

Отчёт по лабораторной работе №7

дисциплина: Архитектура компьютера

Веретенников Дмитрий Олегович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Реализация переходов в NASM	8
4.2	Задания для самостоятельной работы	19
5	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Создание каталога и файла lab7-1.asm	8
4.2	Ввод текста в файл	9
4.3	Создание исполняемого файла и запуск программы	10
4.4	Изменение программы	11
4.5	Проверка правильности выполнения программы	12
4.6	Изменение текста программы	13
4.7	Проверка выполнения команд	14
4.8	Ввод текста программы	15
4.9	Создание исполняемого файла и проверка его работы	16
4.10	Создание файла и его просмотр	17
4.11	Удаление операнда	18
4.12	Выполнение трансляции файла	19
4.13	Программа	20
4.14	Проверка	21

Список таблиц

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

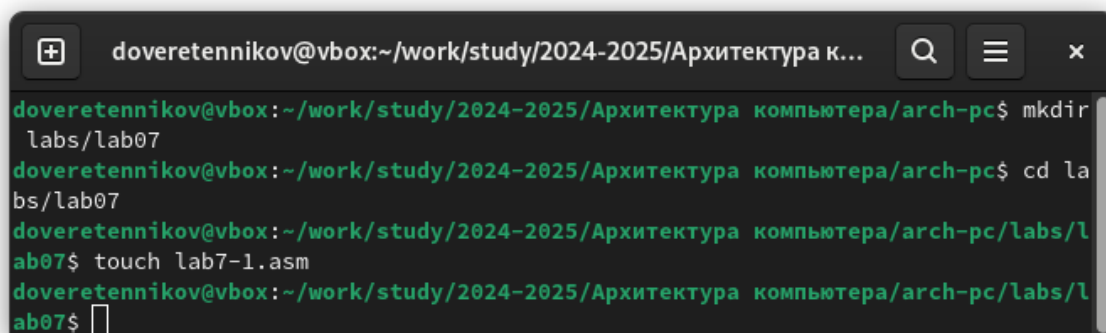
3 Теоретическое введение

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

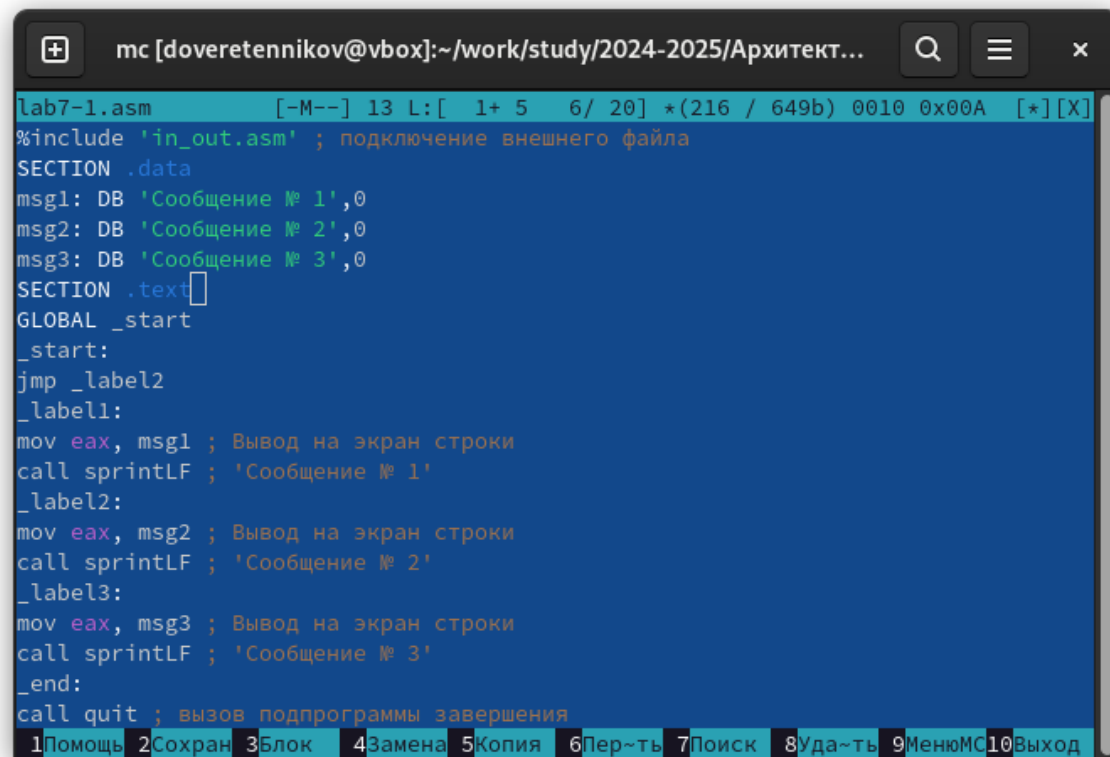
Создаю каталог для программ лабораторной работы №7, перехожу в него и создаю файл lab7-1.asm. (рис. 4.1).



```
doveretennikov@vbox:~/work/study/2024-2025/Архитектура к...
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir labs/lab07
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd labs/lab07
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ touch lab7-1.asm
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 4.1: Создание каталога и файла lab7-1.asm

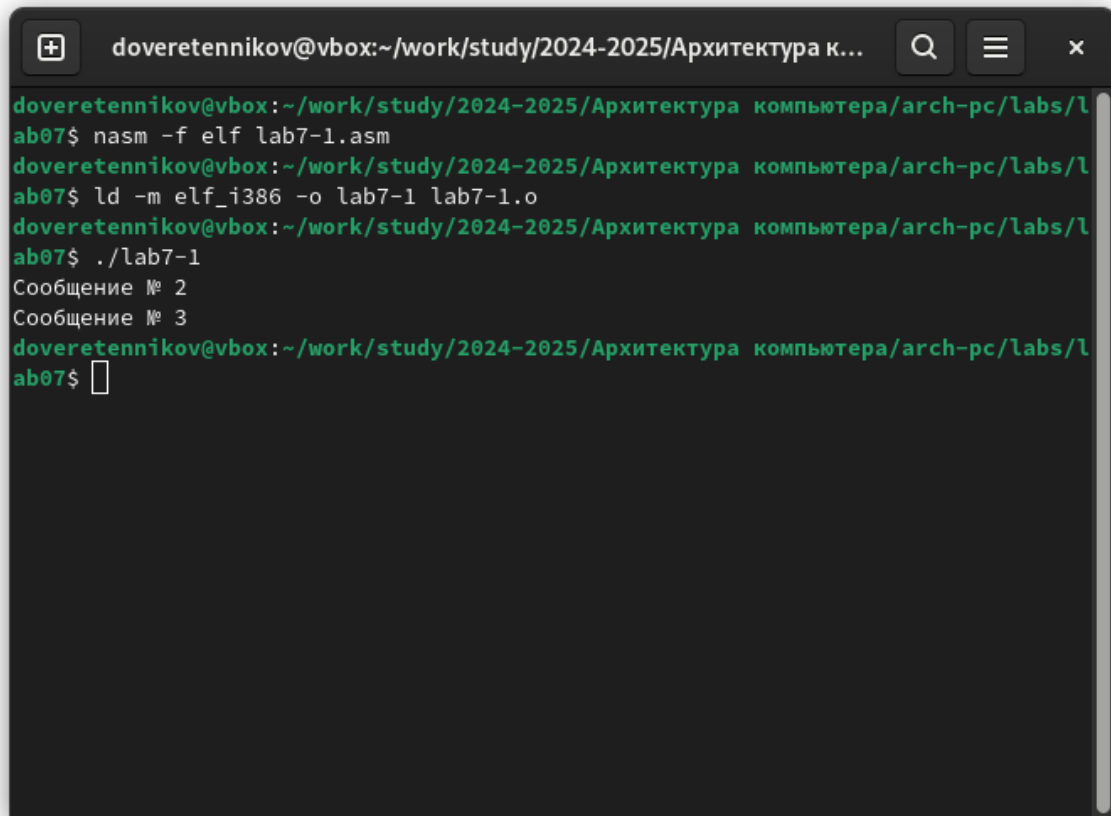
Ввожу в файл lab7-1.asm текст программы листинга 7.1. (рис. 4.2).



```
lab7-1.asm [-M--] 13 L:[ 1+ 5 6/ 20] *(216 / 649b) 0010 0x00A [*][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 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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 4.2: Ввод текста в файл

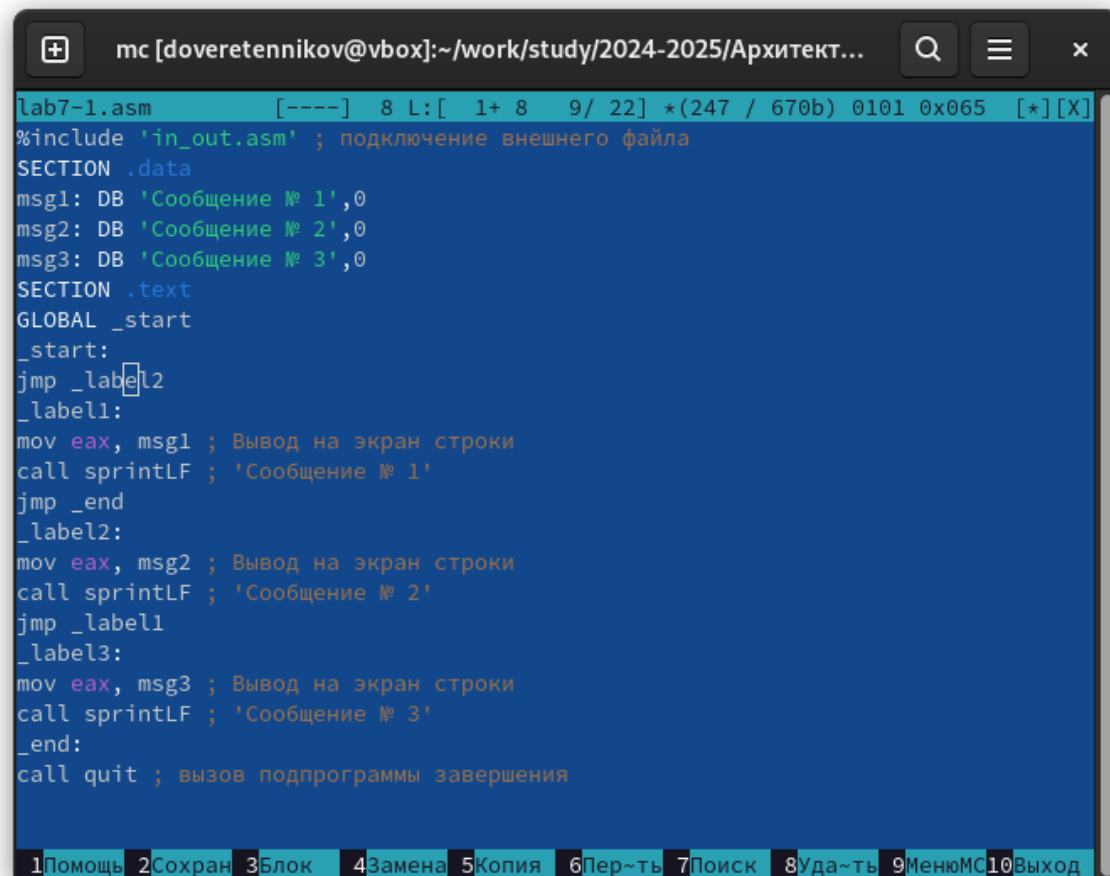
Далее создаю исполняемый файл и запускаю его. (рис. 4.3).

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

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

Рис. 4.3: Создание исполняемого файла и запуск программы

Изменяю текст программы в соответствии с листингом 7.2. (рис. 4.4).

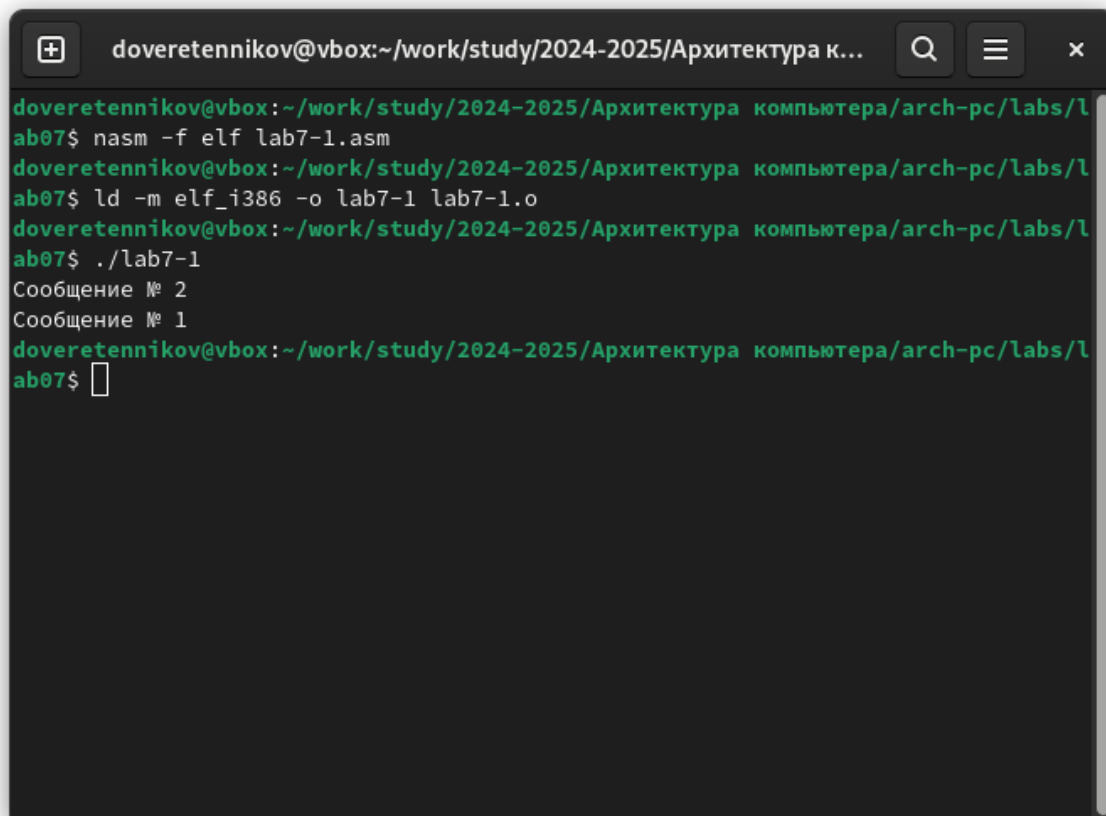


```
lab7-1.asm [----] 8 L:[ 1+ 8 9/ 22] *(247 / 670b) 0101 0x065 [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
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:
call quit ; вызов подпрограммы завершения
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход

Рис. 4.4: Изменение программы

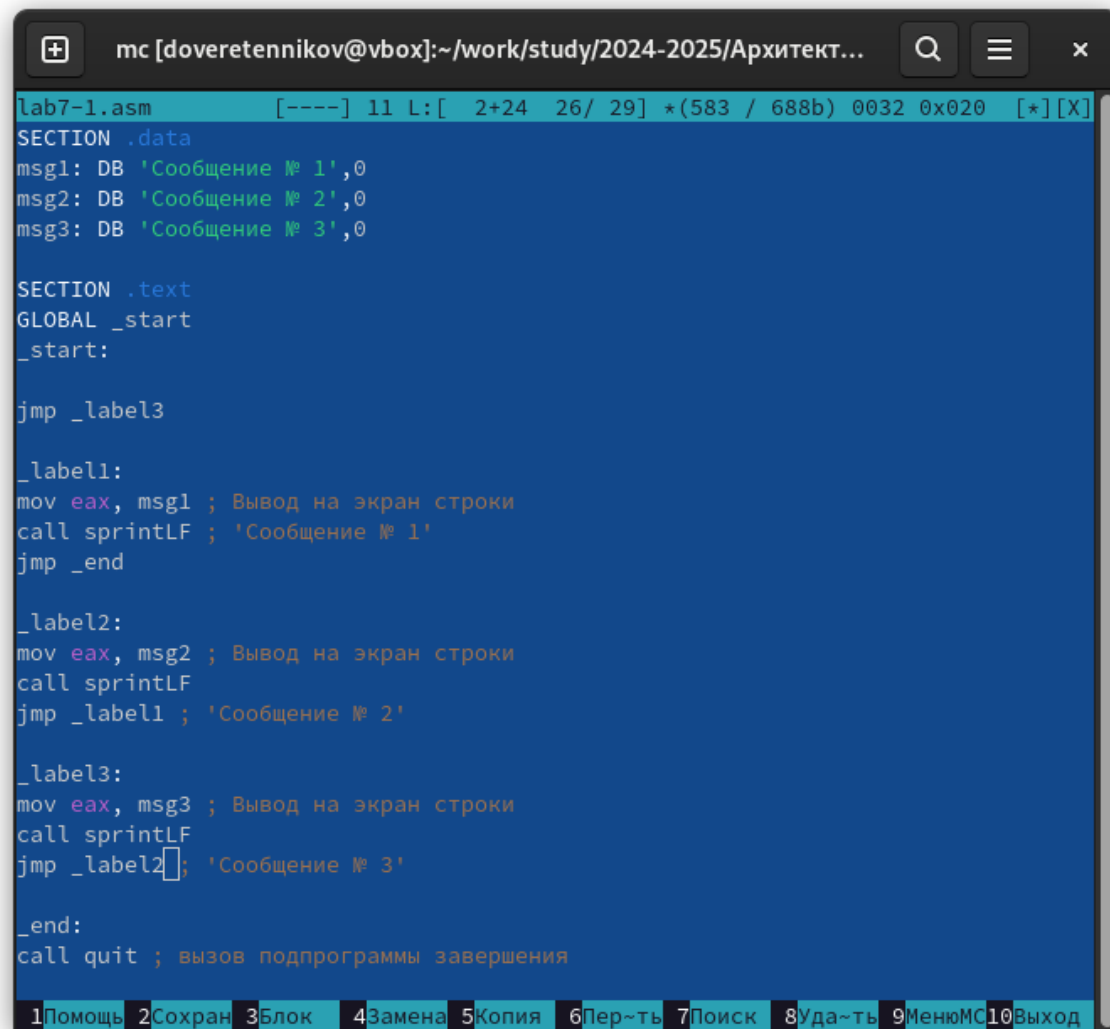
Проверяю правильность выполнения программы. (рис. 4.5).

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

```
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ nasm -f elf lab7-1.asm
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$
```

Рис. 4.5: Проверка правильности выполнения программы

Изменяю текст программы таким образом, чтобы три сообщения выводились в обратном порядке. (рис. 4.6).



```
lab7-1.asm [----] 11 L: [ 2+24 26/ 29] *(583 / 688b) 0032 0x020 [*][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
jmp _label1 ; 'Сообщение № 2'

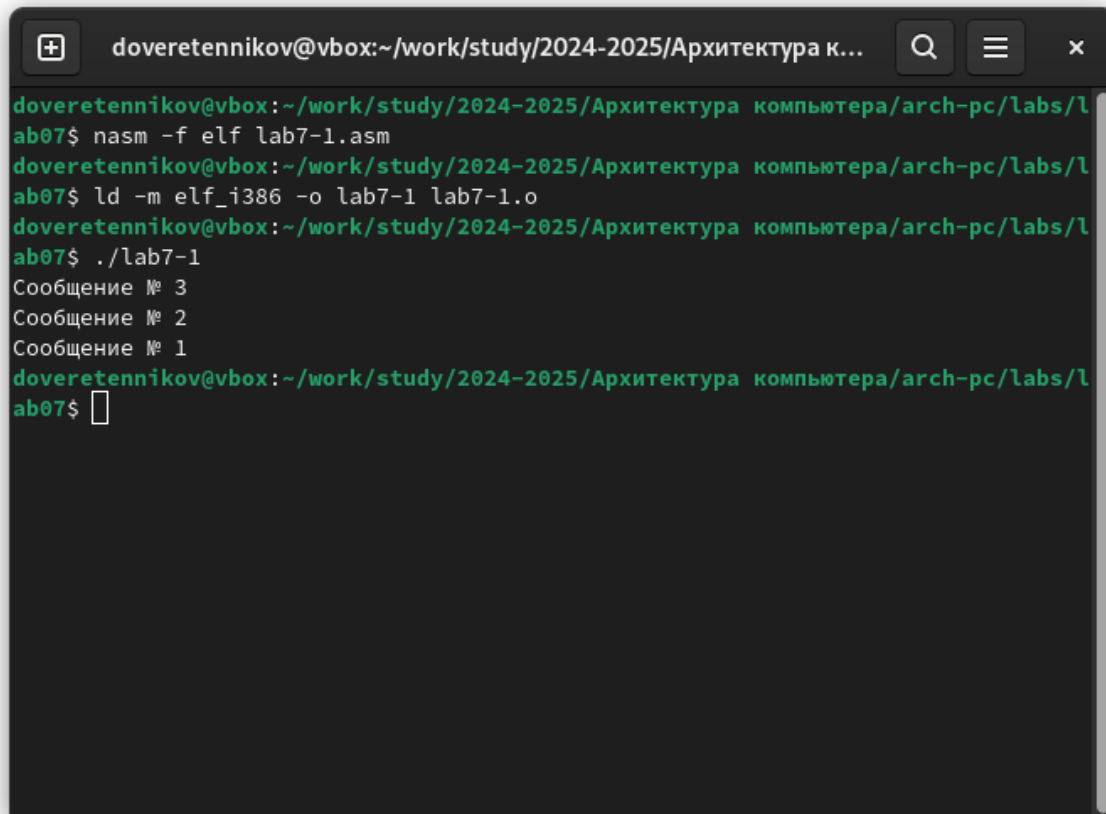
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf
jmp _label2 ; 'Сообщение № 3'

_end:
call quit ; вызов подпрограммы завершения

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 4.6: Изменение текста программы

Проверяю правильность выполнения команд. (рис. 4.7).

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

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

Рис. 4.7: Проверка выполнения команд

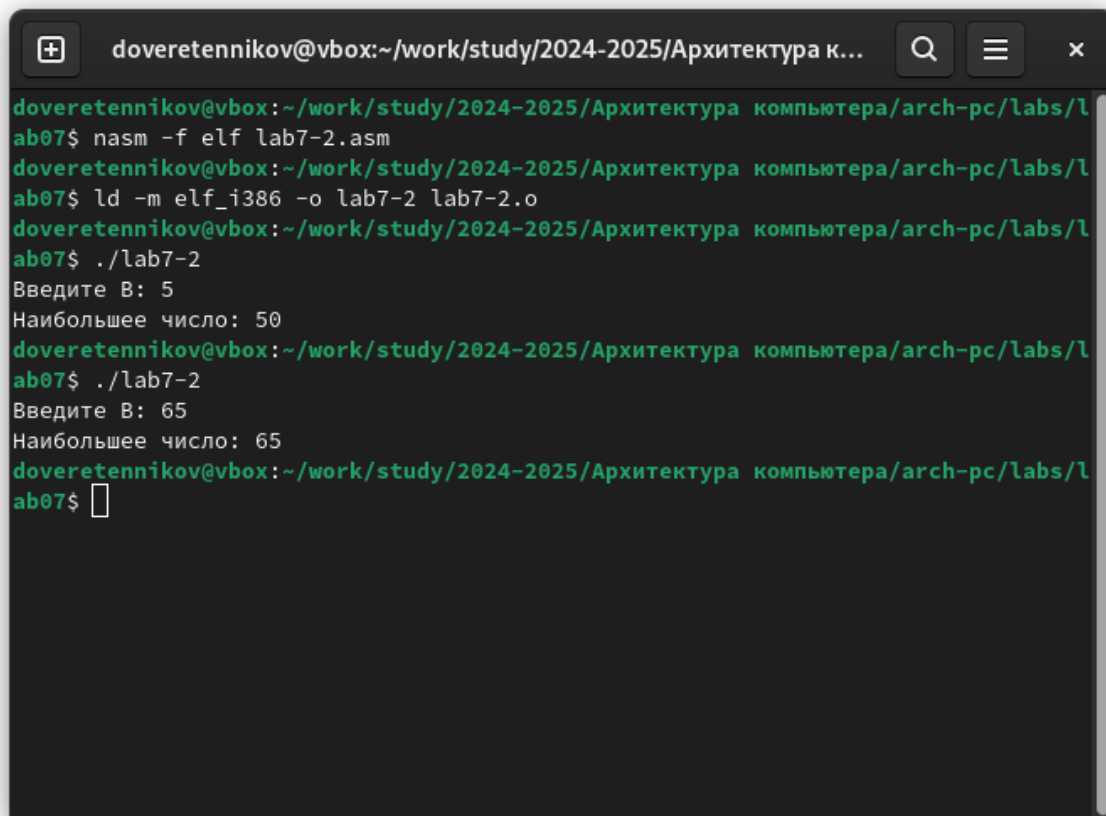
Создаю файл lab7-2.asm и ввожу в него текст программы из листинга 7.3. (рис. 4.8).

```
lab7-2.asm [-M--] 31 L: [ 1+ 3 4/ 49] *(114 /1743b) 0010 0x00A [*] [X]
#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'
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход

Рис. 4.8: Ввод текста программы

Создаю исполняемый файл и проверяю его работу для разных значений В. (рис. 4.9).

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

```
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 5
Наибольшее число: 50
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 65
Наибольшее число: 65
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 4.9: Создание исполняемого файла и проверка его работы

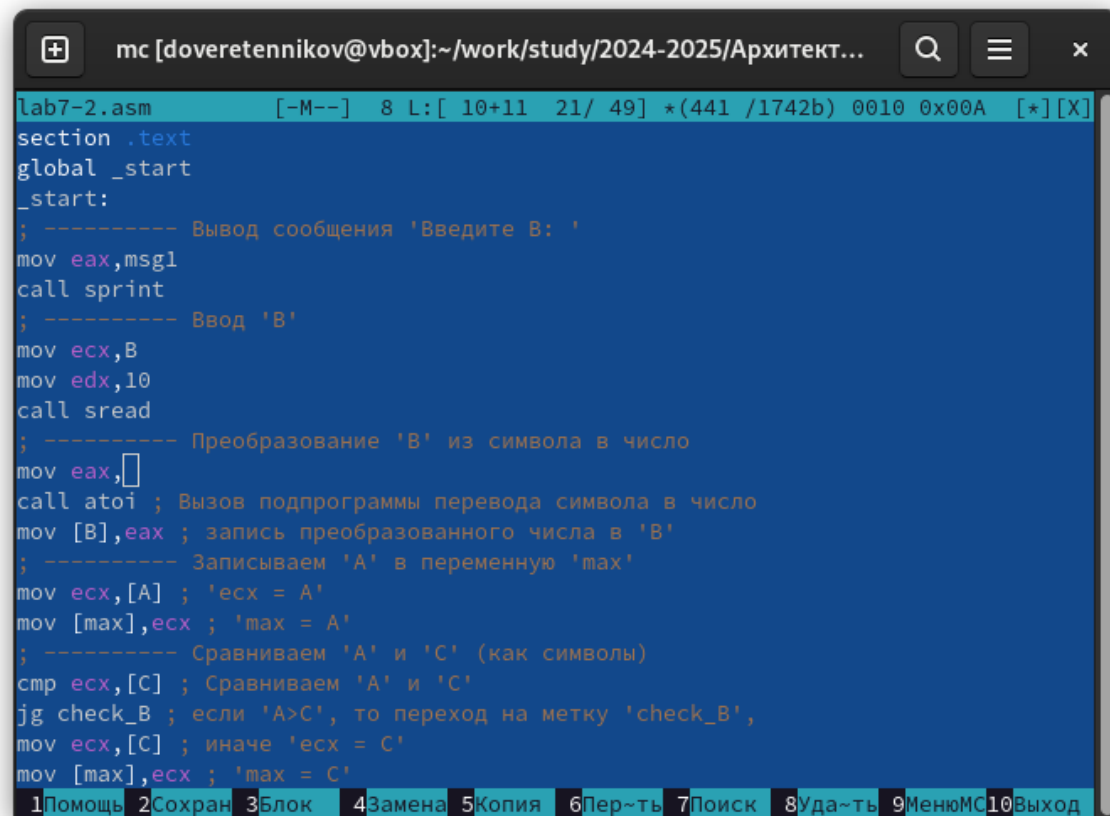
Создаю файл листинга для программы из файла lab7-2.asm и открываю его.
(рис. 4.10).


```
lab7-2.lst [----] 0 L: [ 1+ 0 1/226] *(0 /14548b) 0032 0x020 [*][X]
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
25 00000011 53                        <1> push    ebx
26 00000012 50                        <1> push    eax
27 00000013 E8E8FFFFFF                <1> call    slen

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 4.10: Создание файла и его просмотр

Открываю файл с программой lab7-2.asm и в любой инструкции с двумя операндами удаляю один операнд. (рис. 4.11).

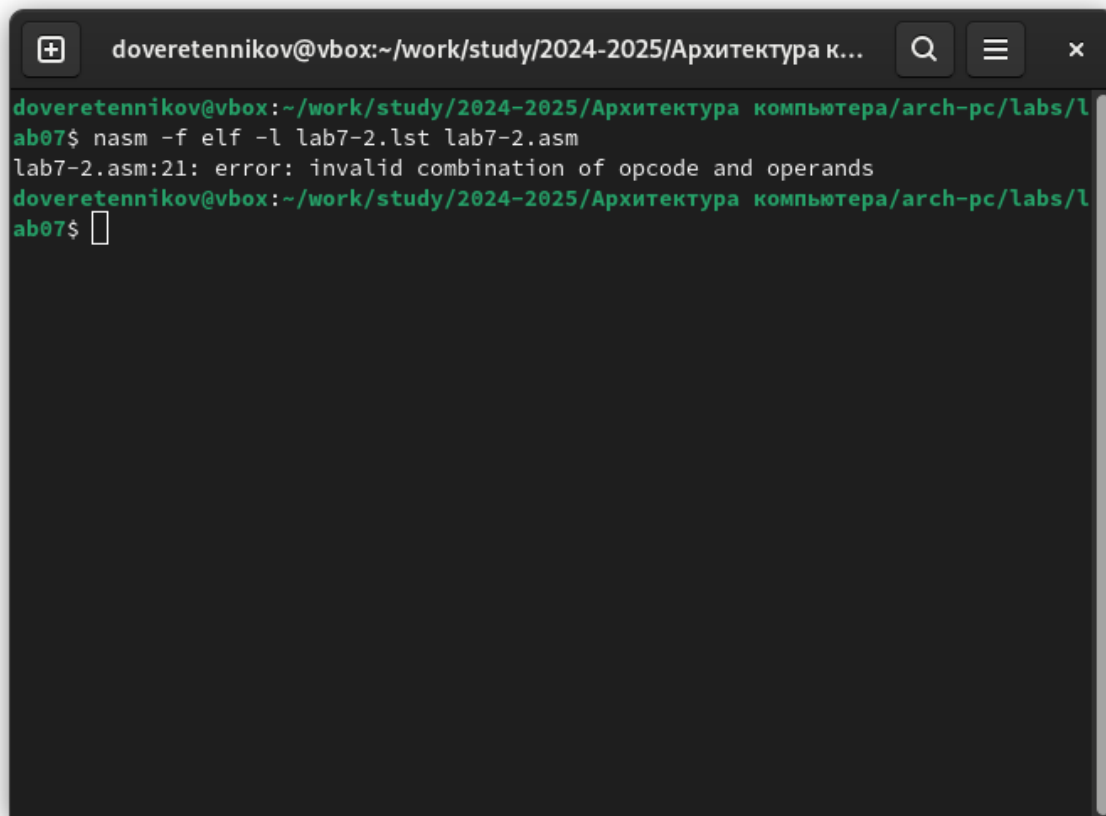


```
lab7-2.asm [-M--] 8 L: [ 10+11 21/ 49] *(441 /1742b) 0010 0x00A [*][X]
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,[]
call atoi ; Вызов подпрограммы 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',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход

Рис. 4.11: Удаление операнда

Пытаюсь выполнить трансляцию файла листинга и получаю ошибку. (рис. 4.12).

A terminal window with a dark background. The title bar shows the user 'doveretennikov@vbox' and the path '~/work/study/2024-2025/Архитектура к...'. The terminal content shows a command 'nasm -f elf -l lab7-2.lst lab7-2.asm' being executed. The output is an error message: 'lab7-2.asm:21: error: invalid combination of opcode and operands'. The prompt 'doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07\$' is visible at the bottom.

```
doveretennikov@vbox:~/work/study/2024-2025/Архитектура к...
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:21: error: invalid combination of opcode and operands
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab07$
```

Рис. 4.12: Выполнение трансляции файла

4.2 Задания для самостоятельной работы

Пишу программу нахождения наименьшей из 3 целочисленных переменных `a`, `b` и `c`. (рис. 4.13).

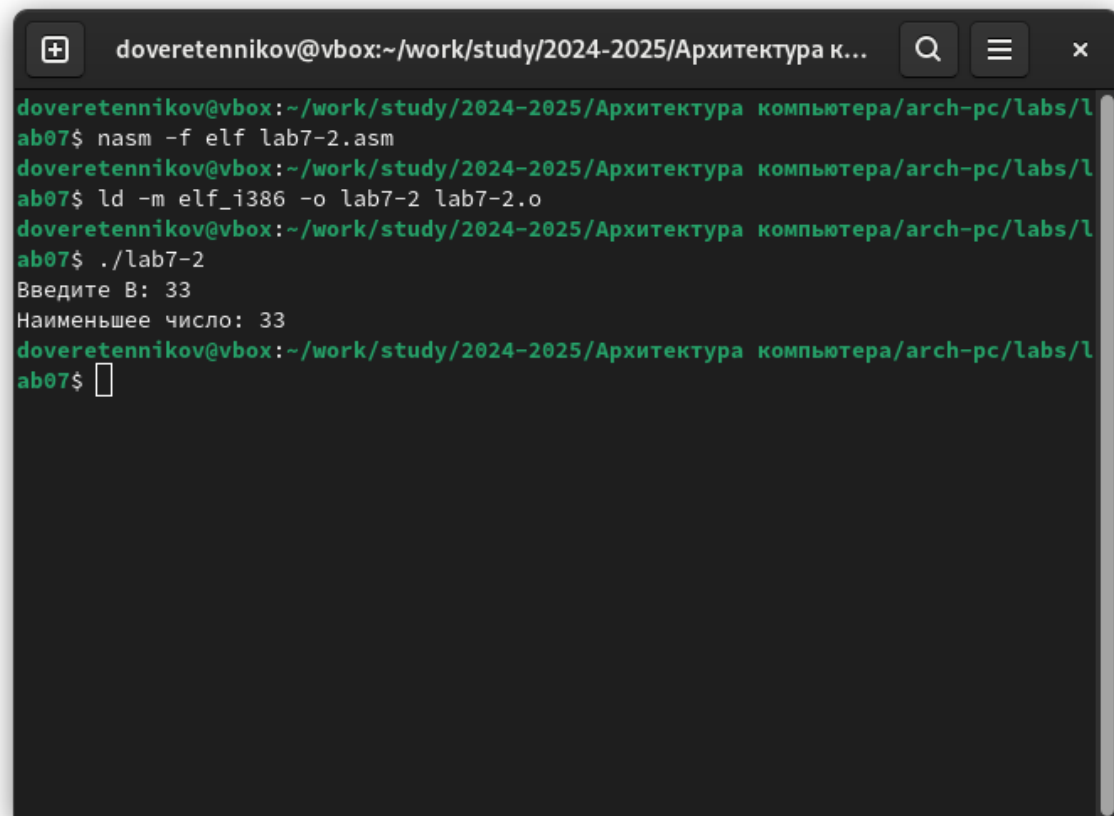
```
lab7-2.asm [----] 0 L:[ 3+ 4 7/ 51] *(135 /1745b) 0010 0x00A [*][X]
msg1 db 'Введите B: ',0h
msg2 db "Наименьшее число: ",0h
A dd '52'
C dd '40'
section .bss
min resb 10
B resb 10

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 [min],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 [min],ecx ; 'max = C'
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,min
call atoi ; Вызов подпрограммы перевода символа в число
mov [min],eax ; запись преобразованного числа в 'max'
; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
mov ecx,[min]
cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
jb fin ; если 'max(A,C)>B', то переход на 'fin',
mov ecx,[B] ; иначе 'ecx = B'
mov [min],ecx
; ----- Вывод результата
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 4.13: Программа

Проверяю правильность работы программы взяв числа из 8 варианта. (рис.

4.14).



```
doveretennikov@vbox:~/work/study/2024-2025/Архитектура к...
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ nasm -f elf lab7-2.asm
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$ ./lab7-2
Введите В: 33
Наименьшее число: 33
doveretennikov@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/l
ab07$
```

Рис. 4.14: Проверка

После этого отправляю файлы на github

5 Выводы

После выполнения 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. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
 17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
 18. — 1120 с. — (Классика Computer Science).