

Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Сахно Никита Вячеславович

Содержание

Цель работы	1
Задание	1
Выполнение лабораторной работы	1
Символьные и численные данные в NASM	1
Выполнение арифметических операций в NASM	6
Ответы на вопросы по программе	9
Выполнение заданий для самостоятельной работы	10
Выводы	11
Список литературы	Ошибка! Закладка не определена.

Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

Выполнение лабораторной работы

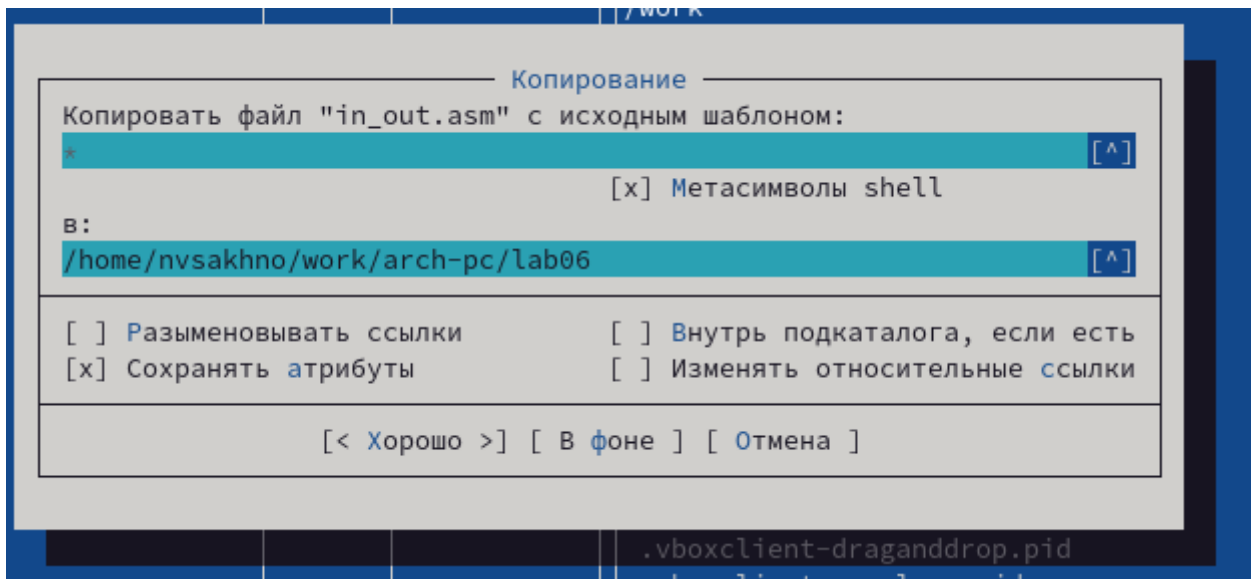
Символьные и численные данные в NASM

С помощью утилиты `mkdir` создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. 1).

```
[nvsakhno@fedora arch-pc]$ mkdir lab06
[nvsakhno@fedora arch-pc]$ cd lab06
[nvsakhno@fedora lab06]$ touch lab6-1.asm
[nvsakhno@fedora lab06]$ ls
lab6-1.asm
[nvsakhno@fedora lab06]$
```

Создание директории

С помощью утилиты touch создаю файл lab6-1.asm и Копирую в текущий каталог файл in_out.asm с помощью midnight commander и функциональной клавишей f5, т.к. он будет использоваться в других программах (рис. 2).



Создание копии файла

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 3).

```
GNU nano 7.2 /home/nvsakhno/work/ar
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call printf
call quit
```

Редактирование файла

Создаю исполняемый файл программы и запускаю его (рис. 4). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-1.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[nvsakhno@fedora lab06]$ ./lab6-1
j
[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 5).

```
GNU nano 7.2
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit
```

Редактирование файла

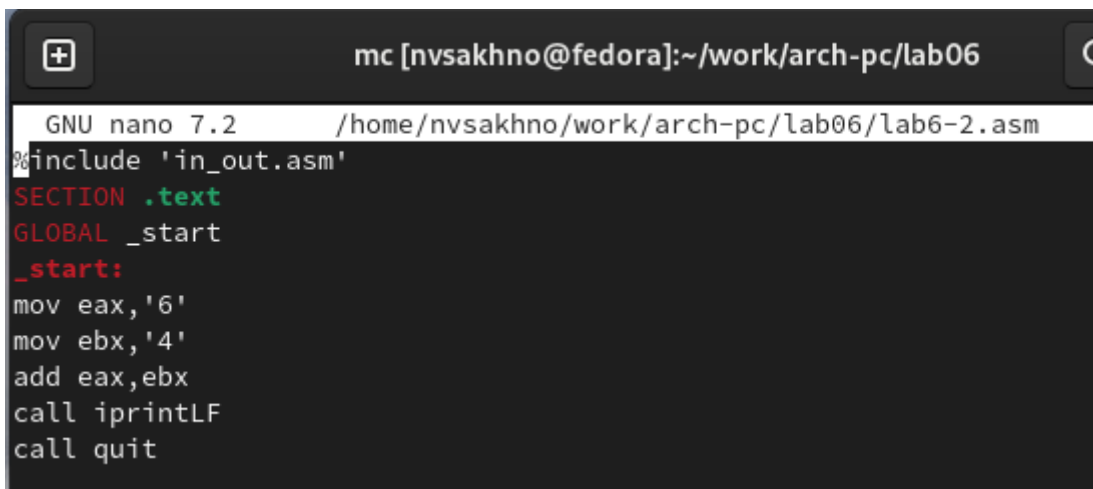
Создаю новый исполняемый файл программы и запускаю его (рис. 6). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-1.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[nvsakhno@fedora lab06]$ ./lab6-1

[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

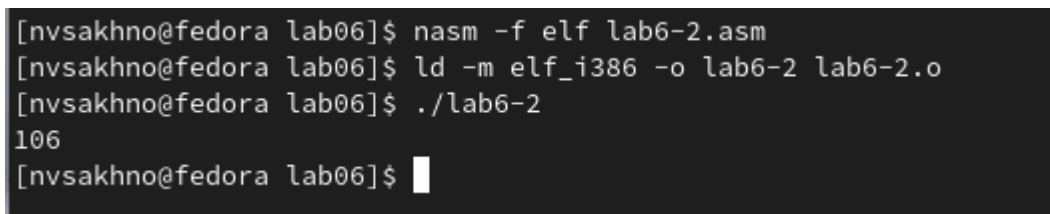
Создаю новый файл lab6-2.asm с помощью утилиты touch и ввожу в файл текст другой программы для вывода значения регистра eax (рис. 7).



```
mc [nvsakhno@fedora]:~/work/arch-pc/lab06
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Редактирование файла

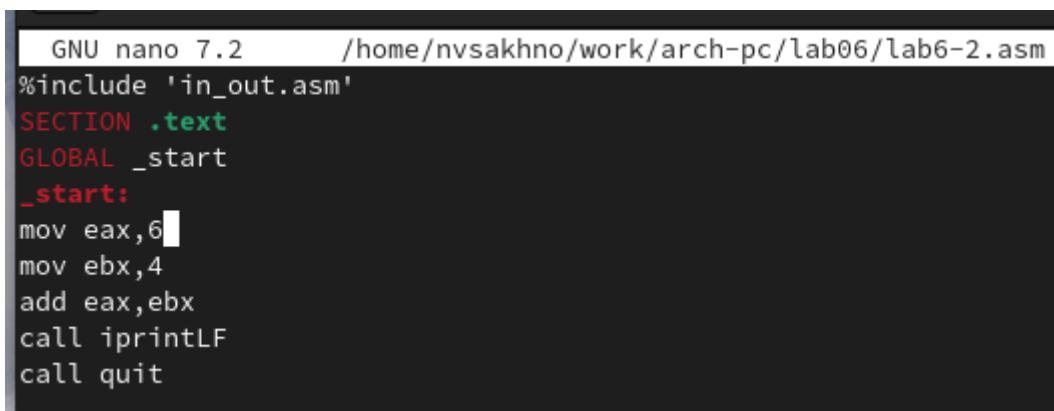
Создаю и запускаю исполняемый файл lab6-2 (рис. 8). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.



```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-2.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[nvsakhno@fedora lab06]$ ./lab6-2
106
[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис.9).



```
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 10). Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-2.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[nvsakhno@fedora lab06]$ ./lab6-2
10
[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

Заменяю в тексте программы функцию `iprintLF` на `iprint` (рис. 11).

```
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 12). Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией `iprintLF`, а `iprint` не добавляет к выводу символ переноса строки, в отличие от `iprintLF`.

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-2.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[nvsakhno@fedora lab06]$ ./lab6-2
10[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

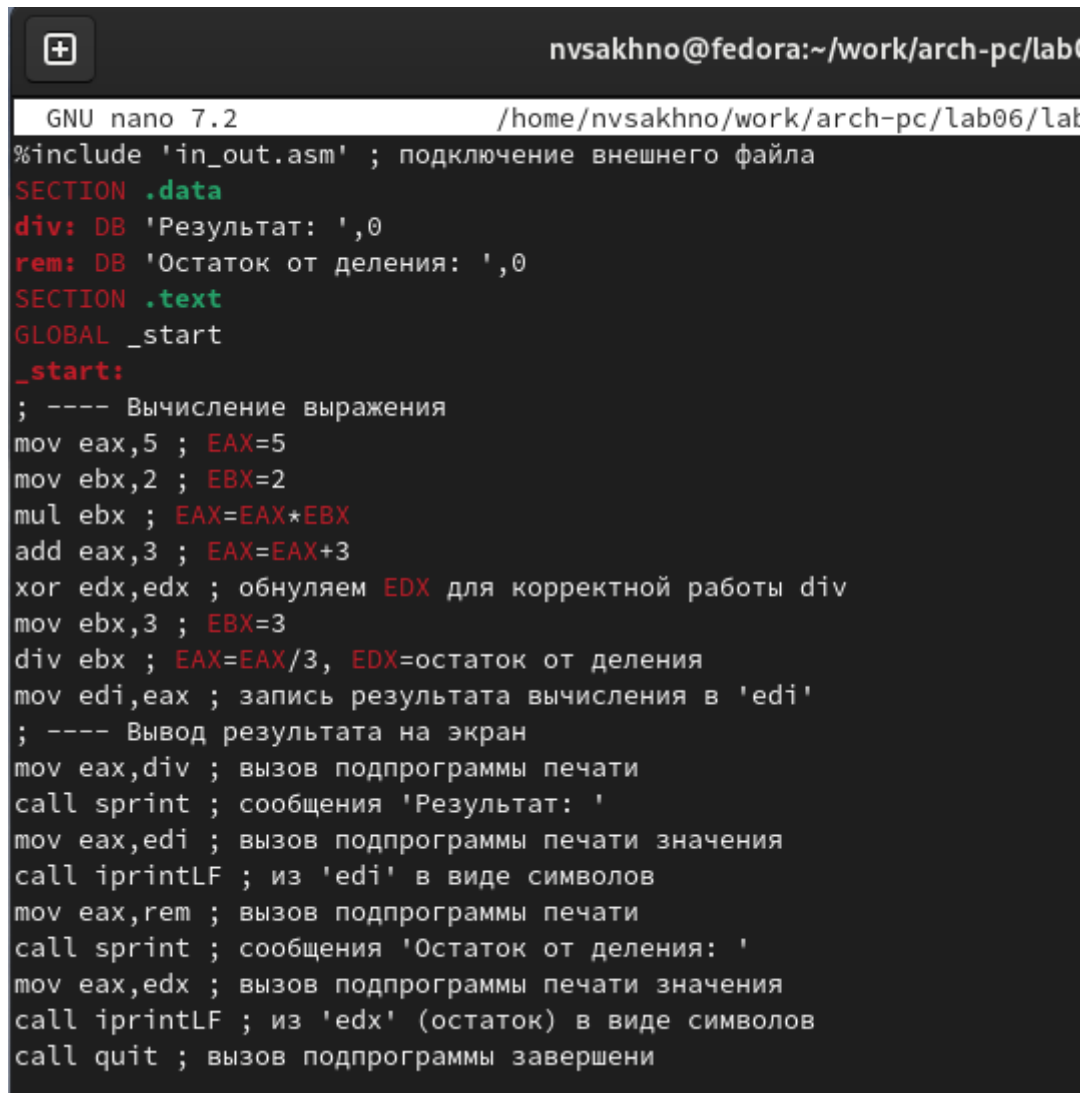
Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью утилиты touch (рис. 13).

```
10[nvsakhno@fedora lab06]$ touch lab6-3.asm
[nvsakhno@fedora lab06]$
```

Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис. 14).



```
nvsakhno@fedora:~/work/arch-pc/lab06/
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab06/lab6-3.asm
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершени
```

Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 15).

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-3.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[nvsakhno@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения $f(x) = (4 * 6 + 2)/5$ (рис. 16).

```
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершени
```

Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 17). Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно.

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-3.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[nvsakhno@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

Создаю файл variant.asm с помощью утилиты touch (рис. 18).

```
[nvsakhno@fedora lab06]$ touch variant.asm
[nvsakhno@fedora lab06]$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o  lab6-3.asm  variant.asm
lab6-1      lab6-1.o    lab6-2.asm  lab6-3    lab6-3.o
```

Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 19).

```

+ nvsakhno@fedora:~/work/arch-pc/lab06
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab06/variant.asm
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x'
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprint
mov eax, edx
call iprintLF
call quit

```

Редактирование файла

Создаю и запускаю исполняемый файл (рис. 20). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 19.

```
[nvsakhno@fedora lab06]$ nasm -f elf variant.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[nvsakhno@fedora lab06]$ ./variant
Введите No студенческого билета:
1132230298
Ваш вариант: 19
[nvsakhno@fedora lab06]$
```

Запуск исполняемого файла

Ответы на вопросы по программе

1. За вывод сообщения "Ваш вариант" отвечают строки кода:

```
mov eax,rem
call sprint
```

2. Инструкция mov esx, x используется, чтобы положить адрес вводимой строки x в регистр esx mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div
mov ebx,20 ; ebx = 20
div ebx ; eax = eax/20, edx - остаток от деления
inc edx ; edx = edx + 1
```

5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Инструкция inc edx увеличивает значение регистра edx на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx
call iprintLF
```

Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью утилиты touch. Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения $(1/3x+5)*7$. Это выражение было под вариантом 19.(рис. 21)

```
GNU nano 7.2 /home/nvsakhno/work/arch-pc/lab06/lab6-4.asm
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция инициированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не инициированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенны
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
add ebx,1/3; eax = eax*1/3 = x*1/3
mov eax,5 ; запись значения 5 в регистр ebx
mul ebx; EAX=EAX*EBX+EAX = (x*1/3+5)
add eax,7; eax = eax*7 = (x*1/3+5)*7
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprint ; из 'edi' в виде символов
Имя файла для записи: /home/nvsakhno/work/arch-pc/lab06/lab6-4.asm
^G Справка M-D Формат DOS M-A Доп. в начало M-
^C Отмена M-M Формат Mac M-P Доп. в конец ^T
```

Написание программы

Создаю и запускаю исполняемый файл (рис. 22).

```
[nvsakhno@fedora lab06]$ nasm -f elf lab6-4.asm
[nvsakhno@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[nvsakhno@fedora lab06]$ ./lab6-4
Введите значение переменной x: 3
Результат: 292[nvsakhno@fedora lab06]$ ./lab6-4
```

Запуск исполняемого файла

Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.