

# **Отчёт по лабораторной работе №6**

**Дисциплина: Архитектура компьютера**

Абакумов Тимофей Александрович

# Содержание

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Цель работы</b>                                      | <b>5</b>  |
| <b>2</b> | <b>Задание</b>  | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>                   | <b>7</b>  |
| 3.1      | Символьные и численные данные в NASM . . . . .          | 7         |
| 3.2      | Выполнение арифметических операций в NASM . . . . .     | 12        |
| 3.3      | Выполнение заданий для самостоятельной работы . . . . . | 16        |
| <b>4</b> | <b>Выводы</b>   | <b>18</b> |

# Список иллюстраций

|      |                                 |    |
|------|---------------------------------|----|
| 3.1  | Создание нового файла . . . . . | 7  |
| 3.2  | Копирование файла . . . . .     | 7  |
| 3.3  | Вставка программы . . . . .     | 8  |
| 3.4  | Запуск программы . . . . .      | 9  |
| 3.5  | Изменение программы . . . . .   | 9  |
| 3.6  | Запуск программы . . . . .      | 9  |
| 3.7  | Создание нового файла . . . . . | 10 |
| 3.8  | Создание нового файла . . . . . | 10 |
| 3.9  | Запуск программы . . . . .      | 11 |
| 3.10 | Изменение программы . . . . .   | 11 |
| 3.11 | Запуск программы . . . . .      | 11 |
| 3.12 | Создание нового файла . . . . . | 12 |
| 3.13 | Изменение программы . . . . .   | 12 |
| 3.14 | Запуск программы . . . . .      | 13 |
| 3.15 | Изменение программы . . . . .   | 14 |
| 3.16 | Запуск программы . . . . .      | 14 |
| 3.17 | Создание файла . . . . .        | 14 |
| 3.18 | Изменение программы . . . . .   | 15 |
| 3.19 | Запуск программы . . . . .      | 16 |
| 3.20 | Создание файла . . . . .        | 16 |
| 3.21 | Изменение программы . . . . .   | 17 |
| 3.22 | Запуск программы . . . . .      | 17 |

## **Список таблиц**

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

### **Порядок выполнения лабораторной работы**

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

## 3 Выполнение лабораторной работы

### 3.1 Символьные и численные данные в NASM

1. Для начала создадим каталог для программ лабораторной работы № 6, перейдём в него и создадим файл lab6-1.asm с помощью утилиты touch (рис. 3.1).

```
taabakumov@dk3n55 ~ $ mkdir ~/work/arch-pc/lab06  
taabakumov@dk3n55 ~ $ cd ~/work/arch-pc/lab06  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-1.asm  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.1: Создание нового файла

2. Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. 3.2).

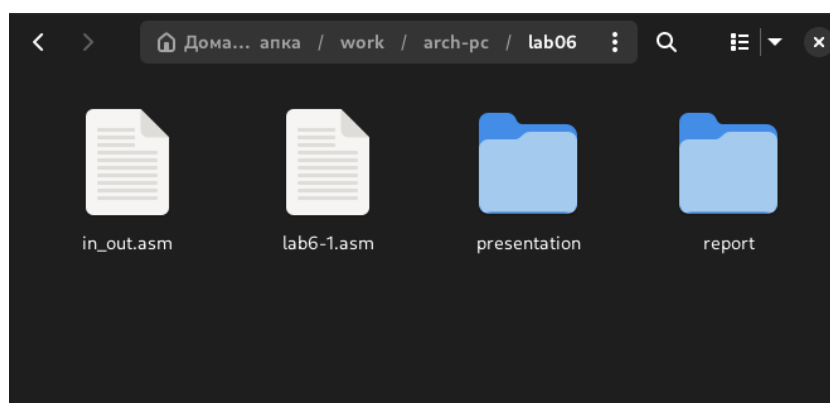
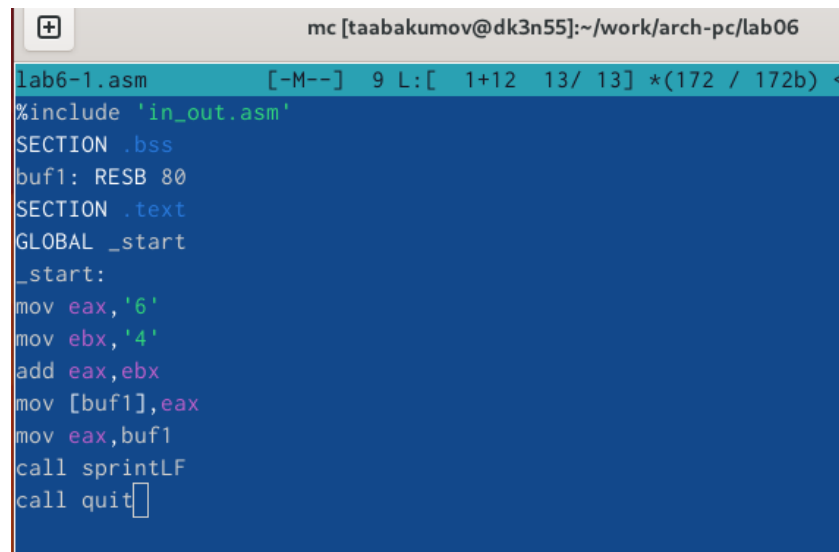


Рис. 3.2: Копирование файла

3. После этого откроем созданный файл lab6-1.asm, вставим в него программу вывода значения регистра eax (рис. 3.3).



```
lab6-1.asm [-M--] 9 L:[ 1+12 13/ 13] *(172 / 172b) <
#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 sprintLF
call quit
```

Рис. 3.3: Вставка программы

Код программы из пункта 3:

```
%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 sprintLF
call quit
```

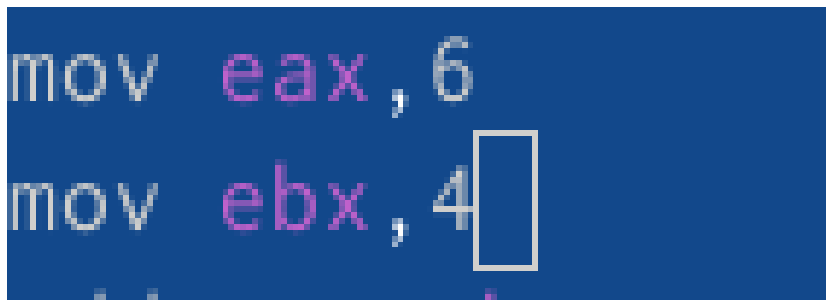


4. Создадим исполняемый файл программы и запустим его. Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6 (рис. 3.4).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 3.4: Запуск программы

5. Далее изменим в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 3.5).



```
mov eax, 6
mov ebx, 4
```

Рис. 3.5: Изменение программы

6. Создадим новый исполняемый файл программы и запустим его. Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 3.6).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-1

taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

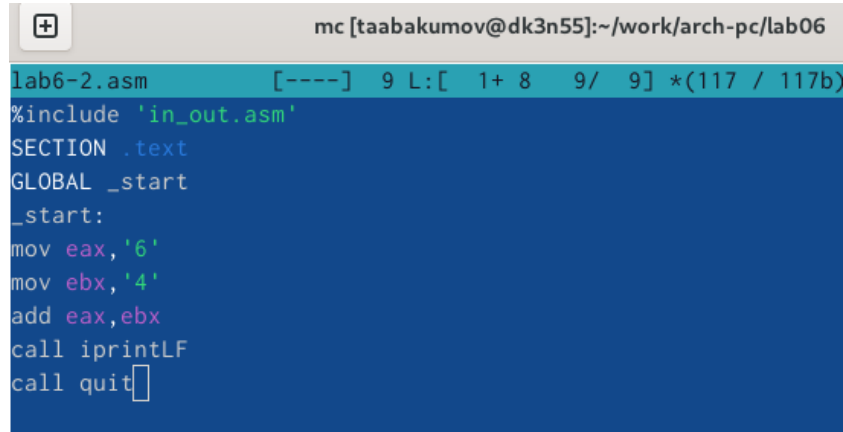
Рис. 3.6: Запуск программы

7. Создадим новый файл lab6-2.asm с помощью утилиты touch (рис. 3.7).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-2.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.7: Создание нового файла

8. Введём в файл текст другой программы для вывода значения регистра `eax` (рис. 3.8).



```
lab6-2.asm [----] 9 L:[ 1+ 8 9/ 9] *(117 / 117b)
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 3.8: Создание нового файла

Код программы из пункта 8:

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

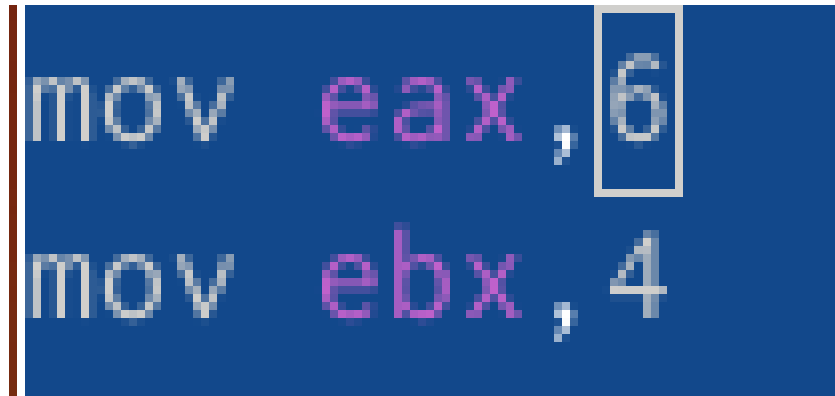
9. Создадим и запустим исполняемый файл `lab6-2` (рис. 3.10). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не

символ, хотя все еще происходит именно сложение кодов символов “6” и “4” (рис. 3.9).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
106
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.9: Запуск программы

10. Заменим в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4. Заменим в тексте программы функцию `iprintLF` на `iprint` (рис. 3.13). (рис. 3.10).



```
mov eax, 6
mov ebx, 4
```

Рис. 3.10: Изменение программы

11. Создадим и запустим новый исполняемый файл. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10 (рис. 3.11).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 3.11: Запуск программы

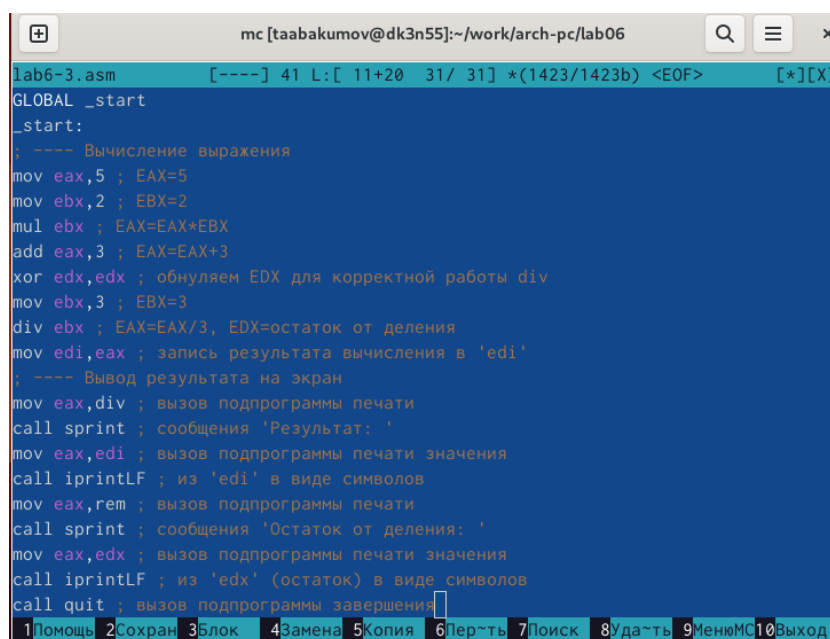
## 3.2 Выполнение арифметических операций в NASM

12. Создадим файл lab6-3.asm с помощью утилиты touch (рис. 3.12).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-3.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.12: Создание нового файла

13. Введём в созданный файл текст программы для вычисления значения выражения  $f(x) = (5 * 2 + 3)/3$  (рис. 3.13).



```
lab6-3.asm  [----] 41 L: [ 11+20  31/ 31] *(1423/1423b) <EOF>  [*][X]
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 завершения
```

Рис. 3.13: Изменение программы

Код программы из пункта 13:

```
%include 'in_out.asm' ; подключение внешнего файла
```

```
SECTION .data
```

```
div: DB 'Результат:',0
```

```
rem: DB 'Остаток от деления:',0
```

```
SECTION .text
```

```

GLOBAL _start
_start:
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'
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 завершения

```

14. Создаю исполняемый файл и запускаю его (рис. 3.14).

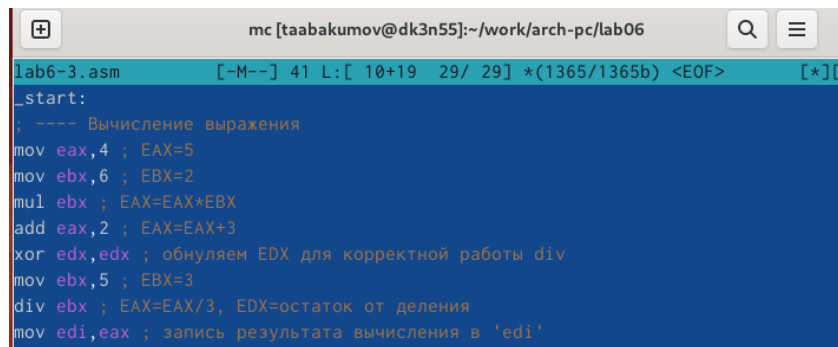
```

taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 3.14: Запуск программы

15. Изменим программу так, чтобы она вычисляла значение выражения  $f(x) = (4 * 6 + 2)/5$  (рис. 3.15).



```
lab6-3.asm [-M--] 41 L:[ 10+19 29/ 29] *(1365/1365b) <EOF> [*]  
_start:  
; ---- Вычисление выражения  
mov eax,4 ; EAX=5  
mov ebx,6 ; EBX=2  
mul ebx ; EAX=EAX*EBX  
add eax,2 ; EAX=EAX+3  
xor edx,edx ; обнуляем EDX для корректной работы div  
mov ebx,5 ; EBX=3  
div ebx ; EAX=EAX/3, EDX=остаток от деления  
mov edi,eax ; запись результата вычисления в 'edi'
```

Рис. 3.15: Изменение программы

16. Создадим и запустим новый исполняемый файл. Я посчитал для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно (рис. 3.16).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-3  
Результат: 5  
Остаток от деления: 1  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.16: Запуск программы

17. Создадим файл variant.asm с помощью утилиты touch (рис. 3.17).

```
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ touch variant.asm  
taabakumov@dk3n55 ~/work/arch-pc/lab06 $
```

Рис. 3.17: Создание файла

18. Введём в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 3.18).

```

variant.asm [----] 39 L: [ 1+15 16/ 25] *(392 / 491b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
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 sprintf
mov eax, edx
call iprintLF
call quit

```

Рис. 3.18: Изменение программы

Код программы из пункта 18:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg: DB 'Введите № студенческого билета:',0
```

```
rem: DB 'Ваш вариант:',0
```

```
SECTION .bss
```

```
x: RESB 80
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax, msg
```

```
call sprintf
```

```
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

```

19. Создаю и запускаю исполняемый файл. Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 9 (рис. 3.19).

```

taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246828
Ваш вариант: 9
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ █

```

Рис. 3.19: Запуск программы

### 3.3 Выполнение заданий для самостоятельной работы

20. Создадим файл lab6-4.asm с помощью утилиты touch (рис. 3.20).

```

taabakumov@dk3n55 ~/work/arch-pc/lab06 $ touch lab6-4.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ █

```

Рис. 3.20: Создание файла

21. Откроем созданный файл для редактирования, введём в него текст программы для вычисления значения выражения  $10+(31x-5)$ . Это выражение было под вариантом 9 (рис. 3.21).



```

lab6-4.asm [----] 37 L: [ 1+20 21/ 28] *(1368/1823b) 0010 0x00A
%include "lab6-4.inc" ; подключение внешнего файла
SECTION .data ; секция инициализированных данных
msg: DB "Введите значение переменной x: ",0
rem: DB "Результат: ",0
SECTION .bss ; секция не инициализированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный размер - 80 байт
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
mov ebx, 31 ; запись значения 31 в регистр ebx
mul ebx ; EAX=EAX*EBX = x*31
add eax, -5 ; eax = eax-5 = (31*x-5)
add eax, 10 ; eax = eax+10 = 10+(31*x)
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
call quit ; вызов подпрограммы завершения

```

Рис. 3.21: Изменение программы

22. Создаю и запускаю исполняемый файл. При вводе значения 1, вывод - 36. Проведём еще один запуск исполняемого файла для проверки работы программы с другим значением на входе. При вводе значения 3, вывод - 98. Программа отработала верно (рис. 3.22).

```

taabakumov@dk3n55 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab-6 lab-4.o
ld: невозможно найти lab-4.o: Нет такого файла или каталога
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab-6 lab-6.o
ld: невозможно найти lab-6.o: Нет такого файла или каталога
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 1
Результат: 36taabakumov@dk3n55 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 3
Результат: 98taabakumov@dk3n55 ~/work/arch-pc/lab06 $

```

Рис. 3.22: Запуск программы

## 4 Выводы

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