие от ADD команда ADС добавляет к результату значение бита флага переноса CF:

*ADD Операнд1, Операнд2*

*ADC Операнд1, Операнд2*

**3. Команды вычитания** – вычитает из первого операнда второй и результат помещает по адресу первого операнда. В отличие от SUB команда SBB вычитает из результата значение бита флага переноса CF. Допустимые варианты те же, что и у сложения:

*SUB Операнд1, Операнд2*

*SBB Операнд1, Операнд 2*

**4. Команды умножения:**

*MUL <Операнд2>*

*IМUL <Операнд2>*

В качестве второго операнда нельзя указать непосредственное значение.

**5. Команды «развертывания» чисел** – операнды в команде не указываются. Операнд и его длина определяются кодом команды и не могут быть изменены. При выполнении команды происходит расширение записи числа до размера результата посредством размножения знакового разряда.

**CBW** ; байт в слово AL -> AX

**CWD** ; слово в двойное слово AX -> DX:AX

**CDQ** ; двойное слово в учетверенное EAX -> EDX:EAX

**CWDE** ; слово в двойное слово AX -> EAX

**6. Команды деления**

*DIV <Операнд2>*

*IDIV <Операнд2>*

В качестве второго операнда нельзя указать непосредственное значение.

**3.** Сформулируйте основные правила построения линейной программы вычисления заданного выражения.

- в операциях сложения и вычитания должен присутствовать хотя бы один регистр;

- в операциях деления и умножения множителем и делителем может быть только регистр. Формат и расположение результата зависят от формата множителя/делителя.

**4.** Почему ввод-вывод на языке ассемблера не программируют с использованием соответствующих машинных команд? Какая библиотека используется для организации ввода вывода в данной лабораторной?

Системные функции read и write, примеры обращения к которым включены в заготовки, осуществляют ввод и вывод данных в виде символьных строк. Для выполнения арифметических операций числа должны быть представлены в памяти в одном из внутренних форматов, в которых знак, если он предусмотрен, кодируется первым битом, а само число записано в двоичной системе счисления. Для преобразования в такое представление и обратно будем использовать специально разработанные подпрограммы:

- подпрограмма преобразования строки, завершающейся байтом 0x0A,

в число StrToInt;

- подпрограмма преобразования числа в строку, завершающуюся бай-

том 0x0A, в число IntToStr.

**5.** Расскажите, какие операции используют при организации ввода-вывода.

Операции преобразования строки в число и число в строку, операции ввода, операции вывода.