|  |  |
| --- | --- |
| 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 и основы компиляции**

**Название лабораторной работы: Программирование целочисленных вычислений**

Студент гр. ИУ6-41Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Самодурова**

(Подпись, дата) (И.О. Фамилия)

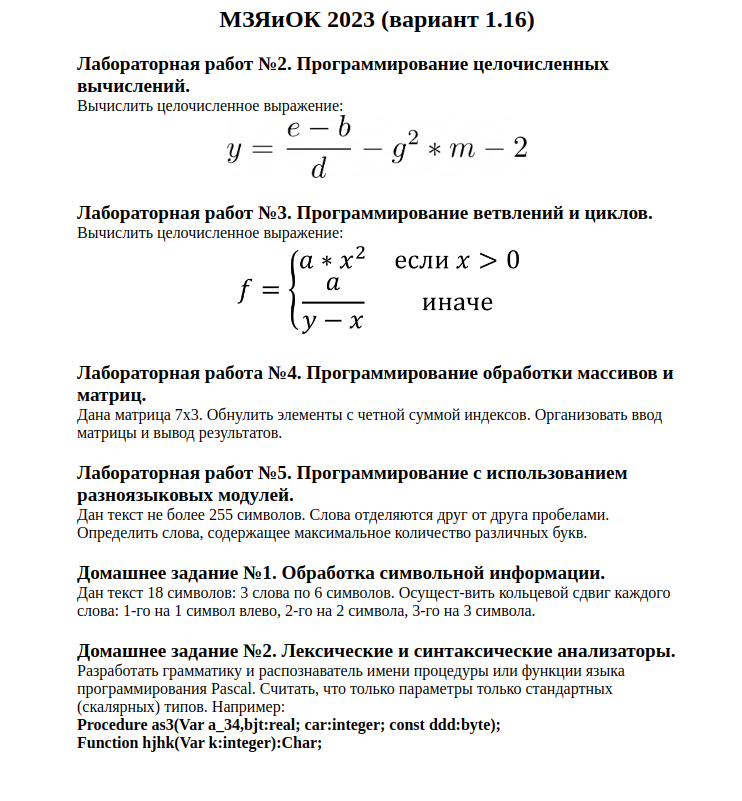
Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. С. Данилюк**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

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

**Текст задания:**



**Код программы:**

%include "io64.asm"

section .data

ExitMsg db "Press Enter to Exit",10

lenExit equ $-ExitMsg

InputMsg db "Input 5 numbers with Enter",10

lenInput equ $-InputMsg

EMsg db "Inter E",10

lenE equ $-EMsg

BMsg db "Inter B",10

lenB equ $-BMsg

DMsg db "Inter D (non 0)",10

lenD equ $-DMsg

QMsg db "Inter Q",10

lenQ equ $-QMsg

MMsg db "Inter M",10

lenM equ $-MMsg

section .bss

InBuf resb 10

lenIn equ $-InBuf

Q resw 1

E resw 1

B resw 1

D resw 1

M resw 1

X resw 1

section .text

global \_start

\_start:

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, InputMsg ; адрес выводимой строки

mov rdx, lenInput ; длина строки

syscall ; вызов системной функции

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, EMsg ; адрес выводимой строки

mov rdx, lenE ; длина строки

syscall ; вызов системной функции

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [E], rax

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, BMsg ; адрес выводимой строки

mov rdx, lenB ; длина строки

syscall ; вызов системной функции

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [B], rax

input: mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, DMsg ; адрес выводимой строки

mov rdx, lenD ; длина строки

syscall ; вызов системной функции

mov eax, 0 ; системная функция 0 (read)

mov edi, 0 ; дескриптор файла stdin=0

mov esi, InBuf ; адрес вводимой строки

mov edx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [D], ax

mov ax, [D]

mov bx, 0

cmp ax, bx

je input

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, QMsg ; адрес выводимой строки

mov rdx, lenQ ; длина строки

syscall ; вызов системной функции

mov eax, 0 ; системная функция 0 (read)

mov edi, 0 ; дескриптор файла stdin=0

mov esi, InBuf ; адрес вводимой строки

mov edx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [Q], ax

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, MMsg ; адрес выводимой строки

mov rdx, lenM ; длина строки

syscall ; вызов системной функции

mov eax, 0 ; системная функция 0 (read)

mov edi, 0 ; дескриптор файла stdin=0

mov esi, InBuf ; адрес вводимой строки

mov edx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [M], eax

mov ax, [Q]

imul ax ;q\*q

mov bx, [M]

imul bx

mov bx, ax

mov ax, [E]

sub ax, [B] ;E-B

mov cx, [D]

cwd

idiv cx ;(E-B)/D

sub ax,bx

sub ax, 2

cwde

mov rsi, InBuf

call IntToStr64

; write

mov rdx, rax

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

syscall

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, ExitMsg ; адрес выводимой строки

mov rdx, lenExit ; длина строки

syscall ; вызов системной функции

; read

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

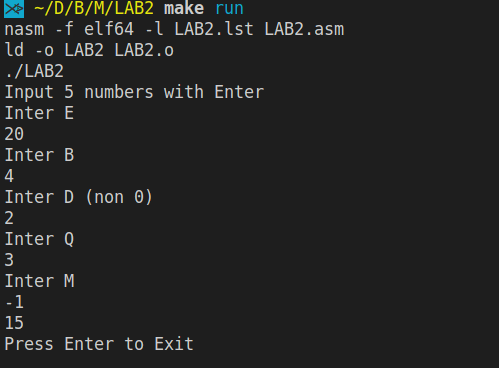
; exit

mov rax, 60 ; системная функция 60 (exit)

xor rdi, rdi ; return code 0

syscall ; вызов системной функции

**Результат работы программы:**

Рисунок 1 — результат выполнения

**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. Процессоры семейства IA-32 поддерживают арифметические операции (сложение, вычитание, умножение и т.п.) над однобайтовыми, двухбайтовыми и четырехбайтовыми целыми числами.

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

Mov, add, sub, div, idiv, mul, imul

Процессоры семейства IA-32 поддерживают арифметические операции (сложение, вычитание, умножение и т.п.) над однобайтовыми, двухбайтовыми и четырехбайтовыми целыми числами.

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

1) Записать в память инициализированные и неинициализированные переменные. 2) Переместить переменные в соответсвующие регистры

3) Выполнить нужные операции

4) Сохранить результат в память

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. Однако 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 этого используется специальная вспомогательная библиотека io64.

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

Подпрограмма преобразования строки, завершающейся байтом 0x0A, в число StrToInt.

Вход: ESI – адрес строки, содержащей запись числа и завершающаяся байтом 0x0A (положительные числа должны вводиться без знака, отрицательные – содержать знак в первой позиции).

Выход: EAX – 32-х разрядное число, EBX содержит 0, если преобразование прошло без ошибок, и 1, если в процессе преобразования обнаружен ввод недопустимого символа или введенное число не попадает в заданный интервал. Подпрограмма преобразования числа в строку, завершающуюся байтом 0x0A, в число IntToStr.

Вход: EAX – число, ESI – адрес области памяти для размещения строки результата (7 байт).

Выход: EAX – размер строки результата – запись числа будет прижата к левой границе области по адресу ESI и завершаться байтом 0x0A.

**Вывод:** в ходе выполнения данной лабораторной работы я изучила форматы машинных команд, команд целочисленной арифметики ассемблера и программирование целочисленных вычислений.