

# **Отчёт по лабораторной работе №7**

**дисциплина: Архитектура компьютера**

Баранова Анна Андреевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Реализация переходов в NASM . . . . .	8
4.2	Изучение структуры файла листинга . . . . .	14
4.3	Задание для самостоятельной работы . . . . .	16
<b>5</b>	<b>Выводы</b>	<b>19</b>

# List of Figures

4.1	Создание каталога для программ лабораторной работы № 7 и создание в нём файла lab7-1.asm . . . . .	8
4.2	Создание каталога для программ лабораторной работы № 7 и создание в нём файла lab7-1.asm . . . . .	8
4.3	Изменения в файле lab7-1.asm . . . . .	9
4.4	Копирование файла in_out.asm в каталог с файлом lab7-1.asm с помощью функциональной клавиши F5 . . . . .	9
4.5	Копирование файла in_out.asm в каталог с файлом lab7-1.asm с помощью функциональной клавиши F5 . . . . .	10
4.6	Создание исполняемого файла и его запуск . . . . .	10
4.7	Создание исполняемого файла и его запуск . . . . .	11
4.8	Изменения в файле lab7-1.asm . . . . .	11
4.9	Создание исполняемого файла и его запуск . . . . .	12
4.10	Изменения в файле lab7-1.asm . . . . .	12
4.11	Создание исполняемого файла и его запуск . . . . .	12
4.12	Создание файла lab7-2.asm . . . . .	12
4.13	Создание файла lab7-2.asm . . . . .	13
4.14	Изменения в файле lab7-2.asm . . . . .	13
4.15	Создание исполняемого файла и его запуск . . . . .	14
4.16	Создание исполняемого файла и его запуск . . . . .	14
4.17	Создание исполняемого файла и его запуск . . . . .	14
4.18	Создание исполняемого файла и его запуск . . . . .	14
4.19	Создание файла листинга для программы из файла lab7-2.asm . . . . .	15
4.20	Открытие файла листинга lab7-2.lst с помощью mcedit . . . . .	15
4.21	Открытие файла листинга lab7-2.lst с помощью mcedit . . . . .	15
4.22	Строка 34 . . . . .	15
4.23	Строка 33 . . . . .	16
4.24	Строка 32 . . . . .	16
4.25	Изменения в файле lab7-2.asm . . . . .	16
4.26	Написание программы нахождения наименьшей из 3 целочисленных переменных a, b, c . . . . .	17
4.27	Создание исполняемого файла и проверка его работы . . . . .	17
4.28	Написание программы, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений . . . . .	18
4.29	Создание исполняемого файла и проверка его работы . . . . .	18

4.30 Создание исполняемого файла и проверка его работы . . . . .	18
--	----

# 1 Цель работы

Изучить команды условного и безусловного переходов. Приобрести навыки написания программ с использованием переходов. Познакомиться с назначением и структурой файла листинга.

## 2 Задание

В ходе выполнения данной лабораторной работы необходимо изучить:

- Команды условного и безусловного переходов;
- файл листинга и его структуру.

Выполнив эту работу, мы изучим команды условного и безусловного переходов, приобретём навыки написания программ с использованием переходов и познакомимся с назначением и структурой файла листинга.

### 3 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

- условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.
- безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

# 4 Выполнение лабораторной работы

## 4.1 Реализация переходов в NASM

Создадим каталог для программ лабораторной работы № 7, перейдём в него и создадим файл lab7-1.asm (рис. 4.1), (рис. 4.2).

```
aabaranova@dk3n55 ~ $ mkdir ~/work/arch-pc/lab07
aabaranova@dk3n55 ~ $ cd ~/work/arch-pc/lab07
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ touch lab7-1.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $
```

Рис. 4.1: Создание каталога для программ лабораторной работы № 7 и создание в нём файла lab7-1.asm

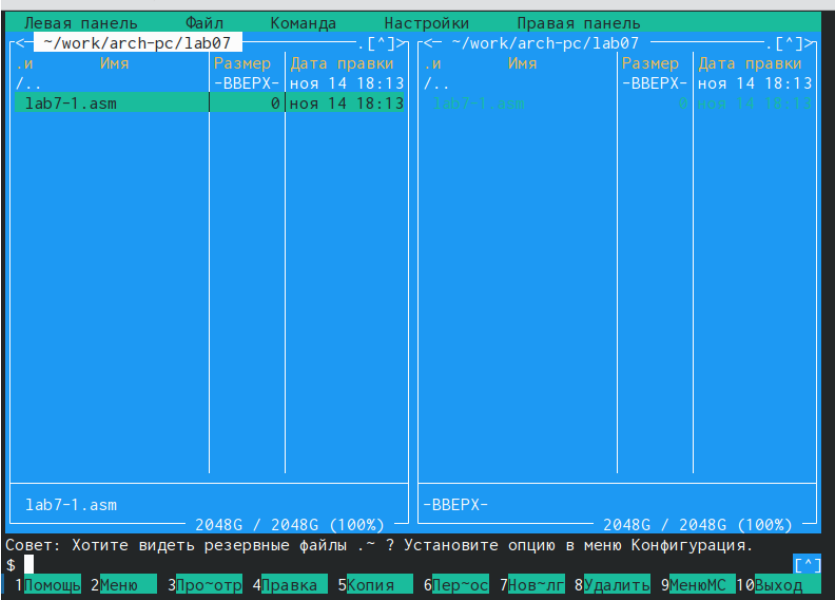


Рис. 4.2: Создание каталога для программ лабораторной работы № 7 и создание в нём файла lab7-1.asm



Введём в файл lab7-1.asm текст программы (рис. 4.3).

```
lab7-1.asm      [-M--] 3 L:[ 1+17 18/ 26] *(376 / 682b) 0097 0x061  [*][X]
%include "in_out.asm" ; подключение внешнего файла

SECTION data
msg1: DB "Сообщение № 1",0
msg2: DB "Сообщение № 2",0
msg3: DB "Сообщение № 3",0

SECTION text
GLOBAL _start
_start:

jmp _label2

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; "Сообщение № 1"

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; "Сообщение № 2"

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; "Сообщение № 3"

_end:
call quit ; вызов подпрограммы завершения

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переть 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 4.3: Изменения в файле lab7-1.asm

Перед созданием исполняемого файла создадим копию файла in\_out.asm в каталоге ~/work/arch-pc/lab07. (рис. 4.4), (рис. 4.5).

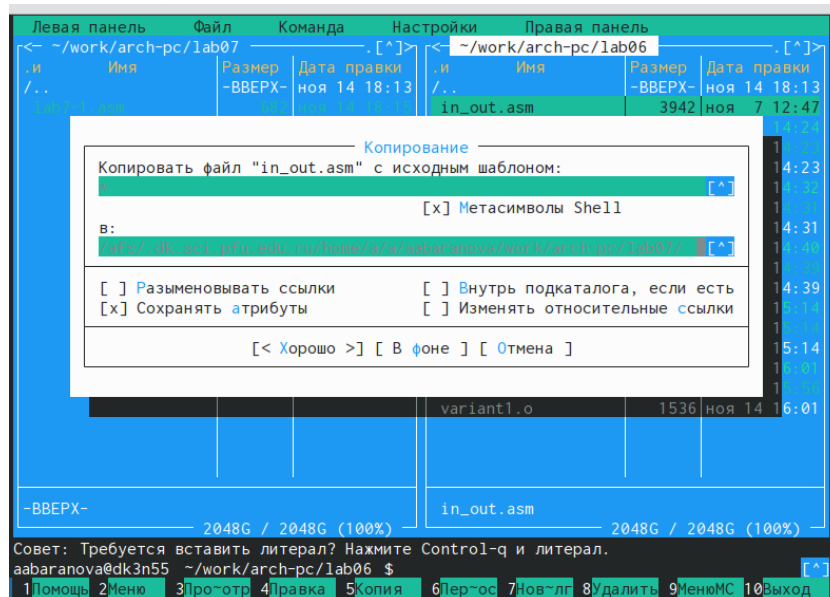


Рис. 4.4: Копирование файла in\_out.asm в каталог с файлом lab7-1.asm с помощью функциональной клавиши F5

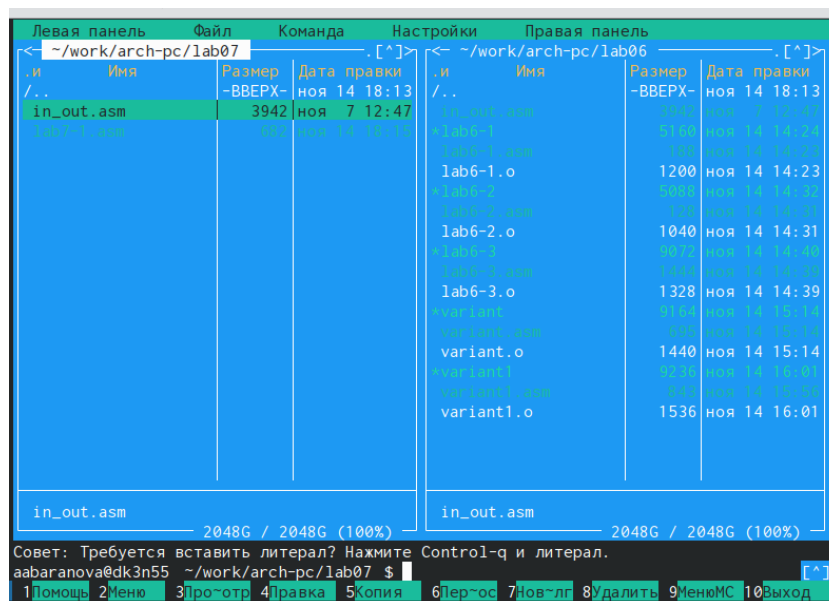


Рис. 4.5: Копирование файла in\_out.asm в каталог с файлом lab7-1.asm с помощью функциональной клавиши F5

Создадим исполняемый файл и запустим его (рис. 4.6), (рис. 4.7).

```
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
aabaranova@dk3n55 ~/work/arch-pc/lab07 $
```

Рис. 4.6: Создание исполняемого файла и его запуск

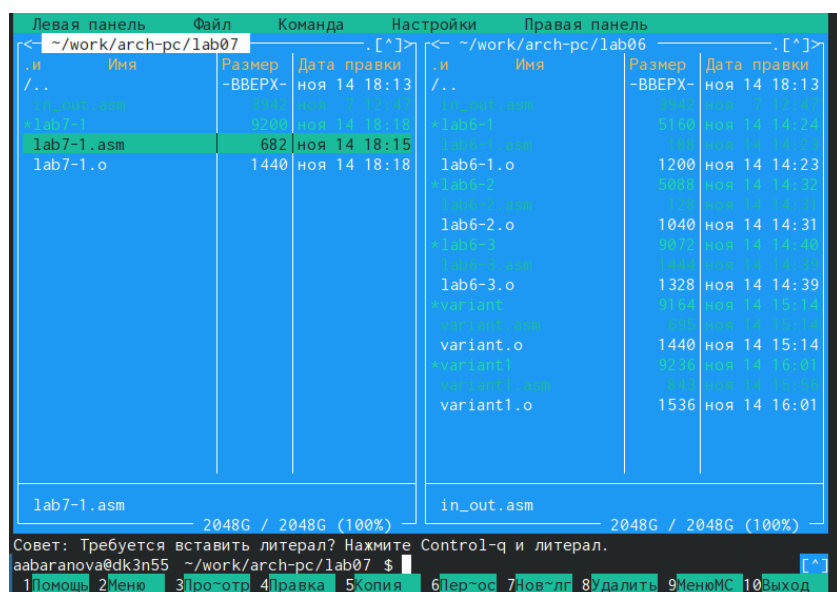


Рис. 4.7: Создание исполняемого файла и его запуск

Изменим текст программы файла lab7-1.asm (рис. 4.8).

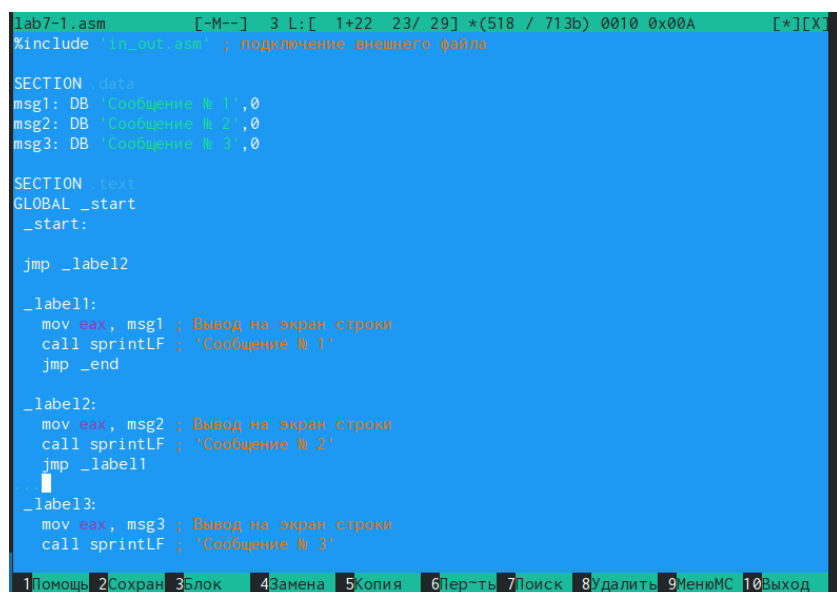


Рис. 4.8: Изменения в файле lab7-1.asm

Создадим исполняемый файл и запустим его (рис. 4.9).

```

aabaranova@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
aabaranova@dk3n55 ~/work/arch-pc/lab07 $

```

Рис. 4.9: Создание исполняемого файла и его запуск

Снова изменим текст программы файла lab7-1.asm, создадим исполняемый файл и запустим его (рис. 4.10), (рис. 4.11).

```

lab7-1.asm  [-M--] 14 L: [ 3+19 22/ 30] *(514 / 728b) 0010 0x00A  [*][X]
SECTION .data
msg1: DB "Сообщение № 1",0
msg2: DB "Сообщение № 2",0
msg3: DB "Сообщение № 3",0

SECTION .text
GLOBAL _start
_start:

jmp _label3

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; "Сообщение № 1"
jmp _end

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; "Сообщение № 2"
jmp _label1

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; "Сообщение № 3"
jmp _label2

_end:
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 4.10: Изменения в файле lab7-1.asm

```

aabaranova@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
aabaranova@dk3n55 ~/work/arch-pc/lab07 $

```

Рис. 4.11: Создание исполняемого файла и его запуск

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. 4.12), (рис. 4.13).

```

aabaranova@dk3n55 ~/work/arch-pc/lab07 $ touch lab7-2.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $

```

Рис. 4.12: Создание файла lab7-2.asm

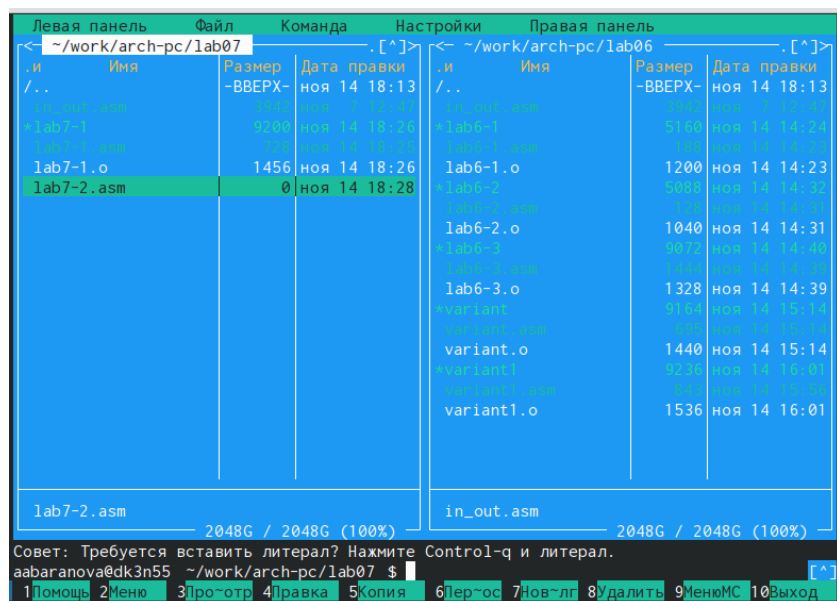


Рис. 4.13: Создание файла lab7-2.asm

Введём в файл lab7-2.asm текст программы (рис. 4.14).

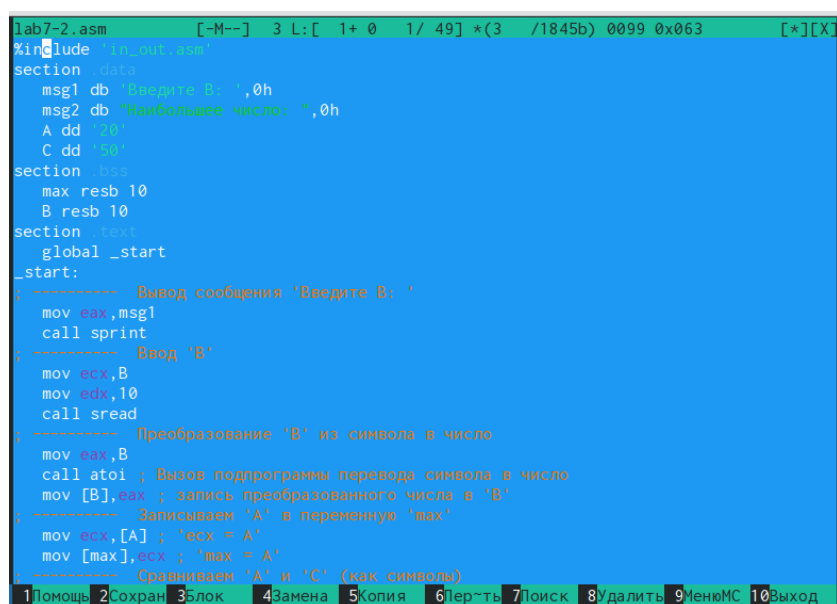


Рис. 4.14: Изменения в файле lab7-2.asm

Создадим исполняемый файл и запустим его и проверим его работу для разных значений В (рис. 4.15), (рис. 4.16), (рис. 4.17), (рис. 4.18).

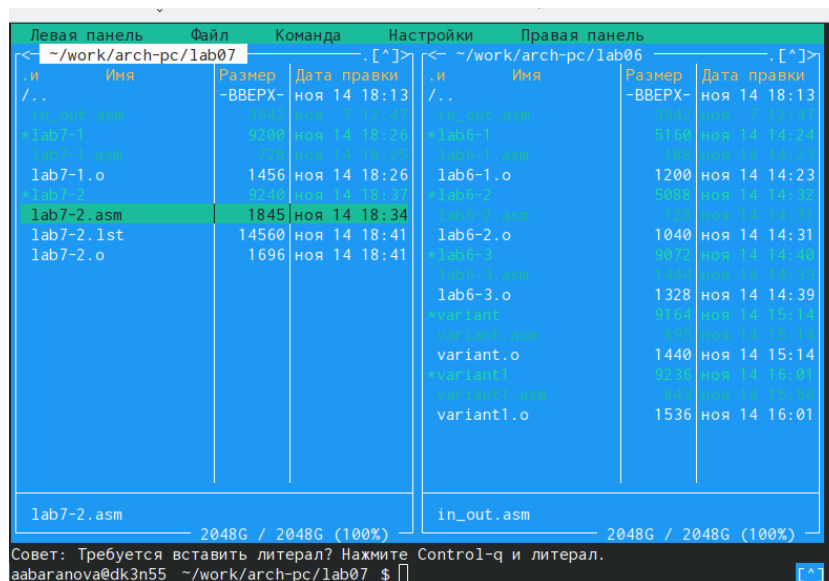


Рис. 4.15: Создание исполняемого файла и его запуск

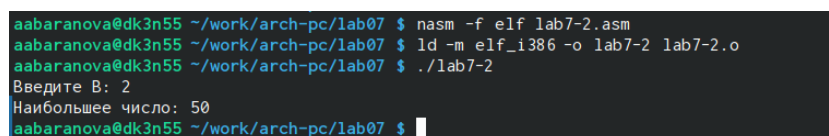


Рис. 4.16: Создание исполняемого файла и его запуск

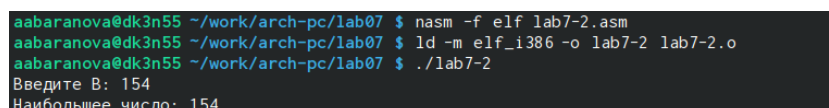


Рис. 4.17: Создание исполняемого файла и его запуск

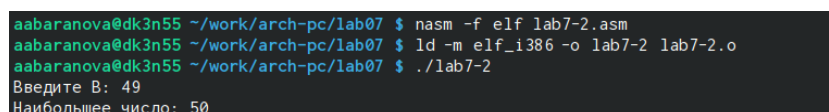


Рис. 4.18: Создание исполняемого файла и его запуск

## 4.2 Изучение структуры файла листинга

Создадим файл листинга для программы из файла lab7-2.asm (рис. 4.19).

```
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
aabaranova@dk3n55 ~/work/arch-pc/lab07 $
```

Рис. 4.19: Создание файла листинга для программы из файла lab7-2.asm

Откроем файл листинга lab7-2.lst с помощью текстового редактора mcedit (рис. 4.20), (рис. 4.21).

```
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ mcedit lab7-2.lst
```

Рис. 4.20: Открытие файла листинга lab7-2.lst с помощью mcedit

```
lab7-2.lst [----] 0 L: [ 1+ 0 1/225] *(0 /14560b) 0032 0x020 [*][X]
1                                     %include 'in_out.asm'
2                                     <1> ;----- slen -----
3                                     <1> ; Функция вычисления длины сообщения
4                                     <1> slen:
5 00000000 53                         <1> push    ebx
6 00000001 89C3                       <1> mov     ebx, eax
7                                     <1>
8                                     <1> nextchar:
9 00000003 803800                     <1> cmp     byte [eax], 0
10 00000006 7403                      <1> jz      finished
11 00000008 40                        <1> inc     eax
12 00000009 EBF8                      <1> jmp     nextchar
13                                     <1>
14                                     <1> finished:
15 0000000B 29D8                      <1> sub     eax, ebx
16 0000000D 5B                        <1> pop     ebx
17 0000000E C3                        <1> ret
18                                     <1>
19                                     <1> ;----- sprint -----
20                                     <1> ; Функция печати сообщения
21                                     <1> ; входные данные: mov eax,<message>
22                                     <1> sprint:
23 0000000F 52                         <1> push    edx
24 00000010 51                         <1> push    ecx
25 00000011 53                         <1> push    ebx
26 00000012 50                         <1> push    eax
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 4.21: Открытие файла листинга lab7-2.lst с помощью mcedit

Эта строка находится на 34 месте, ее адрес “00000022”, машинный код - ‘B804000000’, а ‘mov eax,4’ - исходный текст программы, означающий что в регистр eax запишется 4 (рис. 4.22).

```
34 00000022 B804000000 <1> mov eax, 4
```

Рис. 4.22: Строка 34

Эта строка находится на 33 месте, ее адрес “0000001D”, машинный код -

‘BB01000000’, а ‘mov ebx,1’ - исходный текст программы, означающий что в регистр ebx запишется 1 (рис. 4.23).

```
33 0000001D BB01000000 <1> mov ebx, 1
```

Рис. 4.23: Строка 33

Эта строка находится на 32 месте, ее адрес “0000001B”, машинный код - ‘89C1’, а ‘mov ecx,ecx’ - исходный текст программы, означающий что в регистр ecx запишется значение ecx (рис. 4.24).

```
32 0000001B 89C1 <1> mov ecx, ecx
```

Рис. 4.24: Строка 32

Откроем файл с программой lab7-2.asm и в строке ‘mov ecx,[min]’ удалим ‘[min]’, выполним трансляцию с получением файла листинга (рис. 4.25).

```
aabaranova@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:67: error: invalid combination of opcode and operands
aabaranova@dk3n55 ~/work/arch-pc/lab07 $
```

Рис. 4.25: Изменения в файле lab7-2.asm

## 4.3 Задание для самостоятельной работы

1. Напишем программу нахождения наименьшей из 3 целочисленных переменных a, b, c. Создадим исполняемый файл и проверим его работу (рис. 4.26), (рис. 4.27).



```

lab7-2.asm      [-M-0]  0 L:[ 1+ 0  1/ 79] *(0  /2445b) 0037 0x025  [*][X]
#include "tut1.asm"
section .data
    msg1 db "Введите A: ",0h
    msg2 db "Введите B: ",0h
    msg3 db "Введите C: ",0h
    msg4 db "Наименьшее число: ",0h

section .bss
    min resb 10
    A resb 10
    B resb 10
    C resb 10

section .text
    global _start
_start:
; ----- Вывод сообщения 'Введите A: '
    mov eax,msg1
    call sprint
; ----- Ввод 'A'
    mov ecx,A
    mov edx,10
    call sread
; ----- Преобразование 'A' из символа в число
    mov eax,A
    call atoi ; Вызов подпрограммы перевода символа в число
    mov [A],eax
; ----- Вывод сообщения 'Введите B: '
    mov eax,msg2
    call sprint

```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 4.26: Написание программы нахождения наименьшей из 3 целочисленных переменных a, b, c

```

aabaranova@dk5n60 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ./lab7-2
Введите A: 99
Введите B: 29
Введите C: 26
Наименьшее число: 26
aabaranova@dk5n60 ~/work/arch-pc/lab07 $

```

Рис. 4.27: Создание исполняемого файла и проверка его работы

2. Напишем программу, которая для введенных с клавиатуры значений  $x$  и  $a$  вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений. Создадим исполняемый файл и проверим его работу для значений  $x$  и  $a$  (рис. 4.28), (рис. 4.29), (рис. 4.30).

```

lab7-3.asm [M--] 13 L: [ 1+ 9 10/ 60] *(260 /1442b) 0010 0x00A [*][X]
#include "ft_printf.asm" ; подключение внешнего файла
section .data
    msg1 db "Введите A: ",0h
    msg2 db "Введите X: ",0h
    msg3 db "Результат: ",0h

section .bss
    fin resb 10
    A resb 10
    X resb 10

section .text
    global _start

_start:
; ----- Вывод сообщения 'Введите A: '
    mov eax,msg1
    call sprintf
; ----- Ввод 'A'
    mov ecx,A
    mov edx,10
    call sread
; ----- Преобразование 'A' из символа в число
    mov eax,A
    call atoi ; Вызов подпрограммы перевода символа в число
    mov [A],eax
; ----- Вывод сообщения 'Введите X: '
    mov eax,msg2
    call sprintf
; ----- Ввод 'X'

```

Рис. 4.28: Написание программы, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции  $f(x)$  и выводит результат вычислений

```

aabaranova@dk5n60 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ./lab7-3
Введите A: 7
Введите X: 3
Результат: 21

```

Рис. 4.29: Создание исполняемого файла и проверка его работы

```

aabaranova@dk5n60 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
aabaranova@dk5n60 ~/work/arch-pc/lab07 $ ./lab7-3
Введите A: 4
Введите X: 6
Результат: 1

```

Рис. 4.30: Создание исполняемого файла и проверка его работы

## **5 Выводы**

В ходе выполнения данной лабораторной работы были изучены команды условного и безусловного переходов, приобретены навыки написания программ с использованием переходов и познакомились с назначением и структурой файла листинга.