

**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 измененой 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)

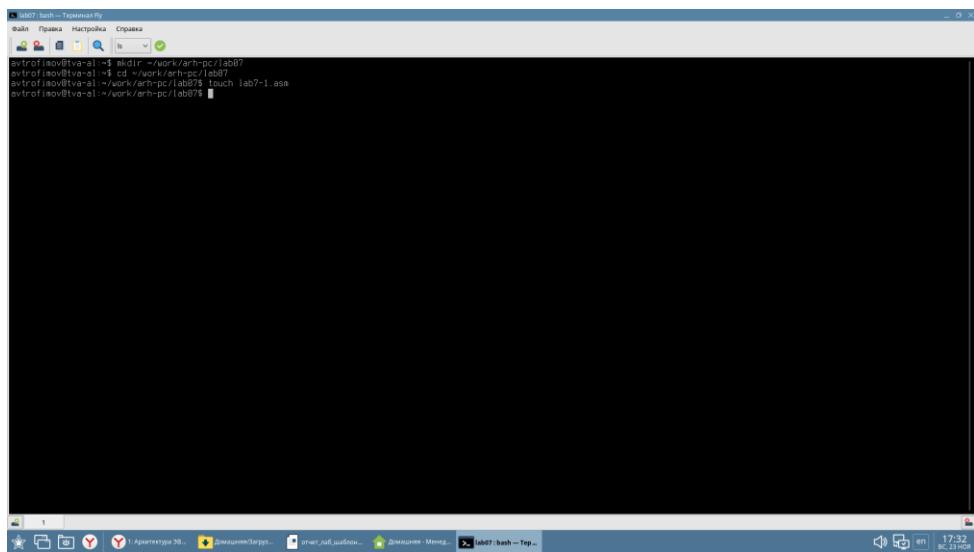


Рис. 4.1: Создание каталога и файла для программы

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

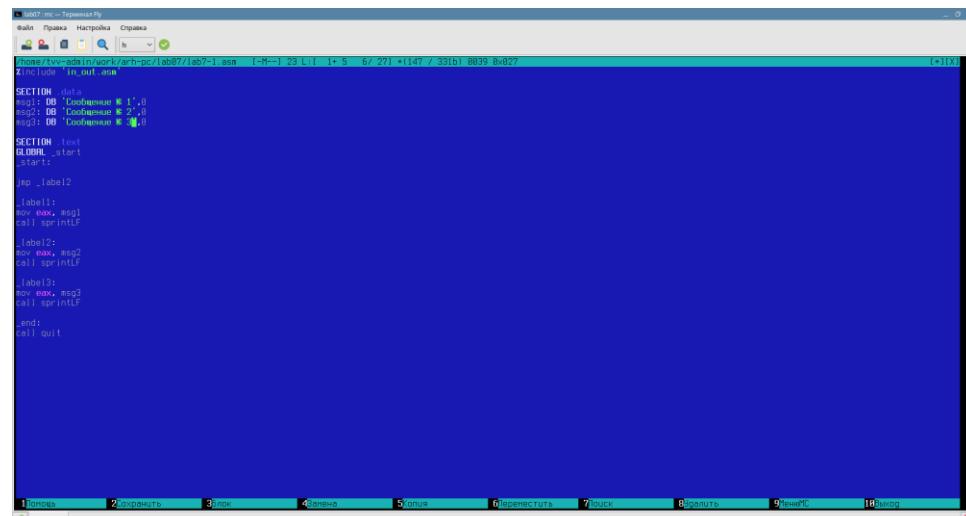


Рис. 4.2: Сохранение программы

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

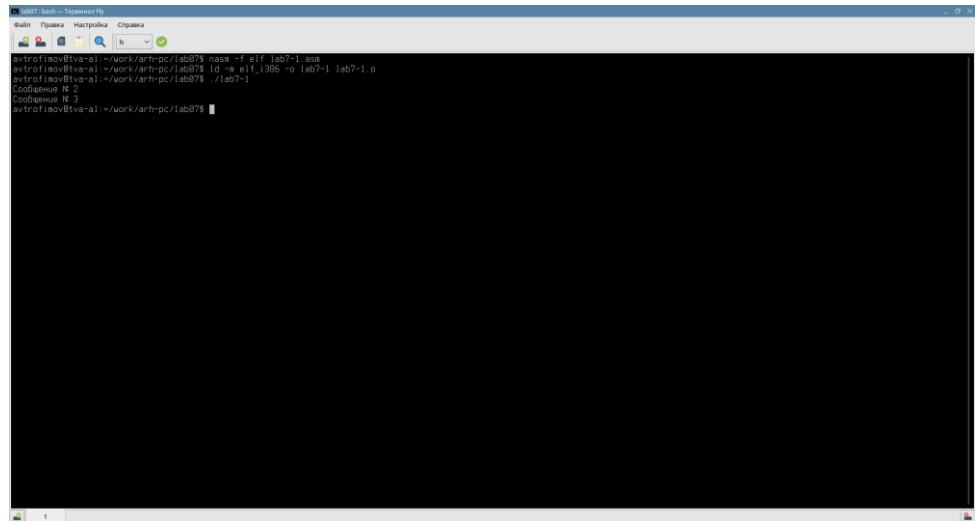
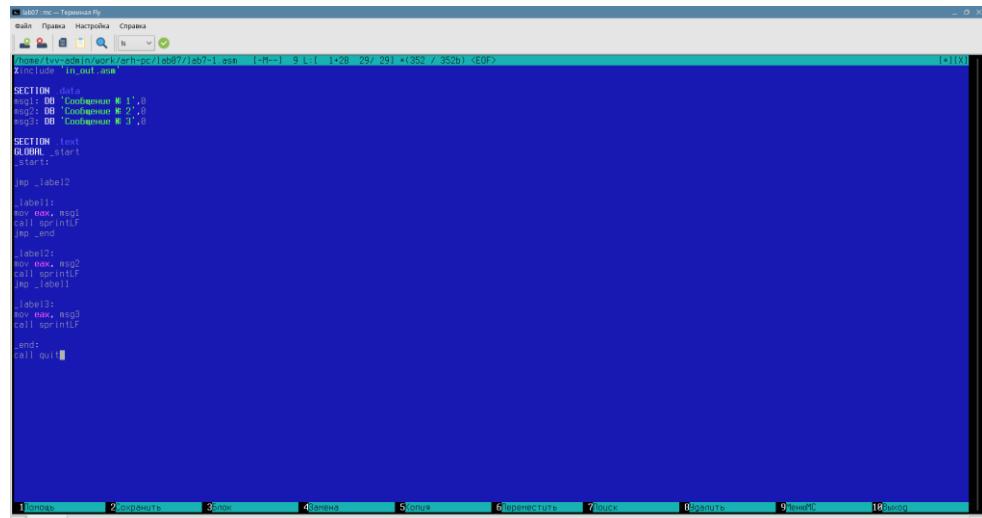


Рис. 4.3: Запуск программы

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



```
SECTION data
msg1: 00 Сообщение № 1
msg2: 00 Сообщение № 2
msg3: 00 Сообщение № 3
SECTION text
GLOBAL _start
_start:
    jmp _label1
_label1:
    mov ebx, msg1
    call sprintLF
    jmp _label2
_label2:
    mov ebx, msg2
    call sprintLF
    jmp _label3
_label3:
    mov ebx, msg3
    call sprintLF
_end:
    call exit
```

Рис. 4.4: Изменение программы

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

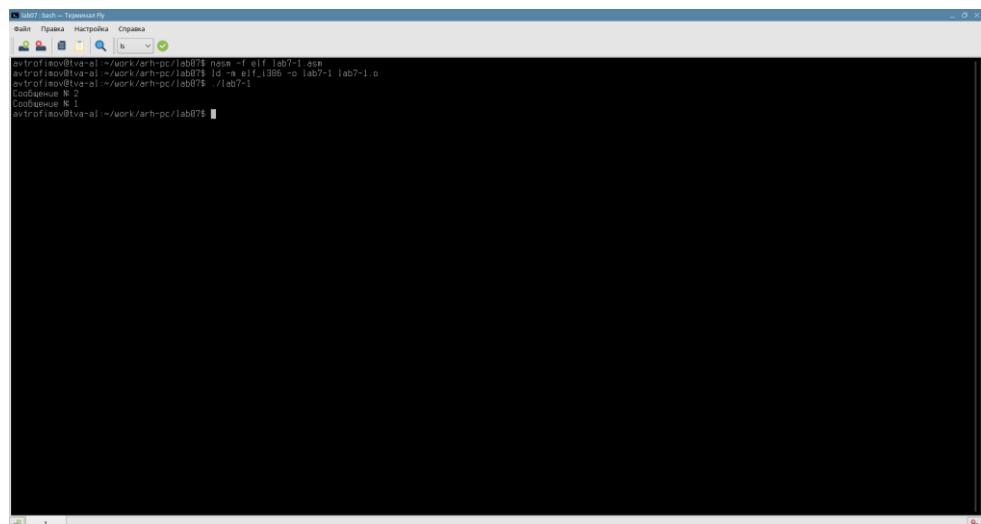


Рис. 4.5: Запуск измененой программы

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

```
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
    call sprintLF
    jmp _label2
_label2:
    mov eax, msg2
    call sprintLF
    jmp _label1
_label3:
    mov eax, msg3
    call sprintLF
    jmp quit
quit:
    call quit
```

A screenshot of a terminal window titled "lab07 ms - Терминал". The window has a menu bar with "Файл", "Правка", "Настройка", and "Справка". The main text area shows assembly code for a modified program. The code defines three strings in the .data section: "Сообщение № 1", "Сообщение № 2", and "Сообщение № 3". The .text section contains the entry point \_start, which jumps to \_label3. From \_label3, it loops back to \_label1, printing msg1 and then jumping to \_label2. From \_label2, it prints msg2 and then jumps back to \_label1. Finally, it reaches \_label3 again, prints msg3, and then calls the quit function. The terminal window has a standard Windows-style title bar and a scroll bar on the right.

Рис. 4.6: Изменение программы

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

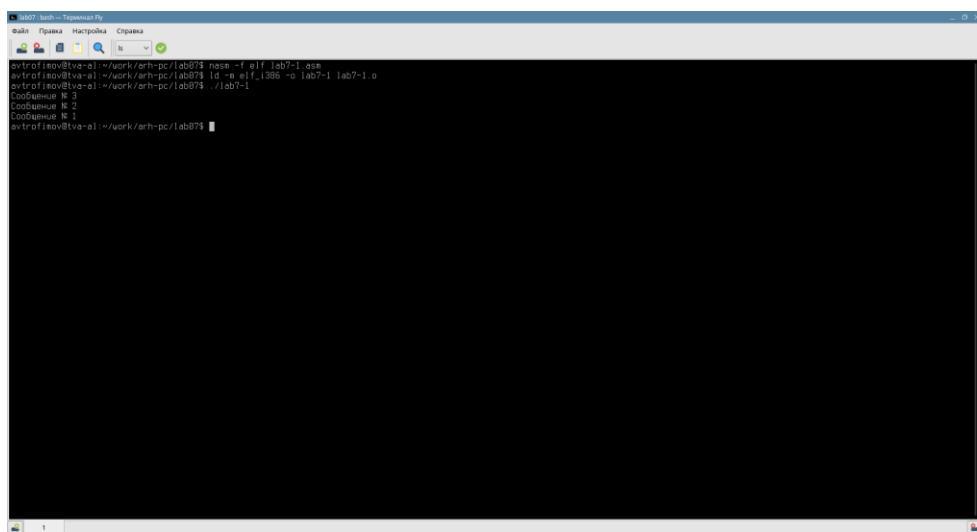


Рис. 4.7: Проверка изменений

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

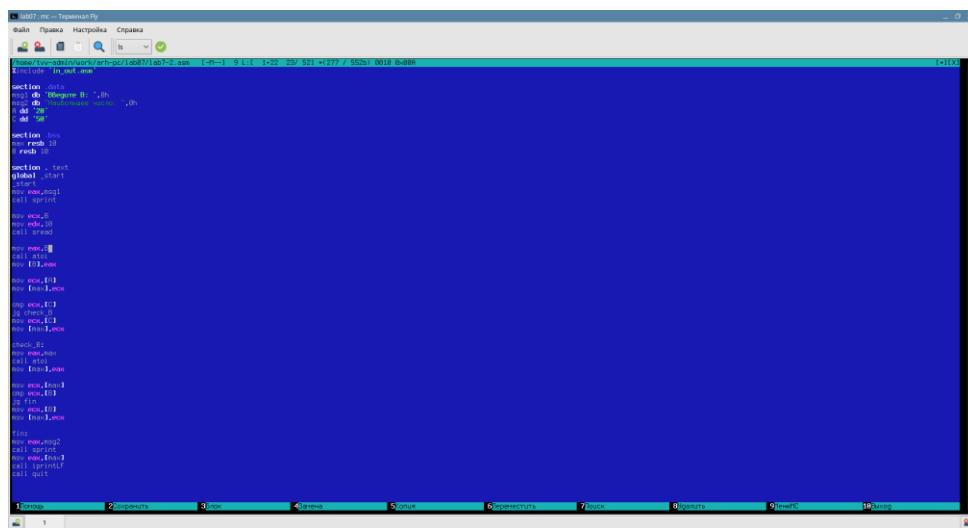


Рис. 4.8: Сохранение новой программы

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

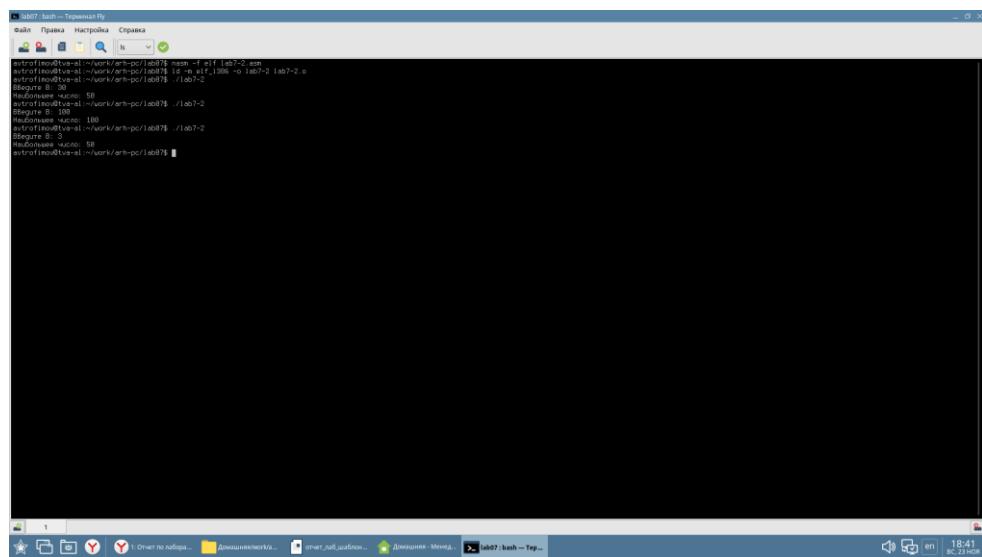


Рис. 4.9: Проверка программы из листинга

## **4.2 Изучение структуры файла листинга**

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

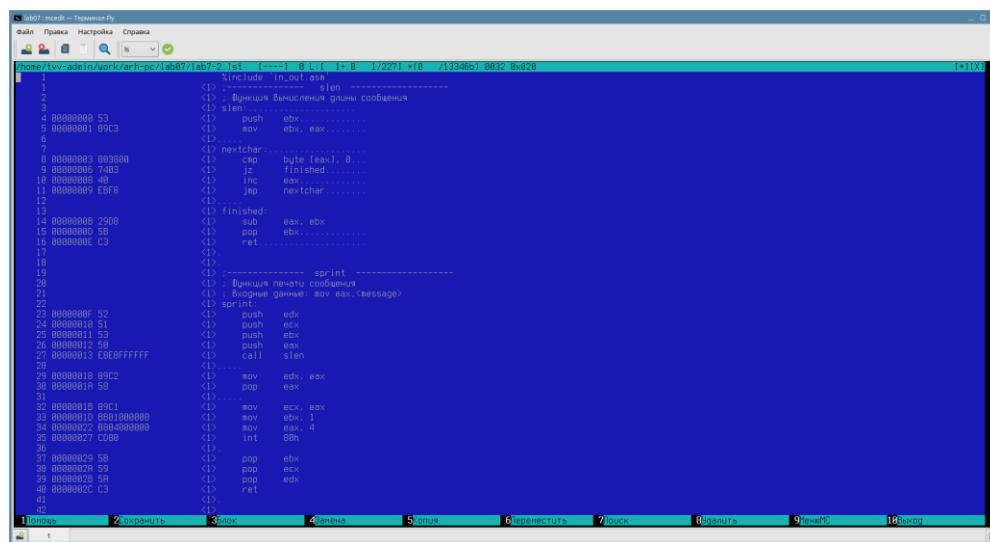


Рис. 4.10: Проверка файла листинга

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).

```
file Правка Настстройка Справка
Программа: /root/work/art-pe/obj-pe/obj7-2.asm (--->| 10 | 1 | 0+28 4B7 52 *(414 / 549D) 0010 000R
0 resb 10
1 resb 10

section test
global start
start:
    mov eax,esp
    call sprint
    mov ecx,B
    mov edx,10
    call srand
    mov eax,B
    call atoi
    mov ebx,10
    mov eax,(B)
    mov [eax],ecx
    cmp eax,[C]
    jg check_B
    mov eax,(C)
    mov [eax],ecx
    check_B:
    add eax,10
    call atoi
    mov [eax],■
    mov ecx,[eax]
    cmp ecx,(B)
    jg fin
    mov eax,(B)
    mov [eax],ecx
    fin:
    mov eax,esp
    call sprint
    mov eax,fin
    call printLF
    call exit
```

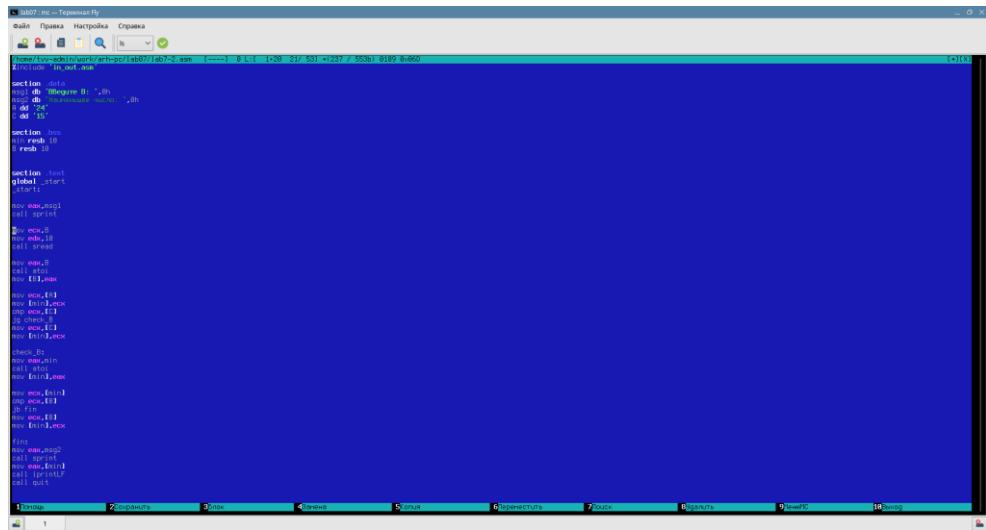
Рис. 4.11: Удаление операнда из программы

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).

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).



The screenshot shows a terminal window titled "lab07.mc - Терминал Ру". The window contains assembly code for a program named "lab07". The code includes sections for data, bss, and text, with various instructions like mov, add, sub, and cmp. There are also global labels for start, check, and init, along with several .data sections containing strings and numbers. The assembly code is written in a standard AT&M assembly syntax.

Рис. 4.13: Первая программа самостоятельной работы

Проверяю корректность написания первой программы (рис. 4.14).

```
avtrofimovtva-al:~/work/arth-pc/lab07$ nasm -f elf lab7-2.asm
avtrofimovtva-al:~/work/arth-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
avtrofimovtva-al:~/work/arth-pc/lab07$ ./lab7-2
0000000000401000
Найденное число: 15
avtrofimovtva-al:~/work/arth-pc/lab07$ ./lab7-2
8Segure B: i
Найденное число: 1
avtrofimovtva-al:~/work/arth-pc/lab07$ ./lab7-2
8Segure B: 16
Найденное число: 15
avtrofimovtva-al:~/work/arth-pc/lab07$
```

Рис. 4.14: Проверка работы первой программы

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

```
File Edit View Favorites Tools Options Help
File Project Properties Search
/home/tvv-sd33/novorka/arm-pc/label/label-3.ess {----} 11 L ( 4+38 42/51 ×15(B / 6965) 8910 B×80A
msg_set_db message значение x : ,0
msg_set_db message значение a : ,0
rem db "Ремонт"
rem db "Ремонт"
section .text
    resb 8B
    resb 8B
    resb 8B

section .text
global start
start:
    mov eax,esp_L
    call sprint
    add esp,4
    mov edx,0B
    call srand
    call _seed
    call atoi
    mov edi,eax
    mov ebx,esp_a
    call sprint
    add esp,4
    mov edx,0B
    call srand
    call _seed
    call atoi
    mov esi,eax
    cmp edi,esi
    jne add_values
    mov eax,0B
    jne print_result
    add_values:
    mov eax,edi
    add eax,esi
    print_result:
    mov edi,eax
    mov ebx,esp_a
    call sprint
    call atoi
    mov edx,edi
    call atoi
    call quit
    call quit

1 home 2 Favorites 3 icon 4 themes 5 icons 6 desktop 7 icons 8 desktop 9 desktop 10 desktop
```

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).

```
avtrotimov@tv-a-s1:~/work/arm-pc/lab07$ nc
avtrotimov@tv-a-s1:~/work/arm-pc/lab07$ nasm -f elf lab7-3.asm
avtrotimov@tv-a-s1:~/work/arm-pc/lab07$ ld -B elf_i386 -o lab7-3 lab7-3.o
Сборка значение x: 7
Сборка значение a: 7
Сборка значение b: 5
Сборка значение d: 4
Результат 4
avtrotimov@tv-a-s1:~/work/arm-pc/lab07$ ./lab7-3
```

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.