Лабораторная работа №9

Понятие подпрограммы. Отладчик GDB

Бунин Арсений Викторович

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

# 2 Задание

1. Реализация подпрограмм в NASM
2. Отладка программ с помощью GDB
3. Выполнить индивидуальное задание по отладке программы на Ассемблере
4. Загрузить файлы на github

# 3 Теоретическое введение

Отладчик GDB (как и любой другой отладчик) позволяет увидеть, что происходит «внутри» программы в момент её выполнения или что делает программа в момент сбоя. GDB может выполнять следующие действия: \* начать выполнение программы, задав всё, что может повлиять на её поведение; \* остановить программу при указанных условиях; \* исследовать, что случилось, когда программа остановилась; \* изменить программу так, чтобы можно было поэкспериментировать с устранением эффектов одной ошибки и продолжить выявление других.

Подпрограмма — это, как правило, функционально законченный участок кода, который можно многократно вызывать из разных мест программы. В отличие от простых переходов из подпрограмм существует возврат на команду, следующую за вызовом. Если в программе встречается одинаковый участок кода, его можно оформить в виде подпрограммы, а во всех нужных местах поставить её вызов. При этом подпрограмма будет содержаться в коде в одном экземпляре, что позволит уменьшить размер кода всей программы.

# 4 Выполнение лабораторной работы

Запустили исполняемый файл в отладчике gdb(рис. [1](#fig:fig1))

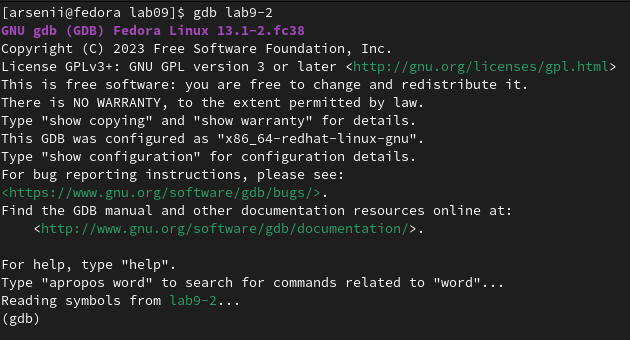


Figure 1: Запуск отладчика

Проверили работу программы в отладчике(рис. [2](#fig:fig2))

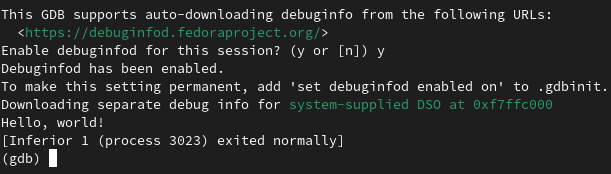


Figure 2: Результат работы программы

Установили первую точку останова (рис. [3](#fig:fig3)).

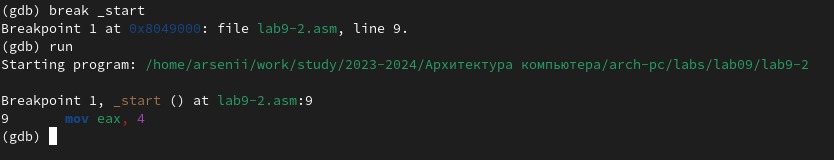


Figure 3: Первая точка останова

Дизассемблирование программы, начиная с точки старта(рис. [4](#fig:fig4)).

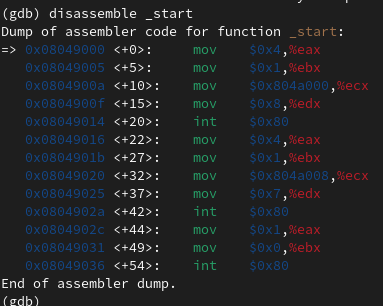


Figure 4: Дизассемблированный код программы

Дизассемблированный код программы со стилем Интел(рис. [5](#fig:fig5))



Figure 5: Стиль Интел

Переход в псевдографический режим (рис. [6](#fig:fig6))

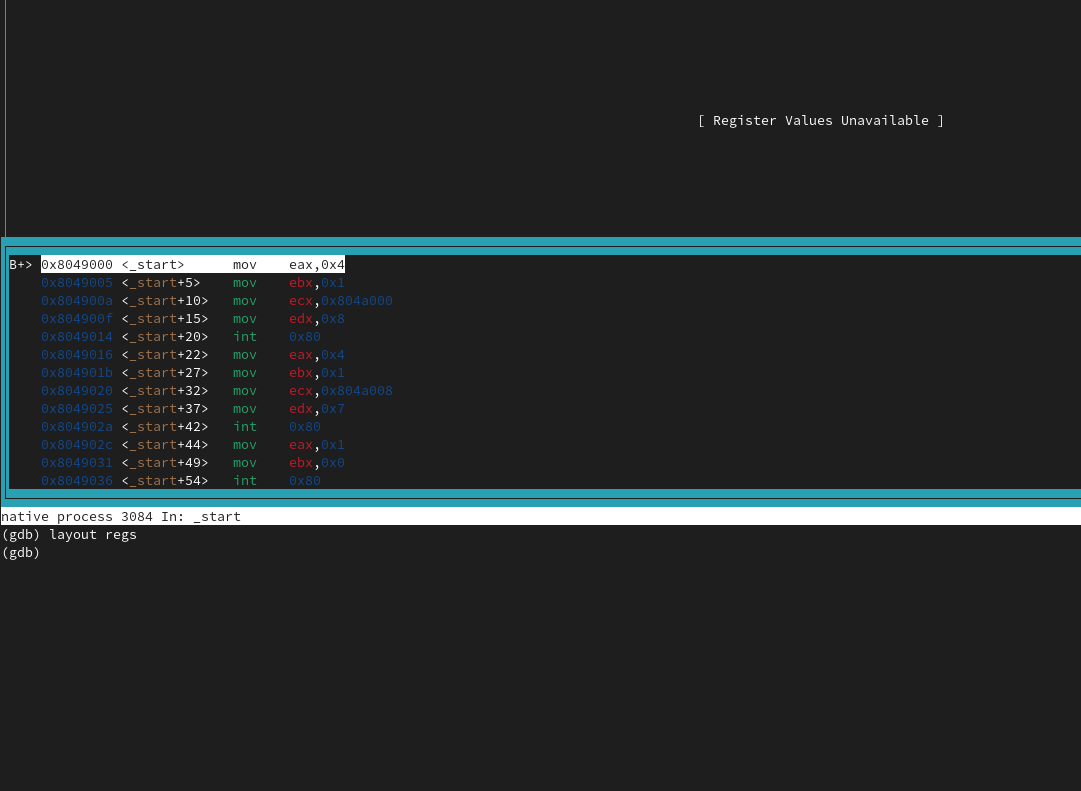


Figure 6: Псевдографический режим

(рис. [7](#fig:fig7))

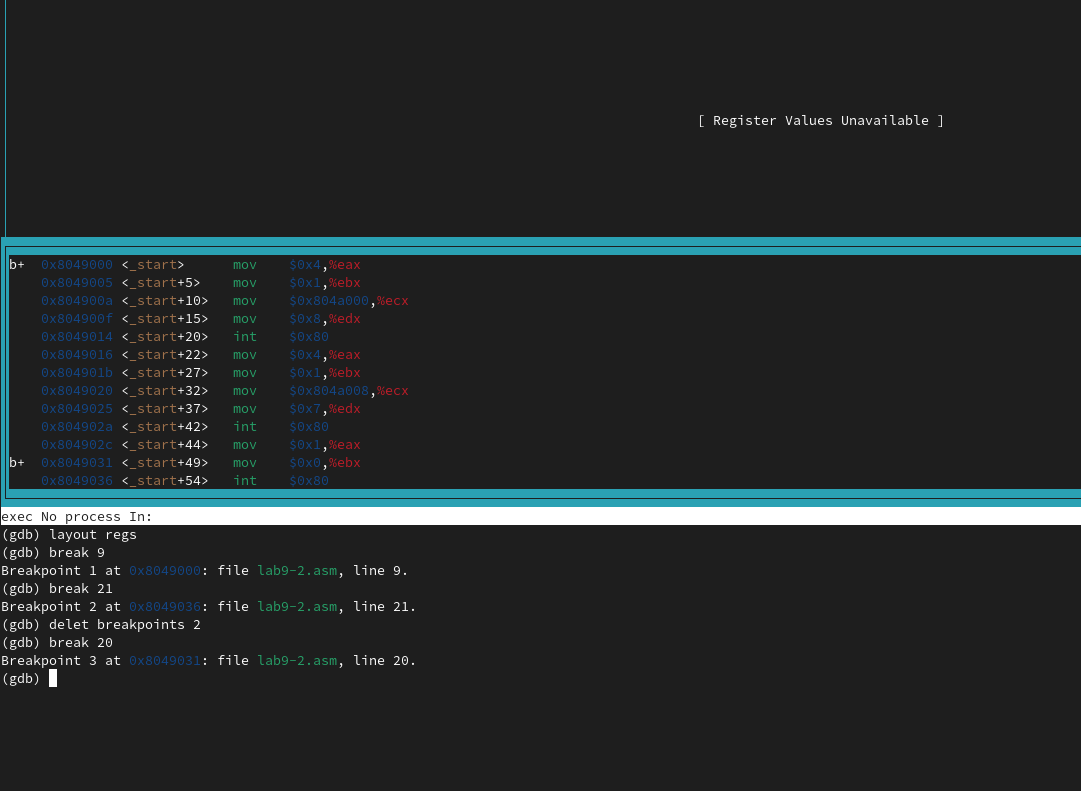


Figure 7: Текст программы

Установили вторую точку останова (рис. [8](#fig:fig8))



Figure 8: Вторая точка останова

Вывод значений переменных (рис. [9](#fig:fig9)) (рис. [10](#fig:fig10))

Figure 9: первая текстовая переменная

Figure 9: первая текстовая переменная

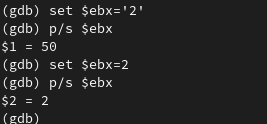


Figure 10: вторая текстовая переменная

Установка значений переменных в программе. Вывод переменной отличается из-за формата данных(рис. [11](#fig:fig11)).

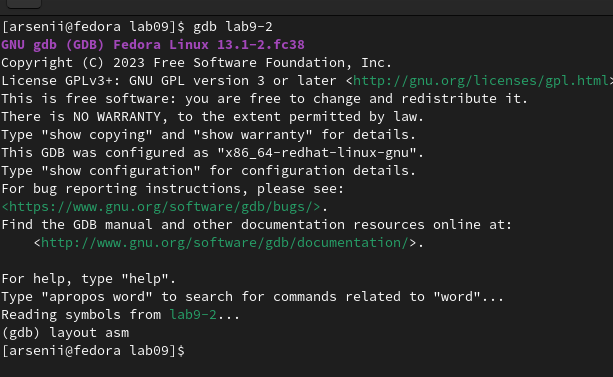


Figure 11: Установка значений переменной

Запуск второй программы с аргументами (рис. [12](#fig:fig12))

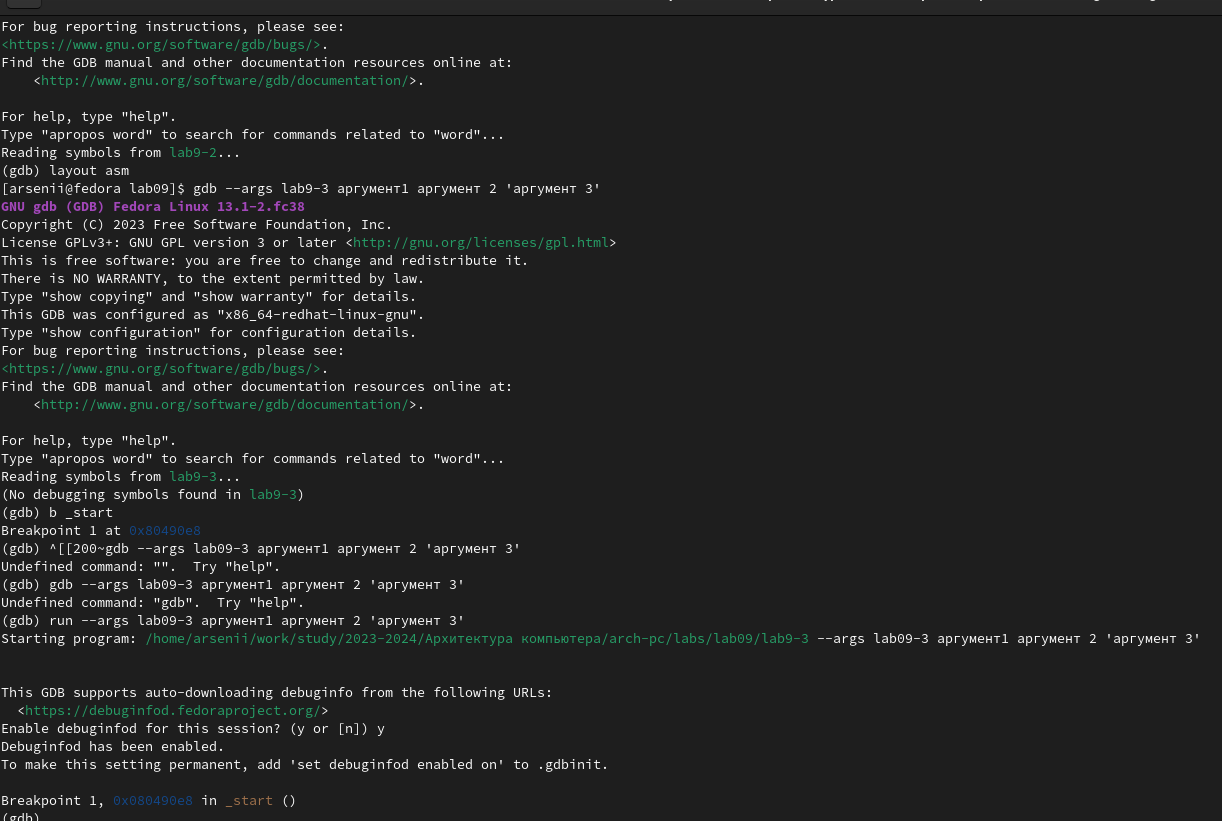


Figure 12: Запуск второй программы

Вывод числа аргументов (рис. [13](#fig:fig13))

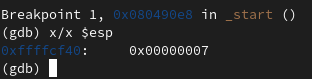


Figure 13: Число переменных

Вывод значений аргументов (рис. [14](#fig:fig14))

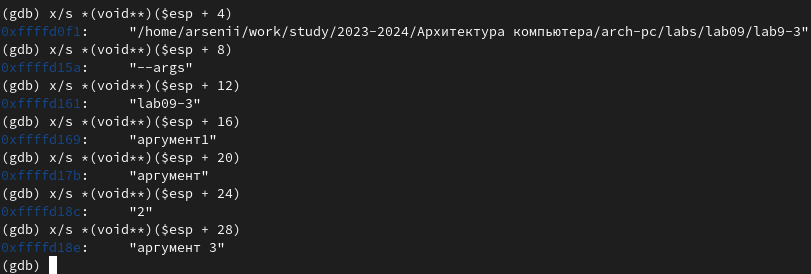


Figure 14: Текст программы

# 5 Самостоятельная работа

1. Преобразовать программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции f(x) как подпрограмму

Код программы (рис. [15](#fig:fig15))

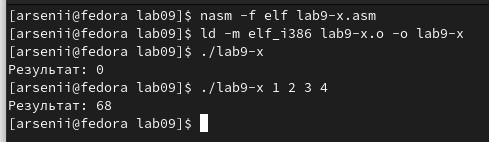


Figure 15: Код программы

Результат работы программы (рис. [16](#fig:fig16))

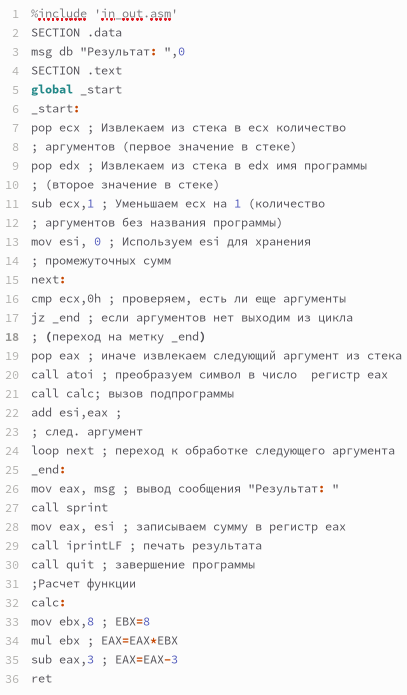


Figure 16: Работа программы

1. Отладка программы с помощью GDB Linux

Программа выдает ошибочное значение (рис. [17](#fig:fig18))

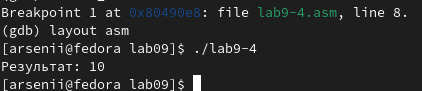


Figure 17: Неправильная работа программы

В отладчике видим, что умножение происходит в регистре eax, тогда как мы положили значение суммы в регистр ebx. Это видно по значениям регистров eax и ebx(рис. [18](#fig:fig17))

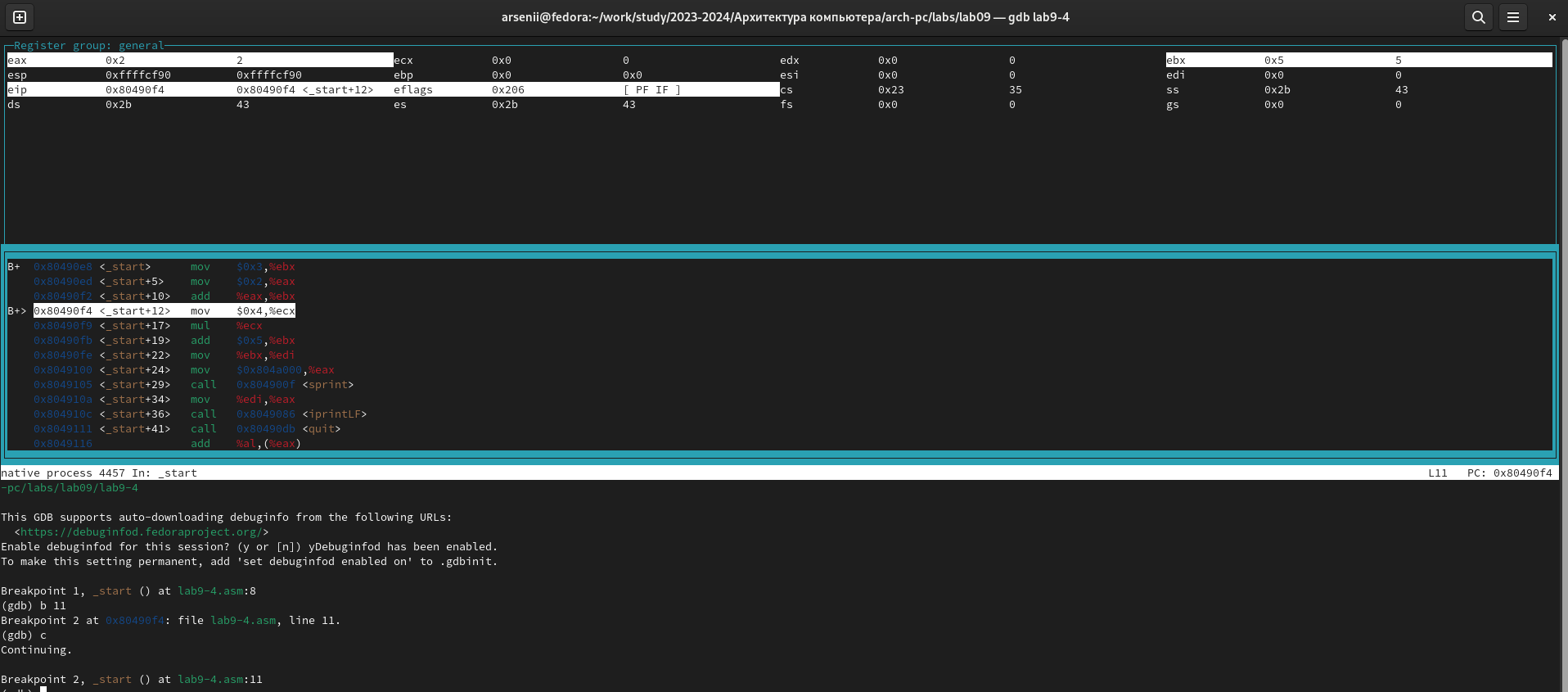


Figure 18: Нахождение ошибки в отладчике

Исправленный код программы (рис. [19](#fig:fig20))

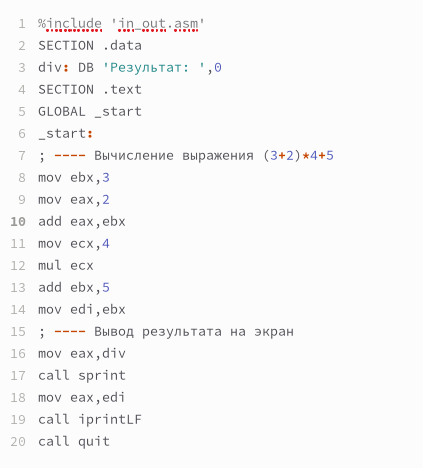


Figure 19: Код программы

Исправленная программа выдает ошибочное значение (рис. [17](#fig:fig18))

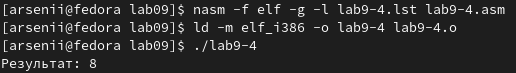


Figure 20: Неправильная работа программы

В отладчике видим, что второе сложение происходит в регистре ebx, тогда как мы положили значение суммы в регистр eax (рис. [21](#fig:fig21))

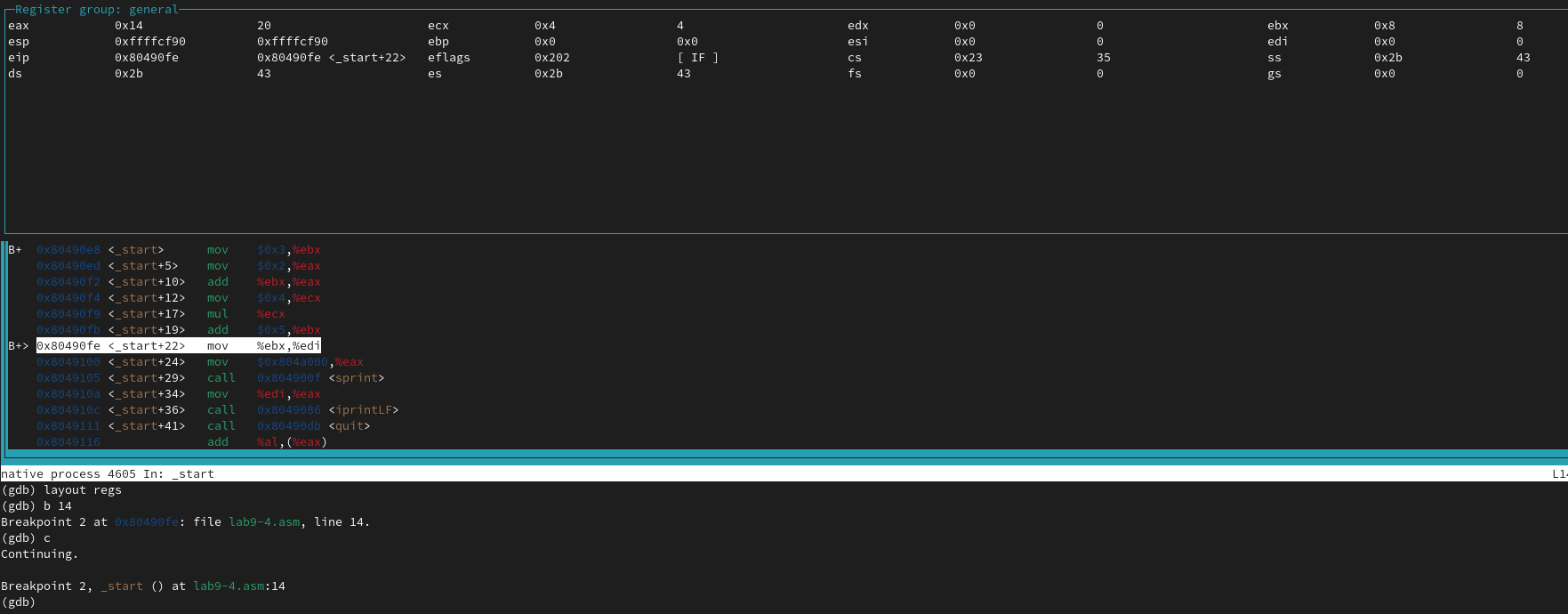


Figure 21: Нахождение ошибки в отладчике

Исправленный код программы (рис. [22](#fig:fig23))

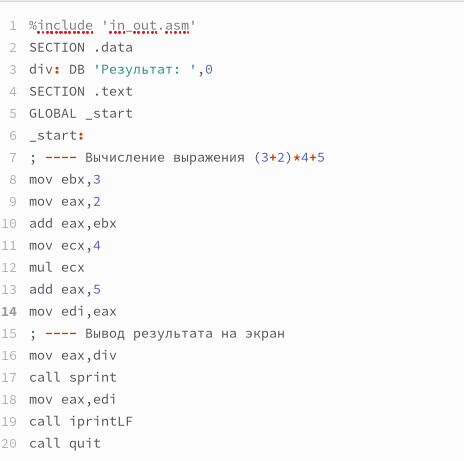


Figure 22: Код программы

Исправленная программа работает корректно (рис. [23](#fig:fig22))

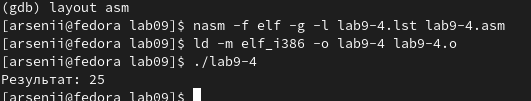


Figure 23: Корректная работа программы

# 6 Выводы

Приобрели навыки написания программ с использованием подпрограмм. Познакомились с методами отладки при помощи GDB и его основными возможностями.

# Список литературы