

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: Жарикова Таисия Александровна

Группа: НКАбд-05-24

МОСКВА

2024 г.

Содержание

1.Цель работы.....	3
2.Задание	4
3.Теоретическое введение.....	5
4.Выполнение лабораторной работы	
4.1.Реализация переходов в NASM	6
4.2.Изучение структуры файла листинга.....	14
4.3.Выполнение заданий для самостоятельной работы.....	18
5.Выводы.....	25
6.Источники	26

1.Цель работы

Целью данной лабораторной работы является 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

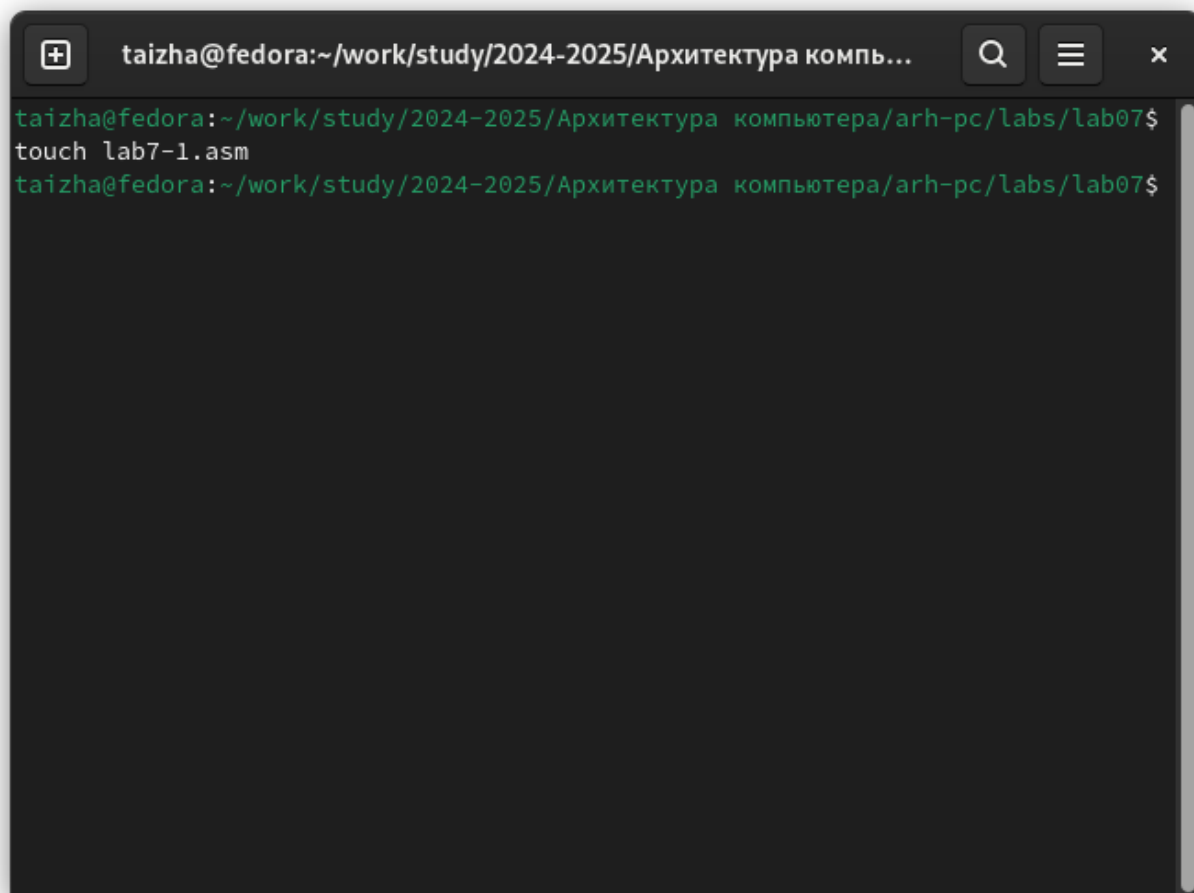
Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

- 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

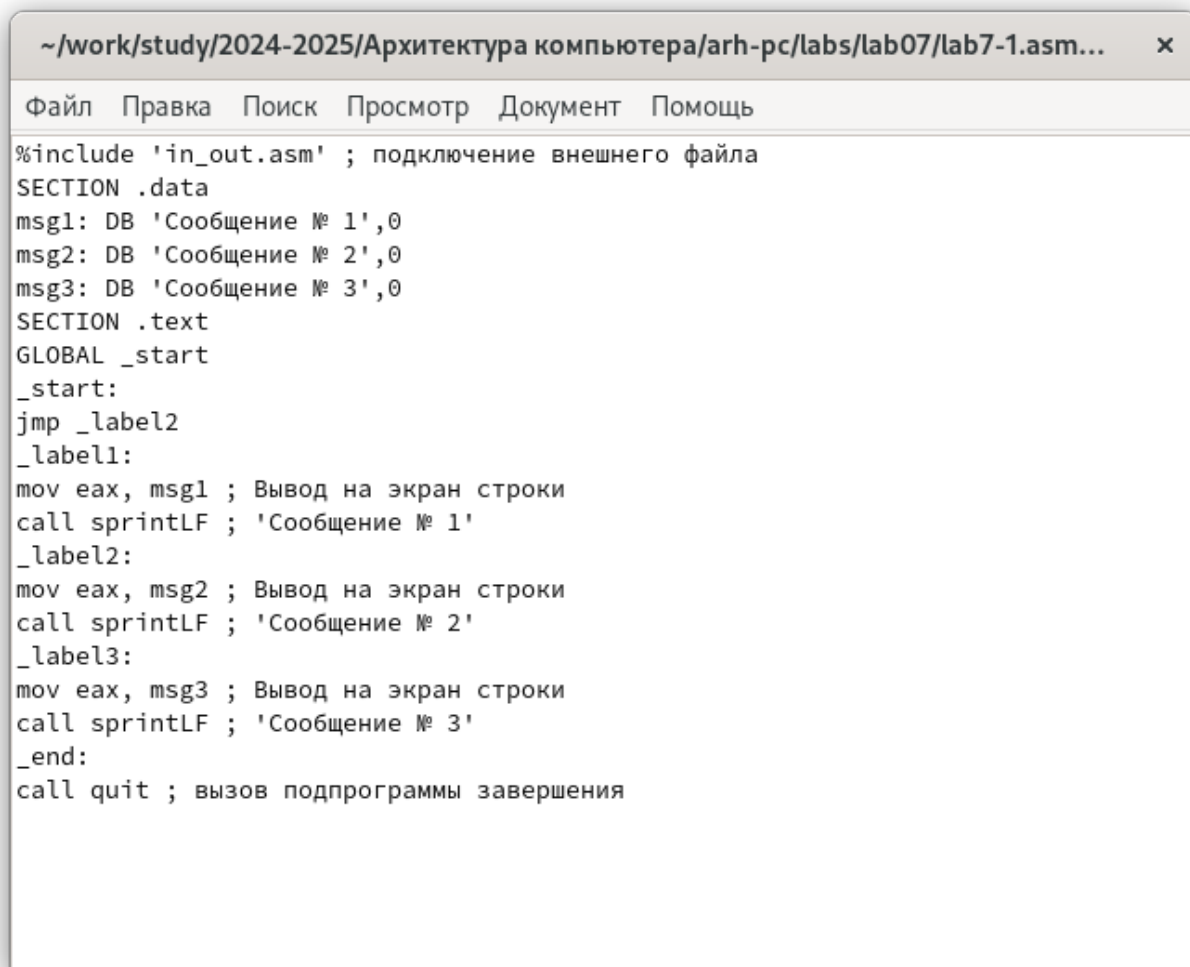
Создаю каталог для программ лабораторной работы №7 и файл для написания программ (рис. 1).



```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...  
taizha@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07$  
touch lab7-1.asm  
taizha@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07$
```

Рис.1 Создание файла для программы

Копирую код из листинга в файл будущей программы (рис. 2)

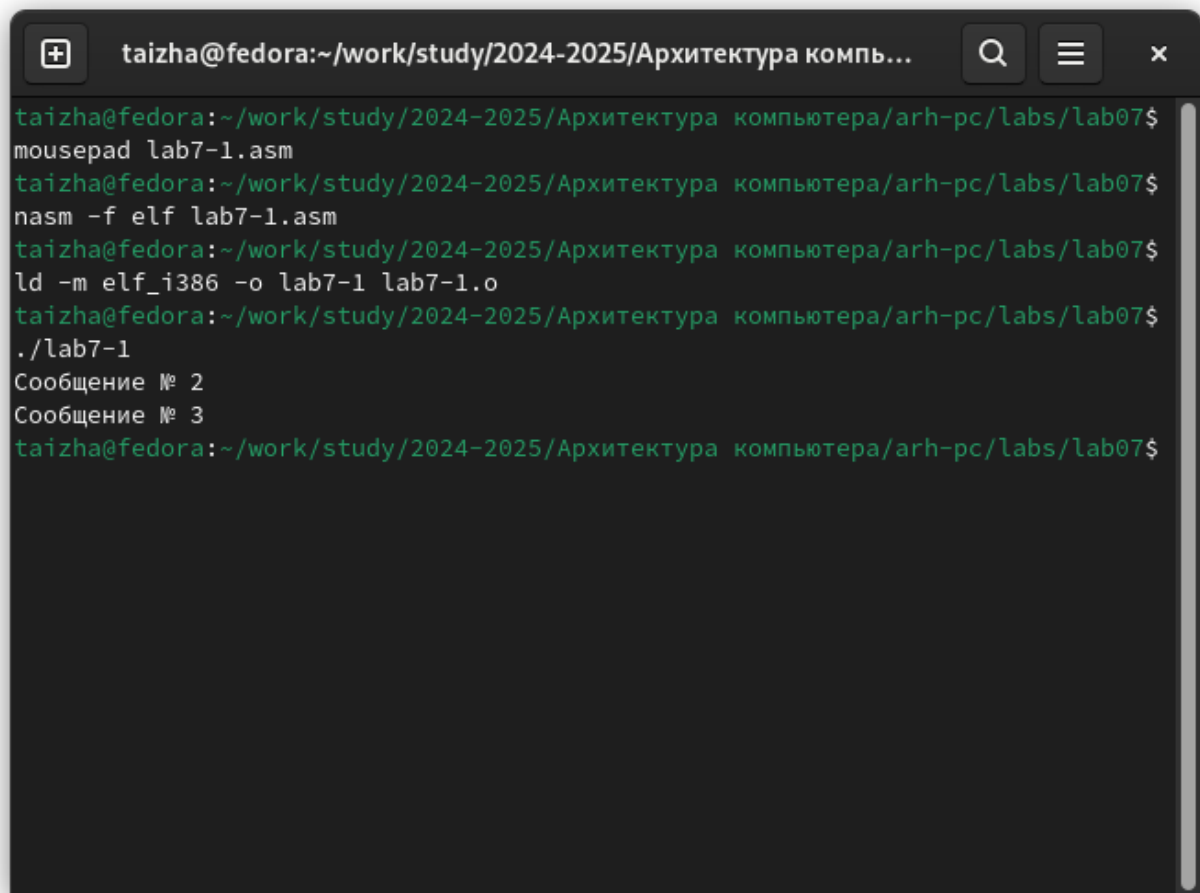


The screenshot shows a window titled `~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-1.asm...` with a menu bar containing `Файл`, `Правка`, `Поиск`, `Просмотр`, `Документ`, and `Помощь`. The main text area contains the following assembly code:

```
%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 sprintLF ; 'Сообщение № 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения
```

Рис.2 Создание программы

При запуске программы я убедилась в том, что безусловный переход действительно изменяет порядок выполнения инструкций (рис. 3).

A terminal window with a dark background and light green text. The window title is "taizha@fedora:~/work/study/2024-2025/Архитектура компь...". The terminal shows the following commands and output:

```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
mousepad lab7-1.asm  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
nasm -f elf lab7-1.asm  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
ld -m elf_i386 -o lab7-1 lab7-1.o  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
./lab7-1  
Сообщение № 2  
Сообщение № 3  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

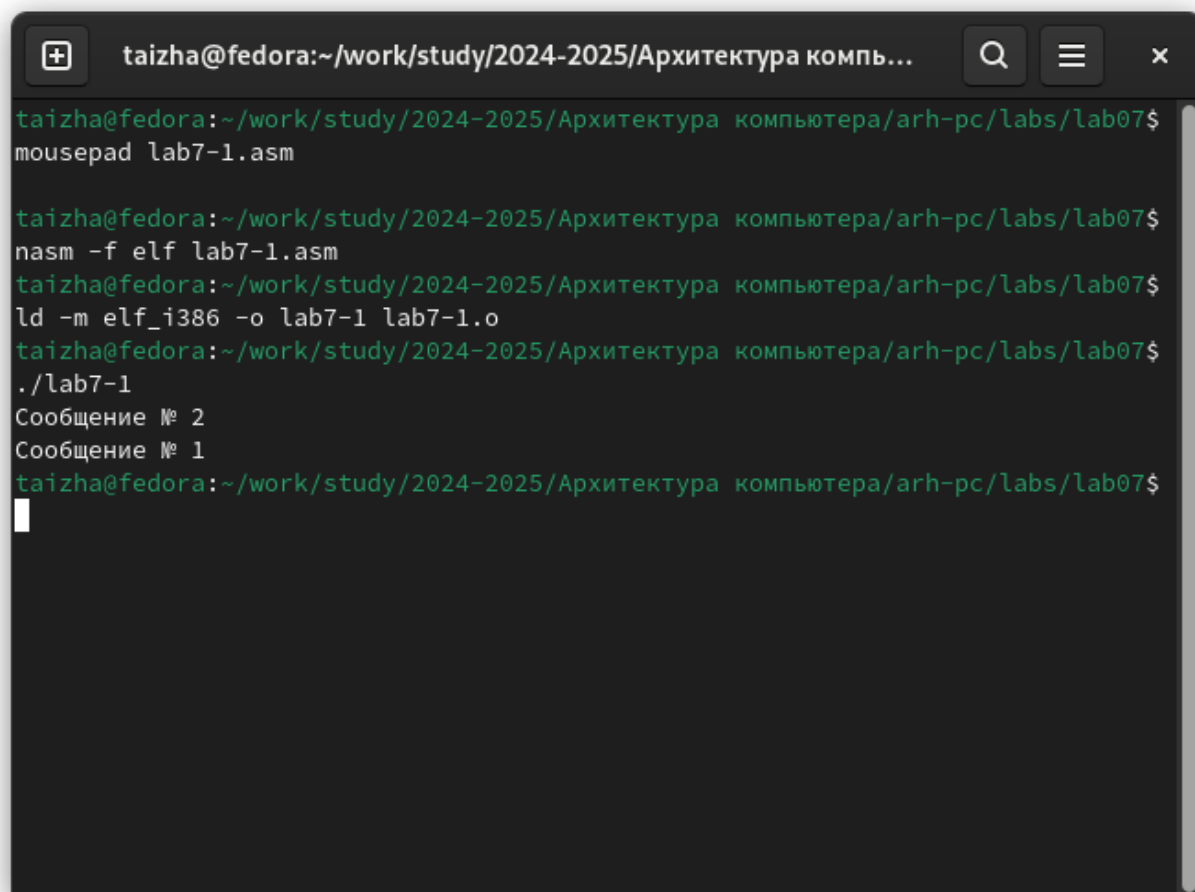
Рис.3 Запуск исполняемого файла

Изменяю программу таким образом, чтобы поменялся порядок выполнения функций (рис. 4).


```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-1.asm... x
Файл  Правка  Поиск  Просмотр  Документ  Помощь
%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 sprintfLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintfLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintfLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения
```

Рис.4 Изменение программы

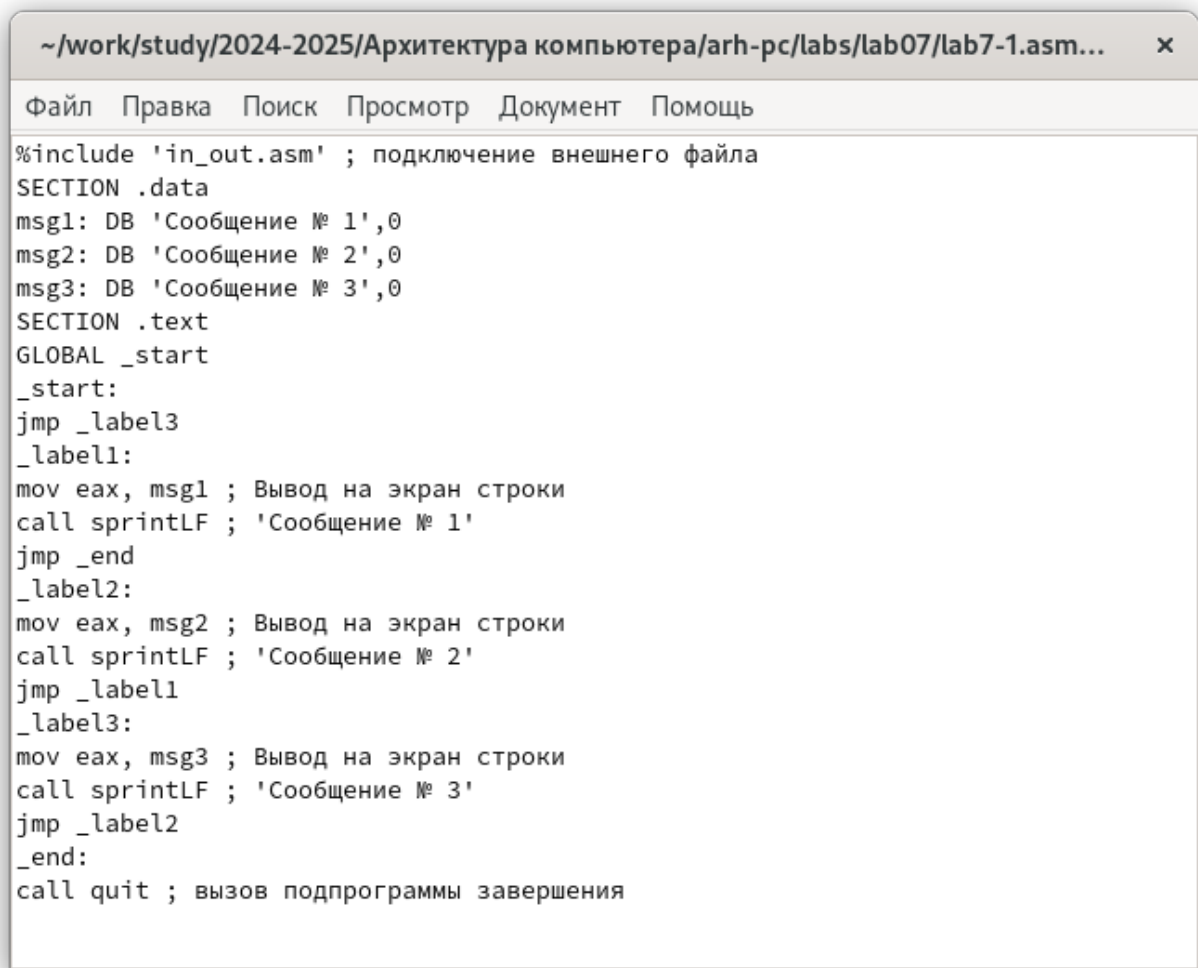
Запускаю программу и проверяю, что примененные изменения верны (рис. 5).

A terminal window with a dark background and light green text. The window title is "taizha@fedora:~/work/study/2024-2025/Архитектура компь...". The terminal shows the following commands and output:

```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ mousepad lab7-1.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ nasm -f elf lab7-1.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ld -m elf_i386 -o lab7-1 lab7-1.o
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ./lab7-1
Сообщение № 2
Сообщение № 1
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

Рис.5 Запуск изменённой программы

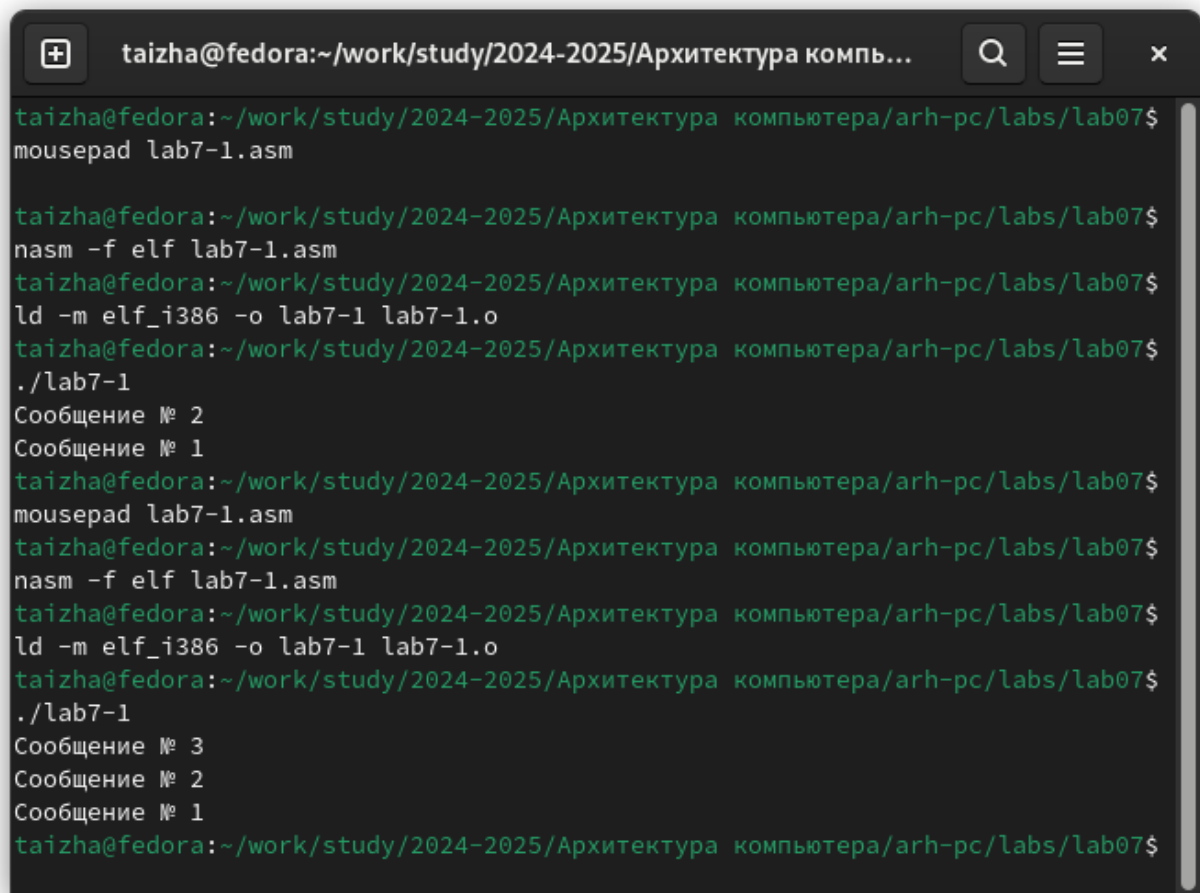
Теперь изменяю текст программы так, чтобы все три сообщения вывелись в обратном порядке (рис. 6)



```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-1.asm... x
Файл  Правка  Поиск  Просмотр  Документ  Помощь
%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 _label3
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения
```

Рис.6 Изменение программы

Работа выполнена корректно, программа в нужном мне порядке выводит сообщения (рис. 7)

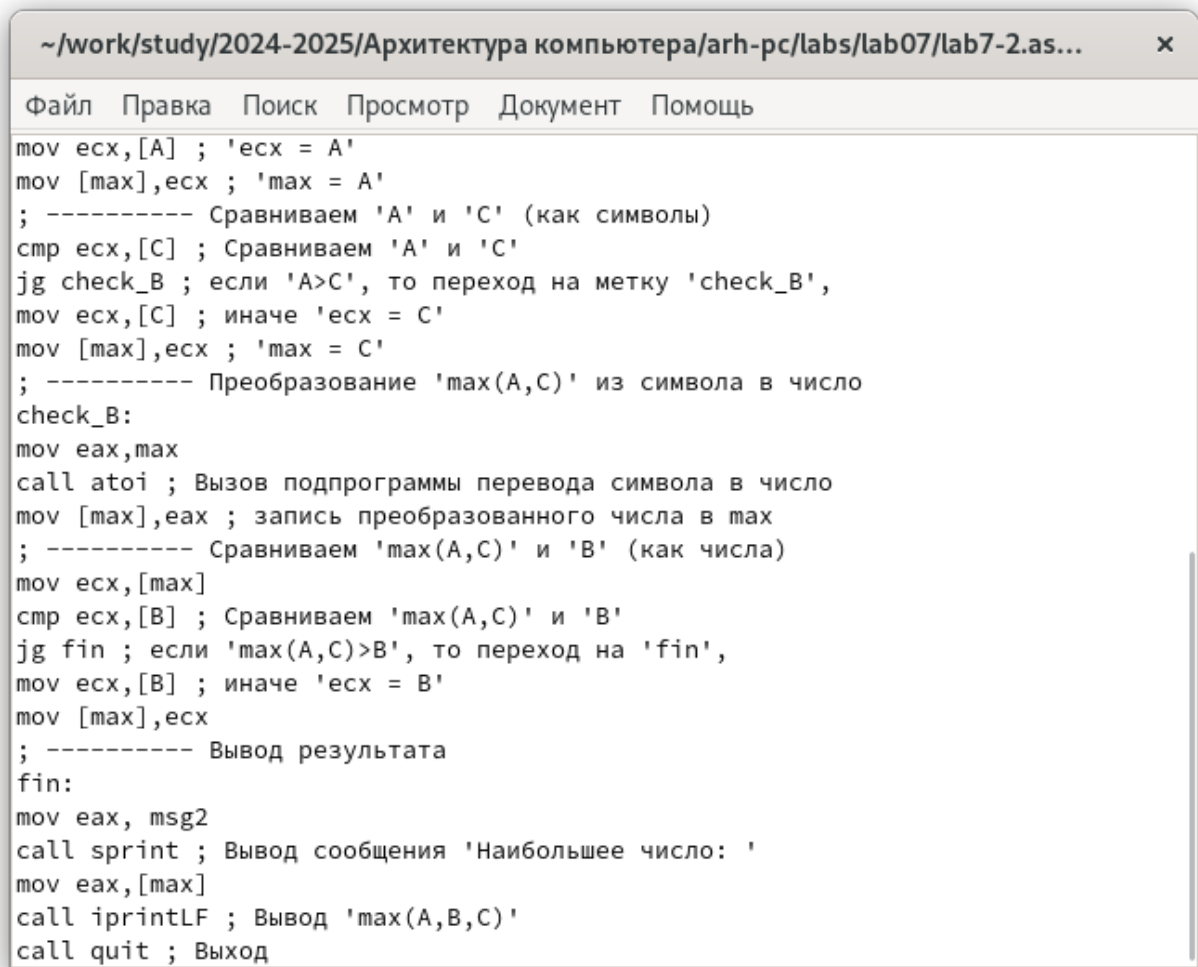


```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
mousepad lab7-1.asm

taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
nasm -f elf lab7-1.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
ld -m elf_i386 -o lab7-1 lab7-1.o
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
./lab7-1
Сообщение № 2
Сообщение № 1
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
mousepad lab7-1.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
nasm -f elf lab7-1.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
ld -m elf_i386 -o lab7-1 lab7-1.o
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

Рис.7 Проверка изменений

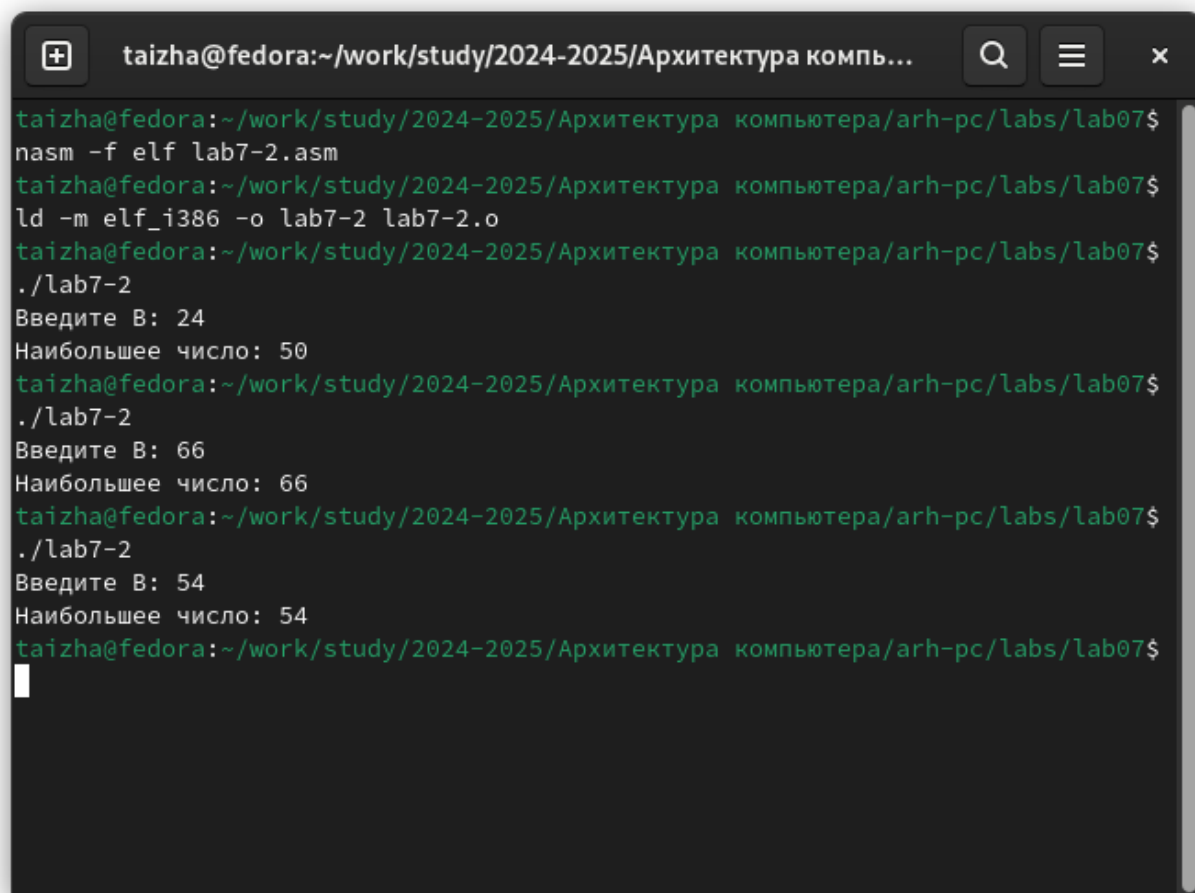
Создаю новый рабочий файл и вставляю в него код из следующего листинга (рис. 8).



```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-2.as... x
Файл  Правка  Поиск  Просмотр  Документ  Помощь
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,max
call atoi ; Вызов подпрограммы перевода символа в число
mov [max],eax ; запись преобразованного числа в max
; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
mov ecx,[max]
cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
jg fin ; если 'max(A,C)>B', то переход на 'fin',
mov ecx,[B] ; иначе 'ecx = B'
mov [max],ecx
; ----- Вывод результата
fin:
mov eax, msg2
call sprint ; Вывод сообщения 'Наибольшее число: '
mov eax,[max]
call iprintLF ; Вывод 'max(A,B,C)'
call quit ; Выход
```

Рис.8 Создание новой программы

Программа выводит значение переменной с максимальным значением, проверяю работу программы с разными входными данными (рис. 9).

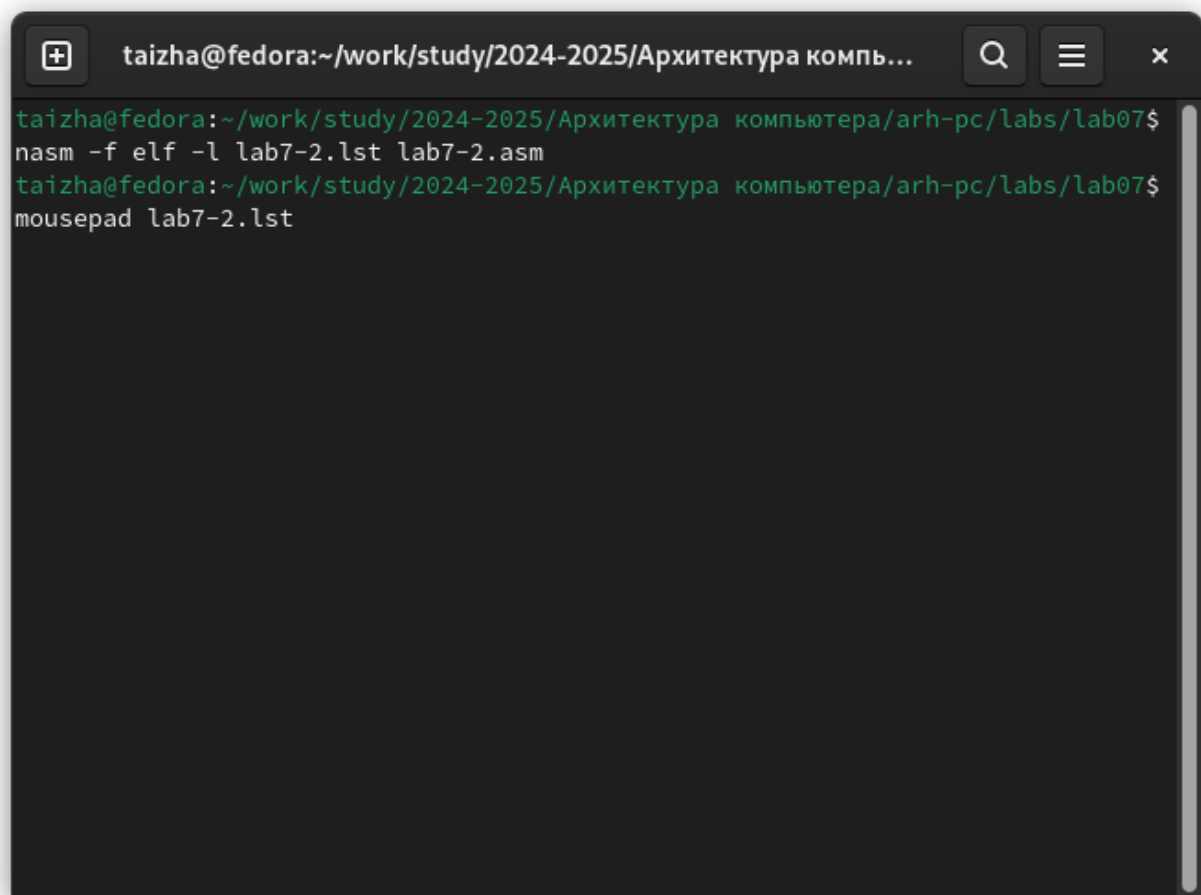


```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ nasm -f elf lab7-2.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ld -m elf_i386 -o lab7-2 lab7-2.o
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ./lab7-2
Введите В: 24
Наибольшее число: 50
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ./lab7-2
Введите В: 66
Наибольшее число: 66
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ./lab7-2
Введите В: 54
Наибольшее число: 54
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

Рис.9 Проверка программы

4.2.Изучение структуры файла листинга

Создаю файл листинга с помощью флага -l команды nasm и открываю его с помощью текстового редактора mousepad (рис. 10, 11).

A terminal window with a dark background and light green text. The window title is "taizha@fedora:~/work/study/2024-2025/Архитектура компь...". The terminal shows three lines of commands: "nasm -f elf -l lab7-2.lst lab7-2.asm", "taizha@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07\$", and "mousepad lab7-2.lst".

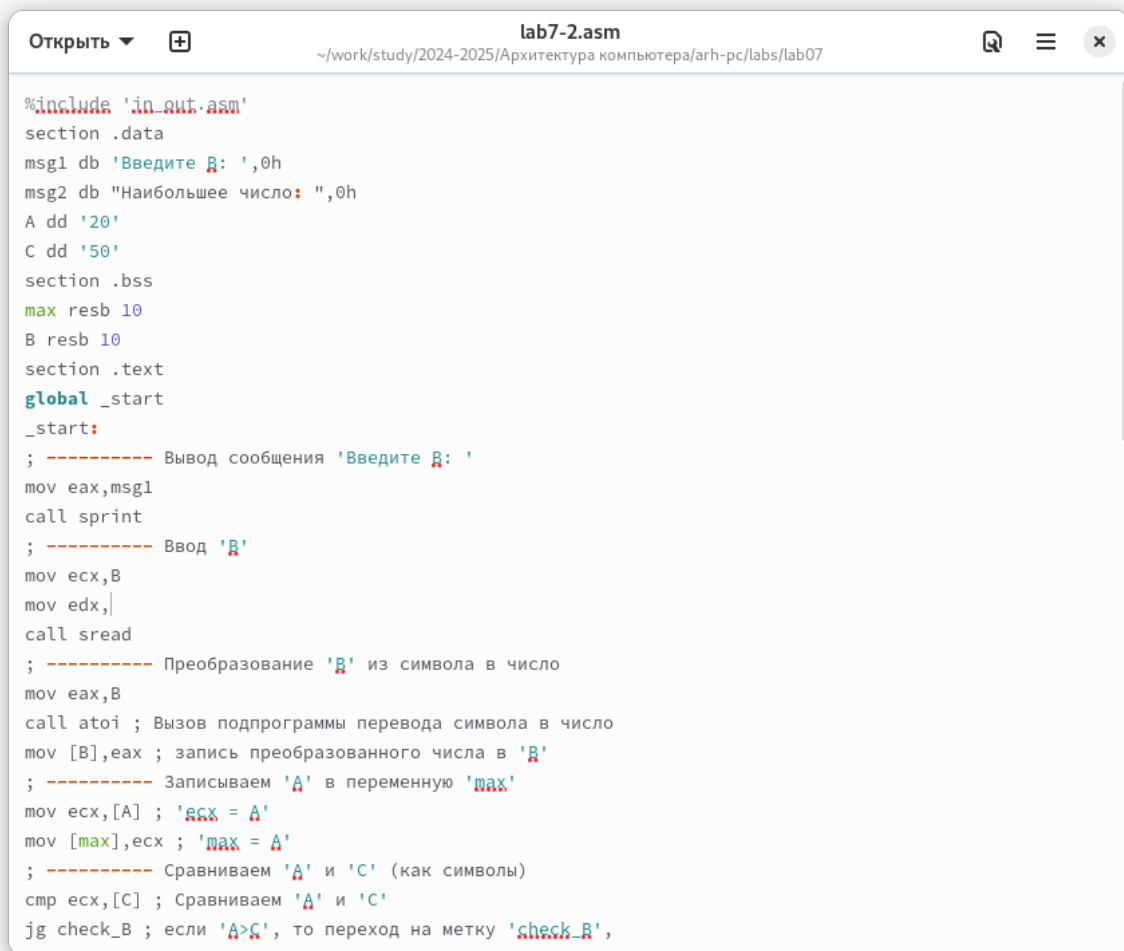
```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...
taizha@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07$
nasm -f elf -l lab7-2.lst lab7-2.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07$
mousepad lab7-2.lst
```

Рис.10 Создание файла

```
1                               %include 'in_out.asm'
1                               <1> ;----- slen -----
2                               <1> ; Функция вычисления длины сообщения
3                               <1> slen:
4 00000000 53                   <1>     push     ebx
5 00000001 89C3                 <1>     mov      ebx, eax
6                               <1>
7                               <1> nextchar:
8 00000003 803800               <1>     cmp      byte [eax], 0
9 00000006 7403                 <1>     jz       finished
10 00000008 40                  <1>     inc      eax
11 00000009 EBF8                <1>     jmp      nextchar
12                               <1>
13                               <1> finished:
14 0000000B 29D8                 <1>     sub      eax, ebx
15 0000000D 5B                  <1>     pop      ebx
16 0000000E C3                  <1>     ret
17                               <1>
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
```

Рис.11 Проверка файла

Первое значение в файле листинга представляет собой номер строки, который может не совпадать с изначальным номером строки в исходном файле. Далее следует адрес — смещение машинного кода относительно начала текущего сегмента, затем сам машинный код. Завершает строку исходный текст программы с комментариями. Для проверки поведения файла листинга вношу изменения: удаляю один операнд из случайной инструкции (рис. 12).



```
lab7-2.asm
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07

%include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,|
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
```

Рис.12 Удаление операнда из программы

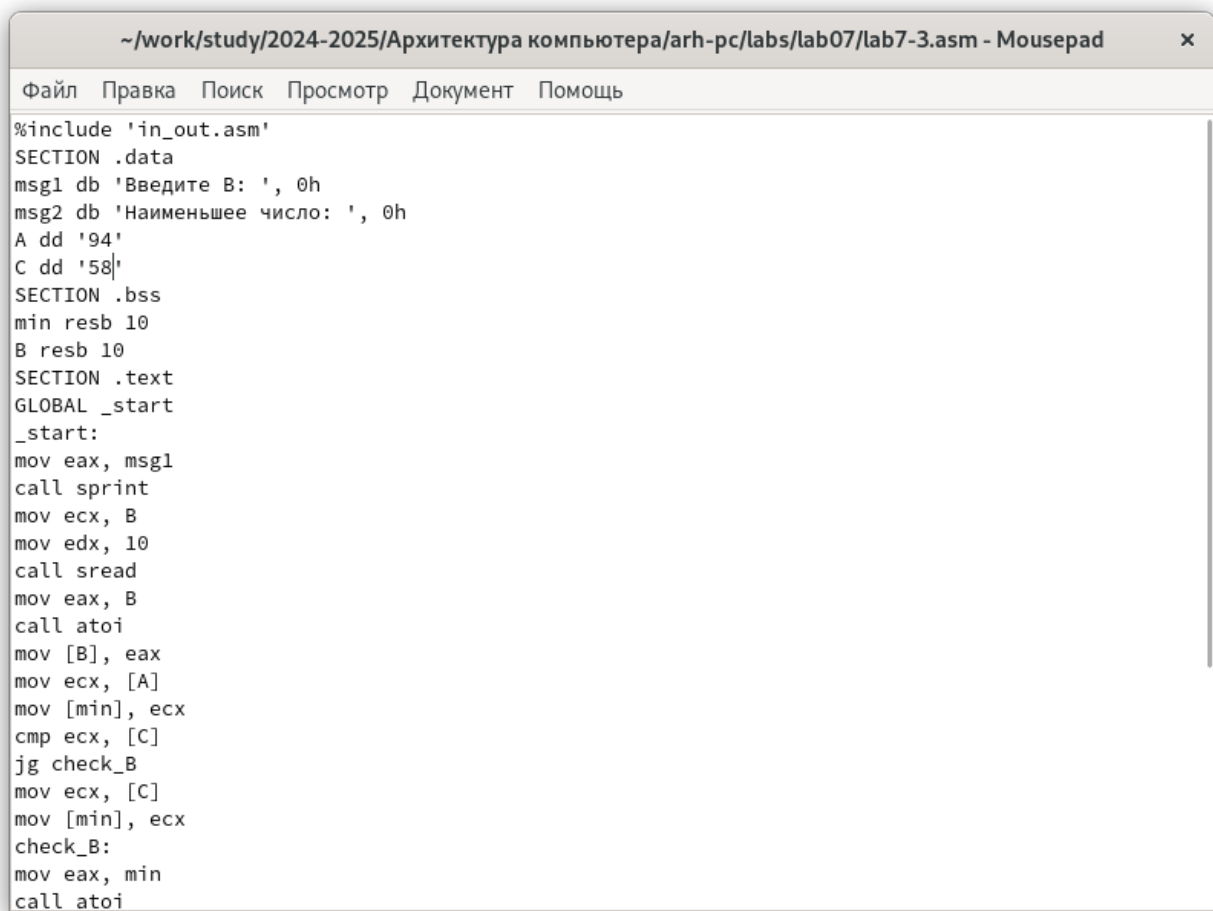
В новом файле листинга показывает ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются (рис. 13).

Адрес	Смещение	Операция	Комментарий
5	00000035 32300000	A dd '20'	
6	00000039 35300000	C dd '50'	
7		section .bss	
8	00000000 <res Ah>	max resb 10	
9	0000000A <res Ah>	B resb 10	
10		section .text	
11		global _start	
12		_start:	
13		; ----- Вывод сообщения 'Введите B: '	
14	000000E8 B8[00000000]	mov eax,msg1	
15	000000ED E81DFFFFFF	call sprintf	
16		; ----- Ввод 'B'	
17	000000F2 B9[0A000000]	mov ecx,B	
18		mov edx,	
18	*****	error: invalid combination of opcode and operands	
19	000000F7 E847FFFFFF	call sread	
20		; ----- Преобразование 'B' из символа в число	
21	000000FC B8[0A000000]	mov eax,B	
22	00000101 E896FFFFFF	call atoi ; Вызов подпрограммы перевода символа в число	
23	00000106 A3[0A000000]	mov [B],eax ; запись преобразованного числа в 'B'	
24		; ----- Записываем 'A' в переменную 'max'	
25	0000010B 8B0D[35000000]	mov ecx,[A] ; 'ecx = A'	
26	00000111 890D[00000000]	mov [max],ecx ; 'max = A'	
27		; ----- Сравниваем 'A' и 'C' (как символы)	
28	00000117 3B0D[39000000]	cmp ecx,[C] ; Сравниваем 'A' и 'C'	
29	0000011D 7F0C	jg check_B ; если 'A>C', то переход на метку 'check_B',	
30	0000011F 8B0D[39000000]	mov ecx,[C] ; иначе 'ecx = C'	
31	00000125 890D[00000000]	mov [max],ecx ; 'max = C'	
32		; ----- Преобразование 'max(A,C)' из символа в число	
33		check_B:	

Рис.13 Просмотр ошибки в файле листинга

4.3.Выполнение заданий для самостоятельной работы

Выполняю 3 вариант. Возвращаю операнд к функции в программе и изменяю ее так, чтобы она выводила переменную с наименьшим значением (рис. 14).



```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-3.asm - Mousepad x
Файл Правка Поиск Просмотр Документ Помощь

%include 'in_out.asm'
SECTION .data
msg1 db 'Введите B: ', 0h
msg2 db 'Наименьшее число: ', 0h
A dd '94'
C dd '58'
SECTION .bss
min resb 10
B resb 10
SECTION .text
GLOBAL _start
_start:
mov eax, msg1
call sprint
mov ecx, B
mov edx, 10
call sread
mov eax, B
call atoi
mov [B], eax
mov ecx, [A]
mov [min], ecx
cmp ecx, [C]
jg check_B
mov ecx, [C]
mov [min], ecx
check_B:
mov eax, min
call atoi
```

Рис.14 Написание программы

Код программы:

```
%include 'in_out.asm'

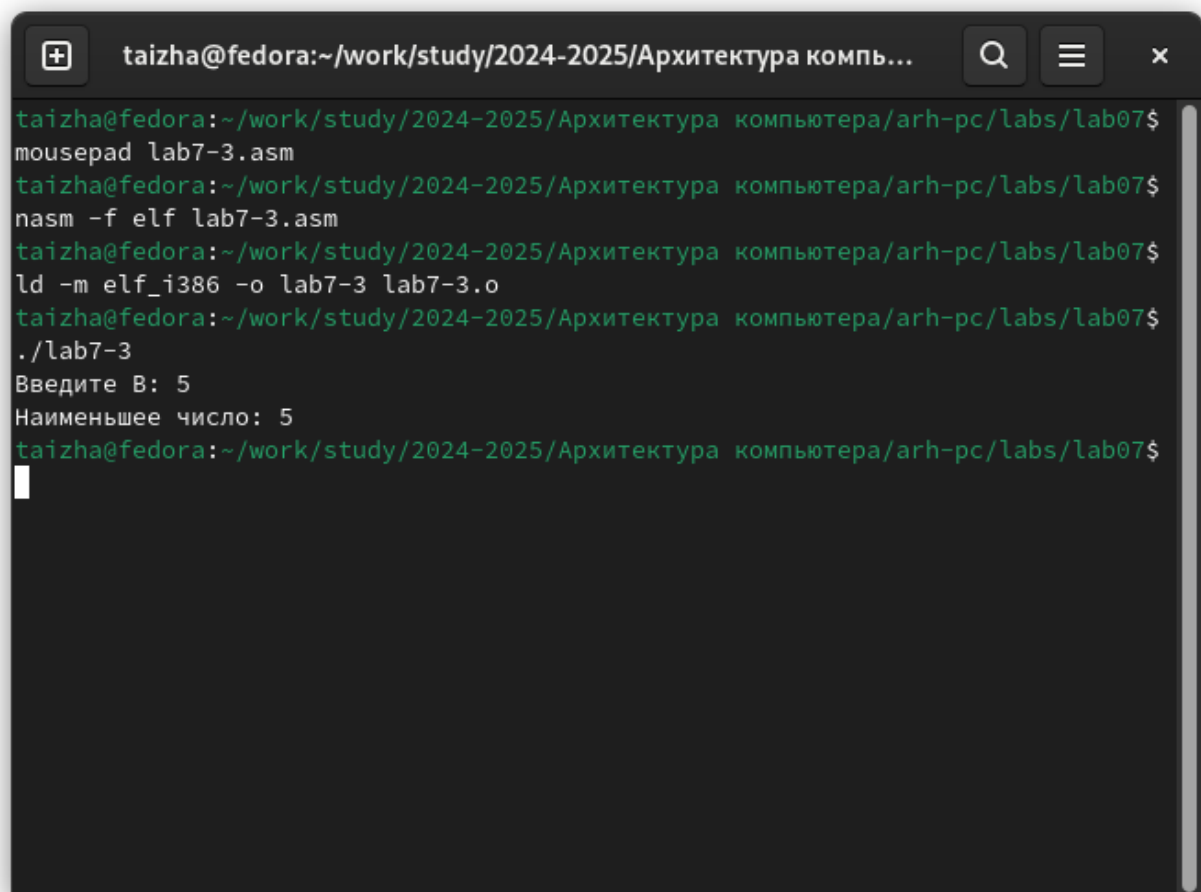
SECTION .data
msg1 db 'Введите B: ', 0h
msg2 db 'Наименьшее число: ', 0h
A dd '8'
C dd '68'
SECTION .bss
min resb 10
B resb 10
SECTION .text
```

```

GLOBAL _start
_start:
mov eax, msg1
call sprint
mov ecx, B
mov edx, 10
call sread
mov eax, B
call atoi
mov [B], eax
mov ecx, [A]
mov [min], ecx
cmp ecx, [C]
jg check_B
mov ecx, [C]
mov [min], ecx
check_B:
mov eax, min
call atoi
mov [min], eax
mov ecx, [min]
cmp ecx, [B]
jb fin
mov ecx, [B]
mov [min], ecx
fin:
mov eax, msg2
call sprint
mov eax, [min]
call iprintLF
call quit

```

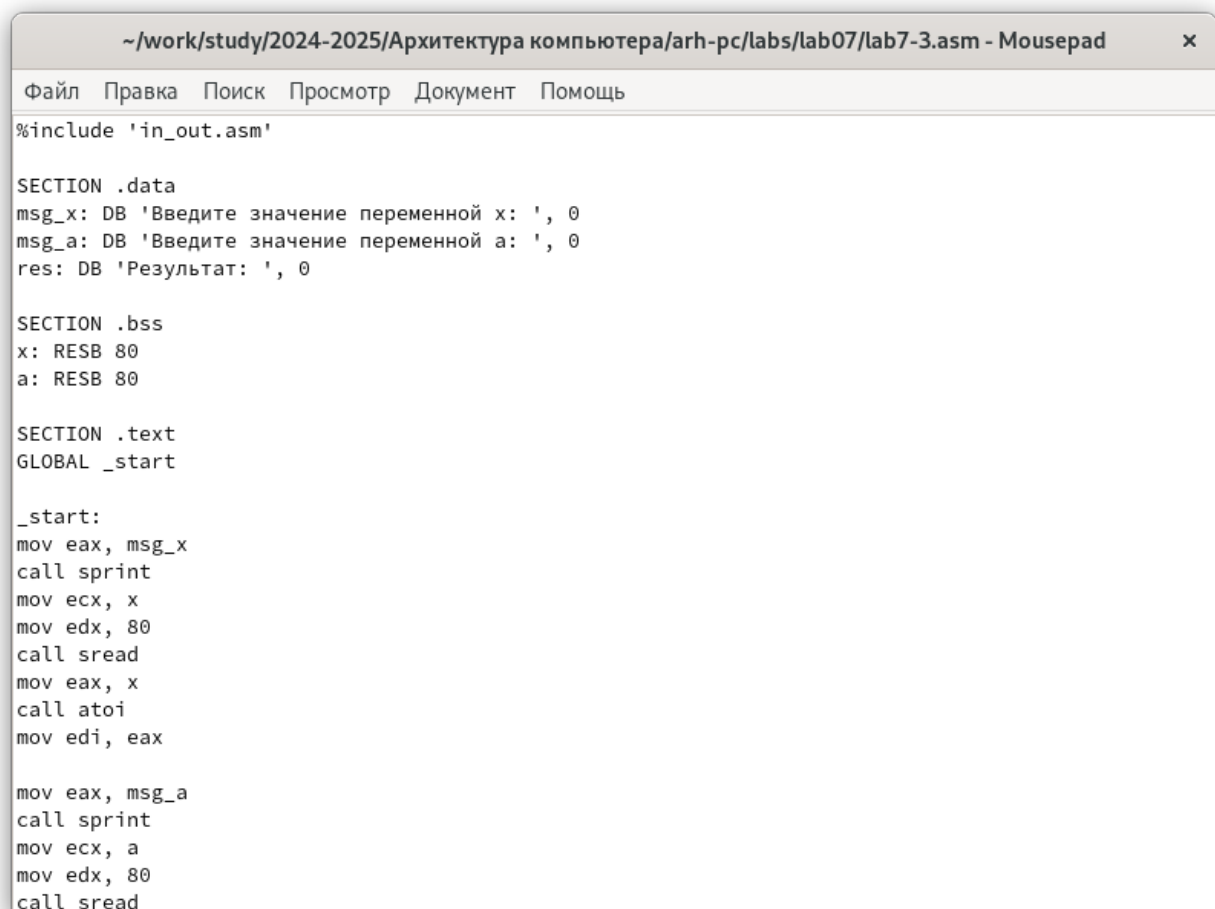
Проверяю корректность написания первой программы (рис. 15).

A terminal window with a dark background and light green text. The window title is "taizha@fedora:~/work/study/2024-2025/Архитектура компь...". The terminal shows the following commands and output:

```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ mousepad lab7-3.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ nasm -f elf lab7-3.asm
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ld -m elf_i386 -o lab7-3 lab7-3.o
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$ ./lab7-3
Введите В: 5
Наименьшее число: 5
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

Рис.15 Запуск программы

Пишу программу, которая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатуры переменных а и х (рис. 16).



```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab07/lab7-3.asm - Mousepad
Файл  Правка  Поиск  Просмотр  Документ  Помощь

%include 'in_out.asm'

SECTION .data
msg_x: DB 'Введите значение переменной x: ', 0
msg_a: DB 'Введите значение переменной a: ', 0
res: DB 'Результат: ', 0

SECTION .bss
x: RESB 80
a: RESB 80

SECTION .text
GLOBAL _start

_start:
mov eax, msg_x
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov edi, eax

mov eax, msg_a
call sprint
mov ecx, a
mov edx, 80
call sread
```

Рис.16 Написание программы

Код программы:

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

```
SECTION .data
```

```
msg_x: DB 'Введите значение переменной x: ', 0
```

```
msg_a: DB 'Введите значение переменной a: ', 0
```

```
res: DB 'Результат: ', 0
```

```
SECTION .bss
```

```
x: RESB 80
```

```
a: RESB 80
```

```
SECTION .text
GLOBAL _start
```

```
_start:
mov eax, msg_x
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov edi, eax
```

```
mov eax, msg_a
call sprint
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov esi, eax
```

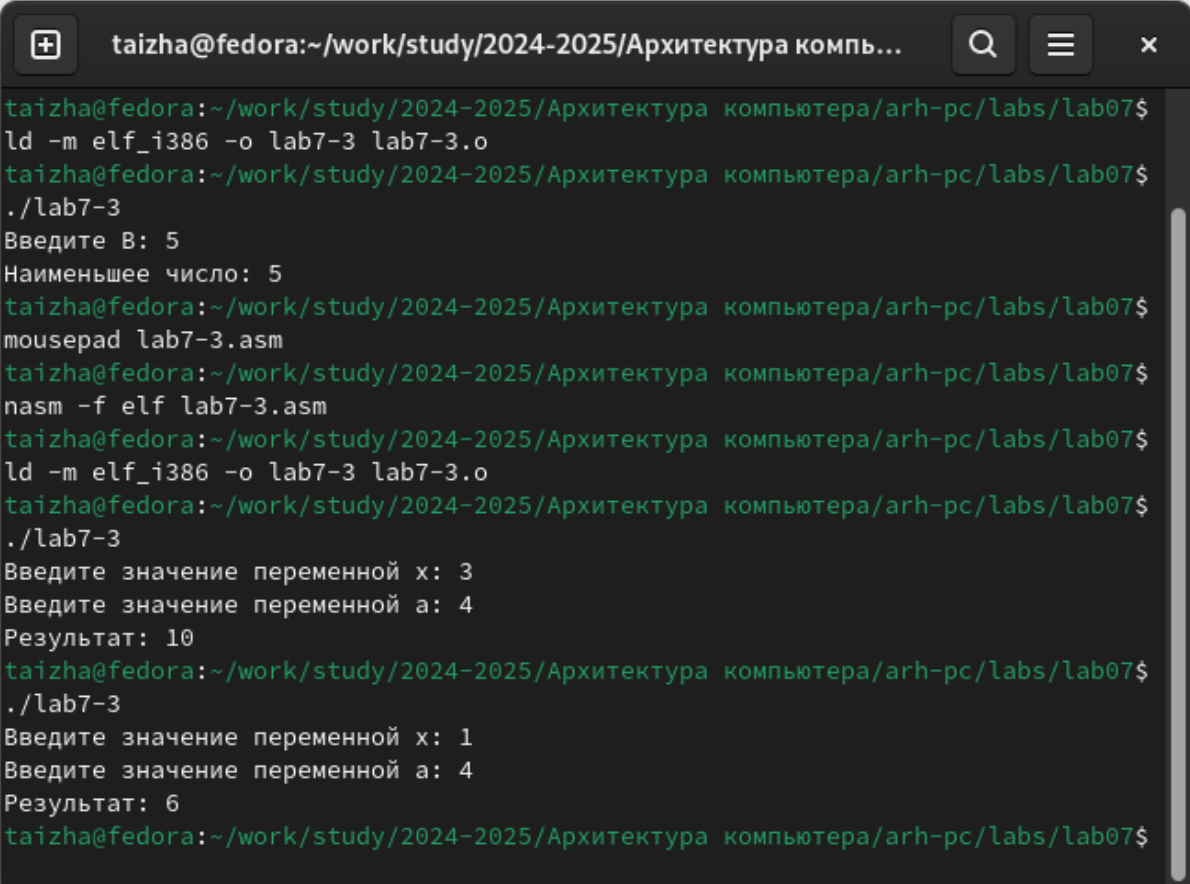
```
cmp esi, 0
je case_a_zero
```

```
mov eax, edi
add eax, eax
add eax, esi
jmp print_result
```

```
case_a_zero:
mov eax, edi
add eax, eax
add eax, 1
```

```
print_result:
mov edi, eax
mov eax, res
call sprint
mov eax, edi
call iprintLF
call quit
```

Транслирую и компоную файл, запускаю и проверяю работу программы для различных значений а и х (рис. 17).

A terminal window with a dark background and green text. The window title is "taizha@fedora:~/work/study/2024-2025/Архитектура компь...". The terminal shows the compilation of an assembly file "lab7-3.asm" into "lab7-3.o" using "ld", followed by running "mousepad lab7-3.asm" and "nasm -f elf lab7-3.asm". Then, the program is executed with "./lab7-3", which prompts for input. The first run takes input 5 and outputs 10. The second run takes input 1 and outputs 6.

```
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
ld -m elf_i386 -o lab7-3 lab7-3.o  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
./lab7-3  
Введите В: 5  
Наименьшее число: 5  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
mousepad lab7-3.asm  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
nasm -f elf lab7-3.asm  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
ld -m elf_i386 -o lab7-3 lab7-3.o  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
./lab7-3  
Введите значение переменной x: 3  
Введите значение переменной a: 4  
Результат: 10  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$  
./lab7-3  
Введите значение переменной x: 1  
Введите значение переменной a: 4  
Результат: 6  
taizha@fedora:~/work/study/2024-2025/Архитектура компь...$
```

Рис.17 Запуск 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.

6.Источники

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)