Отчёт по лабораторной работе 9

Дисциплина: архитектура компьютера

Закиров Нурислам Дамирович

Содержание

# 1 Цель работы

Цель работы состоит в том, чтобы приобрести навыки разработки программ с использованием подпрограмм и приобретение знание методов отладки с помощью GDB и его основных возможностей.

# 2 Выполнение лабораторной работы

## 2.1 Реализация подпрограмм в NASM

Я создал новую директорию и перешел в нее, чтобы выполнить лабораторную работу номер 9. Затем я создал файл lab9-1.asm.

Рассмотрим программу, которая вычисляет арифметическое. Подпрограмма для расчета используется для получения выражения . В этом примере переменная вводится с клавиатуры. а сама формула вычисляется в рамках подпрограммы.(рис. [[1](#fig:001)]) (рис. [[2](#fig:002)])

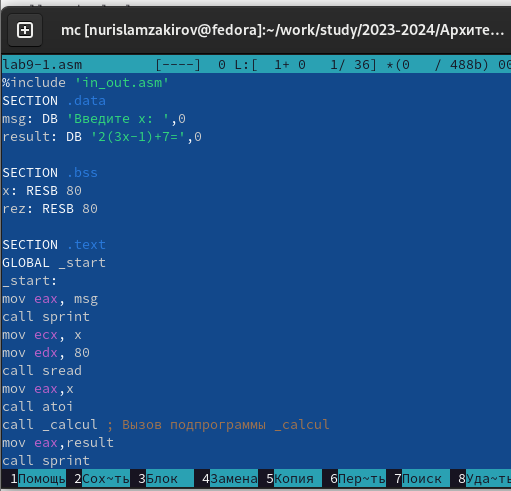


Figure 1: Код программы lab9-1.asm

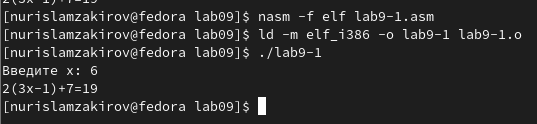


Figure 2: Компиляция и запуск программы lab9-1.asm

Следующим шагом было включение подпрограммы subcalcul в подпрограмму calcul и внесение изменений в текст программы. Это позволяет вычислить составное выражение , в котором также вводится значение с клавиатуры.  
Функции определены следующим образом: , .(рис. [[3](#fig:003)]) (рис. [[4](#fig:004)])

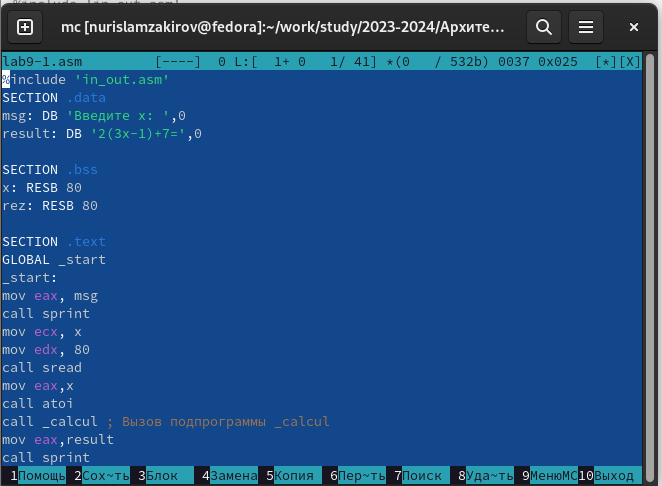


Figure 3: Код программы lab9-1.asm

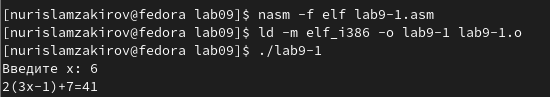


Figure 4: Компиляция и запуск программы lab9-1.asm

## 2.2 Отладка программам с помощью GDB

Я создал файл с именем lab9-2.asm, в котором содержится текст программы из Листинга 9.2. Эта программа отвечает за печать сообщения “Hello world!”.(рис. [[5](#fig:005)])

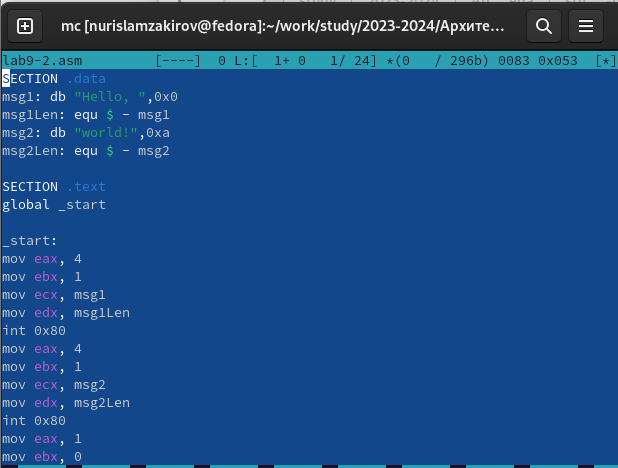


Figure 5: Код программы lab9-2.asm

Следующим шагом я скомпилировал файл и получил исполняемый файл. Чтобы добавить отладочную информацию для работы с отладчиком GDB, использовал ключ “-g”.

После чего я загрузил полученный исполняемый файл в отладчик GDB и проверил его работу, запустив программу с помощью команды “r”. (рис. [[6](#fig:006)])

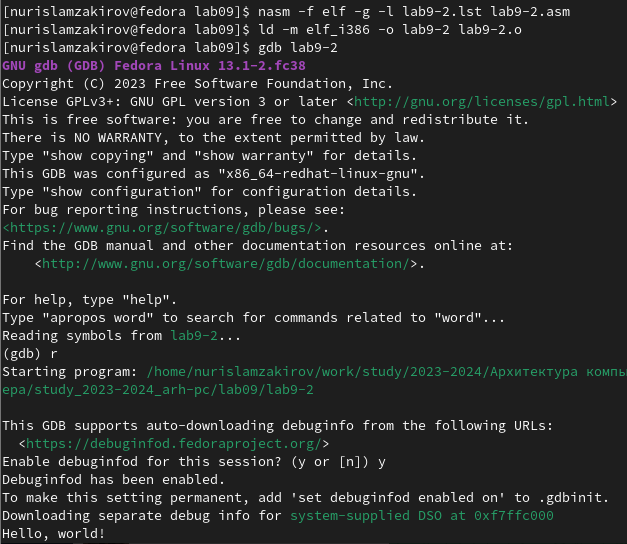


Figure 6: Компиляция и запуск программы lab9-2.asm в отладчике

Чтобы провести более подробный анализ программы, я установил точку остановки на метке «start» и запустил ее. Затем я просмотрел дизассемблированный код программы.(рис. [[7](#fig:007)]) (рис. [[8](#fig:008)])

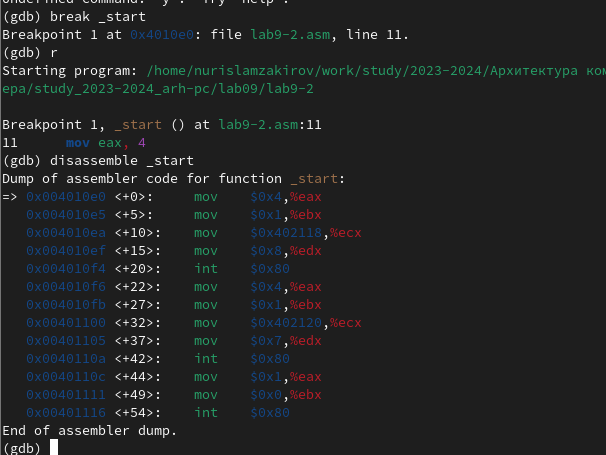


Figure 7: Дизассемблированный код

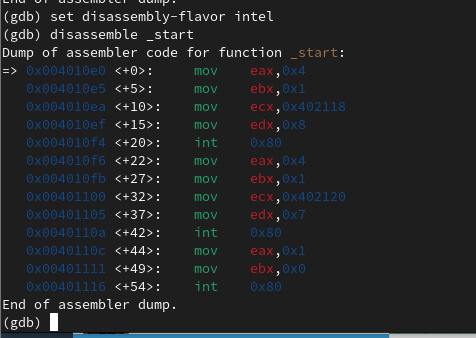


Figure 8: Дизассемблированный код в режиме интел

### 2.2.1 Точки остановки

Я использовал команду «info breakpoints» или «i b», чтобы проверить точку остановки по имени метки «\_start». Затем я установил еще одну точку остановки по адресу инструкции, определив адрес предыдущей инструкции «mov ebx, 0x0». (рис. [[9](#fig:009)])

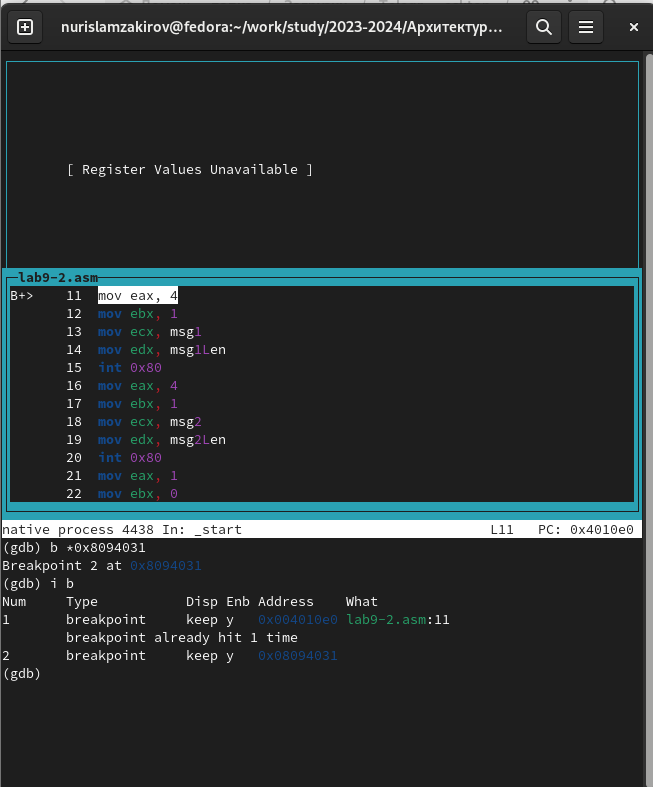


Figure 9: Точка остановки

### 2.2.2 Работа с данными программы в GDB

В отладчике GDB можно просматривать содержимое ячеек памяти и регистров, а также изменять значения регистров и переменных. Выполнил 5 инструкций с помощью команды ‘si’ и отследил изменение значений регистров. (рис. [[10](#fig:010)]) (рис. [[11](#fig:011)])

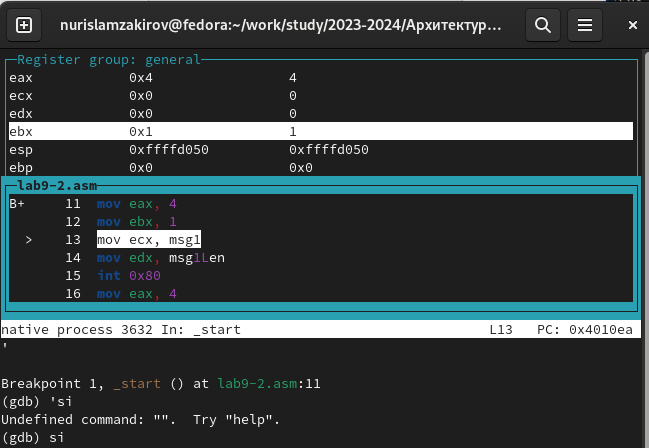


Figure 10: Изменение регистров

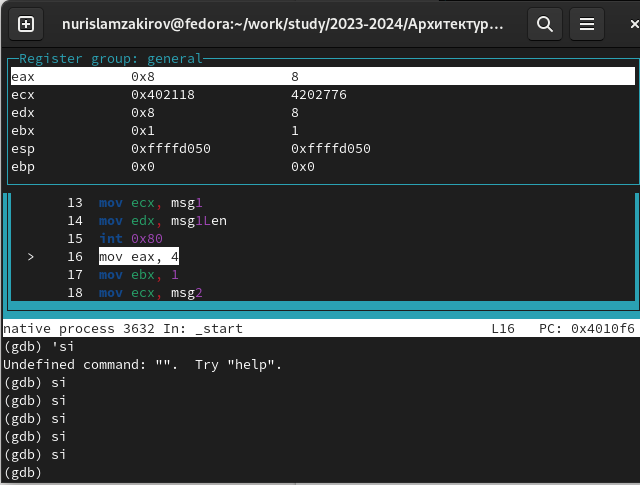


Figure 11: Изменение регистров

Просмотрел значение переменной msg1 по имени и получил нужные данные.

Команда set была использована для изменения значения ячейки памяти или регистра. указав в качестве аргумента имя регистра или адрес. изменил начало переменной msg1. (рис. [[12](#fig:012)])(рис. [[13](#fig:013)])

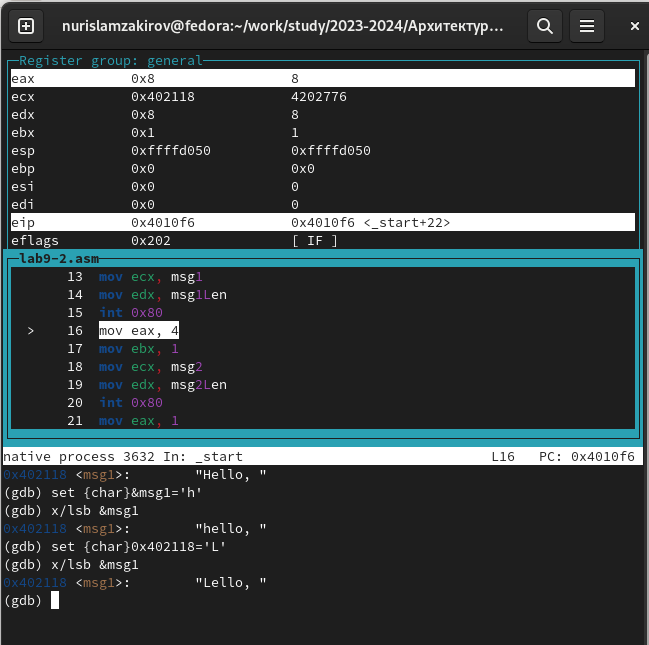


Figure 12: Изменение значения переменной

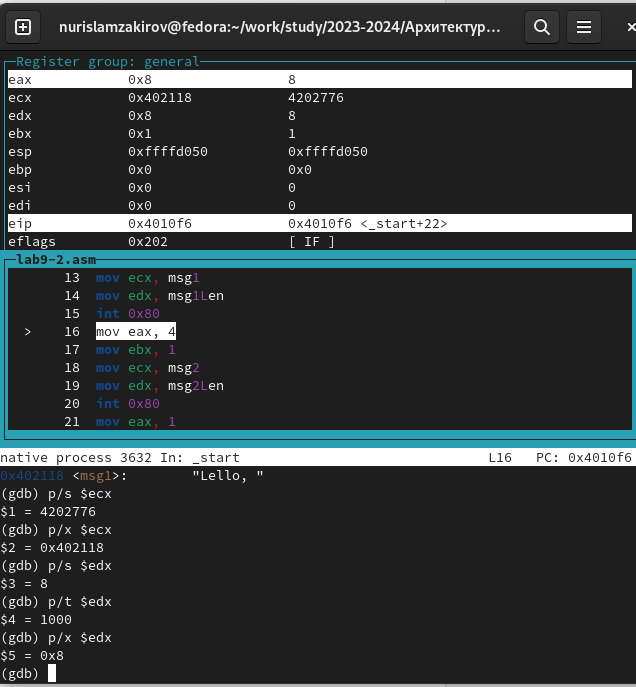


Figure 13: Вывод значения регистра

С помощью команды set изменил значение регистра ebx на нужное значение. (рис. [[14](#fig:014)])

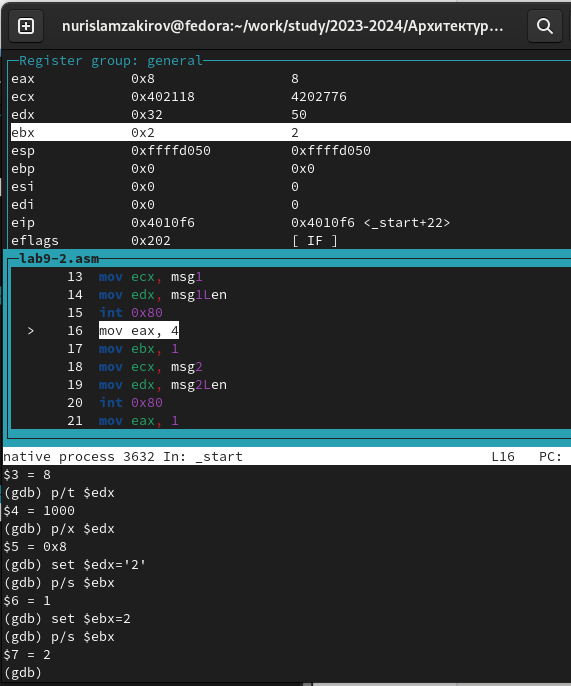


Figure 14: Вывод значения регистра

### 2.2.3 Обработка аргументов командной строки в GDB

Скопировал файл lab8-2.asm, который был создан во время выполнения лабораторной работы номер 8, и он содержит программу, которая позволяет выводить аргументы командной строки. скопировал файл и создал исполняемый файл.

Используя ключ —args, можно загрузить программу с аргументами в gdb, а затем загрузить исполняемый файл в отладчик с этими аргументами.

Установил точку останова и запустил первую инструкцию программы.

В регистре esp хранится адрес вершины стека, который содержит количество аргументов командной строки, в том числе имя программы. По этой ссылке можно найти число, показывающее количество аргументов. В данном случае количество аргументов, включая имя программы lab9-3, а также сами аргументы: аргумент1, аргумент2 и «аргумент 3», видно.

Просмотрел остальные позиции стека. По адресу [esp+4] находится адрес в памяти, где располагается имя программы. По адресу [esp+8] хранится адрес первого аргумента, по адресу [esp+12] - второго и так далее. (рис. [[15](#fig:015)])

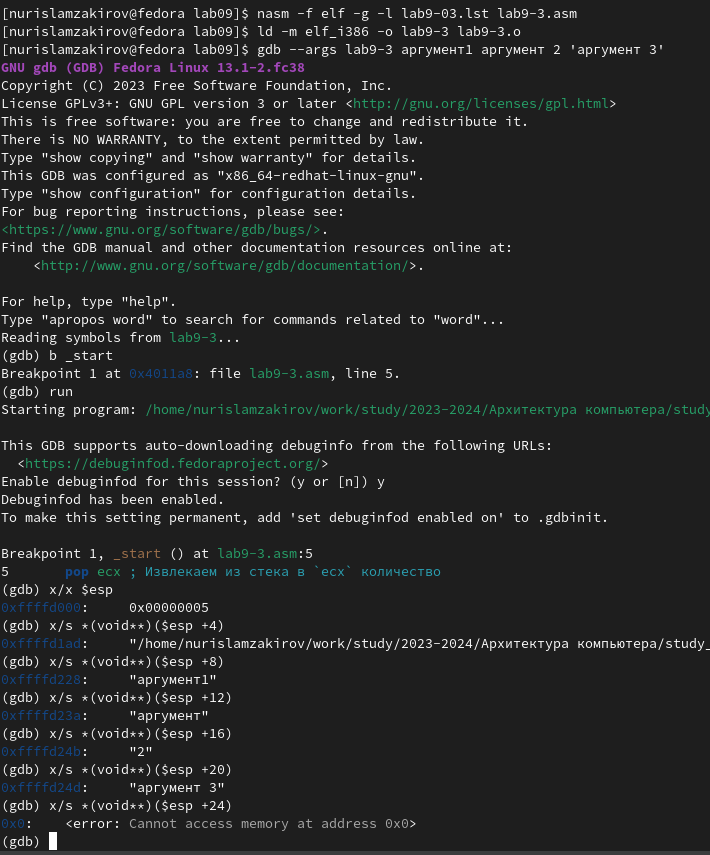


Figure 15: Вывод значения регистра

Шаг изменения адреса равен 4, так как каждый следующий адрес на стеке находится на расстоянии 4 байт от предыдущего ([esp+4], [esp+8], [esp+12]).

## 2.3 Задание для самостоятельной работы

Изменил программы из лабораторной работы No8 (Задание No1 для самостоятельной работы), добавив подпрограмму для вычисления значения функции f(x).Мой вариант был 1. (рис. [[16](#fig:016)]) (рис. [[17](#fig:017)])



Figure 16: Код программы prog-1.asm

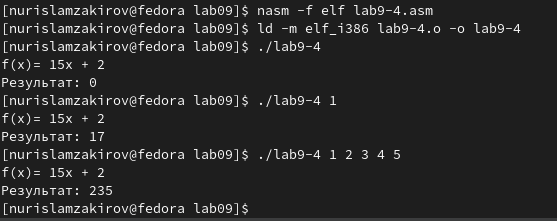


Figure 17: Компиляция и запуск программы prog-1.asm

В листинге приведена программа вычисления выражения . При запуске данная программа дает неверный результат. Проверил это, анализируя изменения значений регистров с помощью отладчика GDB.

Определил ошибку - перепутан порядок аргументов у инструкции add. Также обнаружил, что по окончании работы в edi отправляется ebx вместо eax.(рис. [[18](#fig:018)])

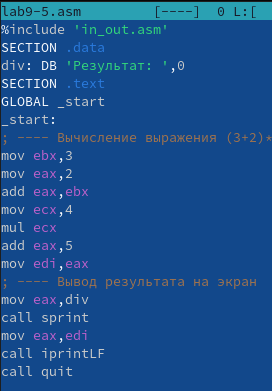


Figure 18: Код с ошибкой

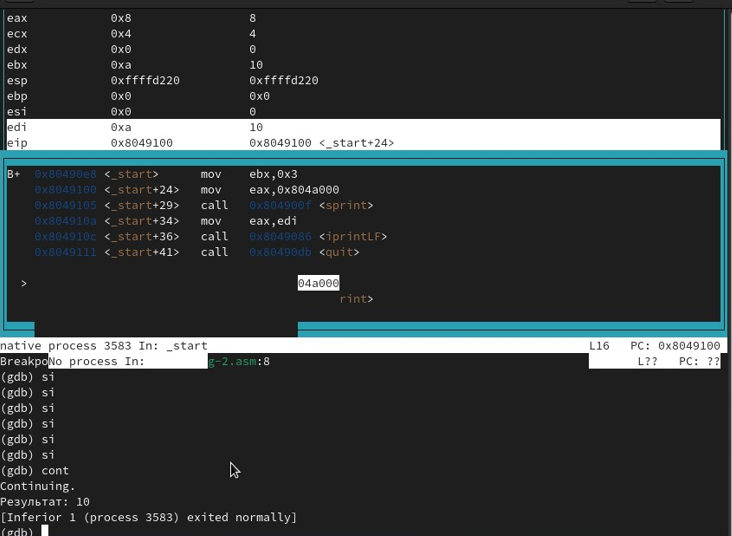


Figure 19: Отладка

Отмечу, что перепутан порядок аргументов у инструкции add и что по окончании работы в edi отправляется ebx вместо eax (рис. [[19](#fig:019)])

Исправленный код программы (рис. [[20](#fig:020)]) (рис. [[21](#fig:021)])

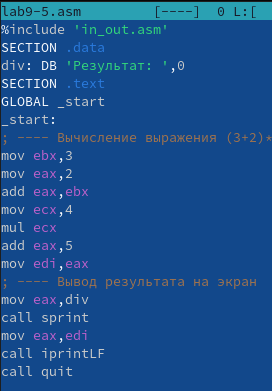


Figure 20: Код исправлен

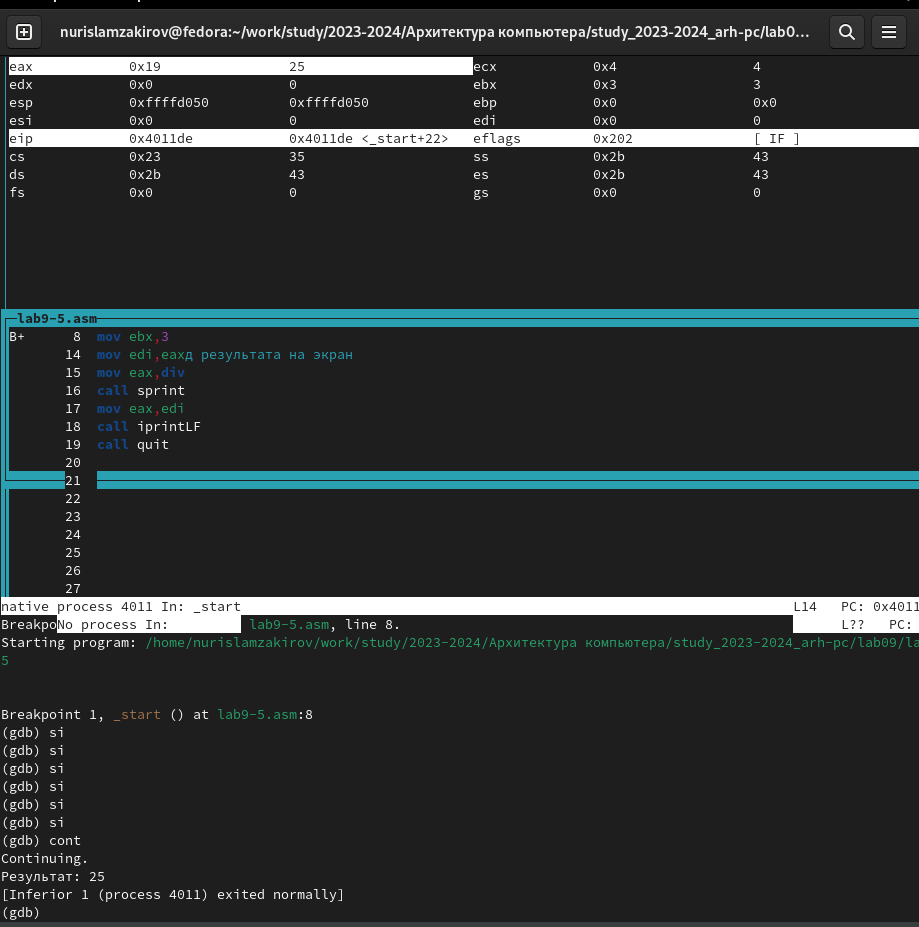


Figure 21: Проверка работы

# 3 Выводы

В ходе лаболаторной работы, я приобрел навыки разработки программ с использованием подпрограмм и знание методов отладки с помощью GDB и его основных возможностей. # Список литературы 1. [Лабораторная работа №7](https://esystem.rudn.ru/pluginfile.php/2089096/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%969.%20%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B.%20%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA%20..pdf)