Отчёт по лабораторной работе №7

Дисциплина: архитектура компьютера

Терещенкова Маргарита Владимировна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Программа с использованием инструкции jmp (листинг 1)
2. Программа с использованием инструкции jmp (листинг 2)
3. Изучение структуры файла листинга
4. Выполнение заданий для самостоятельной 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

1. Создаю каталог для программ лабораторной работы № 7 с помощью команды **mkdir**.

Рис. 1: Создание каталога

Рис. 1: Создание каталога

Перехожу в него и создаю файл lab7-1.asm с помощью утилиты **touch**.

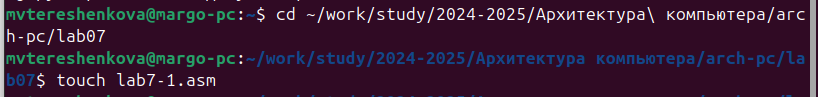


Рис. 2: Создание файла

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, так как он будет использоваться в других программах. И проверяю наличие файла в данной директории с помощью команды ls.

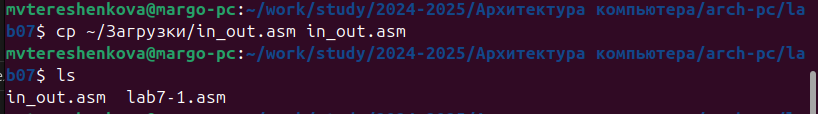


Рис. 3: Копирование файла

2.Открываю созданный файл lab7-1.asm, вставляю в него текст программы из листинга 7.1 (**Программа с использованием инструкции jmp**).

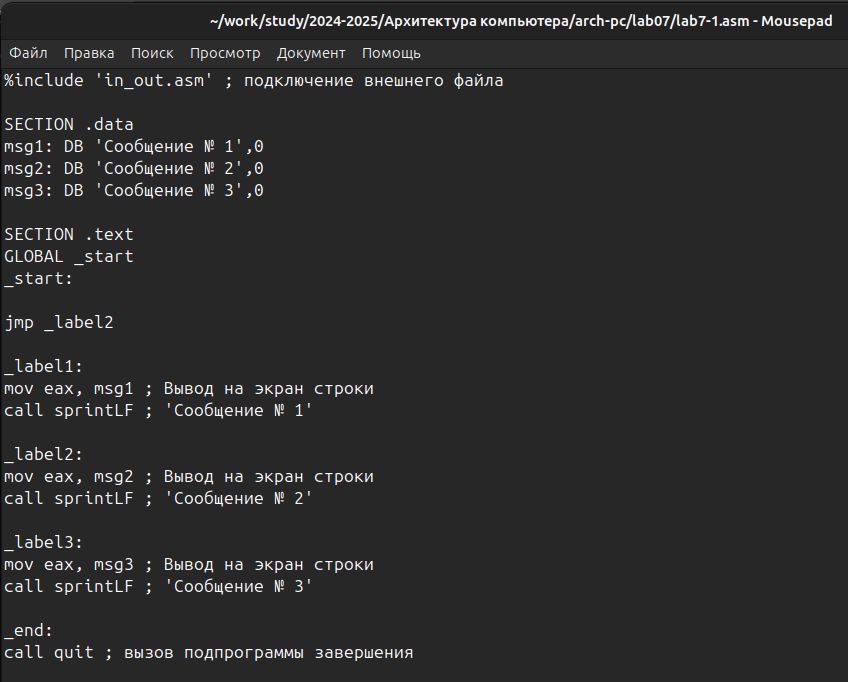


Рис. 4: Редактирование файла

Создаю исполняемый файл программы и запускаю его.

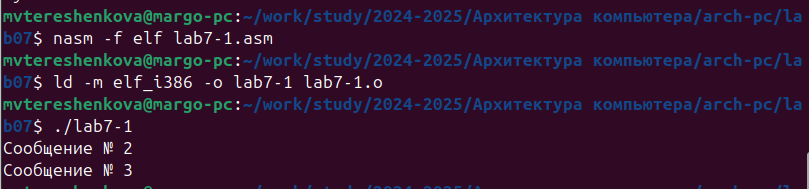


Рис. 5: Запуск исполняемого файла

Таким образом, использование инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения.

Изменила программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Изменяю текст программы в соответствии с листингом 7.2(**Программа с использованием инструкции jmp**).

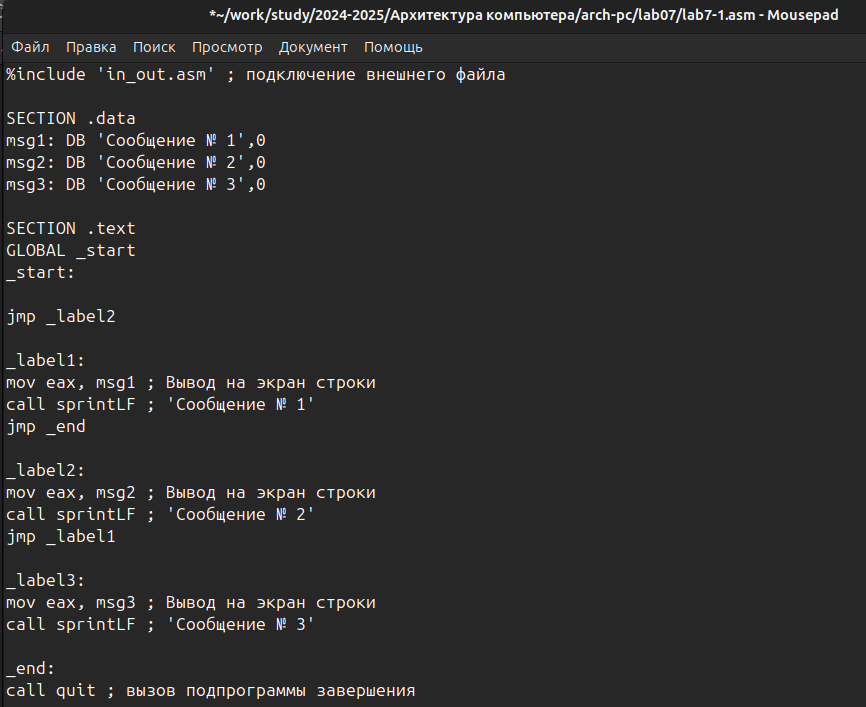


Рис. 6: Редактирование файла

Создаю исполняемый файл программы и запускаю его.

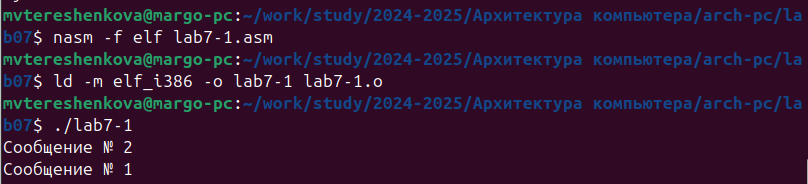


Рис. 7: Запуск исполняемого файла

Изменила текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим:

mvtereshenkova@margo-pc:~$ ./lab7-1

Сообщение № 3

Сообщение № 2

Сообщение № 1

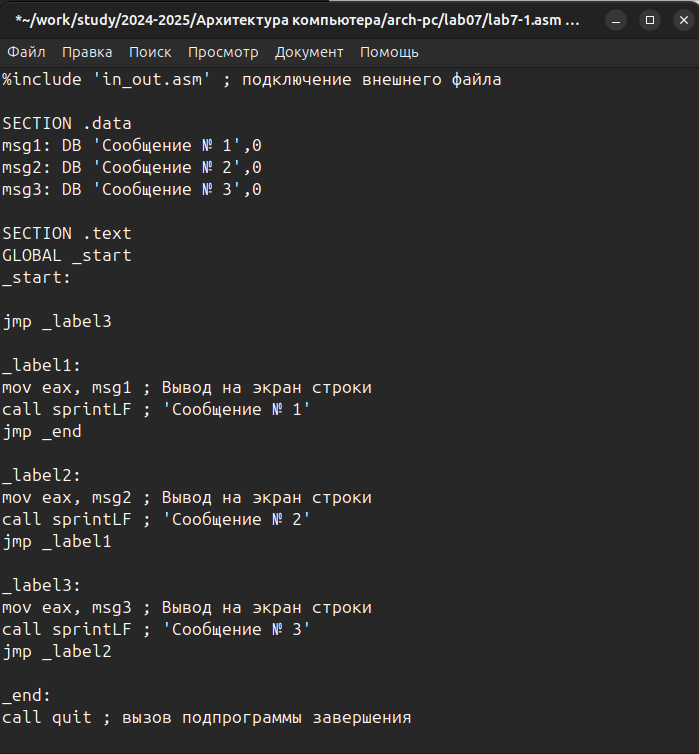


Рис. 8: Редактирование файла

Создаю исполняемый файл программы и запускаю его.

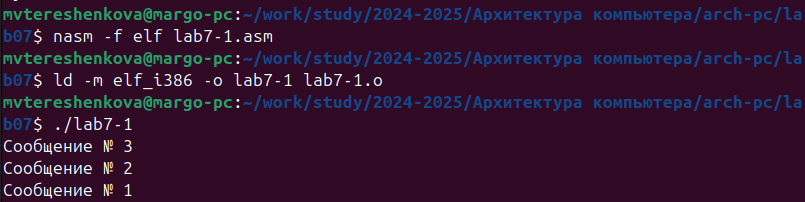


Рис. 9: Запуск исполняемого файла

1. Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучаю текст программы из листинга 7.3 (**Программа, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C**) и ввожу в lab7-2.asm.

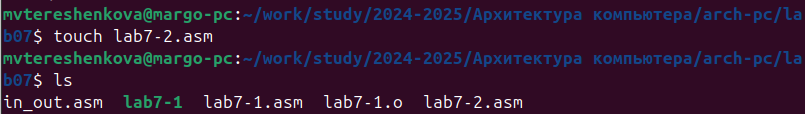


Рис. 10: Создание файла

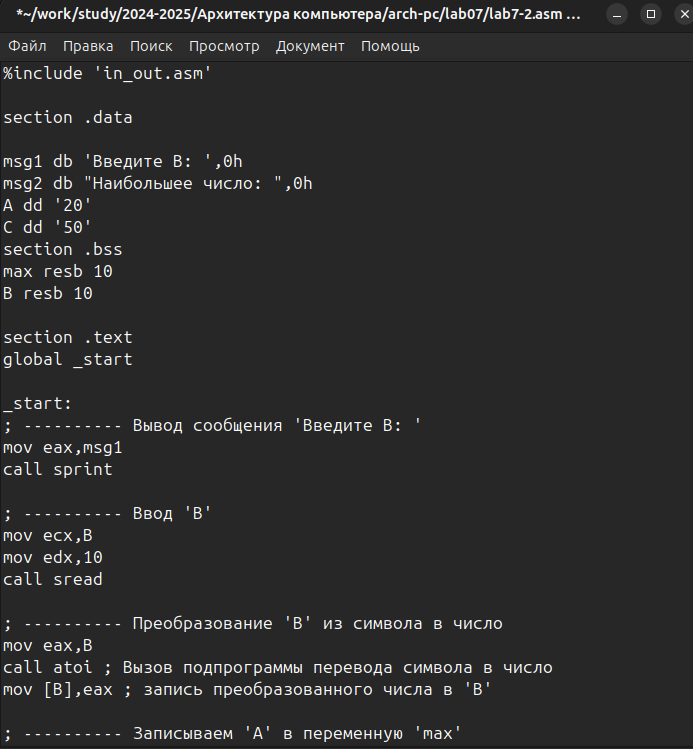


Рис. 11: Редактирование файла

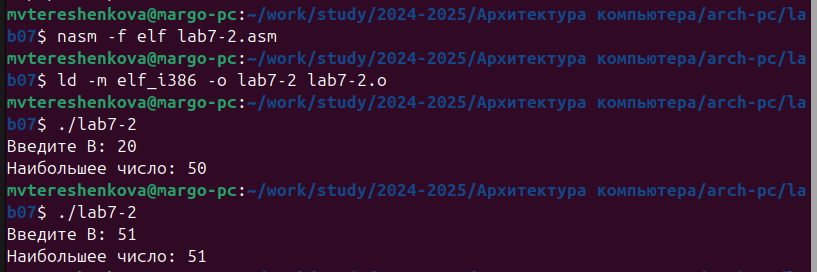


Рис. 12: Запуск исполняемого файла

При введении числа до 50, программа выводит наибольшее число 50, при введении числа больше 50, программа выводит введенное нами число. Программа сравнивает число A (значение 20) и C (значение 50) и инициализирует переменную max значением большего из них. Сравнивает текущее значение max с введённым числом B и обновляет max, если B больше. Выводит сообщение “Наибольшее число:” и затем значение переменной max, которая содержит наибольшее из трёх чисел: A, B и C.

## 4.2 Изучение структуры файлы листинга

1. Создаю файл листинга для программы из файла lab7-2.asm. Открываю его через mousepad.

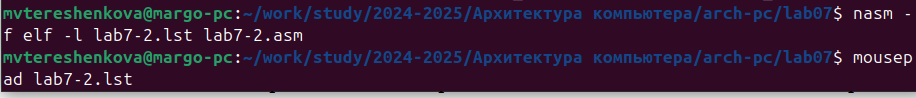


Рис. 13: Создание файла

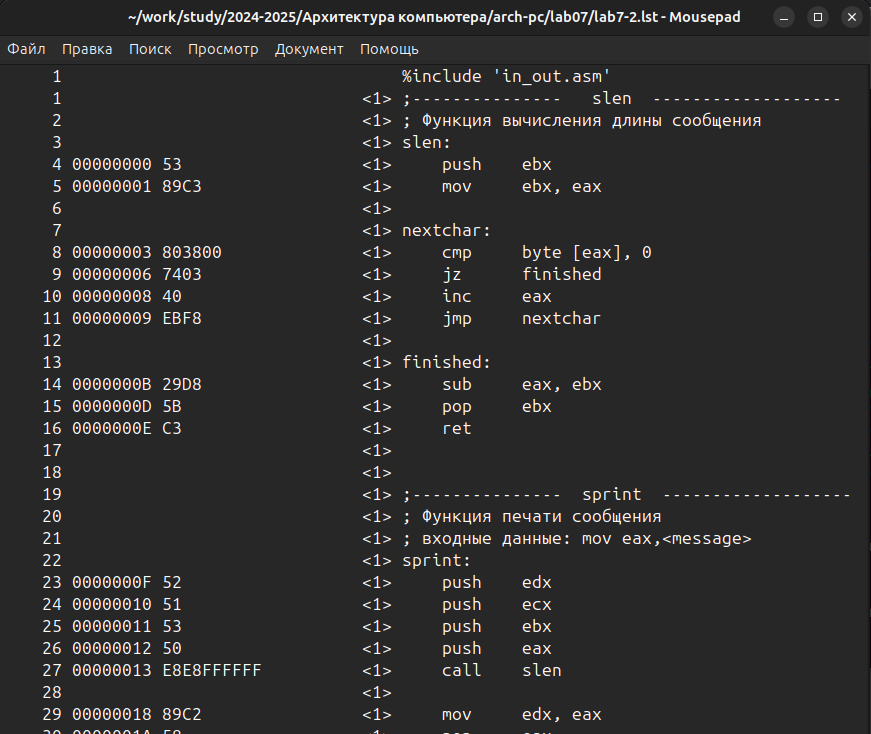


Рис. 14: Открытие файла в mousepad

При компиляции и сборке программы на ассемблере создаются следующие файлы:

* Объектный файл (.o): Это промежуточный файл, содержащий машинный код, но ещё не готовый для выполнения.
* Исполняемый файл: После связывания объектных файлов с библиотеками (например, с помощью ld), создается исполняемый файл, который можно запустить.
* Файл листинга (.lst): Это текстовый файл, который включает исходный код программы вместе с адресами и скомпилированным машинным кодом. В этом файле обычно содержатся комментарии и информация о процессе компиляции.

В файл листинга могут быть добавлены следующие элементы:

* Исходный код: Полный исходный код программы, как он написан в ассемблере.
* Адреса: Для каждой инструкции будут указаны адреса в памяти, по которым эти инструкции будут располагаться после компиляции.
* Машинный код: Бинарный код, соответствующий каждой инструкции, представленный в шестнадцатеричном формате.
* Комментарии: Комментарии из исходного кода, которые могут помочь понять логику программы.
* Информация о секциях: Данные о том, как разделены секции кода (.text, .data, .bss и т.д.) и их размеры.
* Ошибки и предупреждения: Если при компиляции были обнаружены ошибки или предупреждения, они также могут быть записаны в файл 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. Открыла файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Дальше выполнила трансляцию с получением файла листинга: **nasm -f elf -l lab7-2.lst lab7-2.asm**

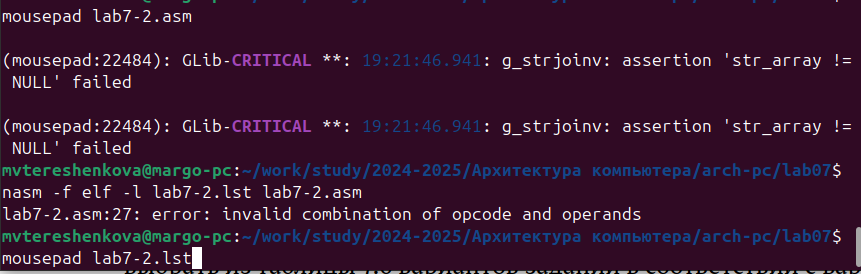


Рис. 15: Удаление операнда из программы и запуск программмы

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

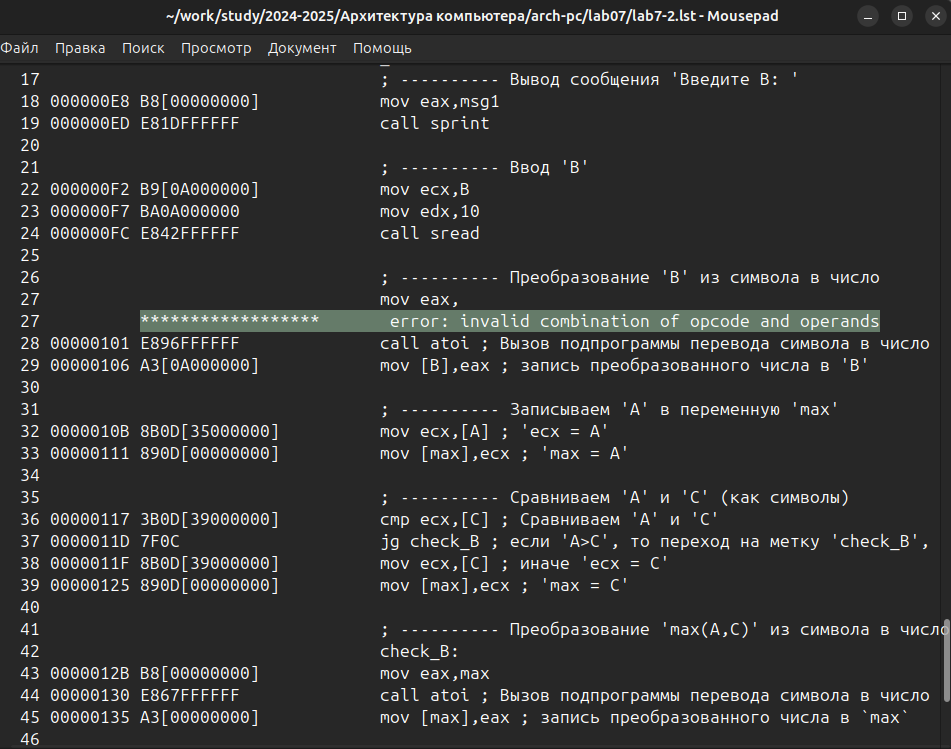


Рис. 16: Просмотр ошибки в файле листинга

# 5 Задания для самостоятельной работы

1. Мой вариант номер 4. Создаю файл с названием lab7-3.asm, написала программу для нахождения наименьшего из 3 переменных, значения переменных беру, исходя из своего варианта, полученного в ходе лабораторной работы номер 6. Сама программа прикреплена в ТУИС и в Git Hub.

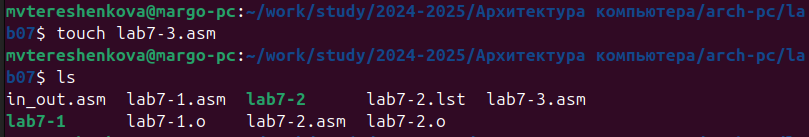


Рис. 17: Создание файла

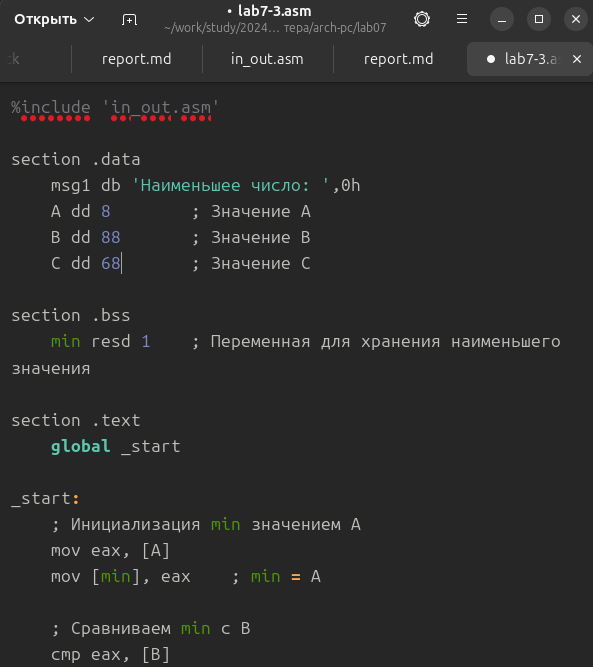


Рис. 18: Редактирование файла

Проверяю работу программы, программа работает корректно.

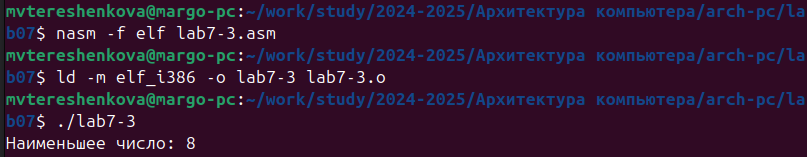


Рис. 19: Запуск исполняемого файла

1. Создаю файл с названием lab7-4.asm.

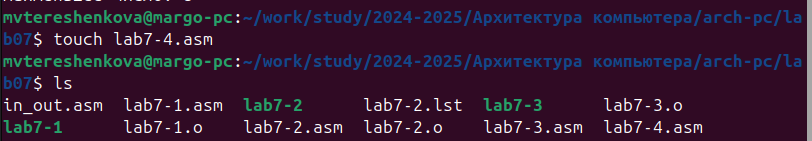


Рис. 20: Создание файла

Написала программу для вычисления *f(x)*. Сама программа прикреплена в ТУИС и в Git Hub.

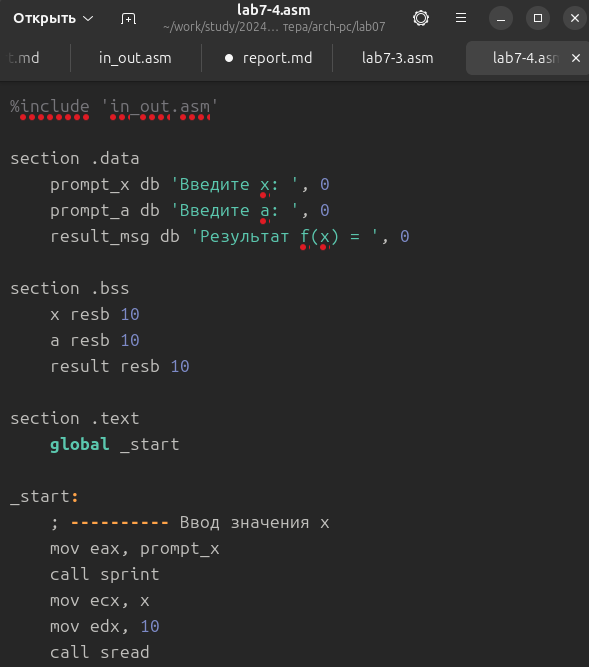


Рис. 21: Написание программы

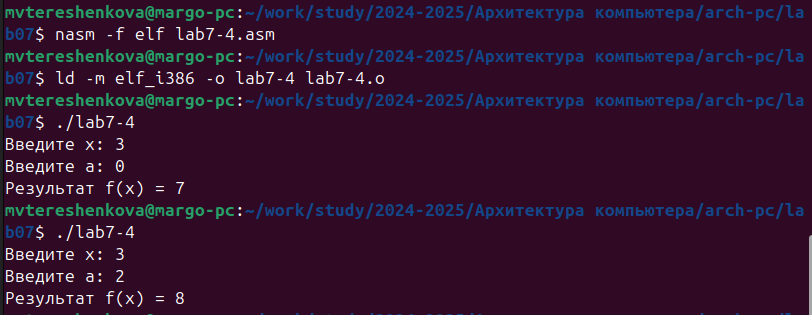


Рис. 22: Запуск исполняемого файла

Программа работает корректно; проверила, подставив соответсвующие x и a, указанные для моего варианта.

# 6 Выводы

Благодаря данной лабораторной работе изучила команды условного и безусловного переходов; приобрела навыки написания программ с использованием переходов; познакомилась с назначением и структурой файла листинга.

# Список литературы

1. Архитектура компьютера