

Лабораторная работа №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Габидов Ислам Магомедович

Содержание

1 Цель работы

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

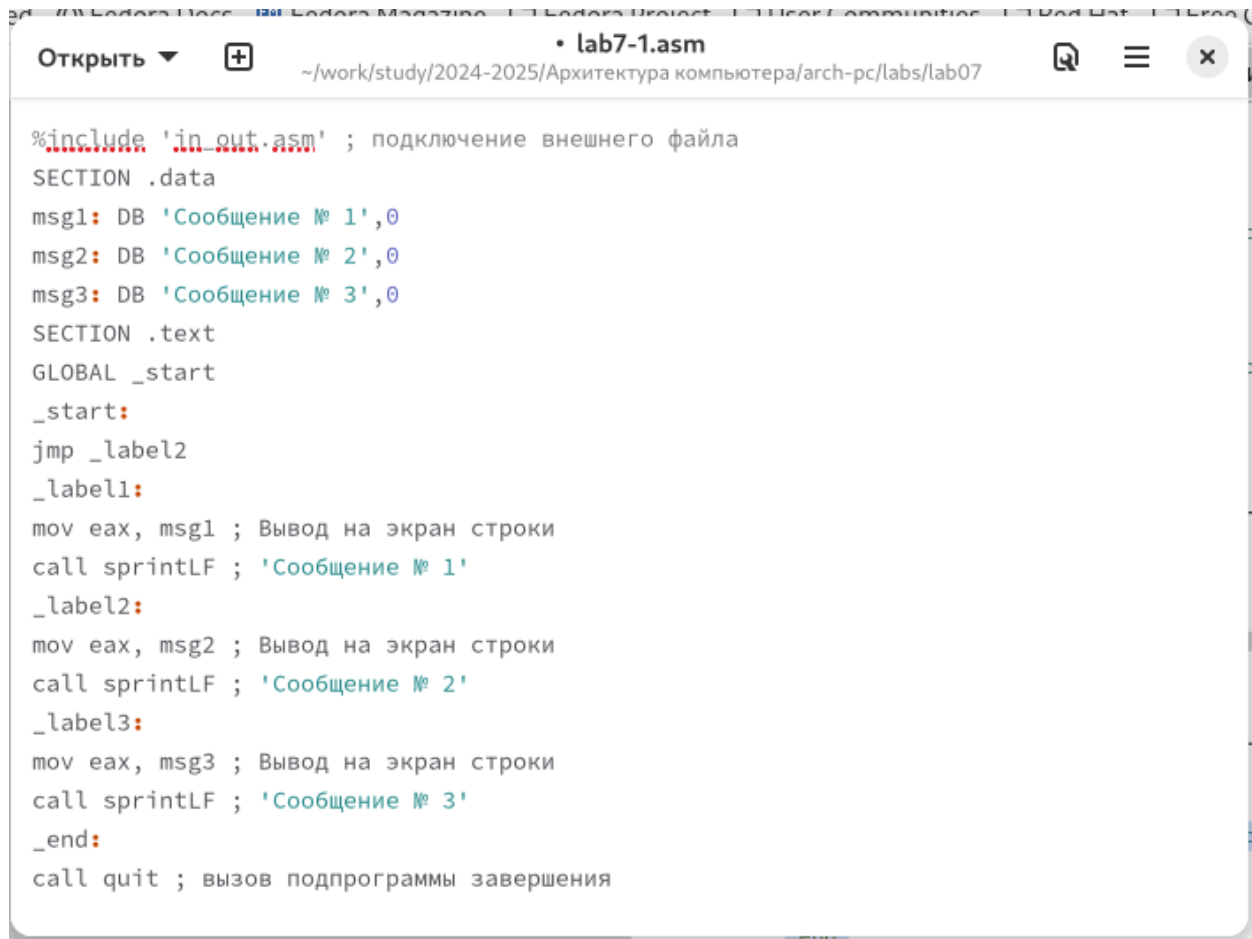
2 Выполнение лабораторной работы

- 1) Я перешел в каталог lab7 и внутри создал файл lab7-1.asm

```
исламгабидов@fedora:~$ cd work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab07
исламгабидов@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ touch lab7-1.asm
исламгабидов@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ls
lab7-1.asm
исламгабидов@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 1: Создание файла lab7-1.asm

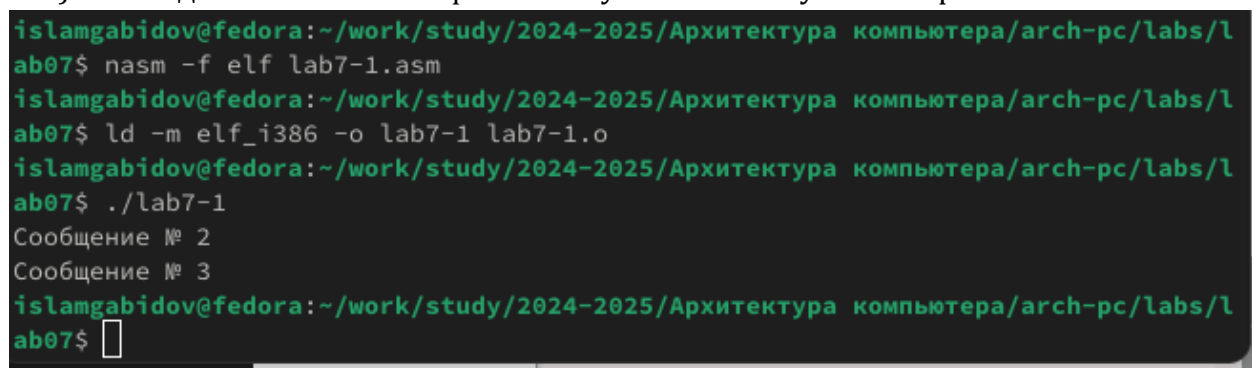
- 2) Я ввел в файл текст программы и запустил его.



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

Рис. 2: Текст в файле lab7-1.asm

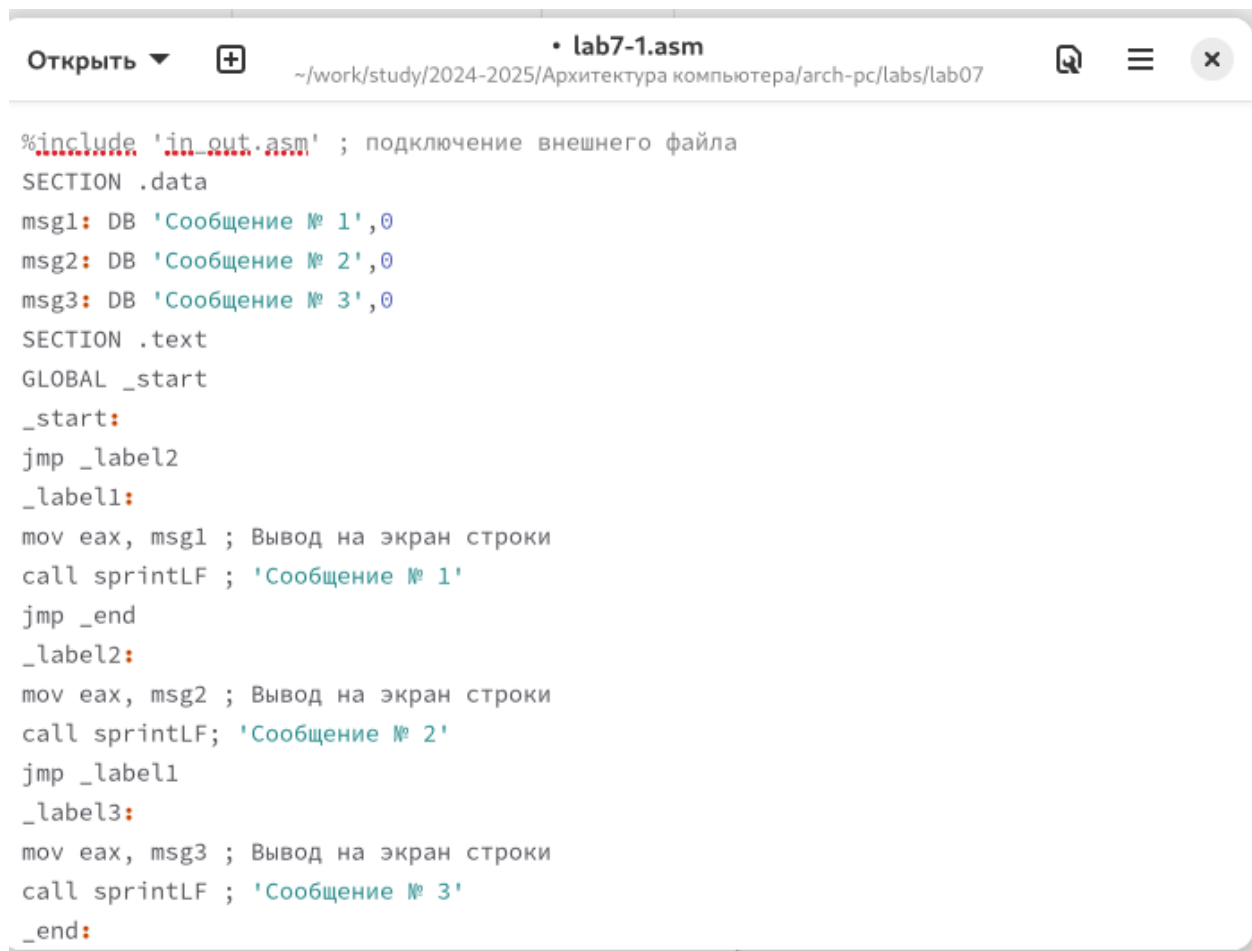
3) Я создал исполняемый файл и запустил его. Результат верный



```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 3: Запуск программы lab7-1

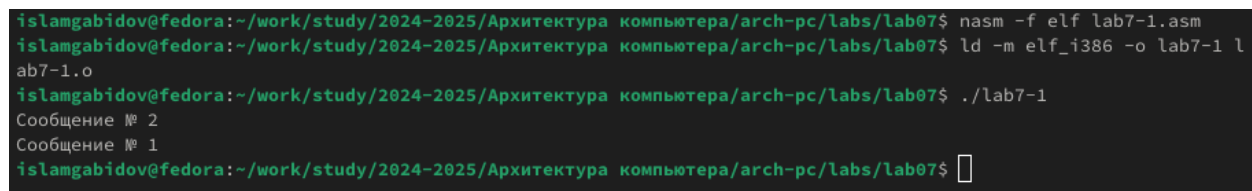
4) Я изменил текст программы чтобы выводился нужный ответ и создал исполняемый файл.



```
Открыть ▾ + • lab7-1.asm ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07

%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'
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'
_end:
```

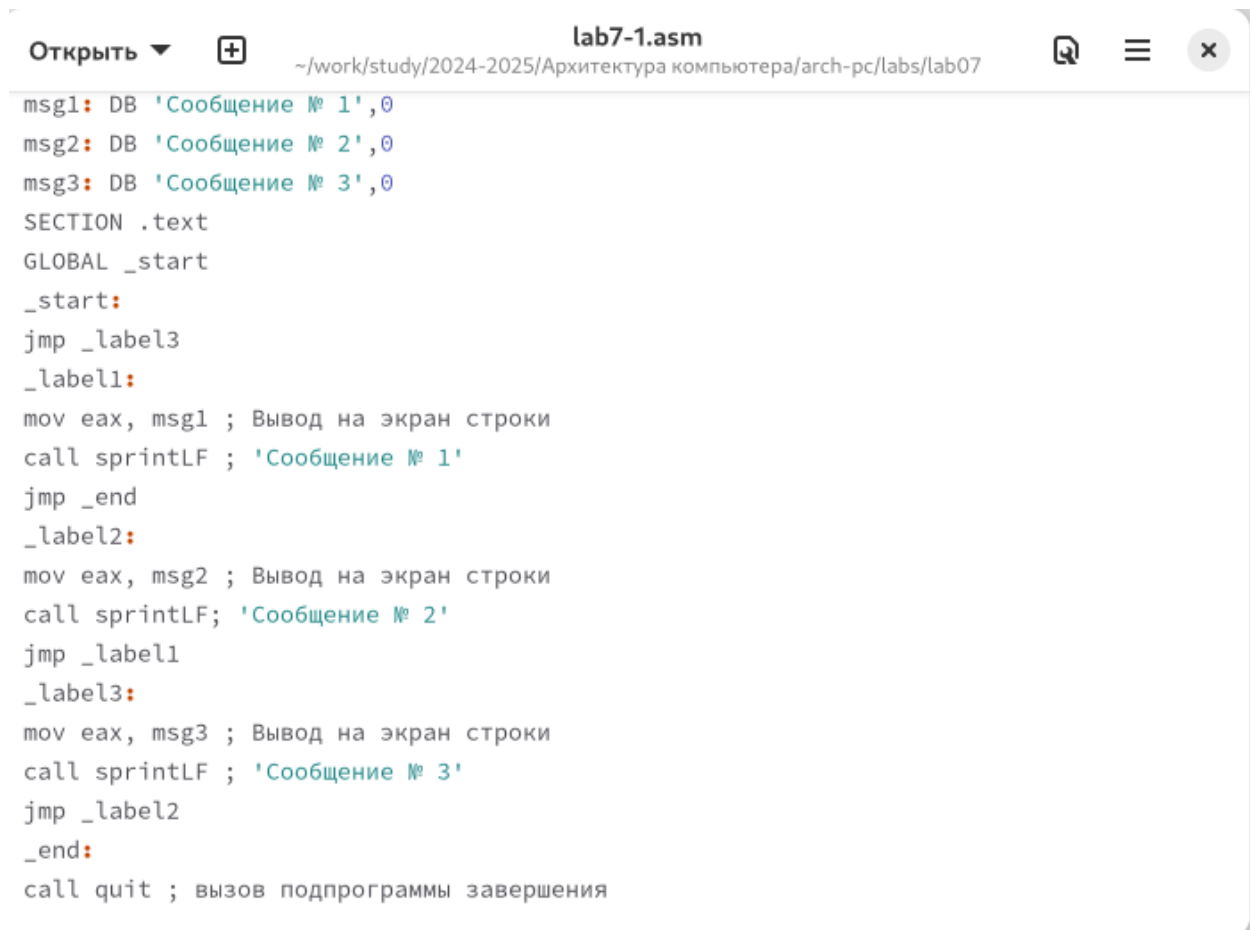
Рис. 4: Изменение текста



```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 l
ab7-1.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 5: Проверка работы программы

5)Я изменил текст программы чтобы сначала выводило сообщение 3,затем 2, затем 1.







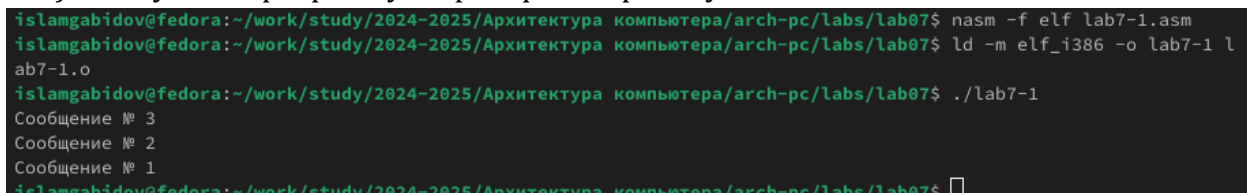
```
Открыть ▾  lab7-1.asm ~/.work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07   
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
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:
call quit ; вызов подпрограммы завершения
```

Рис. 6: Изменение текста

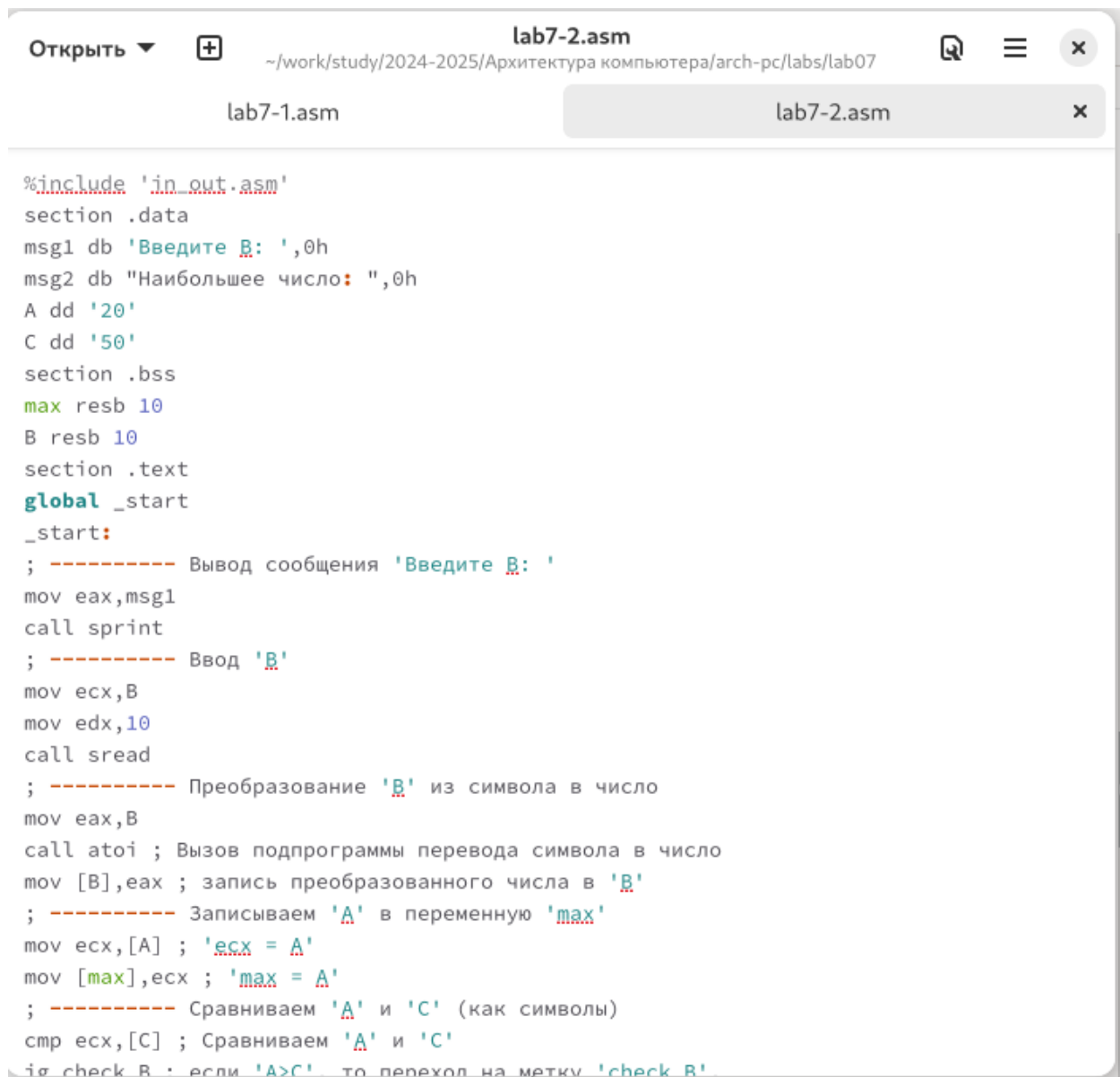
6) Запустил программу и проверил ее работу.



```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 7: Запуск программы

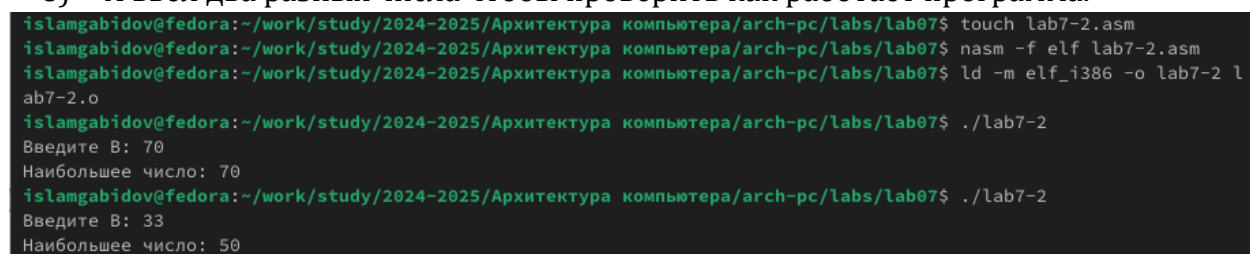
7) Я создал файл lab7-2.asm и написал текст программы.



```
%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,10
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'
; ----- Если 'A' больше 'C', то переход на метку 'check B'
```

Рис. 8: Текст программы для сравнения чисел

8) Я ввел два разных числа чтобы проверить как работает программа.



```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ touch lab7-2.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 70
Наибольшее число: 70
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 33
Наибольшее число: 50
```

Рис. 9: Программа для сравнения чисел

9) Я создал файл листинга lab7-2.lst и открыл его.

```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ mcedit lab7-2.lst
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 10: Файл листинга lab7-2.lst

10) Я понял как работает файл и какие ответы выдает

11) Эта строка находится на 24 месте, ее адрес "00000101", Машинный код - B8 [0A000000], а mov eax,B - исходный текст программы, означающий что в регистр eax мы вносим значения переменной B.

```
24 00000101 B8[0A000000]          mov eax,B
```

Рис. 11: Объяснения первой строки

2) Эта строка находится на 38 месте, ее адрес "00000134", Машинный код - E863FFFFFF, а call atoi - исходный текст программы, означающий что символ лежащий в строке выше переводится в число.

```
37 00000121 B8[00000000]          mov eax,max
38 00000134 E863FFFFFF          call atoi
```

Рис. 12: Объяснения второй строки

3) Эта строка находится на 50 месте, ее адрес "00000162", Машинный код - A1[00000000], а mov eax,[max] - исходный текст программы, означающий что число хранившееся в переменной max записывается в регистр eax.

```
49 0000015B E8B7E1F7          call sprintf
50 00000162 A1[00000000]          mov eax,[max]
```

Рис. 13: Объяснения третьей строки

11) В строке mov eax,max я убрал max и попробовал создать файл. Выдало ошибку, так как для программы нужно два операнда.

```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:34: error: invalid combination of opcode and operands
lab7-2.asm:36: error: invalid combination of opcode and operands
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 14: Создание файла без одного операнда

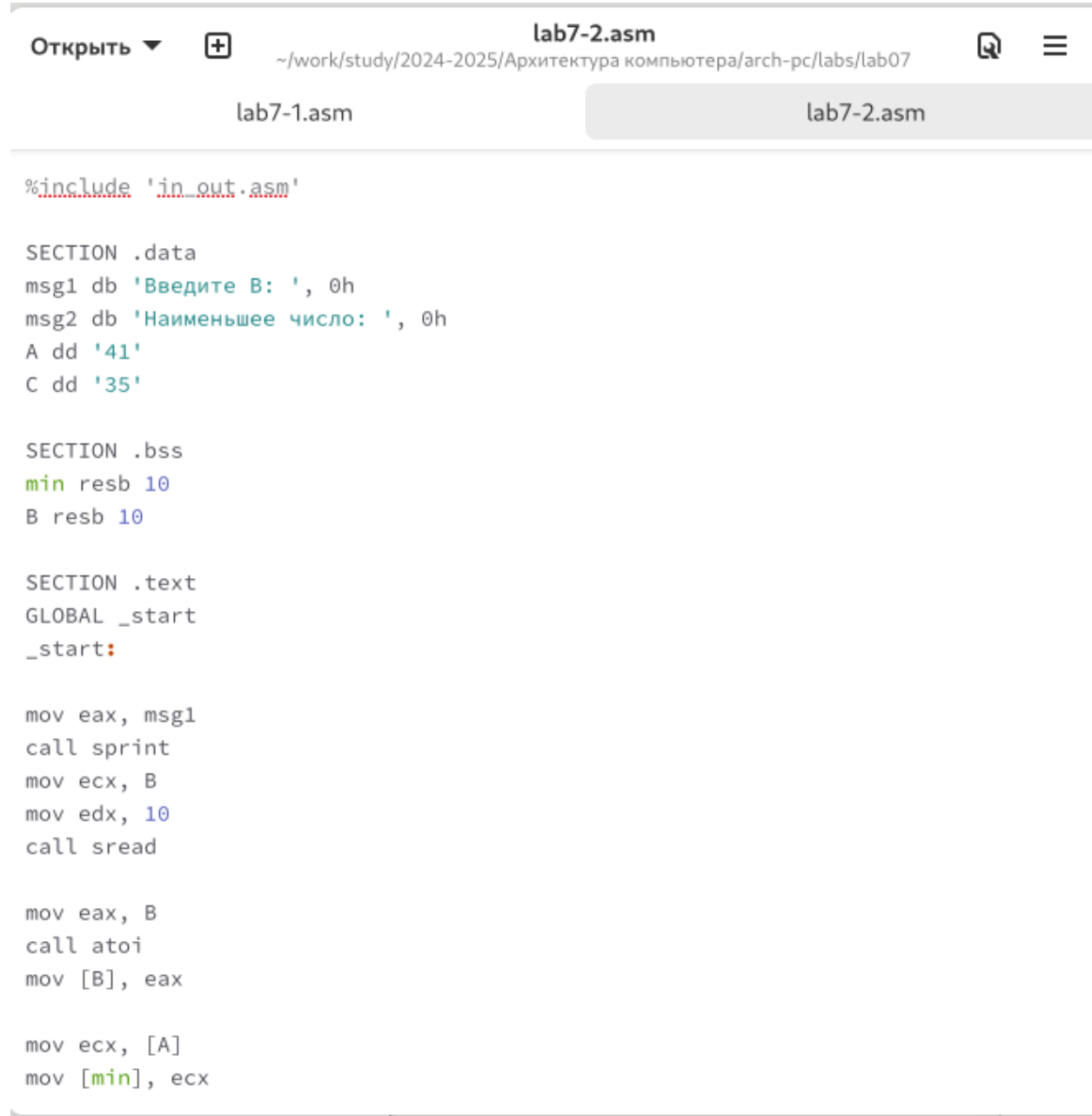
12) В файле листинга показывает где именно ошибка и с чем она связана.

```
37                                mov eax
37  *****                      error: invalid combination of opcode and operands
```

Рис. 15: Файл листинга без одного операнда

3 Самостоятельная работа.

- 1) Я написал программу для нахождения меньшего из трех чисел. Для большего удобства я сделал ввод чисел с клавиатуры. Мой вариант был десятый, поэтому числа были :41,62,35 Программа вывела меньшее из этих чисел.



```
%include 'in_out.asm'

SECTION .data
msg1 db 'Введите B: ', 0h
msg2 db 'Наименьшее число: ', 0h
A dd '41'
C dd '35'

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
```

Рис. 16: Текст программы

```

islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 62
Наименьшее число: 35
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$

```

Рис. 17: Результат работы программы

2) Я написал программу, чтобы она вычисляла выражение при введенных X и A.

Открыть ▾

+

lab7-3

~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07

🔍

☰

✕

lab7-1.asm

lab7-2.asm

lab7-3 ✕

```

%include 'in_out.asm'

SECTION .data
prim1 DB 'x-2 , x>2' ,0
prim2 DB '3a, x<=2',0
X1 DB 'Введите значение X:', 0
A1 DB 'Введите значение a:', 0
otv DB 'Ответ: ', 0

SECTION .bss
X RESB 20
A RESB 20
F RESB 20
SECTION .text
GLOBAL _start
_start:

mov eax,prim1
call sprintf
mov eax,prim2
call sprintf

mov eax,X1
call sprintf

mov ecx,X
mov edx,10
call sread

```

Рис. 18: Текст программы


```
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-3.asm
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-3
Введите значение x: 3
Введите значение a: 0
Результат: 0
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-3
Введите значение x: 1
Введите значение a: 2
Результат: 3
islamgabidov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 19: Проверка работы программы

4 Вывод

Я изучил команды условного и безусловного перехода. Приобрел навыки написания программ с переходами.