Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»

Кафедра «Программная инженерия и вычислительная техника»

«Машино-зависимые языки программирования»

Отчет

по лабораторной работе №3

«Ввод-вывод на ассемблере (в DOS и Linux)»

|  |  |
| --- | --- |
|  | Выполнила |
|  | студентка группы ИКПИ-25 |
|  | А. Ю. Яскунова |
|  |  |
|  | Проверил |
|  | Ст. преподаватель |
|  | И.Л. Савельев |
|  |  |

2023 г.

**1. Задание**

Реализовать целочисленные вычисления, сделанные в лабораторной работе №1 или №2(по указанию преподавателя), полностью используя язык Ассемблера для организации корректного ввода-вывода информации:

• исходные данные должны вводиться с проверкой правильности вводимых символов;

• входные данные и результат должны быть проверены на область допустимых значений;

• при наличии ошибки должно быть выдано соответствующее сообщение.

**Вариант №26 из ЛР № 2**

b\*a-20, если a < b

20, если a = b

9\*a/b, если a > b

**2. Текст программы**

**2.1 Модуль program.asm**

%include 'asm\_calc.asm'

section .data

promptA db "Enter a: ", 0

promptB db "Enter b: ", 0

promptError db "Error! Not unsigned int! ", 0

promptTask db "b\*a-20, если a<b 20, если a=b 9\*a/b, если a>b ", 0

a dd 0

b dd 0

result\_asm\_num dd 0

section .bss

buffer resb 16

section .text

global main

main:

mov eax, 4

mov ebx, 1

mov ecx, promptTask

mov edx, 60

int 0x80

; вывод приветствия и ввод первого числа

mov eax, 4

mov ebx, 1

mov ecx, promptA

mov edx, 8

int 0x80

call read\_num

mov [a], eax

mov eax, 4

mov ebx, 1

mov ecx, promptB

mov edx, 8

int 0x80

call read\_num

mov [b], eax

call calculate\_expression\_asm

; завершение программы

mov eax, 1

xor ebx, ebx

int 0x80

ret

write\_eax:

lea edi, [buffer + 15]

mov byte [edi], 0x0A ; newline character

test eax, eax,

jns convert\_write

mov byte [edi], '-'

neg eax

convert\_write:

dec edi

xor edx, edx

mov ecx, 10

div ecx

add dl, '0'

mov [edi], dl

test eax, eax

jnz convert\_write

; Write to console

mov eax, 4

mov ebx, 1

lea ecx, [edi]

lea edx, [buffer + 16]

sub edx, ecx

int 0x80

ret

read\_num:

; Read from console

mov eax, 3

mov ebx, 0

lea ecx, [buffer]

mov edx, 16

int 0x80

; Convert string to number

xor eax, eax

lea esi, [buffer]

convert\_read:

movzx edx, byte [esi]

cmp dl, 0x0A ; newline character

je done

imul eax, 10

sub edx, '0'

add eax, edx

inc esi

jmp convert\_read

done:

cmp eax, 0

jl error

cmp eax, 65535

jg error

ret

error:

mov eax, 4

mov ebx, 1

mov ecx, promptError

mov edx, 25

int 0x80

jmp exit

exit:

ret

**2.2 Модуль asm\_calc.asm**

; b\*a-20, если a<b 20, если a=b 9\*a/b, если a>b

calculate\_expression\_asm:

mov eax, [a]

mov ebx, [b]

cmp eax, ebx ; Сравнение a и b

jl less\_than ; Если a < b, переход к метке less\_than

je equal ; Если a = b, переход к метке equal

jg greater\_than ; Если a > b, переход к метке greater\_than

less\_than:

mov eax, dword [a] ; Загрузка значения a в EAX

imul eax, dword [b] ; Умножение EAX на b

sub eax, 20 ; Вычитание 20 из EAX

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

call write\_eax

ret

equal:

mov eax, 20 ; Загрузка значения 20 в EAX

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

call write\_eax

ret

greater\_than:

cmp ebx, 0

je error\_calc

mov eax, 9 ; Загрузка значения 9 в EAX

imul eax, dword [a] ; Умножение EAX на a

cdq ; Расширение знака из EAX в EDX:EAX

idiv dword [b] ; Деление EDX:EAX на b

call write\_eax

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

ret

error\_calc:

ret

**2.3. Модуль makefile**

all:

nasm -f elf64 -F dwarf -g program.asm

gcc -m64 -o program program.o -no-pie

**3. Сборка проекта**

make

**4. Выполнение программы**

**4.1. Запуск программы**

./program

**4.2. Входные данные**

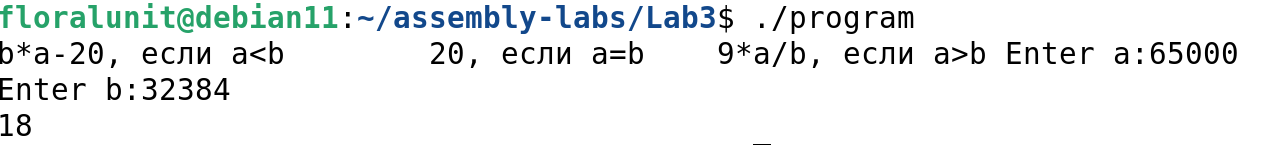
a = 65000

b= 32384

**4.3. Ожидаемый результат выполнения**

18

**4.4. Результат выполнения**



**5. Вывод**

Результат выполнения программы соответствует ожидаемому результату. Работа выполнена в полном объеме.