

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина:     Архитектура компьютера     -

Студент: Трофимов Владислав Алексеевич

Группа: НКАбд-06-25

МОСКВА

2025 г.

# Содержание

1 Цель работы .....	3
2 Задание .....	4
3 Теоретическое введение .....	5
4 Выполнение лабораторной работы.....	6
4.1 Реализация переходов в NASM .....	7
4.2 Изучение структуры файла листинга.....	8
4.3 Задания для самостоятельной работы.....	9
5 Выводы.....	10
Список литературы .....	10

# Список иллюстраций

- Рис. 4.1: Создание каталога и файла для программы
- Рис. 4.2: Сохранение программы
- Рис. 4.3: Запуск программы
- Рис. 4.4: Изменение программы
- Рис. 4.5: Запуск измененной программы
- Рис. 4.6: Изменение программы
- Рис. 4.7: Проверка изменений
- Рис. 4.8: Сохранение новой программы
- Рис. 4.9: Проверка программы из листинга
- Рис. 4.10: Проверка файла листинга
- Рис. 4.11: Удаление операнда из программы
- Рис. 4.12: Просмотр ошибки в файле листинга
- Рис. 4.13: Первая программа самостоятельной работы
- Рис. 4.14: Проверка работы первой программы
- Рис. 4.15: Вторая программа самостоятельной работы
- ис. 4.16: Проверка работы второй программы

# 1 Цель работы

Изучение команд условного и безусловного переходов.  
Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

## **2 Задание**

1. Реализация переходов в NASM
2. Изучение структуры файлов листинга
3. Самостоятельное написание программ по материалам лабораторной работы

### 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.

## 4 Выполнение лабораторной работы

### 4.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7 (рис. 4.1)

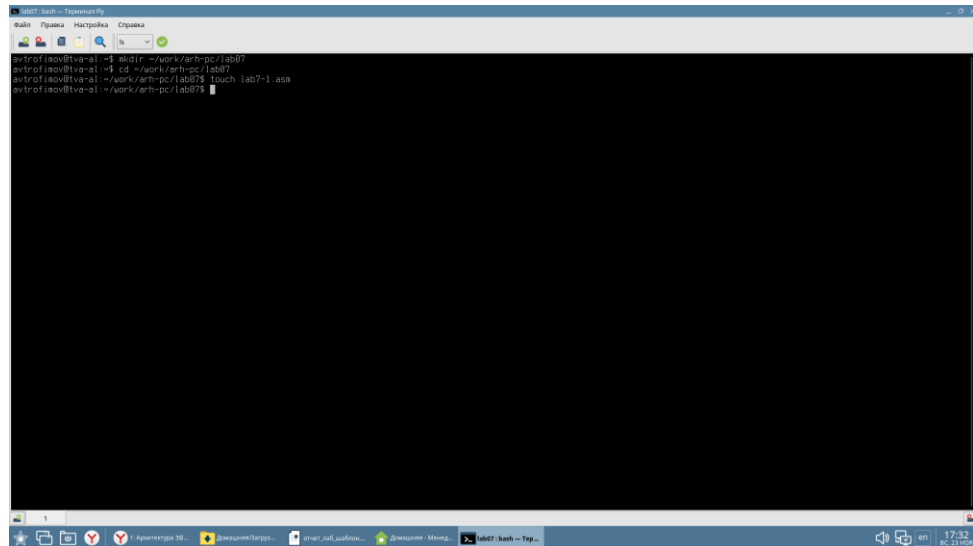


Рис. 4.1: Создание каталога и файла для программы

Копирую код из листинга в файл будущей программы. (рис. 4.2).

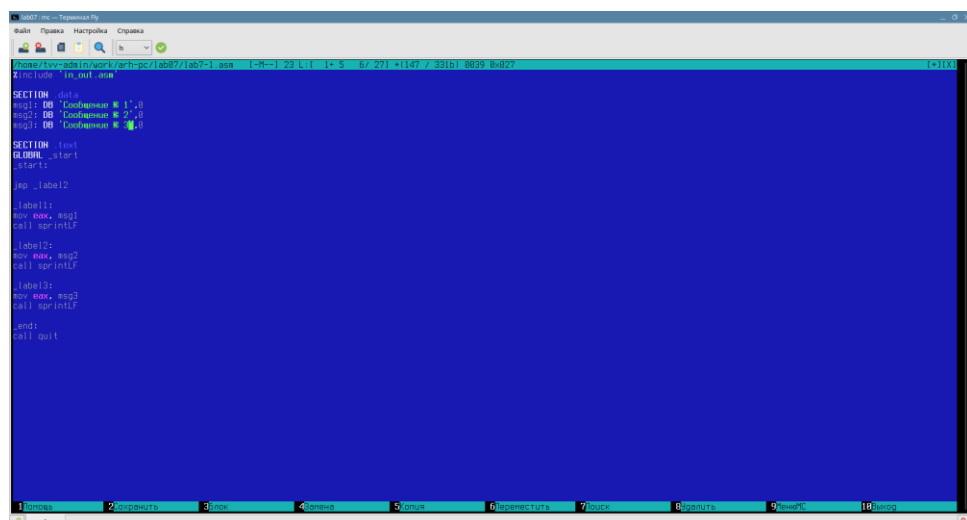


Рис. 4.2: Сохранение программы

При запуске программы я убедился в том, что безусловный переход действительно изменяет порядок выполнения инструкций (рис. 4.3).

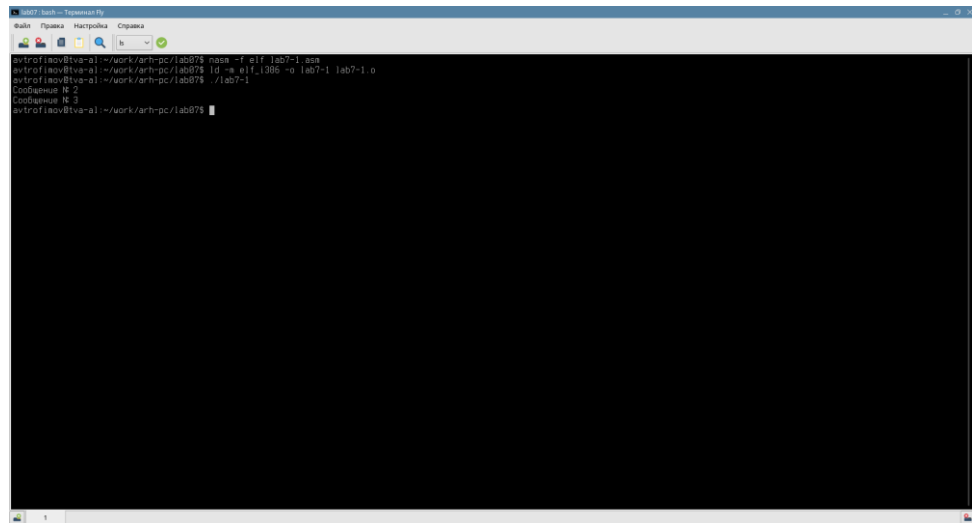


Рис. 4.3: Запуск программы

Изменяю программу таким образом, чтобы поменялся порядок выполнения функций (рис. 4.4).

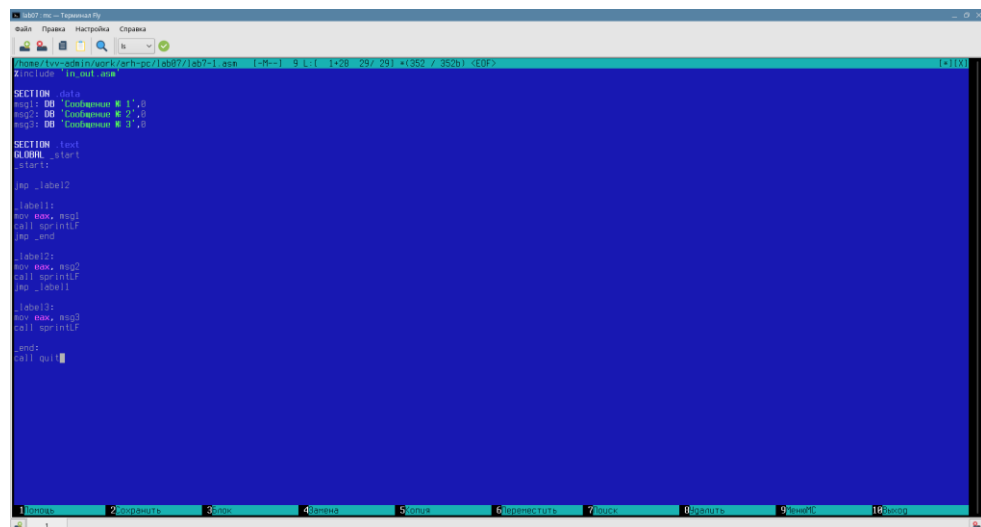


Рис. 4.4: Изменение программы

Запускаю программу и проверяю, что примененные изменения верны (рис. 4.5).



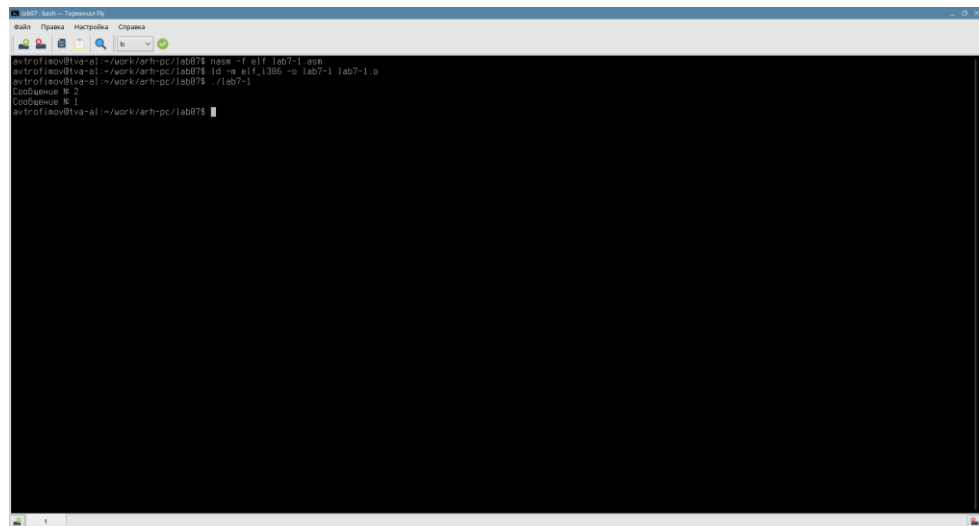


Рис. 4.5: Запуск измененной программы

Теперь изменяю текст программы так, чтобы все три сообщения вывелись в обратном порядке (рис. 4.6).

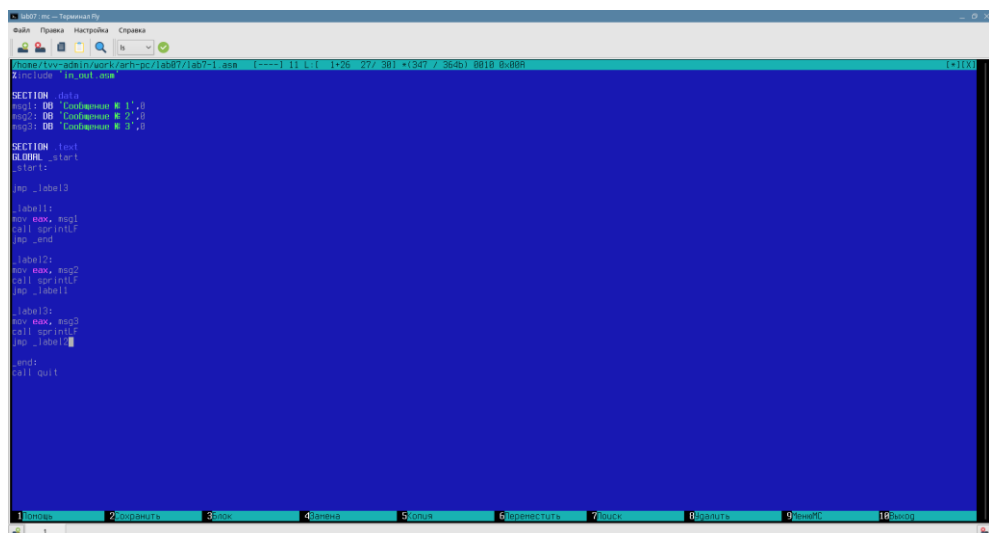


Рис. 4.6: Изменение программы

Работа выполнена корректно, программа в нужном мне порядке выводит сообщения (рис. 4.7).

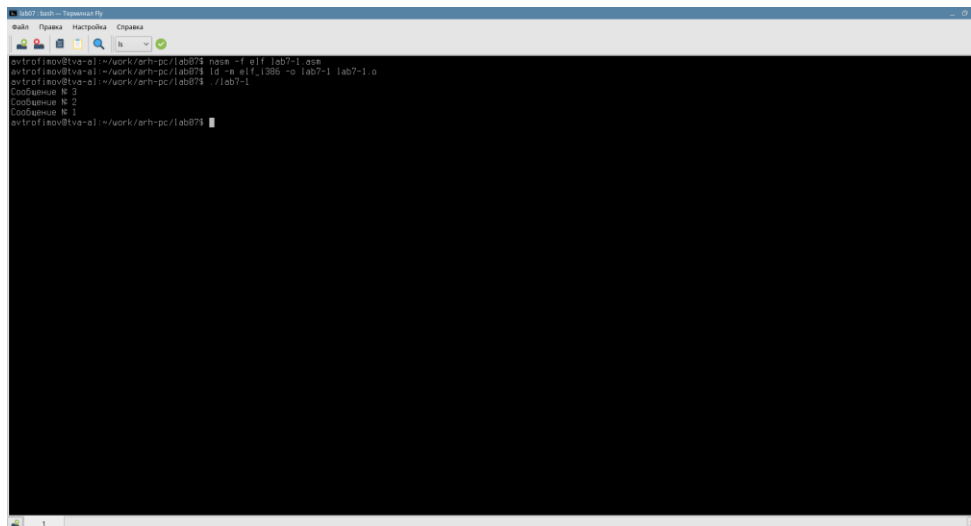


Рис. 4.7: Проверка изменений

Создаю новый рабочий файл и вставляю в него код из следующего листинга (рис. 4.8).

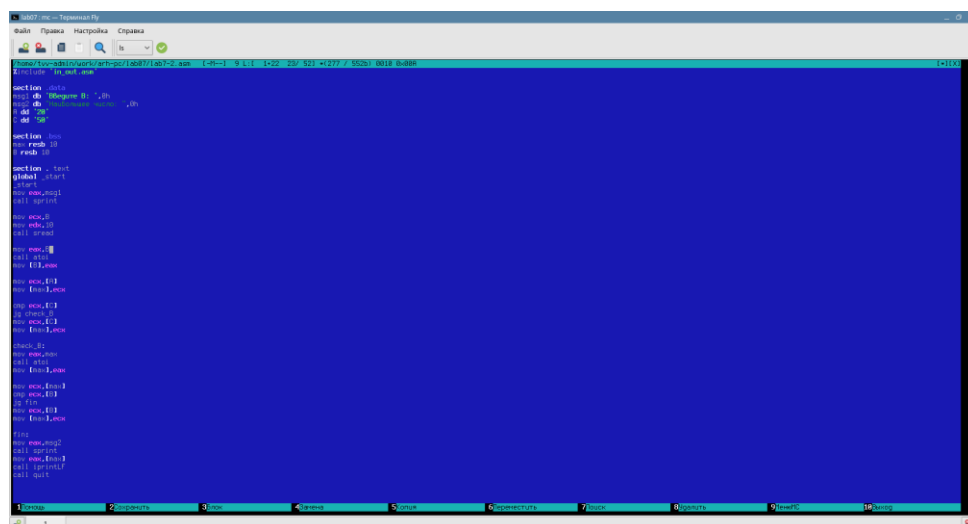


Рис. 4.8: Сохранение новой программы

Программа выводит значение переменной с максимальным значением, проверяю работу программы с разными входными данными (рис. 4.9).

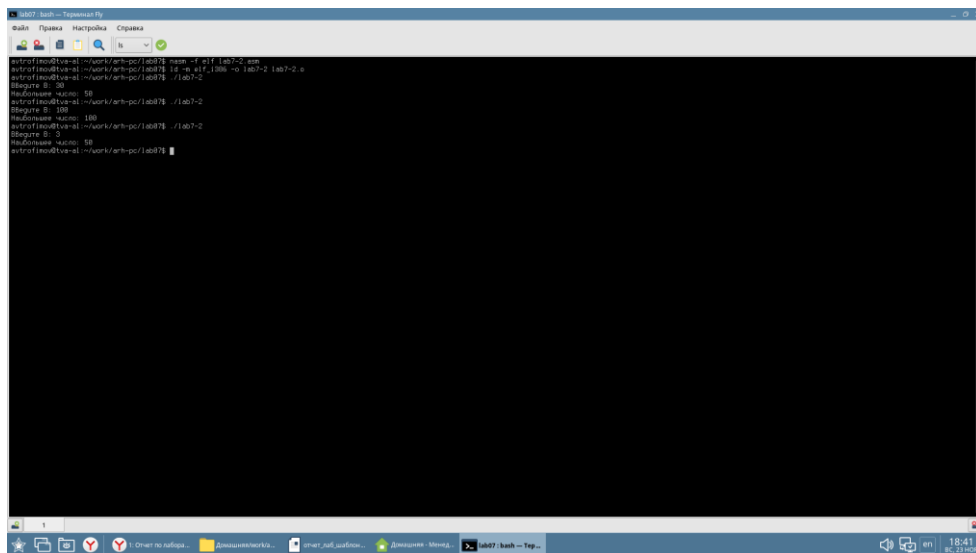


Рис. 4.9: Проверка программы из листинга

## 4.2 Изучение структуры файла листинга

Создаю файл листинга с помощью флага `-l` команды `nasm` и открываю его с помощью текстового редактора `msedit` (рис. 4.10).

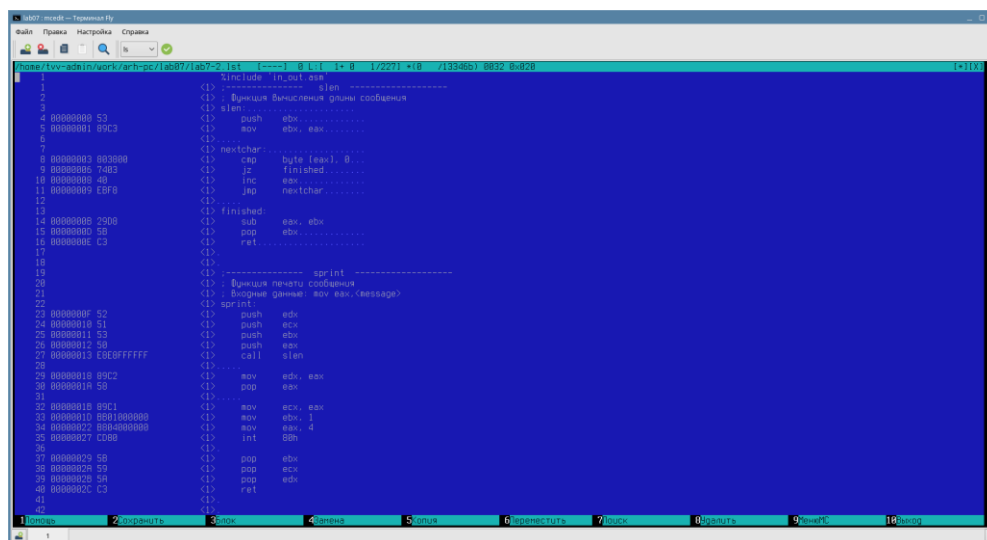
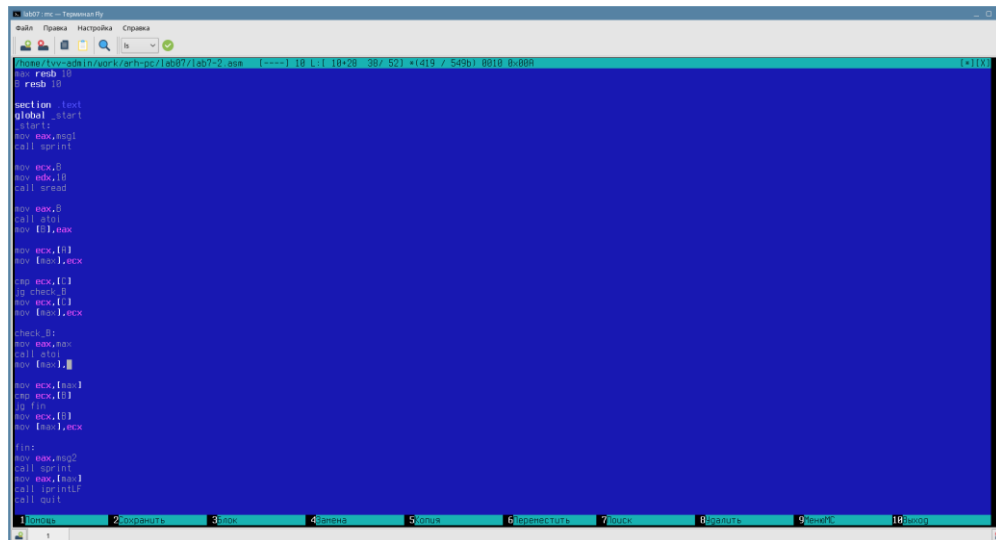


Рис. 4.10: Проверка файла листинга

Первое значение в файле листинга - номер строки, и он может вовсе не совпадать с номером строки изначального

файла. Второе вхождение - адрес, смещение машинного кода относительно начала текущего сегмента, затем непосредственно идет сам машинный код, а заключает строку исходный текст программы с комментариями.

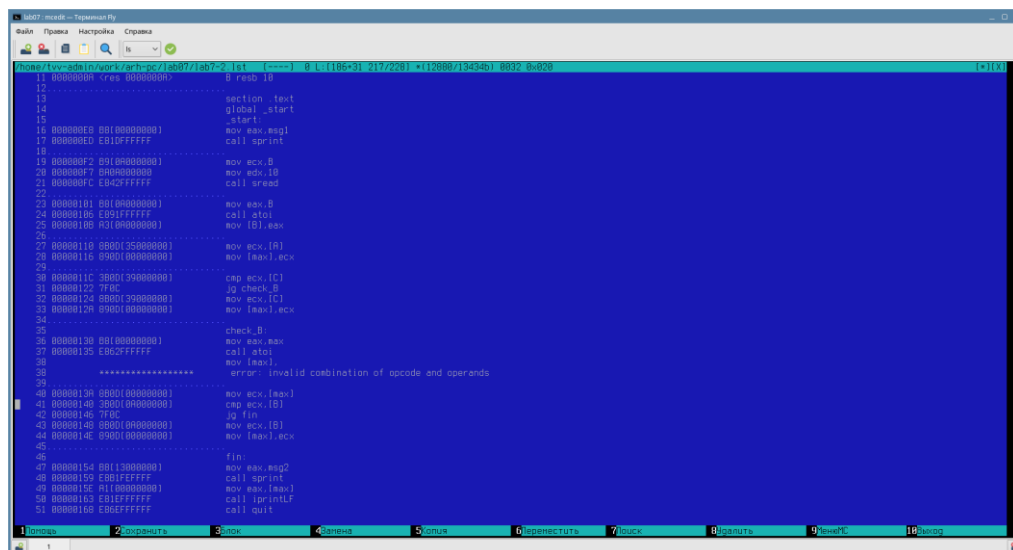
Удаляю один операнд из случайной инструкции, чтобы проверить поведение файла листинга в дальнейшем (рис. 4.11).



```
home/uv-admin/work/arb-pc/ab07/ab7-2.asm [----] 0 L:186*31 217/2281 +12888/13434b) 8832 8x828
mov ebx, 10
mov ebx, 10
section .text
global _start
_start:
mov eax, esp
call sprint
mov ecx, 8
mov ebx, 10
call read
mov ecx, 8
call atoi
mov [0], eax
mov ecx, [0]
mov [eax], ecx
cpl ecx, [0]
jg check_0
mov ecx, [0]
mov [eax], ecx
check_0:
mov eax, eax
call atoi
mov [eax], [0]
mov ecx, [eax]
cpl ecx, [0]
jg fin
mov ecx, [0]
mov [eax], ecx
fin:
mov eax, esp
call sprint
mov [eax], [eax]
call printf
call quit
```

Рис. 4.11: Удаление операнда из программы

В новом файле листинга показывает ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются. (рис. 4.12).



```
home/uv-admin/work/arb-pc/ab07/ab7-2.lst [----] 0 L:186*31 217/2281 +12888/13434b) 8832 8x828
1 00000000 0x00000000 0 read 10
12
13
14 section .text
15 global _start
16 _start:
17 mov eax, esp
18 call sprint
19
20 mov ecx, 8
21 mov ebx, 10
22 call read
23
24 mov ecx, 8
25 call atoi
26 mov [0], eax
27
28 mov ecx, [0]
29 mov [eax], ecx
30
31 cpl ecx, [0]
32 jg check_0
33 mov ecx, [0]
34 mov [eax], ecx
35
36 check_0:
37 mov eax, eax
38 call atoi
39 mov [eax], [0]
40
41 mov ecx, [eax]
42 cpl ecx, [0]
43 jg fin
44 mov ecx, [0]
45 mov [eax], ecx
46
47 fin:
48 mov eax, esp
49 call sprint
50 mov [eax], [eax]
51 call printf
52 call quit
53
54 error: invalid combination of opcode and operands
```

Рис. 4.12: Просмотр ошибки в файле листинга

### 4.3 Задания для самостоятельной работы

Так как я не получал никакого варианта во время выполнения седьмой лабораторной работы, то буду использовать свой вариант из предыдущей лабораторной работы. Возвращаю операнд к функции в программе и изменяю ее так, чтобы она выводила переменную с наименьшим значением (рис. 4.13).

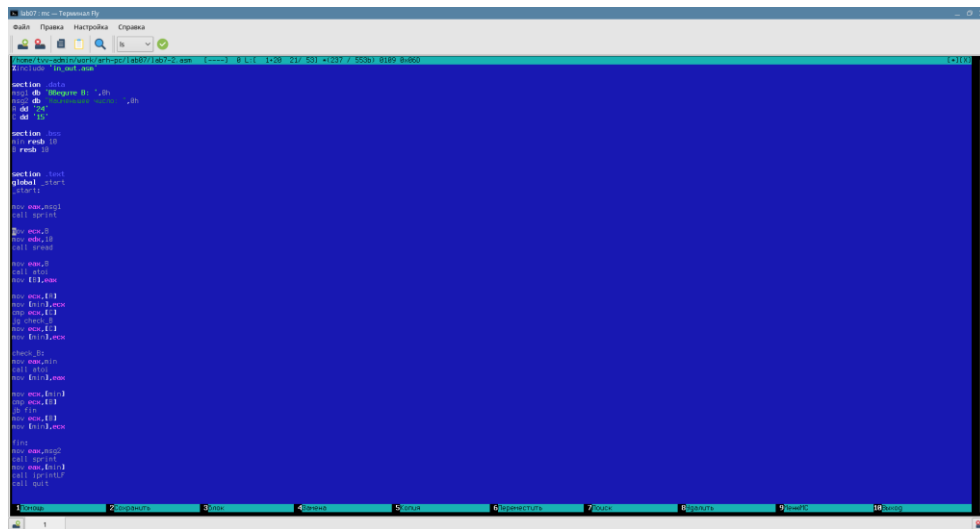


Рис. 4.13: Первая программа самостоятельной работы

Проверяю корректность написания первой программы (рис. 4.14).

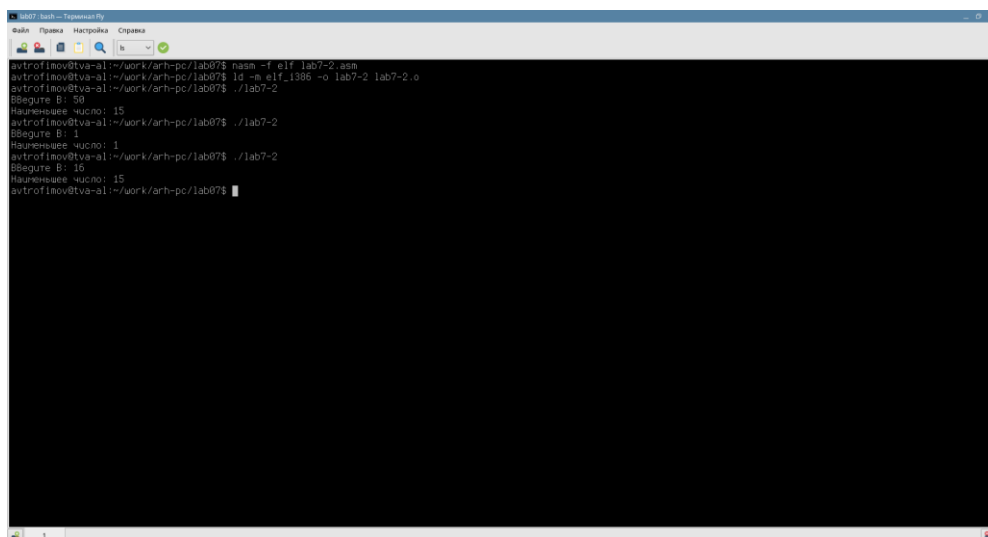


Рис. 4.14: Проверка работы первой программы

Пишу программу, которая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатуры переменных  $a$  и  $x$  (рис. 4.15).

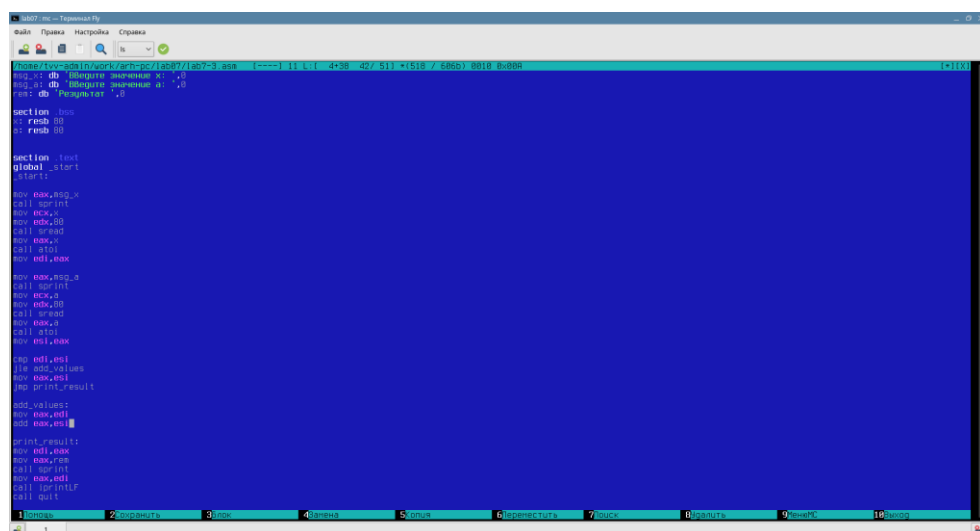
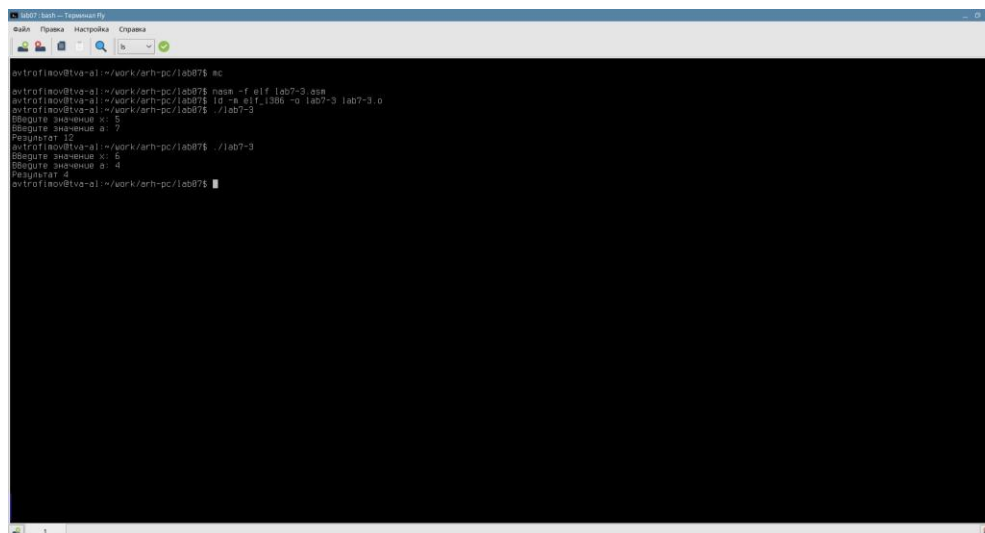


Рис. 4.15: Вторая программа самостоятельной работы

Транслирую и компоную файл, запускаю и проверяю работу программы для различных значений  $a$  и  $x$  (рис. 4.16).



```
avtrofiarov@va-el:~/work/erh-pc/lab875$ nc
avtrofiarov@va-el:~/work/erh-pc/lab875$ nasm -f elf lab7-3.asm
avtrofiarov@va-el:~/work/erh-pc/lab875$ ld -r elf_1386 -o lab7-3 lab7-3.o
avtrofiarov@va-el:~/work/erh-pc/lab875$ ./lab7-3
ISegute znachenie x: 5
ISegute znachenie a: 7
Результат 12
avtrofiarov@va-el:~/work/erh-pc/lab875$ ./lab7-3
ISegute znachenie x: 5
ISegute znachenie a: 4
Результат 4
avtrofiarov@va-el:~/work/erh-pc/lab875$
```

Рис. 4.16: Проверка работы второй программы

## 5 Выводы

При выполнении лабораторной работы я изучил команды условных и безусловных переходов, а также приобрел навыки написания программ с использованием переходов, познакомился с назначением и структурой файлов листинга.

## Список литературы

[1. Курс ТУИС](#)

[2. Лабораторная работа №7](#)

3. Программирование на языке ассемблера NASM Столяров А.  
В.