**Российский университет дружбы народов**

**ответ по лабораторной работе NO.7**

Студент:Чжан Сюйтун

Группа:НММбд-04-23

**Москва**

**2023 Г**

**Содержание**

1. [Цель работы](#_bookmark0) 5
2. [Теоретическое введение](#_bookmark1) 6
   1. [Команды безусловного перехода](#_bookmark2) . . . . . . . . . . . . . . . . . . 6
3. [Лабораторной работы](#_bookmark3) 7
   1. [Реализация переходов в NASM](#_bookmark4) . . . . . . . . . . . . . . . . . . . 7
   2. [Изучение структуры файлы листинга](#_bookmark13) 13
4. [Задание для самостоятельной работы](#_bookmark19) 17
5. [Выводы](#_bookmark24) 22
6. [Список литературы](#_bookmark25) 23

**Список иллюстраций**

* 1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
  3. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
  4. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  5. 10
  6. 10
  7. 11
  8. 12
  9. 13
  10. 13
  11. 14
  12. 15
  13. 16
  14. 18
  15. 18
  16. 20
  17. 21

**Список таблиц**

# Цель работы

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

# Теоретическое введение

Для реализации ветвлений в ассемблере используются так называе- мые 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 опреде- ленную точку программы без каких-либо условий.

## Команды безусловного перехода

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление:

jmp <адрес\_перехода>

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.

# Лабораторной работы

## Реализация переходов в NASM

Создайте каталог для программам лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm:

mkdir ~/work/arch-pc/lab07 cd ~/work/arch-pc/lab07 touch lab7-1.asm

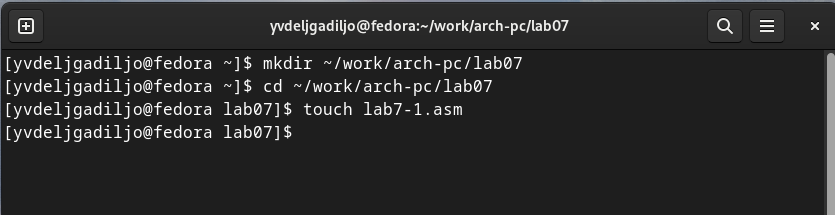


Рис. 3.1:

Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab7-1.asm текст программы из листинга 7.1.

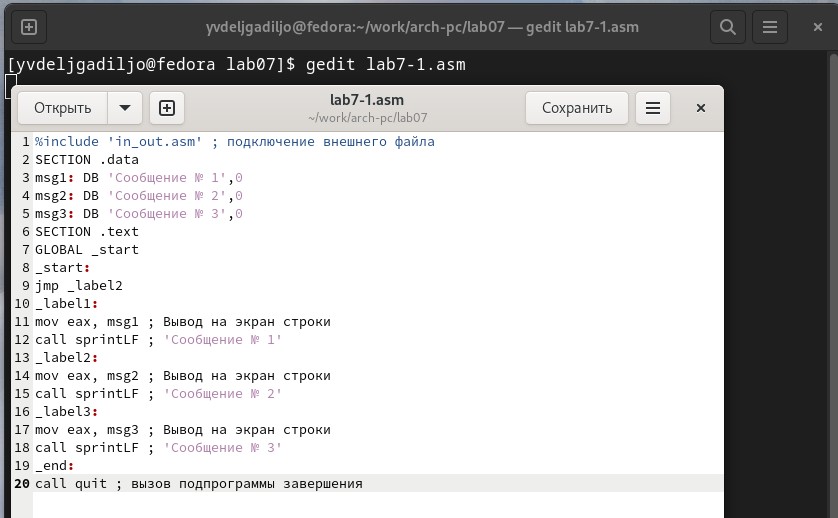


Рис. 3.2:

Создайте исполняемый файл и запустите его. Результат работы данной про- граммы будет следующим:

user@dk4n31:~$ ./lab7-1 Сообщение № 2

Сообщение № 3 user@dk4n31:~$

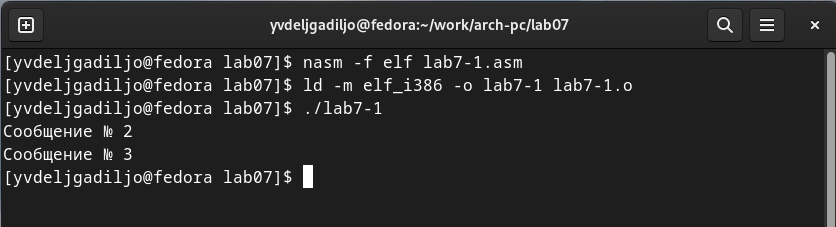


Рис. 3.3:

Таким образом, использование инструкции jmp \_label2 меняет порядок испол-

нения

инструкций и позволяет выполнить инструкции начиная с метки \_label2, про- пустив вывод первого сообщения.

Инструкция jmp позволяет осуществлять переходы не только вперед но и назад. Изменим 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). Измените текст программы в соответствии с листингом 7.2

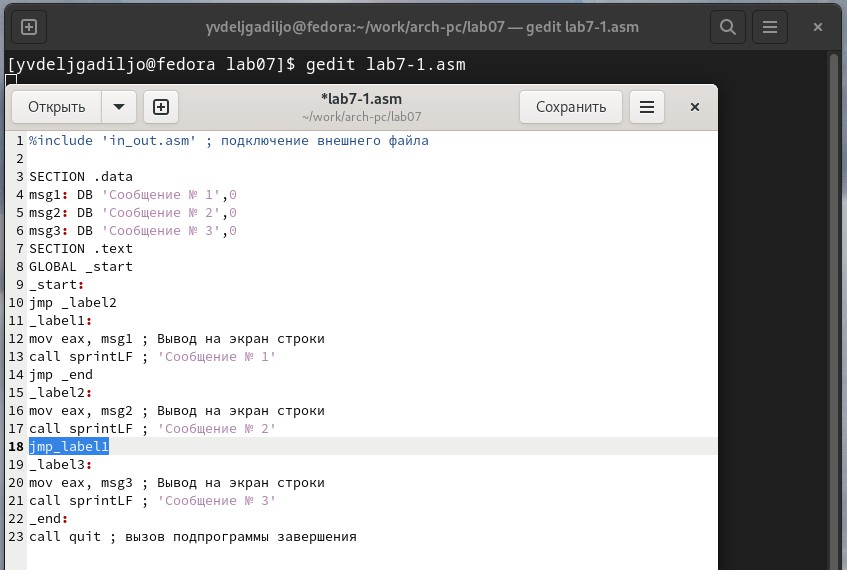


Рис. 3.4:

Создайте исполняемый файл и проверьте его работу.

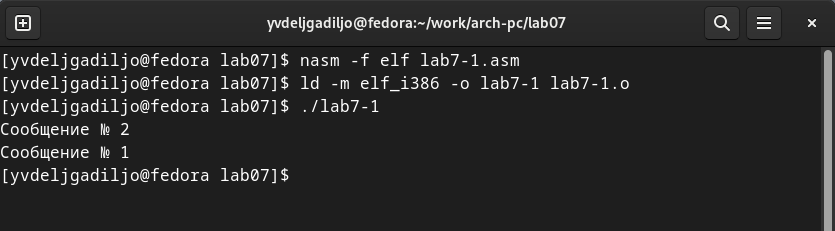


Рис. 3.5:

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

user@dk4n31:~$ ./lab7-1 Сообщение № 3

Сообщение № 2

Сообщение № 1 user@dk4n31:~$

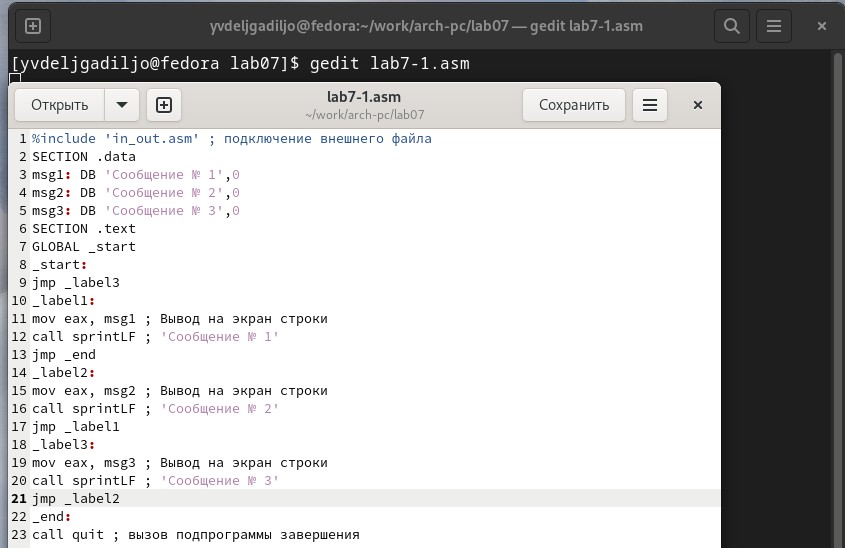


Рис. 3.6:

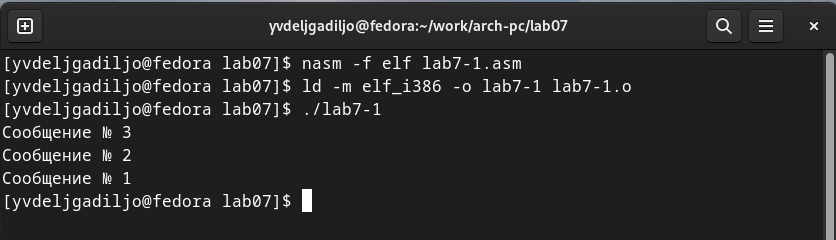


Рис. 3.7:

Использование инструкции jmp приводит к переходу в любом случае. Однако, 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 наи- большую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры.

Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm.

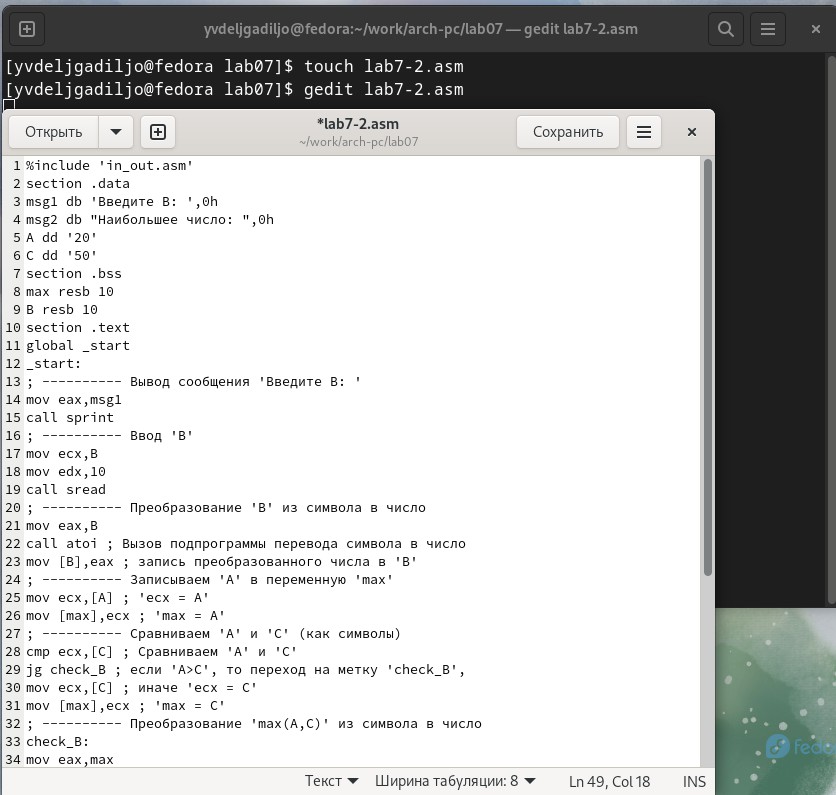


Рис. 3.8:

Создайте исполняемый файл и проверьте его работу для разных значений B. 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 демонстрации того, как сравниваются данные.

Данную программу можно упростить и сравнивать все 3 переменные как сим- волы (т.е. не использовать функцию atoi). Однако если переменные преобразо- вать из символов числа, над ними можно корректно проводить арифметические операции.

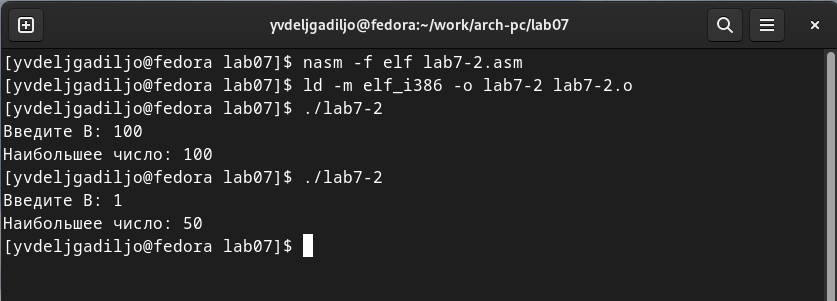


Рис. 3.9:

## 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 lab7-2.asm nasm -f elf -l lab7-2.lst lab7-2.asm

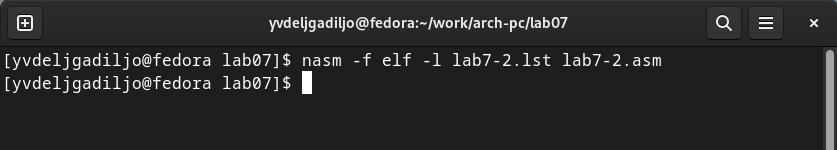


Рис. 3.10:

Откройте файл листинга lab7-2.lst с помощью любого текстового редактора, например mcedit:

mcedit lab7-2.lst

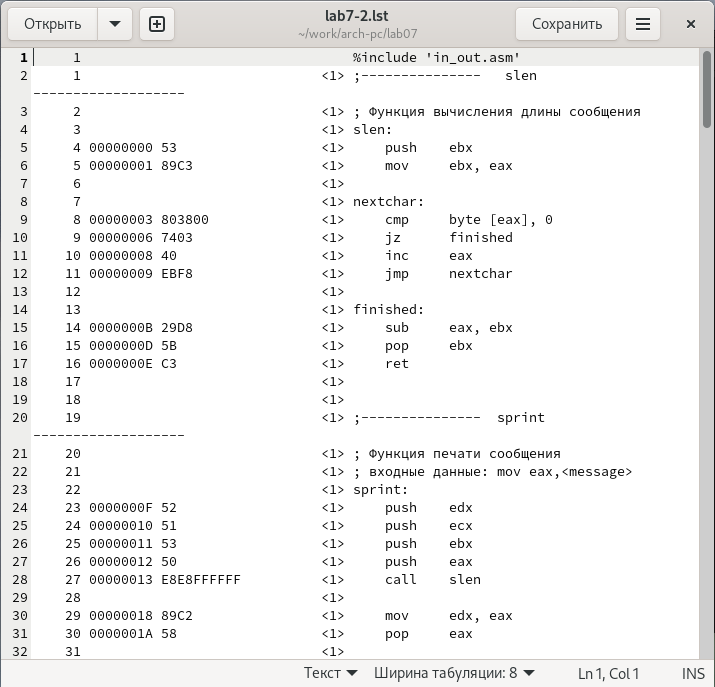


Рис. 3.11:

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

в строке 9 содержится собственно номер сторки [9], адресс [00000003], машин- ный код [803800] и содержимое строки кода [cmp byte [eax], 0] в строке 11 содер- жится номер сторки [11], адресс [00000008], машинный код [40] и содержимое строки кода [inc eax] в строке 24 содержится номер сторки [24], адресс [0000000F], машинный код [52] и содержимое строки кода [push edx].

Откройте файл с программой lab7-2.asm и в любой инструкции с двумя опе- рандами удалить один операнд.

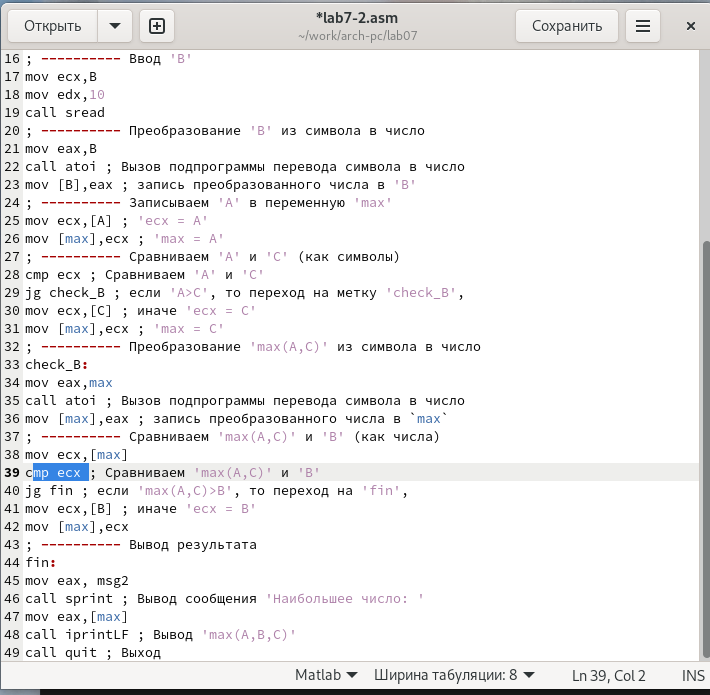


Рис. 3.12:

Выполните трансляцию с получением файла листинга: nasm -f elf -l lab7-2.lst lab7-2.asm

Какие выходные файлы создаются в этом случае? Что добавляется в листинге? Oшибка в файле листинга

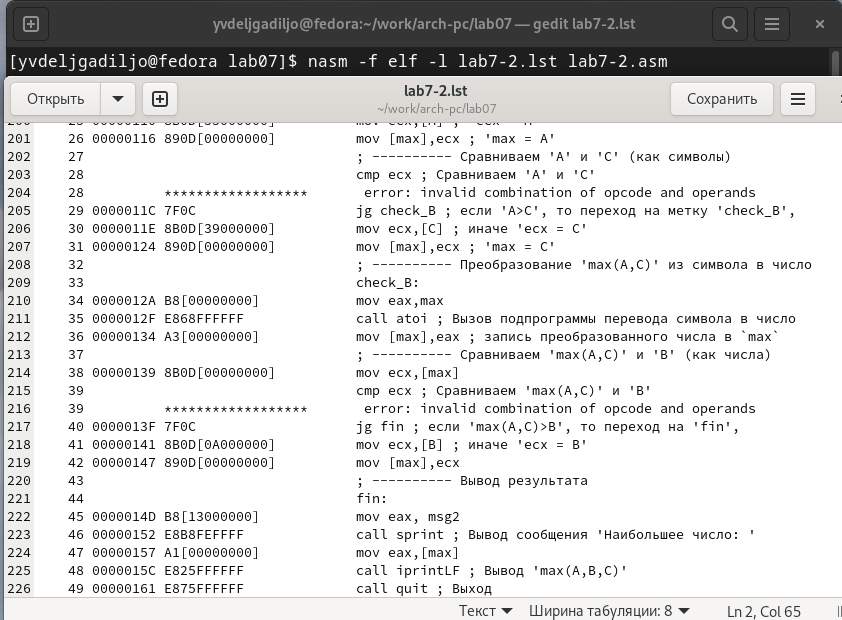


Рис. 3.13:

# Задание для самостоятельной работы

Напишите программу нахождения наименьшей из 3 целочисленных перемен- ных a,b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариан- том, полученным при выполнении лабораторной работы № 7 (Вар 19). Создайте исполняемый файл и проверьте его работу.

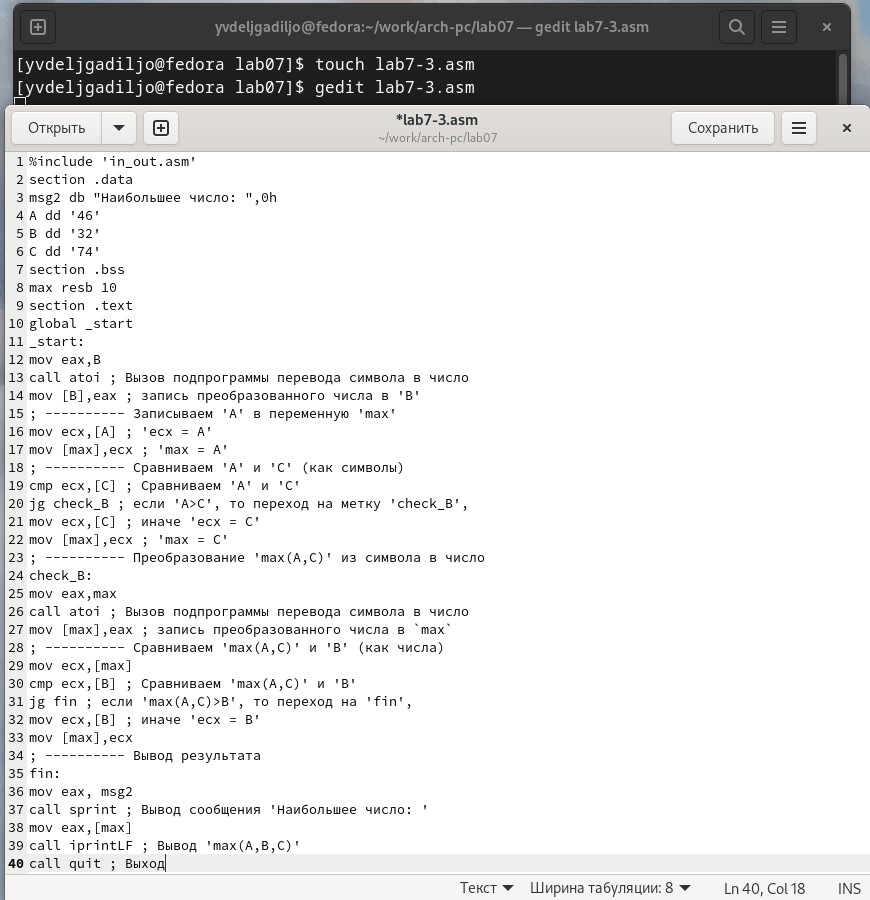


Рис. 4.1:

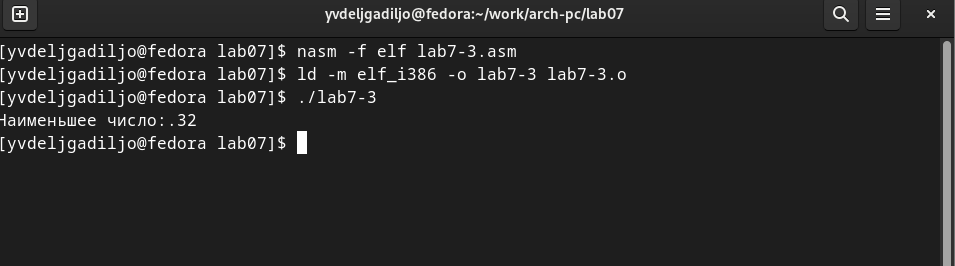


Рис. 4.2:

Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет

значение заданной функции f(x) и выводит результат вычислений. Вид функ- ции f(x)

выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полу- ченным

при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте

его работу для значений x и a из 7.6.Вар 19: a + x, x > a

x, x ≤ a Код программы

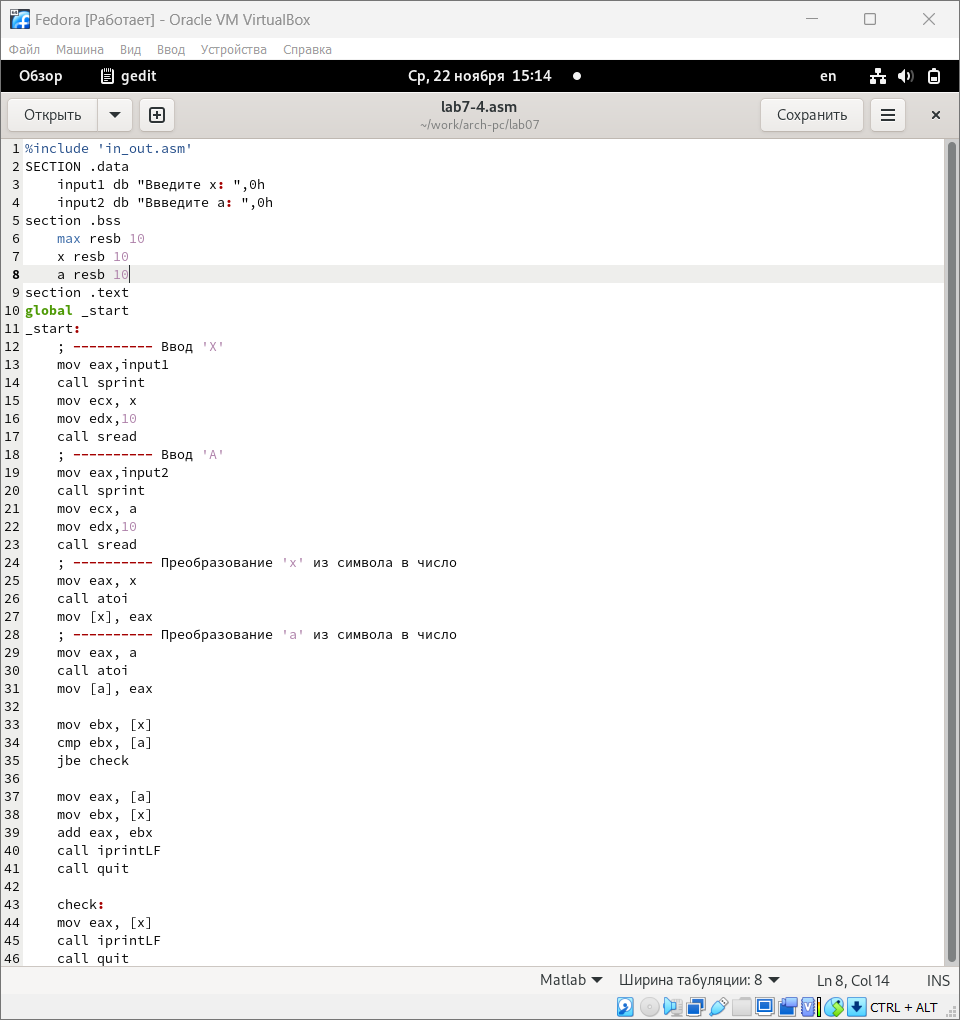


Рис. 4.3:

Результат выполнения программы

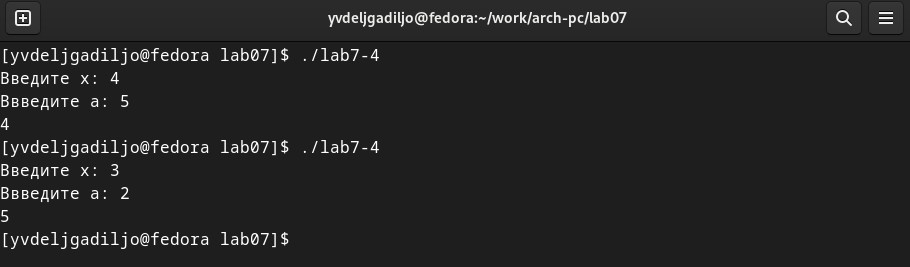


Рис. 4.4:

# Выводы

Я изучила команды условного и безусловного переходов и научилась писать программы с использованием этих переходов.