Лабораторная работа №8

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Вершинина Ангелина Алексеевна

Содержание

# 1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навы- ков написания программ с использованием переходов. Знакомство с назначе- нием и структурой файла листинга.

# 2 Задание

Написание программ, используя полученные знания

# 3 Теоретическое введение

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление: mp . Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предварительно помещен указатель перехода. Кроме того, в качестве операнда можно использовать имя регистра, в таком случае переход будет осу- ществляться по адресу, хранящемуся в этом регистре

# 4 Выполнение лабораторной работы

## 4.1 Реализация переходов в NASM

Создайте каталог для программам лабораторной работы No 8, перейдите в него и создайте файл lab8-1.asm (рис. 1)

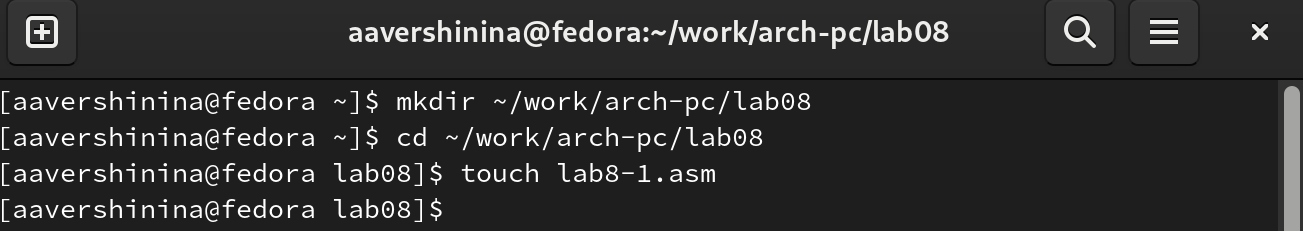


Рис. 1: Создание каталога

Инструкция jmp в NASM используется для реализации безусловных пере- ходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab8-1.asm текст программы из листинга 8.1 (рис. 2)

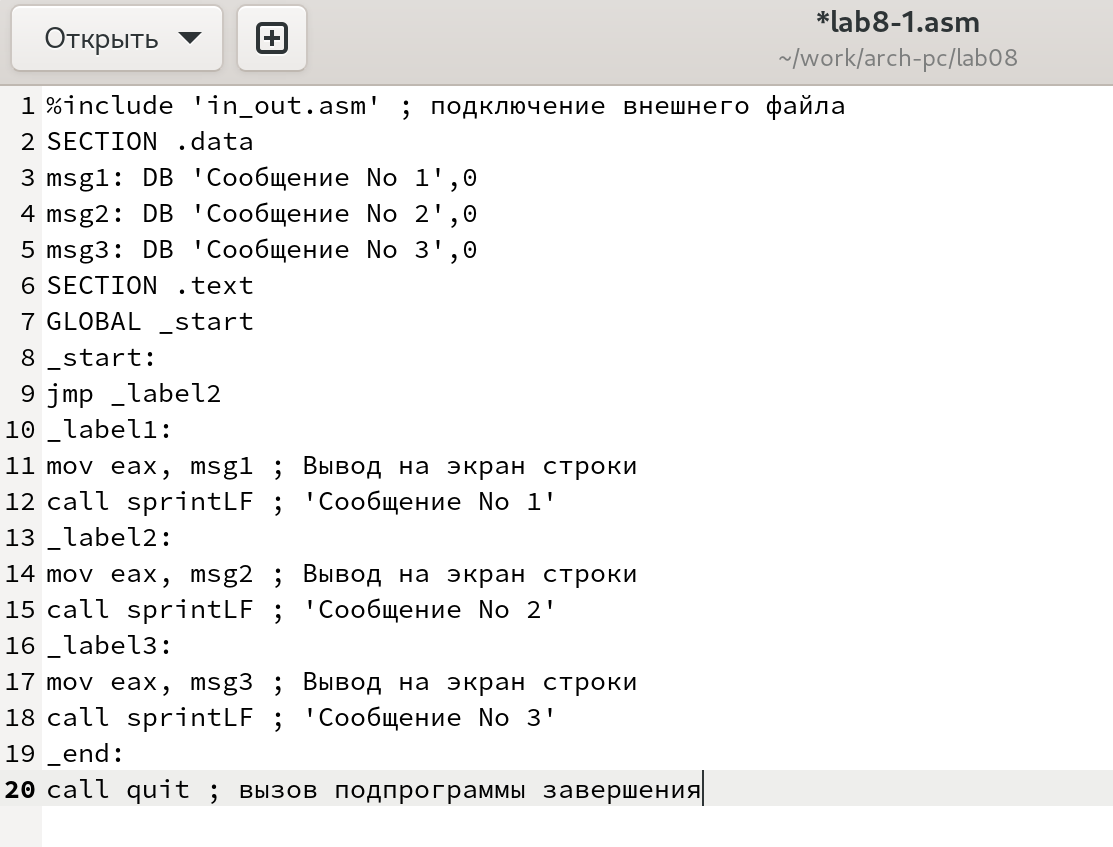


Рис. 2: Текст программы

Создайте исполняемый файл и запустите его (рис. 3)

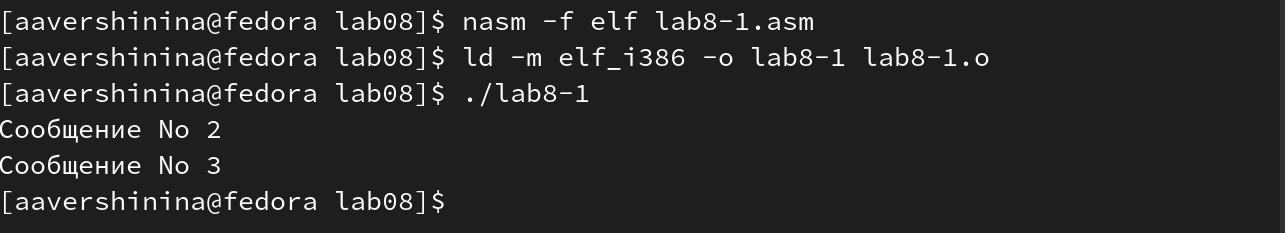


Рис. 3: Создание исполняемого файла и его запуск

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

Инструкция jmp позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу. Для этого в текст программы после вывода сообщения No 2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения No 1) и после вывода сообщения No 1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Измените текст программы в соответствии с листингом 8.2. (рис. 3)

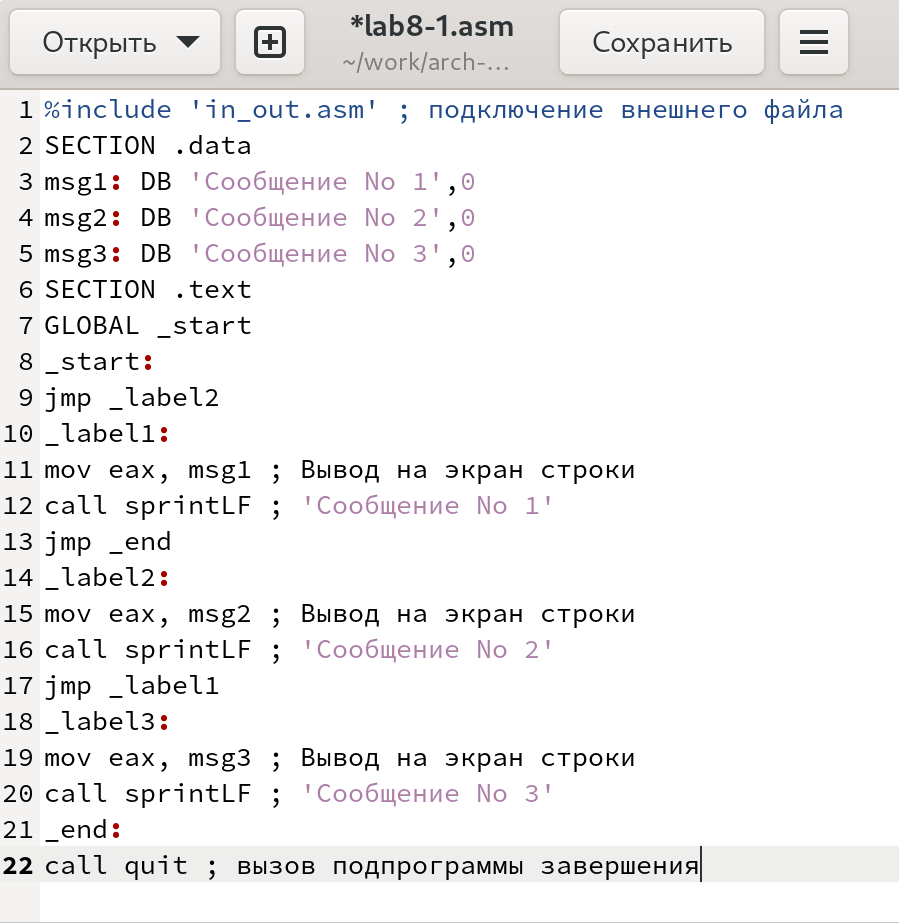


Рис. 4: Текст программы

Создайте исполняемый файл и запустите его (рис. 5)

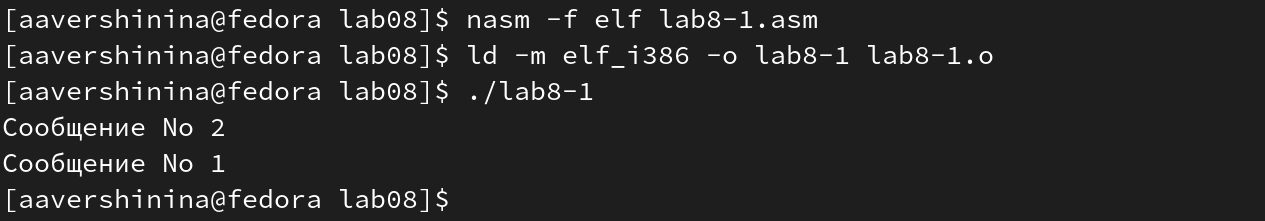


Рис. 5: Создание исполняемого файла и его запуск

Измените текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим:(рис. 6)

Сообщение No 3  
Сообщение No 2  
Сообщение No 1

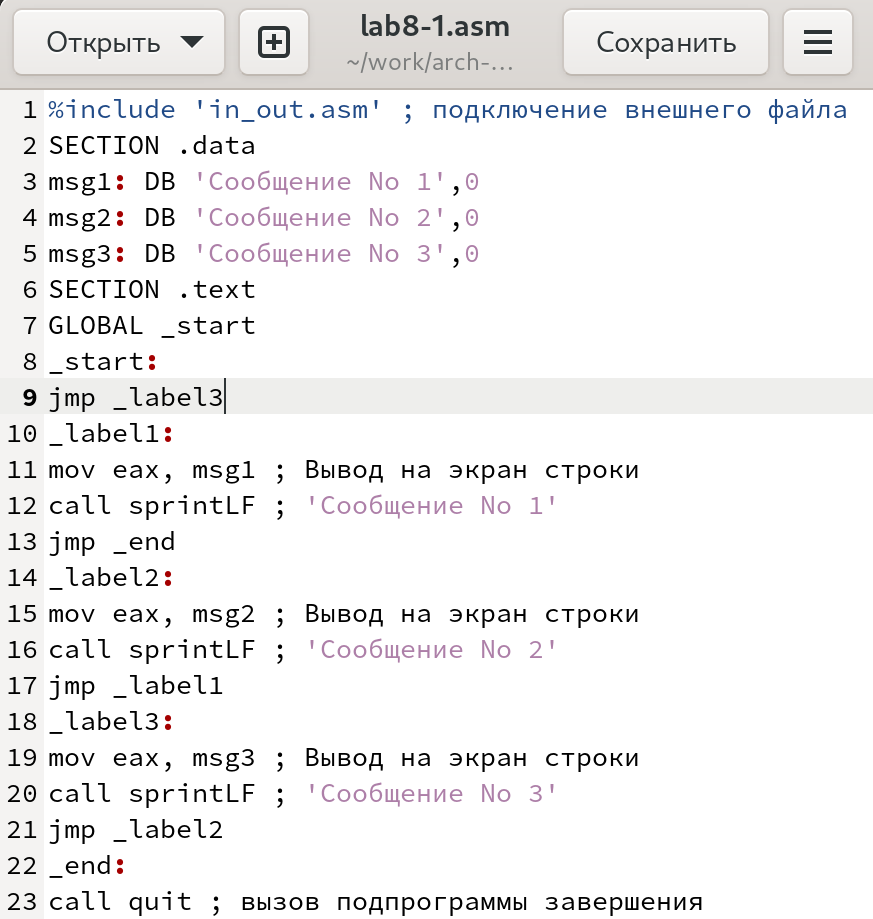


Рис. 6: Текст программы

Создайте исполняемый файл и запустите его (рис. 7)

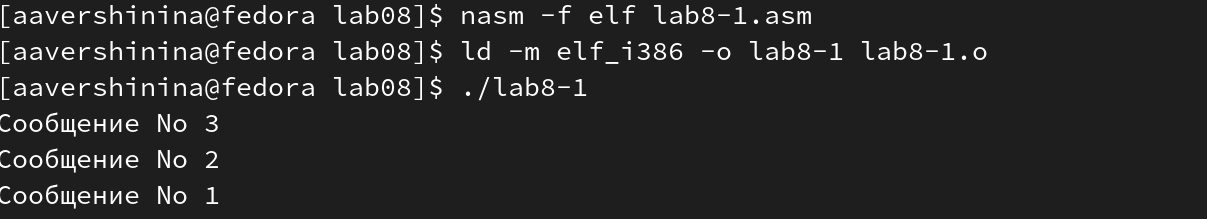


Рис. 7: Создание исполняемого файла и его запуск

Использование инструкции jmp приводит к переходу в любом случае. Од- нако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры.

Создайте файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Внимательно изучите текст программы из листинга 8.3 и введите в lab8-2.asm. (рис. 8)

Рис. 8: Создание файла

Рис. 8: Создание файла

Создайте исполняемый файл и запустите его (рис. 9)

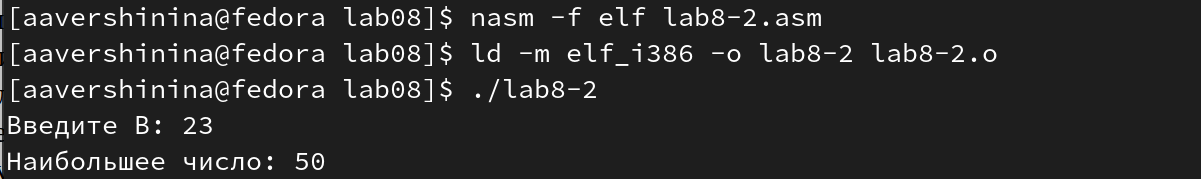


Рис. 9: Создание исполняемого файла и его запуск

Обратите внимание, в данном примере переменные A и С сравниваются как символы, а переменная B и максимум из A и С как числа (для этого используется функция atoi преобразования символа в число). Это сделано для демонстра- ции того, как сравниваются данные. Данную программу можно упростить и сравнивать все 3 переменные как символы (т.е. не использовать функцию atoi). Однако если переменные преобразовать из символов числа, над ними можно корректно проводить арифметические операции.

## 4.2 Изучение структуры файлы листинга

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создайте файл листинга для программы из файла lab8-2.asm (рис. 10)

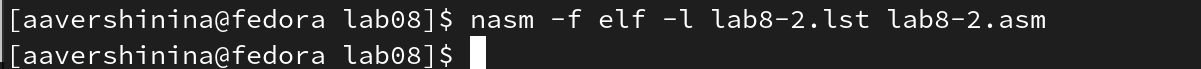


Рис. 10: Создание файла листинга

Откройте файл листинга lab8-2.lst с помощью любого текстового редактора, например mcedit (рис. 11)

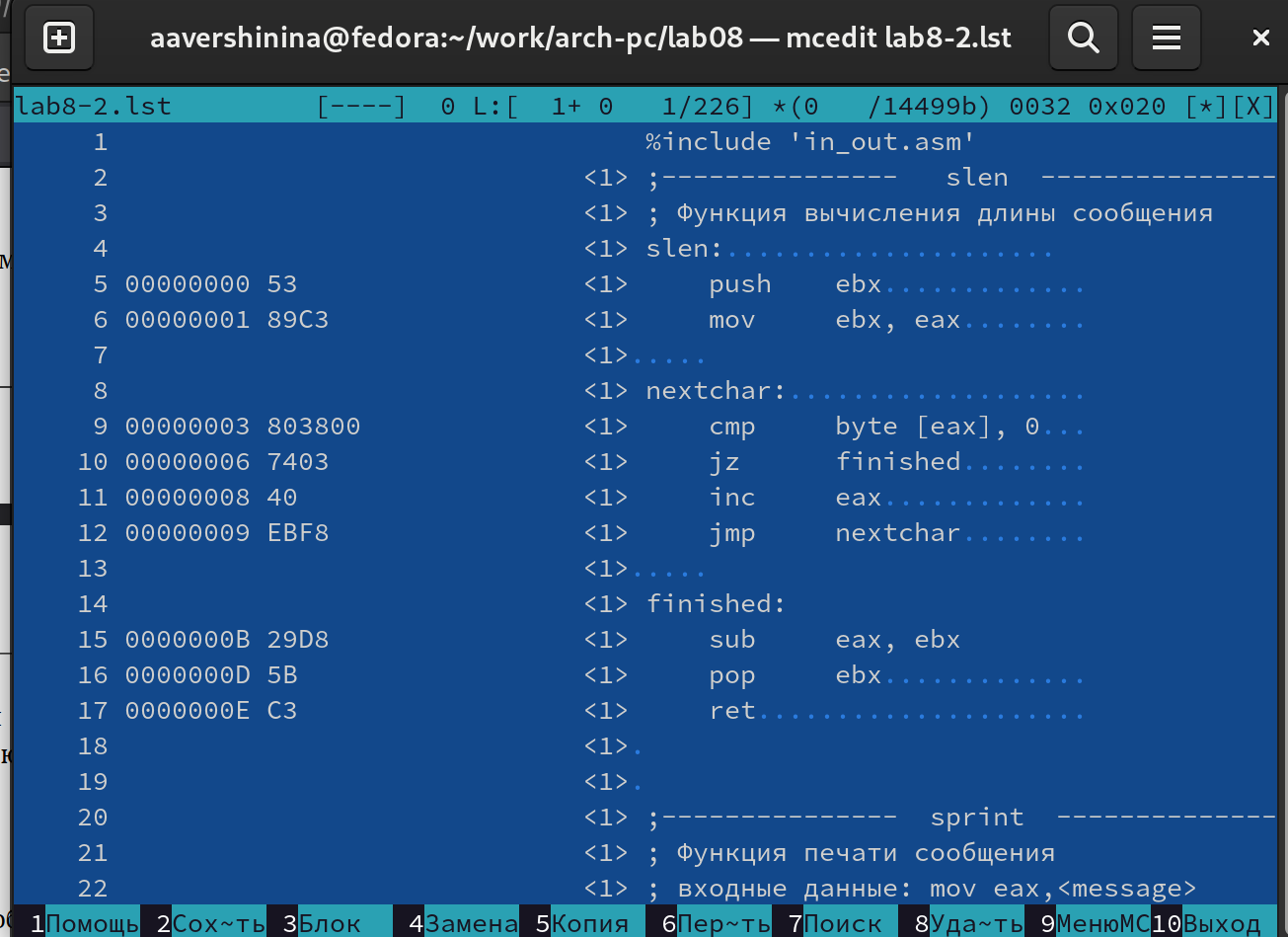


Рис. 11: Открытие файла листинга

Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержимое трёх строк файла листинга по выбору.

1. 6(номер строки) 00000001(адрес в сегменте кода) 89С3(машинный код) mov ebx,eax(исходный текст программы) (рис. 12)
2. 11(номер строки) 00000008(адрес в сегменте кода) 40(машинный код) inc eax(исходный текст программы) (рис. 13)
3. 54(номер строки) 0000003B(адрес в сегменте кода) E8CFFFFFFF(машинный код) call sprint(исходный текст программы) (рис. 14)

Рис. 12: Строка 6

Рис. 12: Строка 6

Рис. 13: Строка 11

Рис. 13: Строка 11

Рис. 14: Строка 54

Рис. 14: Строка 54

Откройте файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалить один операнд.(рис. 15) Выполните трансляцию с получением файла листинга (рис. 16). В результате система выдает ошибку и создает файлы lab8-2 и lab8-2.lsl.

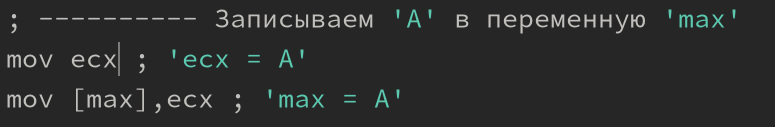


Рис. 15: Удаление операнда

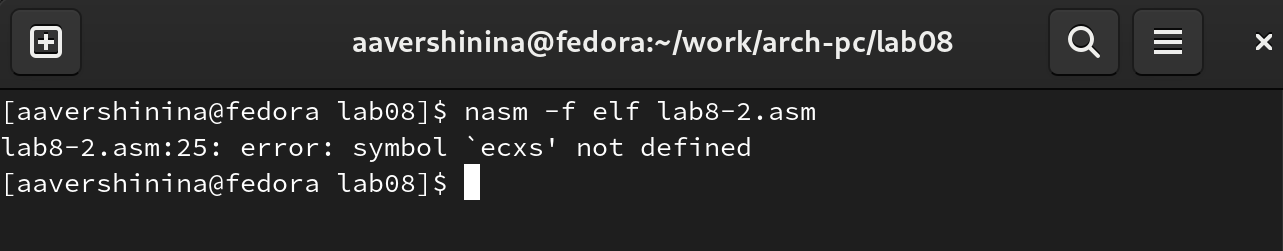


Рис. 16: Трансляция файла

## 4.3 Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных пере- менных a, b и с. Значения переменных выбрать из табл. 8.5 в соответствии с вариантом 12, полученным при выполнении лабораторной работы No 7. (рис. 17) Создайте исполняемый файл и проверьте его работу. (рис. 18)

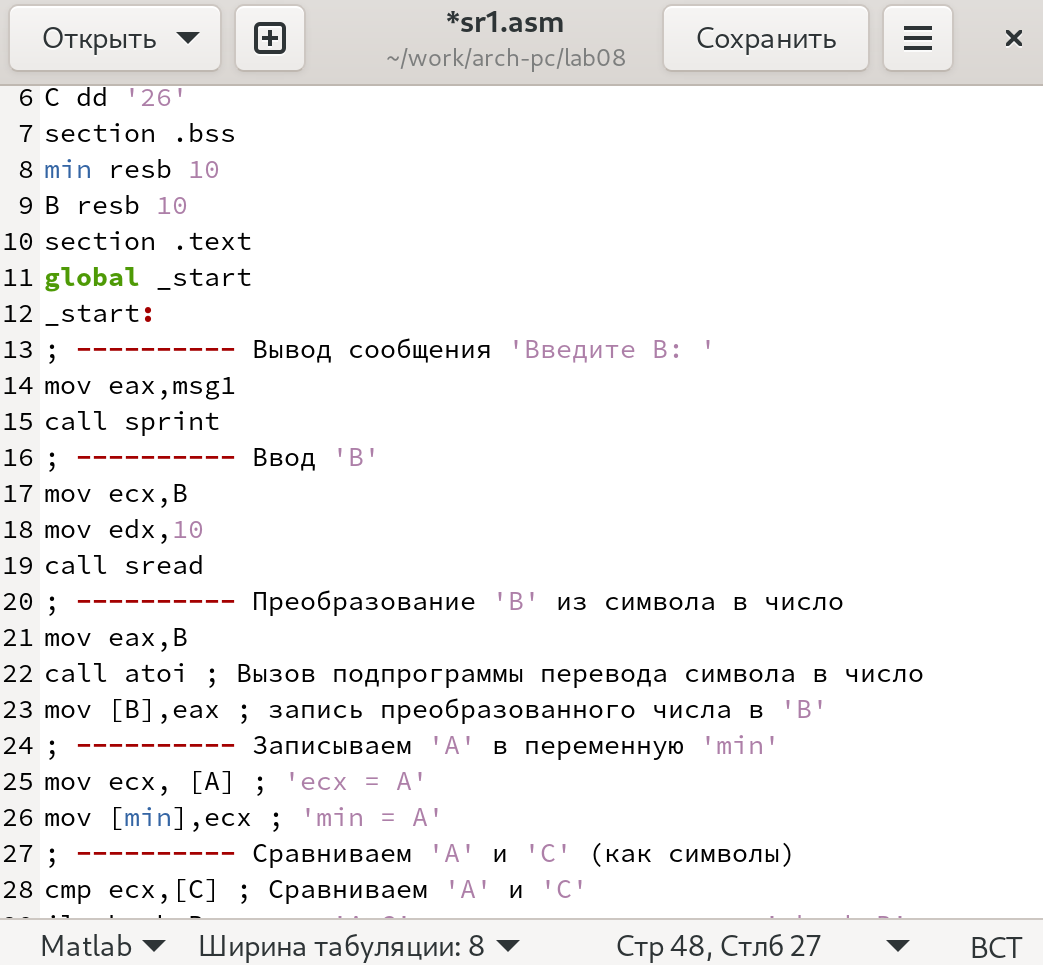


Рис. 17: Текст программы



Рис. 18: Создание исполняемого файла и его запуск

1. Напишите программу, которая для введенных с клавиатуры значений 𝑥 и 𝑎 вычисляет значение заданной функции 𝑓(𝑥) и выводит результат вы- числений. Вид функции 𝑓(𝑥) выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом 12, полученным при выполнении лабораторной работы No 7. Создайте исполняемый файл и проверьте его работу для зна- чений 𝑥 и 𝑎 из 8.6. (рис. 19)

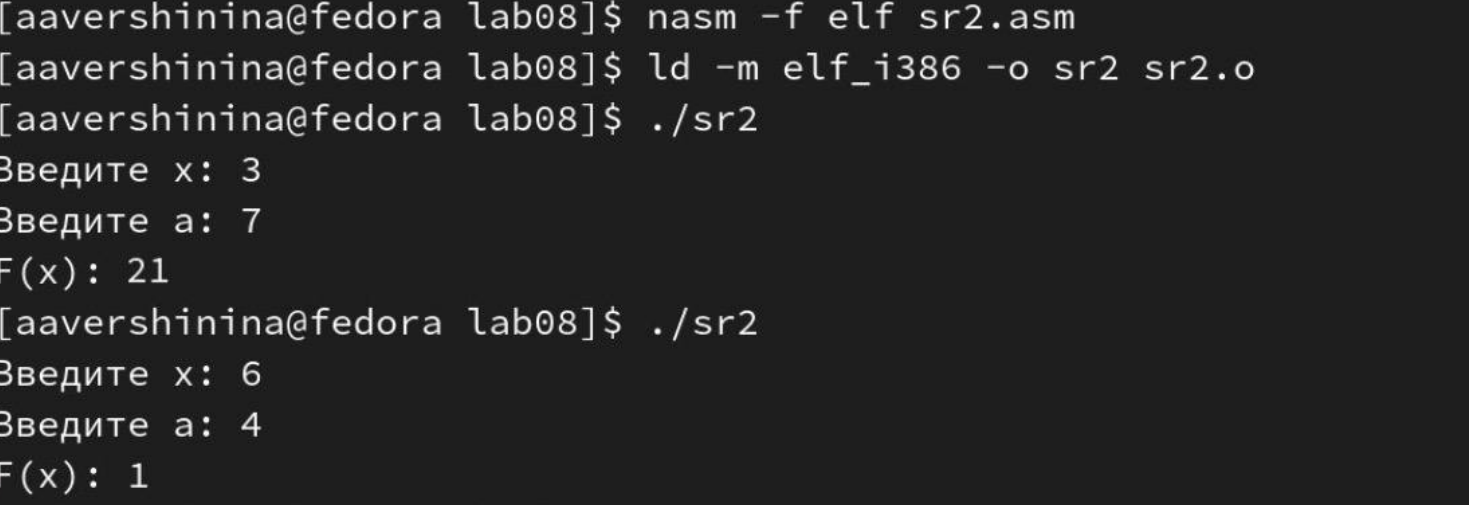


Рис. 19: Создание исполняемого файла и его запуск

# 5 Выводы

В результате выполнения лабораторной работы я изучила команды условного и безусловного переходов. Приобрела навыки написания программ с использованием переходов. Ознакомилась с назначением и структурой файла листинга.

# Список литературы