**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: Томилова Валентина

Группа: НКАбд-06-25

### МОСКВА

2025 г.

Оглавление

[Список таблиц 1 Цель работы 4](#_bookmark0)

1. [​Задания 6](#_bookmark1)
2. [​Теоретическое введение 7](#_bookmark2)
3. [​Выполнение работы 8](#_bookmark3)

[Реализация переходов в NASM 8](#_bookmark4)

[Изучение структуры файлы листинга 13](#_bookmark23)

[Объяснение строк листинга 13](#_bookmark27)

[Задание для самостоятельной работы 15](#_bookmark31)

1. [​Выводы 19](#_TOC_250001)

[Список литературы 20](#_TOC_250000)

Список изображений

[Рисунок 0.1 (создание каталога) 7](#_bookmark5)

[Рисунок 0.2 (переход в каталог) 7](#_bookmark6)

[Рисунок 0.3 (создаем файл) 7](#_bookmark7)

[Рисунок 0.4 (текст программы) 7](#_bookmark8)

[Рисунок 0.5 (создание исполняемого файла) 8](#_bookmark9)

[Рисунок 0.6 (запуск исполняемого файла) 8](#_bookmark10)

[Рисунок 0.7 (результат работы файла) 8](#_bookmark11)

[Рисунок 0.8 (измененный текст программы) 8](#_bookmark12)

[Рисунок 0.9 (создание и запуск исполняемого файла) 9](#_bookmark13)

[Рисунок 0.10 (измененный текст программы) 9](#_bookmark14)

[Рисунок 0.11 (создание исполняемого файла) 9](#_bookmark15)

[Рисунок 0.12(запуск исполняемого файла) 10](#_bookmark16)

[Рисунок 0.13(создание файла) 10](#_bookmark17)

[Рисунок 0.14 (текст программы) 10](#_bookmark18)

[Рисунок 0.15 (текст программы) 11](#_bookmark19)

[Рисунок 0.16 (создание исполняемого файла и проверка работы) 11](#_bookmark20)

[Рисунок 0.17(создание исполняемого файла и проверка работы) 11](#_bookmark21)

[Рисунок 0.18(создание исполняемого файла и проверка работы) 11](#_bookmark22)

[Рисунок 0.19 (создание файла листинга) 12](#_bookmark24)

[Рисунок 0.20 (открытие листинга) 12](#_bookmark25)

[Рисунок 0.21 (открытие листинга) 12](#_bookmark26)

[Рисунок 0.22 (текст программы с удалением операнды) 13](#_bookmark28)

[Рисунок 0.23 (создание файла листинга) 13](#_bookmark29)

[Рисунок 0.24 (открытие листинга) 13](#_bookmark30)

[Рисунок 0.25 (текст программы) 14](#_bookmark32)

[Рисунок 0.26 (текст программы) 14](#_bookmark33)

[Рисунок 0.27 (текст программы) 14](#_bookmark34)

[Рисунок 0.28 (создание исполняемого файла и проверка его работы) 15](#_bookmark35)

Рисунок 0.29 (текст программы) 15

Рисунок 0.30 (текст программы) 16

Рисунок 0.31 (создание исполняемого файла) 16

Рисунок 0.32 (создание исполняемого файла) 16

Рисунок 0.33 (запуск программы) 16

Рисунок 0.34 (проверка результатов) 16

Рисунок 0.35 (проверка результатов) 16

### 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.

# Задания

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

# Теоретическое введение

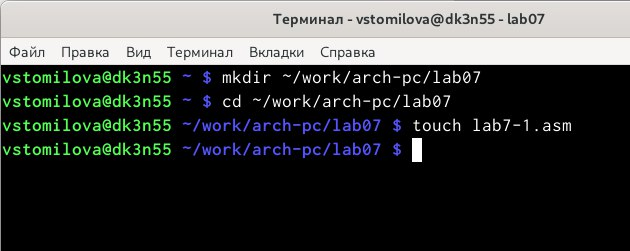
Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

* 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.

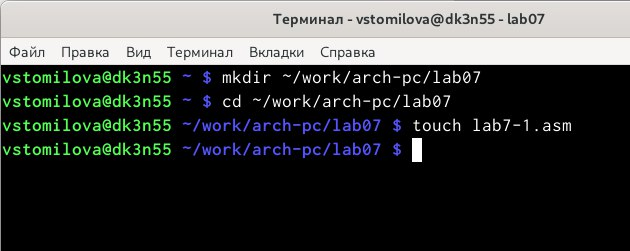
# Выполнение работы

## Реализация переходов в NASM

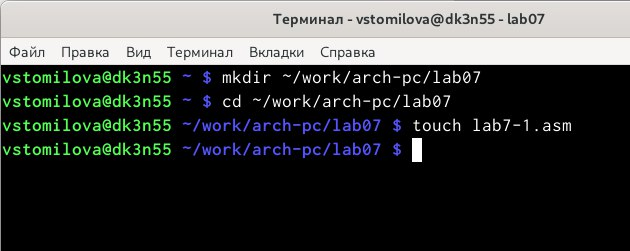
Создадим каталог для программам лабораторной работы № 7, перейдем в него и создадим файл lab7-1.asm (рисунок 0.1-0.3)



**Рисунок 0.1 (создание каталога)**

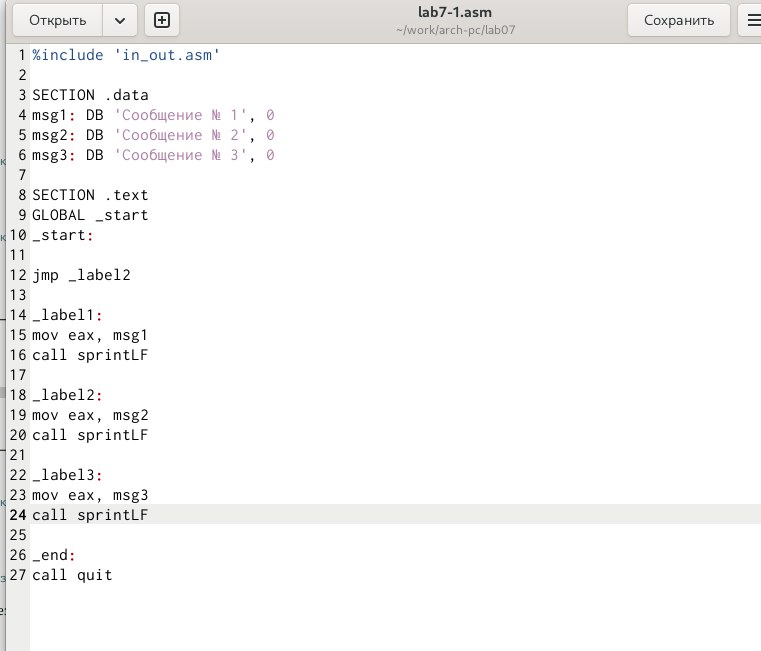


**Рисунок 0.2 (переход в каталог)**



**Рисунок 0.3 (создаем файл)**

Введем в файл lab7-1.asm текст программы из листинга (рисунок 0.4)

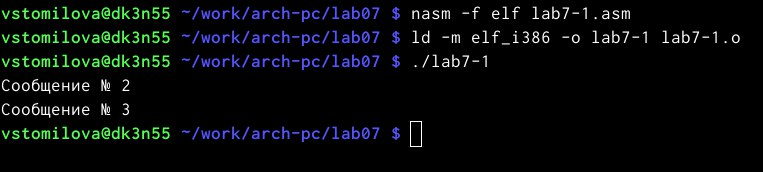


**Рисунок 0.4 (текст программы)**

Создадим исполняемый файл и запустим его (рисунок 0.5-0.6)

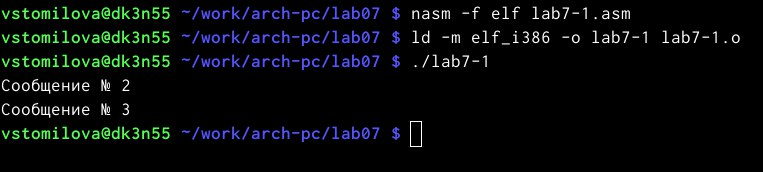


**Рисунок 0.5 (создание исполняемого файла)**



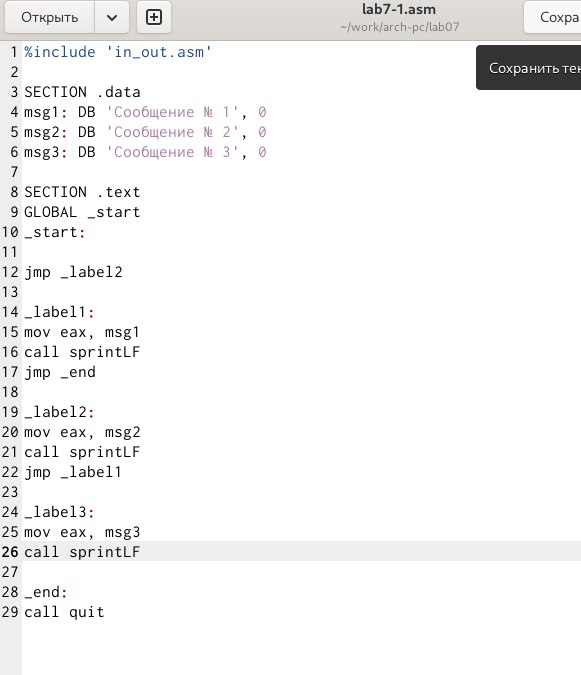
**Рисунок 0.6 (запуск исполняемого файла)**

Результат работы данной программы будет следующим (рисунок 0.7)



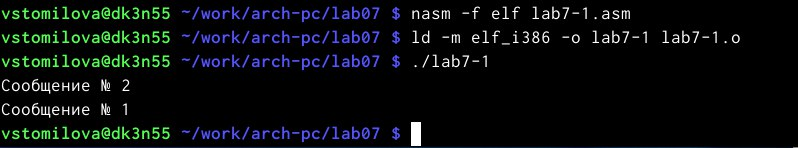
**Рисунок 0.7 (результат работы файла)**

Изменим текст программы в соответствии с листингом (рисунок 0.8)



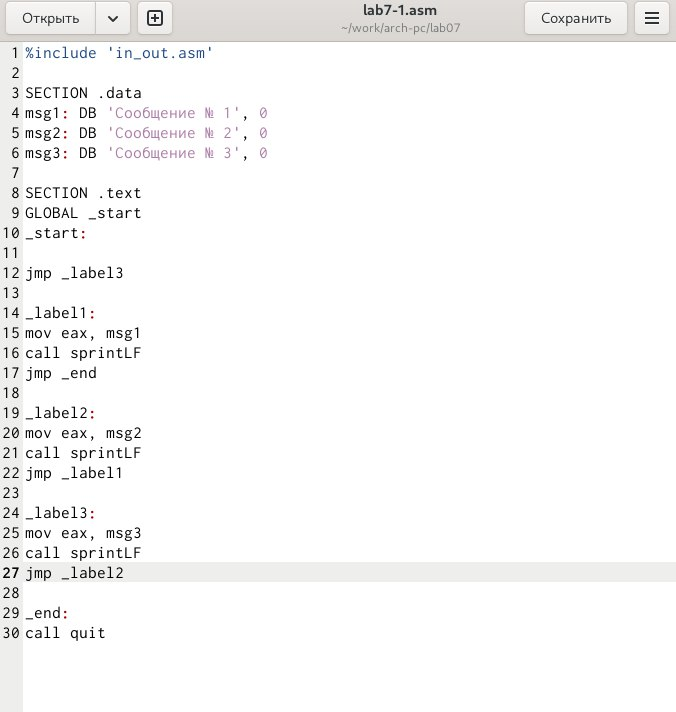
**Рисунок 0.8 (измененный текст программы)**

Создадим исполняемый файл и проверим его работу (рисунок 0.9)



**Рисунок 0.9 (создание и запуск исполняемого файла)**

Изменим текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим (рисунок 0.10)



**Рисунок 0.10 (измененный текст программы)**



**Рисунок 0.11 (создание исполняемого файла)**

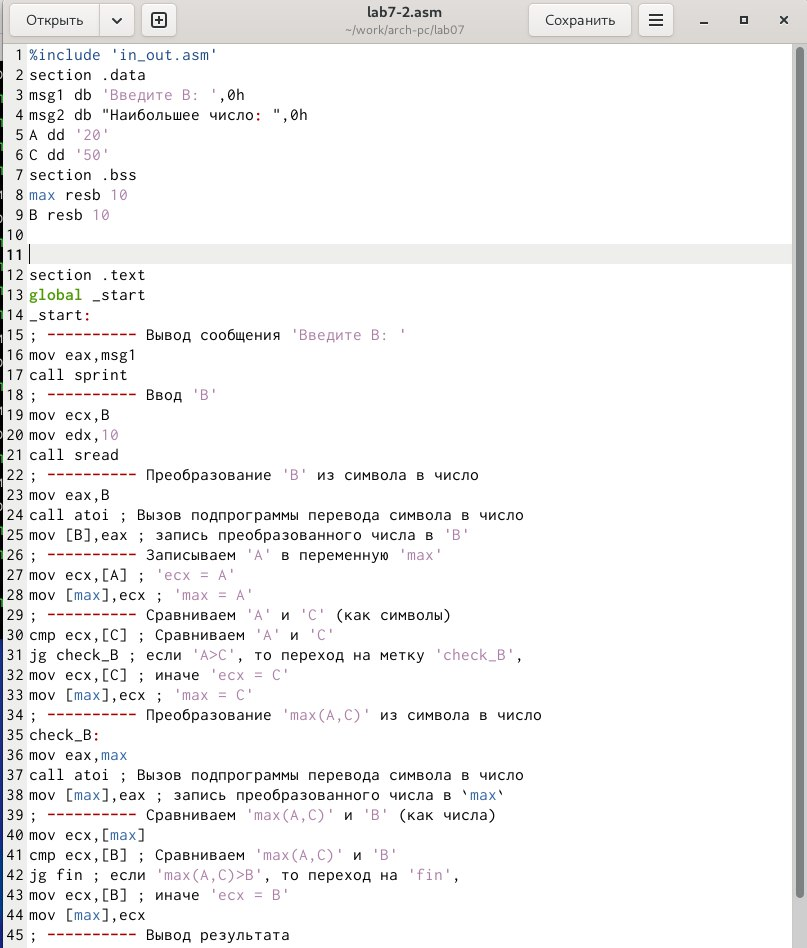


**Рисунок 0.12(запуск исполняемого файла)**

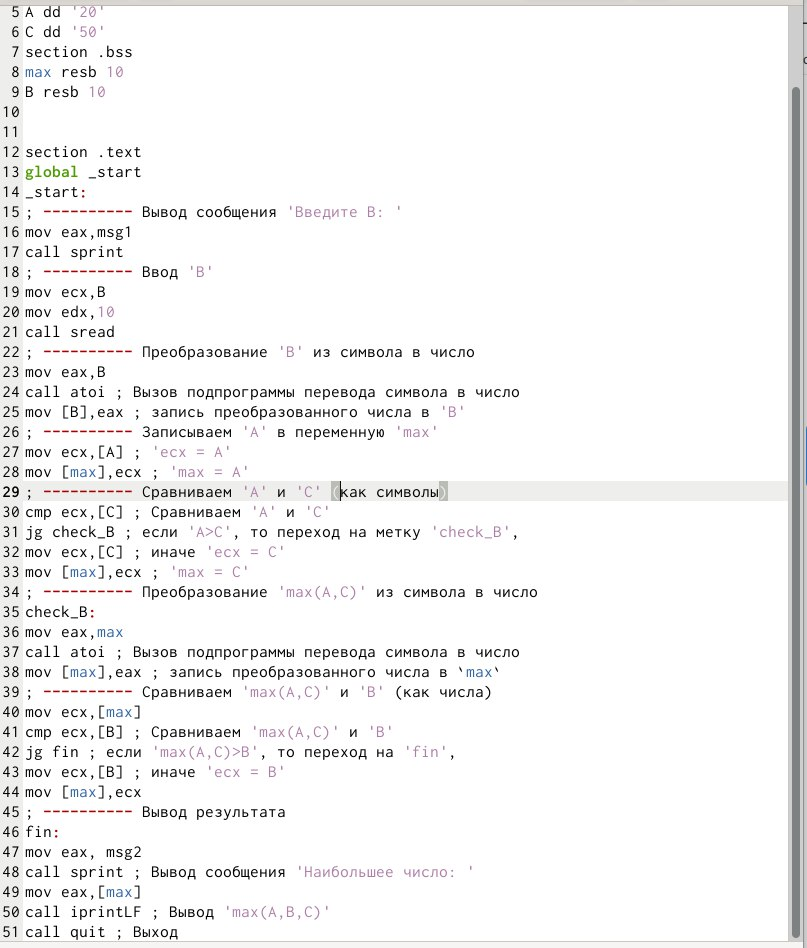
Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучим текст программы из листинга 7.3 и введем в lab7-2.asm (рисунок 0.13-0.15)



**Рисунок 0.13(создание файла)**

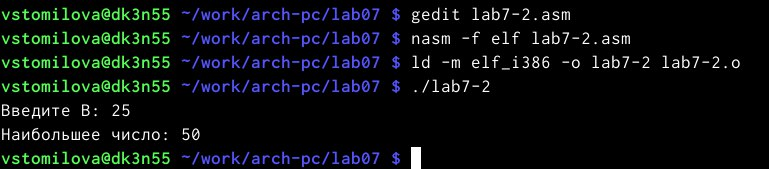
****

**Рисунок 0.14 (текст программы)**

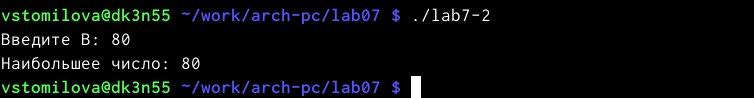


**Рисунок 0.15 (текст программы)**

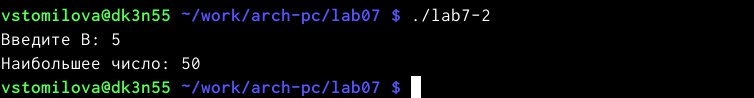
Создадим исполняемый файл и проверим его работу для разных значений B (рисунок 0.16-0.18)



**Рисунок 0.16 (создание исполняемого файла и проверка работы)**



**Рисунок 0.17(создание исполняемого файла и проверка работы)**



**Рисунок 0.18(создание исполняемого файла и проверка работы)**

## Изучение структуры файлы листинга

Создадим файл листинга для программы из файла lab7-2.asm (рисунок 0.19-

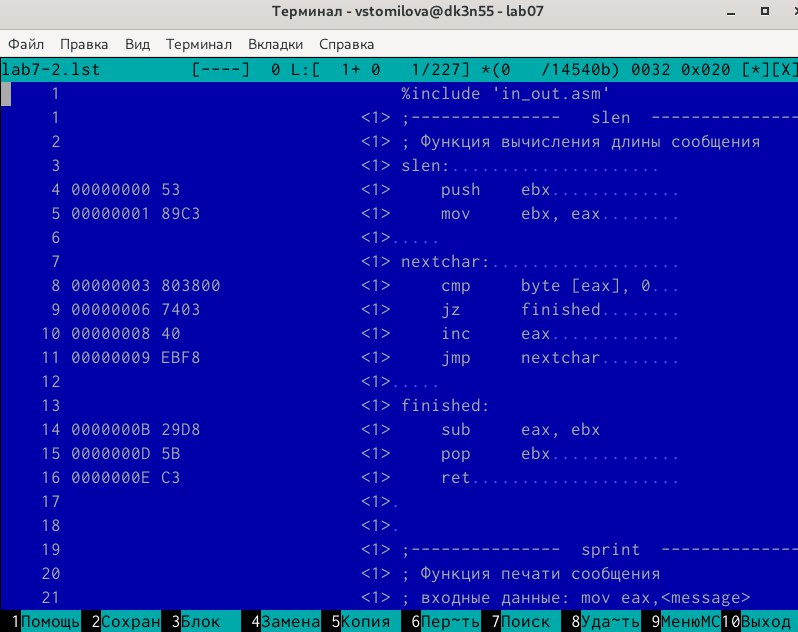
0.21)



**Рисунок 0.19 (создание файла листинга)**



**Рисунок 0.20 (открытие листинга)**



**Рисунок 0.21 (открытие листинга)**

## Объяснение строк листинга:

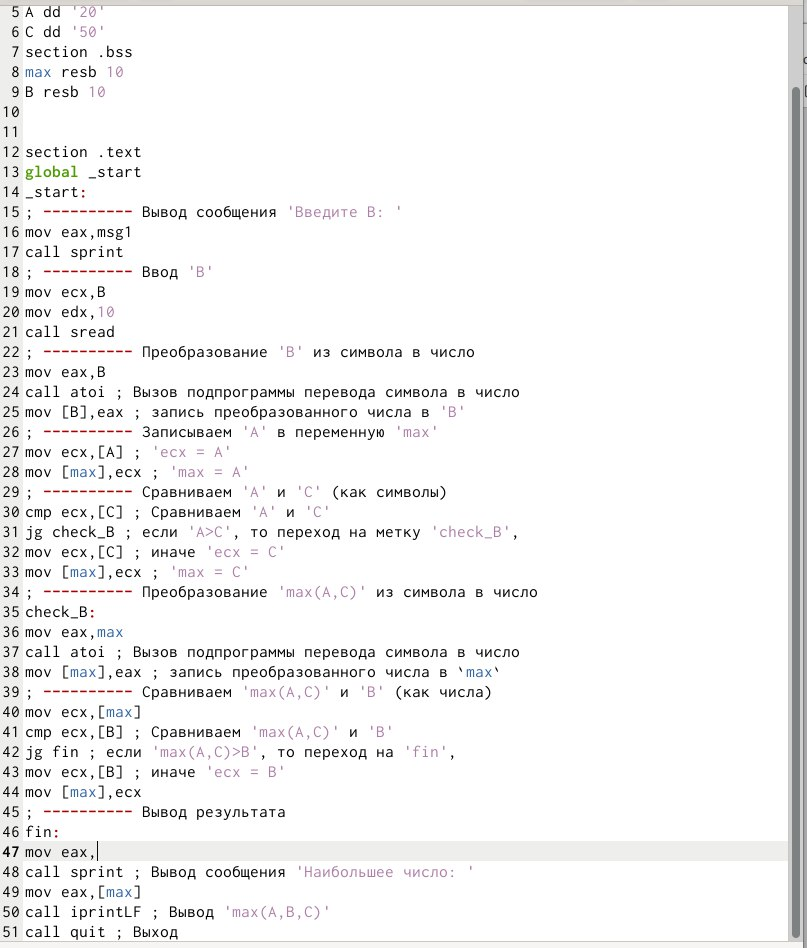
Самое первое значение в файле листинга – номер строки—это номер строки файла листинга (нужно помнить, что номер строки в файле листинга может не соответствовать номеру строки в файле с исходным текстом программы)

Второе вхождение- адрес—это смещение машинного кода от начала текущего сегмента

Далее идут машинный код, представляющий собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, и исходный текст программы — это просто строка исходной программы вместе с комментариями

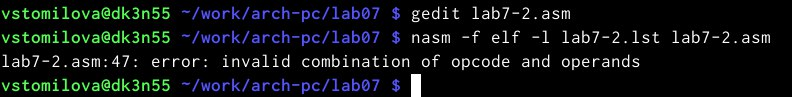
Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами

удалим один операнд. Выполним трансляцию с получением файла листинга (рисунок 0.22-0.24)

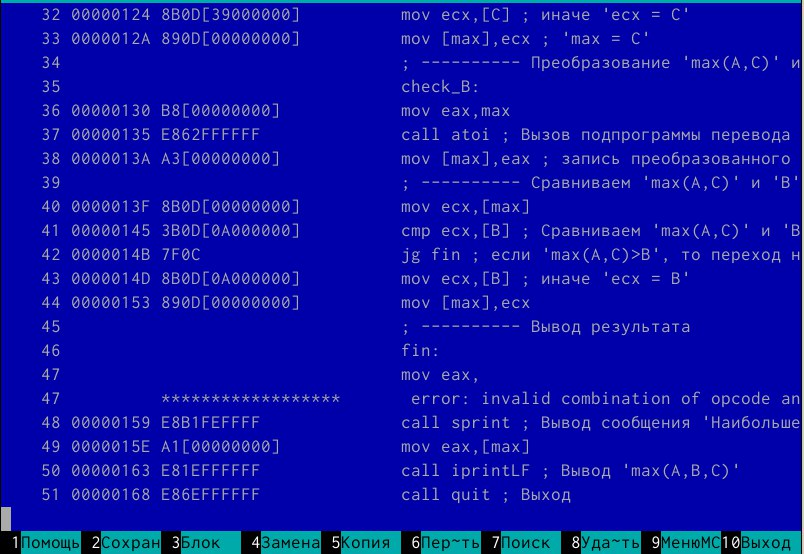


**Рисунок 0.22 (текст программы с удалением операнды)**

В 47 строке удалим один операнд



**Рисунок 0.23 (создание файла листинга)**



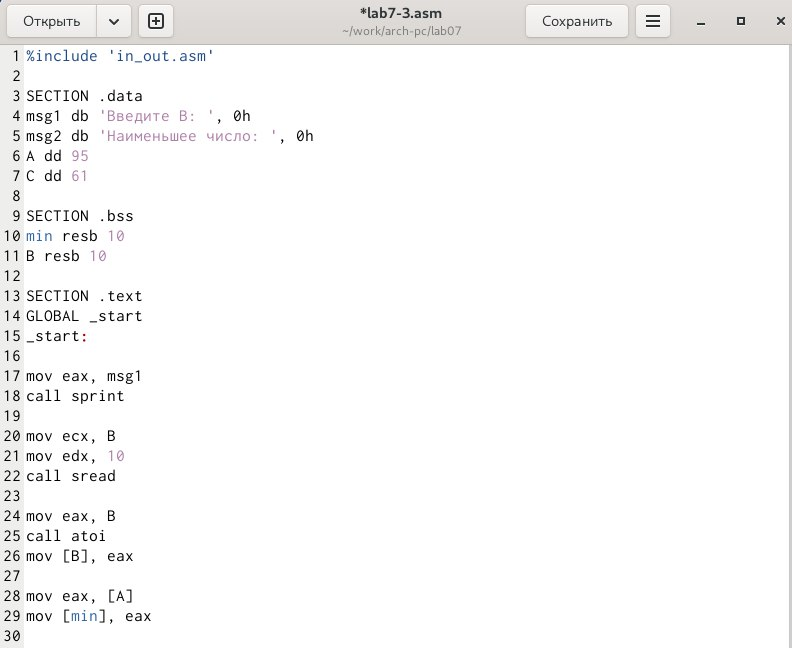
**Рисунок 0.24 (открытие листинга)**

В новом файле листинга показывает ошибку, возникшую при попытке трансляции файла. При этом никакие выходные файлы, помимо файла листинга, не появляются

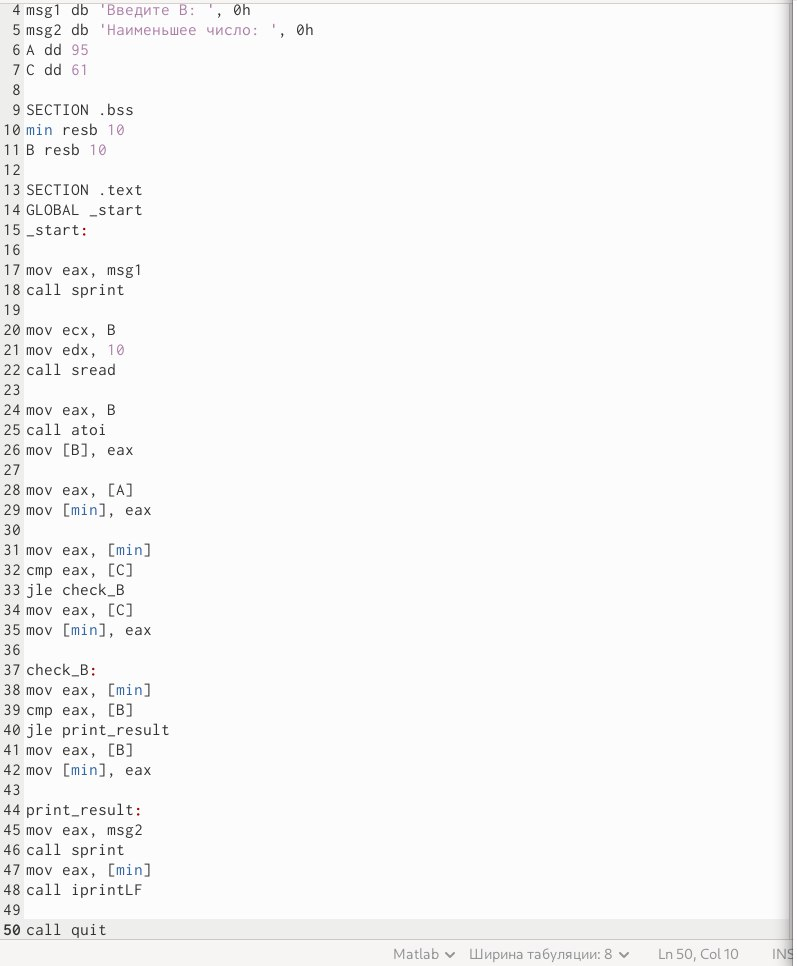
## Задание для самостоятельной работы

Напишем программу нахождения наименьшей из 3 целочисленных

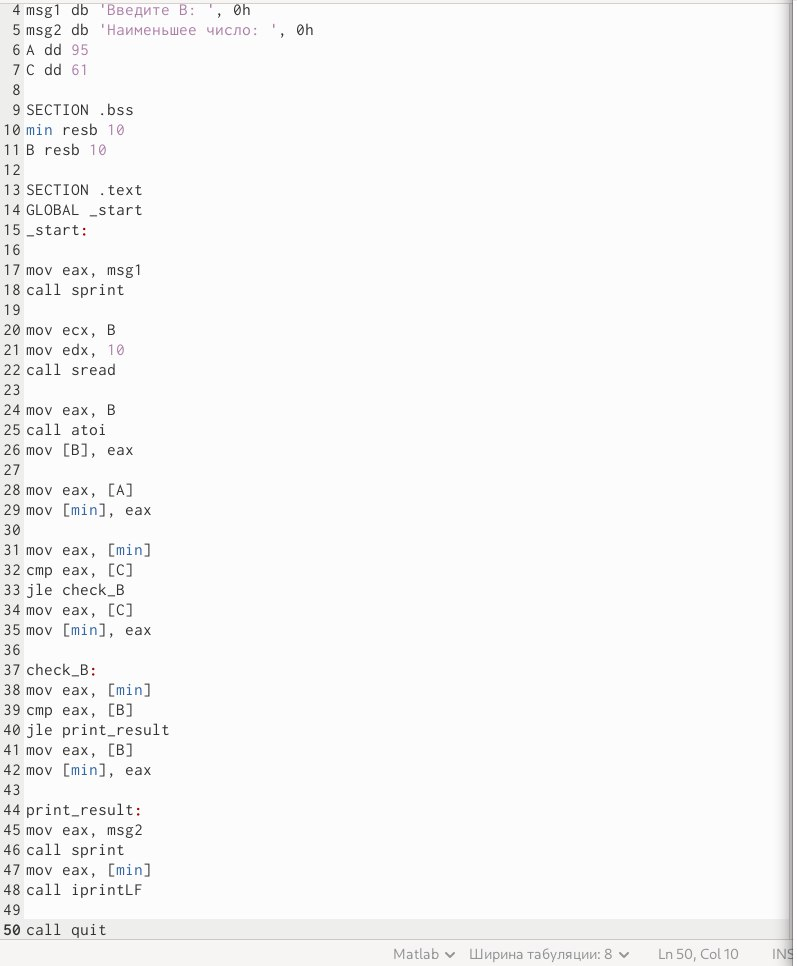
переменных 𝑎, 𝑏 и с. Значения переменных 95,2,61 (рисунок 0.25-0.27)



**Рисунок 0.25 (текст программы)**

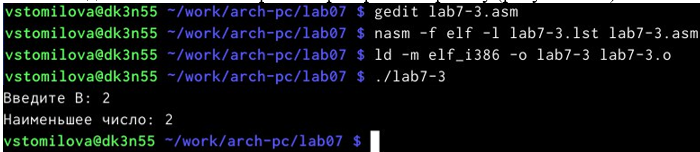


**Рисунок 0.26 (текст программы)**



**Рисунок 0.27 (текст программы)**

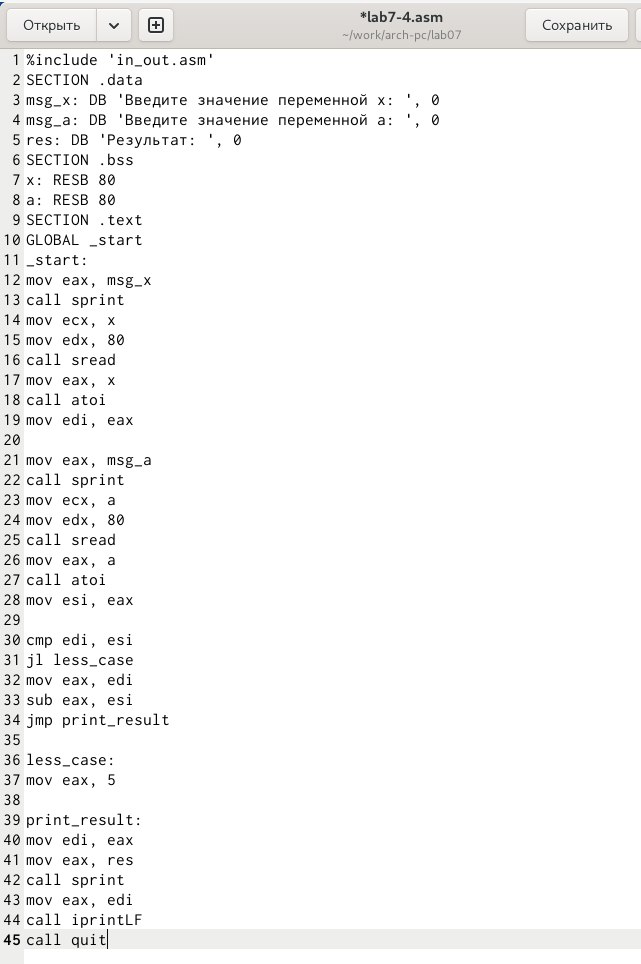
Создайте исполняемый файл и проверьте его работу (рисунок 0.28)



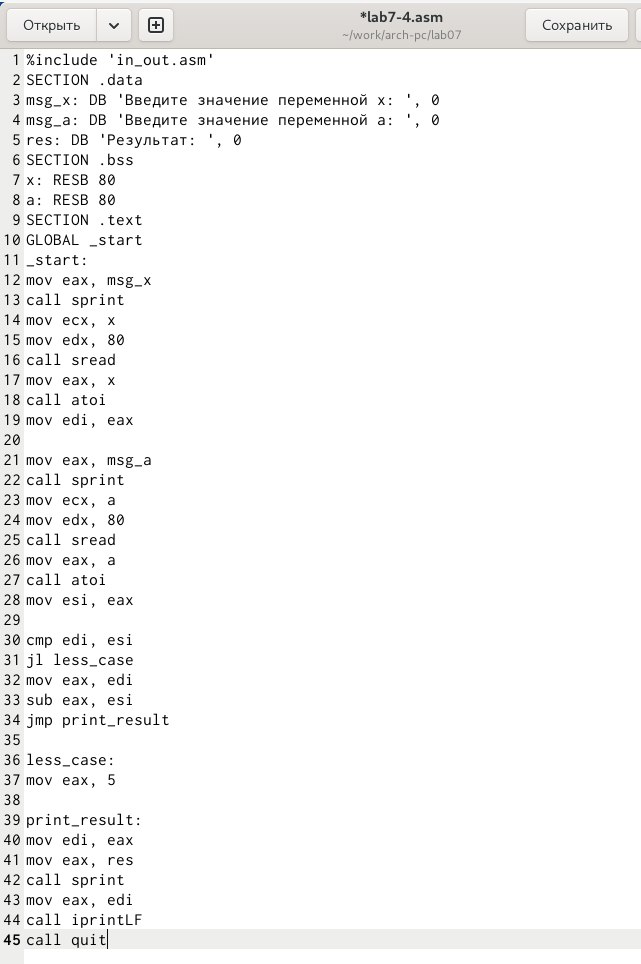
**Рисунок 0.28 (создание исполняемого файла и проверка его работы)**

Напишите программу, которая для введенных с клавиатуры значений 𝑥 и 𝑎 вычисляет значение заданной функции 𝑓(𝑥) и выводит результат вычислений. Вид функции 𝑓(𝑥) (x-a,x>=a;

5, x<a)(рисунок 0.29-0.30)



**Рисунок 0.29 (текст программы)**



**Рисунок 0.30 (текст программы)**

Создадим исполняемый файл и проверим его работу для значений 𝑥 и 𝑎

(рисунок 0.31-0.35)



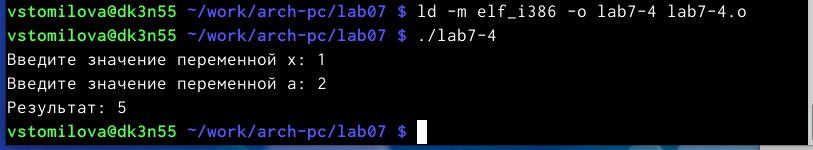
**Рисунок 0.31 (создание исполняемого файла)**



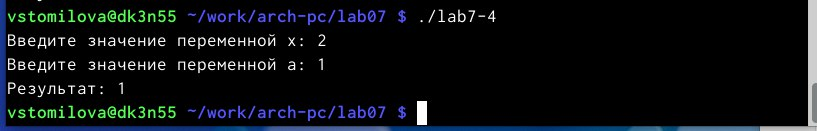
**Рисунок 0.32 (создание исполняемого файла)**



**Рисунок 0.33 (запуск программы)**



**Рисунок 0.34 (проверка результатов)**



**Рисунок 0.35 (проверка результатов)**

# Выводы

Мы изучили команды условного и безусловного перехода, приобрели навыки написания программ с использованием переходов, ознакомились с назначением и структурой файла листинга

# Список литературы

* 1. GDB:TheGNUProjectDebugger.—URL:https://[www.gnu.org/software/gdb/.](http://www.gnu.org/software/gdb/)
  2. GNUBashManual.—2016.—URL:https://[www.gnu.org/software/bash/manual/.](http://www.gnu.org/software/bash/manual/)
  3. Midnight CommanderDevelopment Center.—2021.—URL: https://midnight- commander. org/.
  4. NASMAssemblyLanguageTutorials.—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.

* 1. RobbinsA. Bash Pocket Reference.—O’Reilly Media,2016.—156 с.—ISBN 978-1491941591.
  2. TheNASMdocumentation.—2021.—URL:https://[www.nasm.us/docs.php.](http://www.nasm.us/docs.php)
  3. Zarrelli G. Mastering Bash.—Packt Publishing,2017.—502 с.—ISBN 9781784396879.
  4. Колдаев В. Д.,Лупин С. А.Архитектура ЭВМ.—М. : Форум,2018.
  5. Куляс О. Л.,Никитин К. А. Курс программирования наASSEMBLER.— М. : Солон-Пресс, 2017.
  6. НовожиловО.П.Архитектура ЭВМисистем.—М.:Юрайт,2016.

1 2 . Р а с ш и р е н н ы й а с с е м б л е р : N A S M . — 2 0 2 1 . — URL:https://[www.opennet.ru/docs/RUS/nasm/.](http://www.opennet.ru/docs/RUS/nasm/)

1. Робачевский А.,Немнюгин С.,Стесик О. Операционная система UNIX.— 2-е изд.—БХВ Петербург, 2010.—656 с.—ISBN 978-5-94157-538-1.
2. СтоляровА.Программированиенаязыкеассемблера NASMдляОСUnix.—2- еизд.— М. : МАКСПресс,2011.—URL:[http://www.stolyarov.info/books/asm\_unix.](http://www.stolyarov.info/books/asm_unix)
3. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
4. Таненбаум Э.,Бос Х. Современные операционные системы.—4-е изд.— СПб.: Питер, 2015. —1120 с.—(Классика Computer Science)