Отчёта по лабораторной работе 10

Понятие подпрограммы. Отладчик GDB.

Чесноков Артемий Павлович НПИбд-02-22

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1. Создаим файл lab10-1.asm:
2. Введём в этот файл программу из лабораторной(Листинг 10.1). (рис. 1, 2)

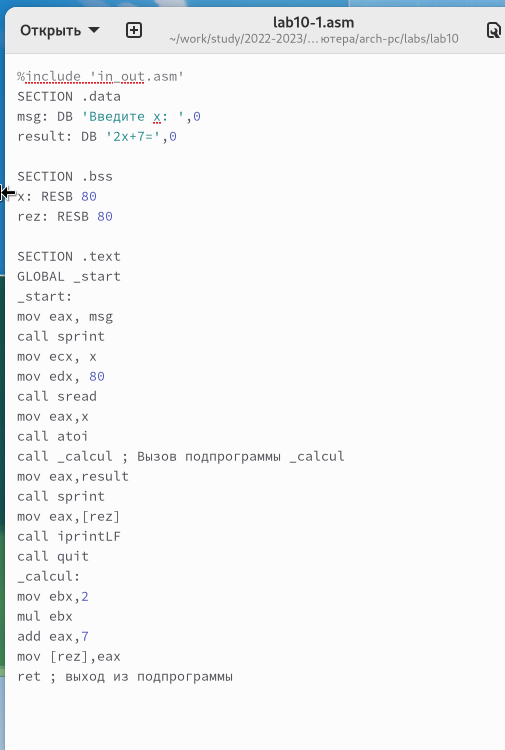


Рис. 1: Файл lab10-1.asm

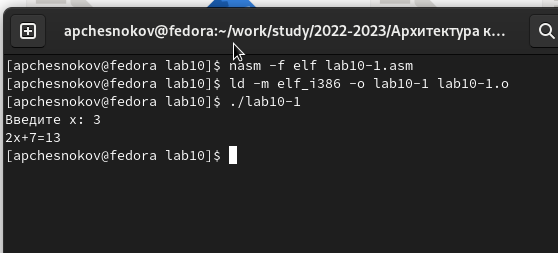


Рис. 2: Работа программы lab10-1.asm

1. Изменим текст программы, по задаче из лабораторной, чтобы вычислялось f(g(x)), где f(x) = 2x + 7, g(x) = 3x − 1(рис. 3, 4)

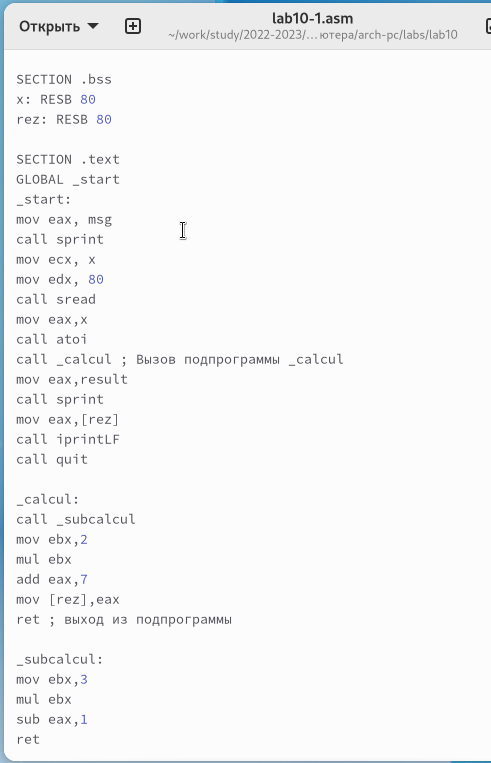


Рис. 3: Файл lab10-1.asm

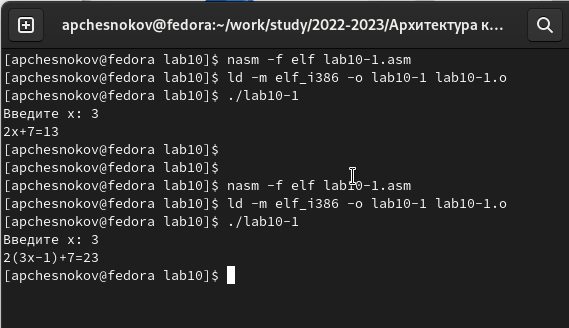


Рис. 4: Работа программы lab10-1.asm

1. Создадим lab10-2.asm с текстом программы из Листинга 10.2: (рис. 5)

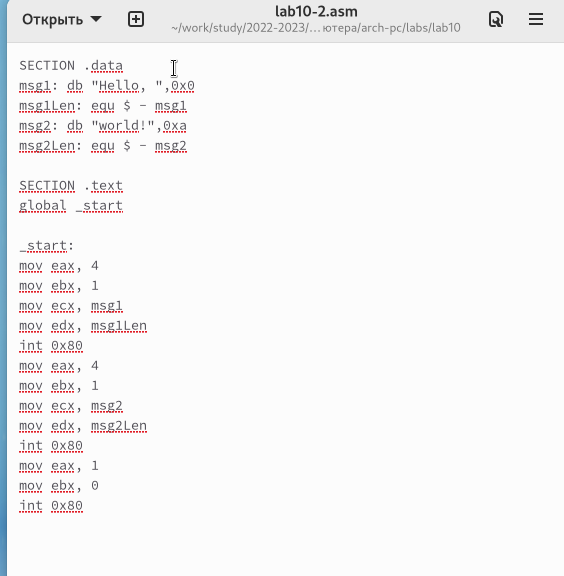


Рис. 5: Файл lab10-2.asm

Проверьте работу программы, запустив ее в оболочке GDB:(рис. 6)

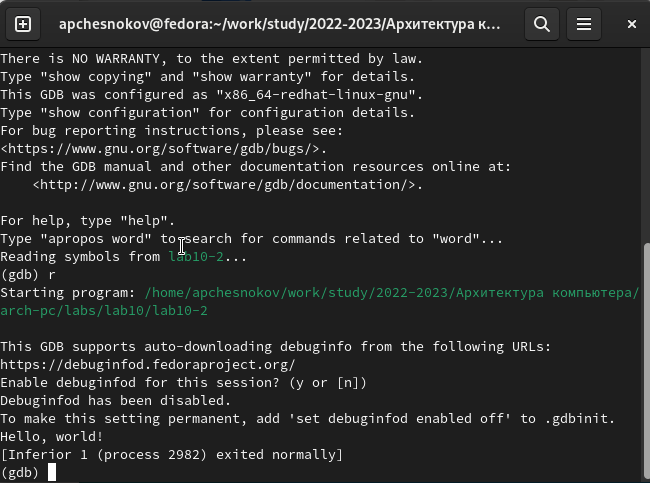


Рис. 6: Работа программы lab10-2.asm в отладчике

Для более подробного анализа программы установите брейкпоинт на метку start и запустим её. Так же посмотрим диассемблированный код (рис. 7, 8)

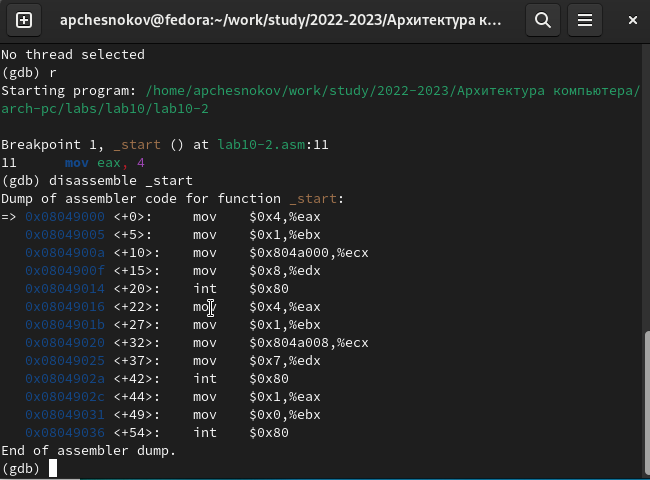


Рис. 7: дисассимилированный код

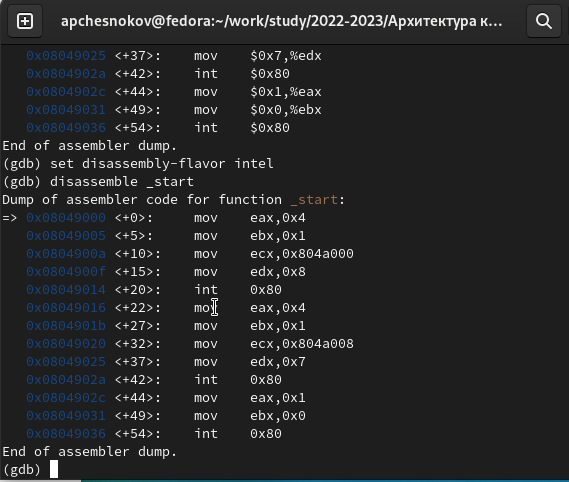


Рис. 8: дисассимилированный код в режиме интел

На предыдущих шагах была установлена точка останова по имени метки (\_start). Проверьте это с помощью команды info breakpoints (кратко i b) Установим еще одну точку останова по адресу инструкции. Адрес инструкции можно увидеть в средней части экрана в левом столбце соответствующей инструкции. Определите адрес предпоследней инструкции (mov ebx,0x0) и установите точку.(рис. 9)

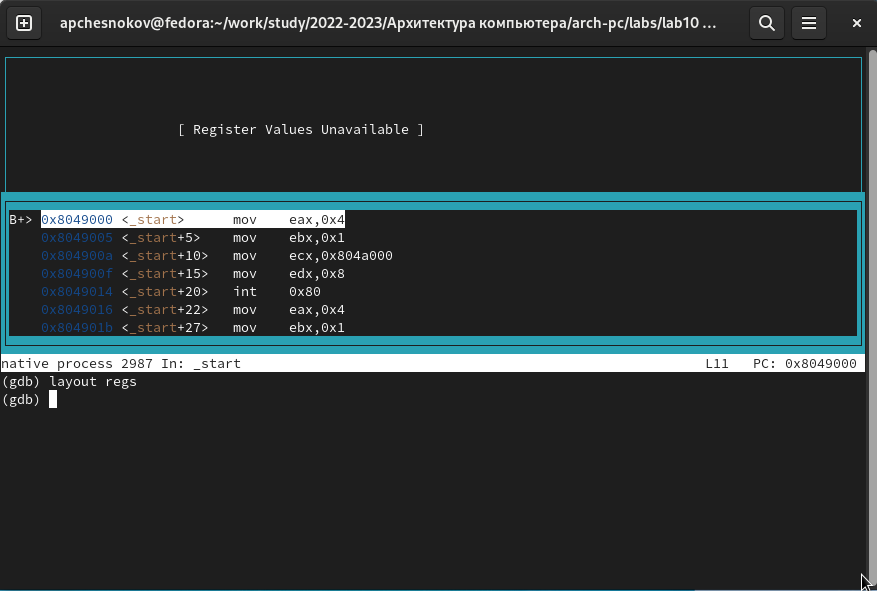


Рис. 9: точка остановки

Выполним инструкций с помощью команды si и проследим за изменением значений регистров. (рис. 11 12)

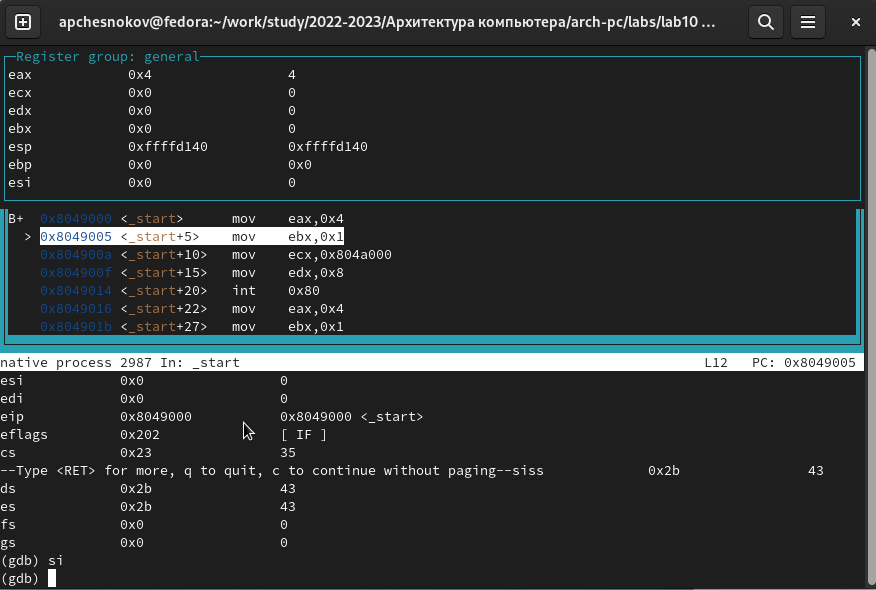


Рис. 10: изменение регистров

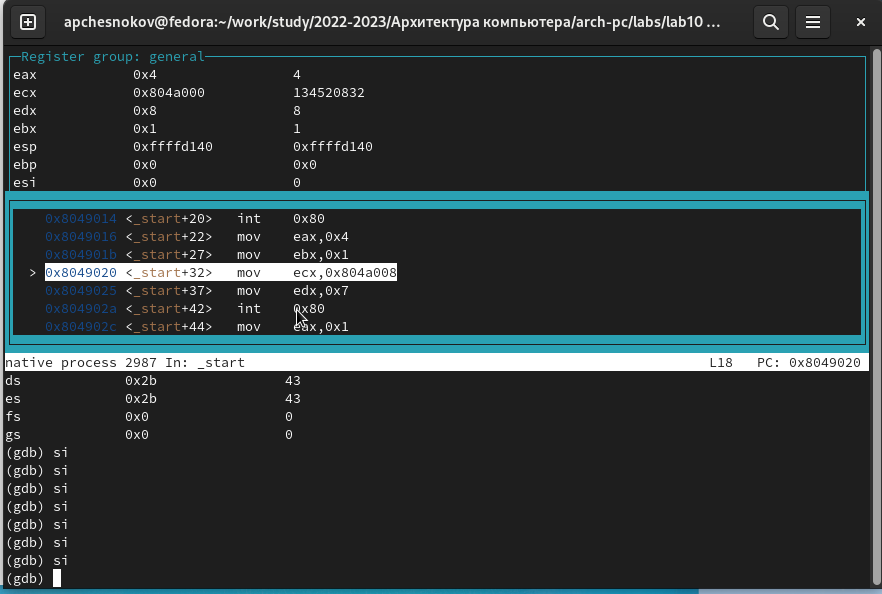


Рис. 11: изменение регистров

Посмотрим значение переменной msg1 Посмотрим значение переменной msg2 Измените первый символ переменной msg1 Замените любой символ во второй переменной msg2. (рис. 12)

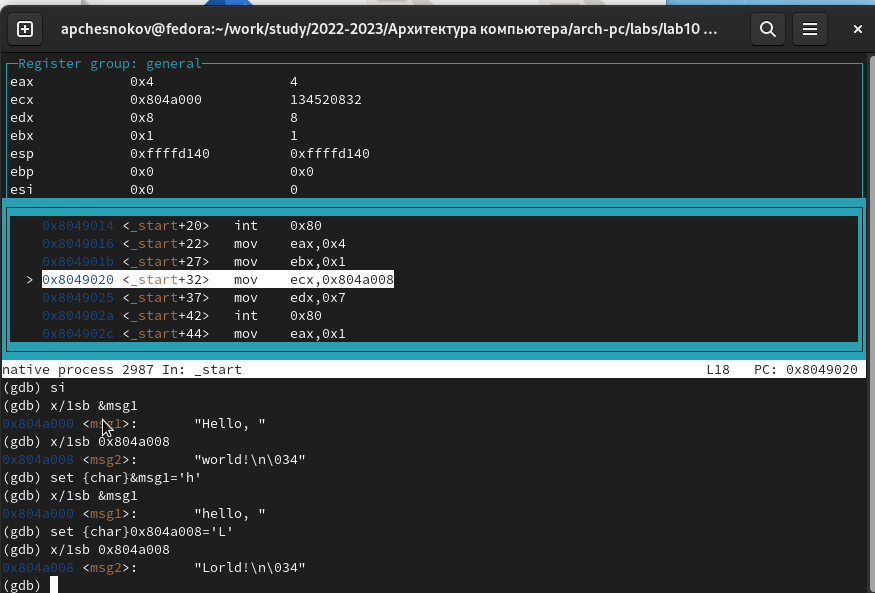


Рис. 12: изменение значения переменной

Выведете в различных форматах значение регистра edx. С помощью команды set изменим значение регистра ebx:(рис. 13)

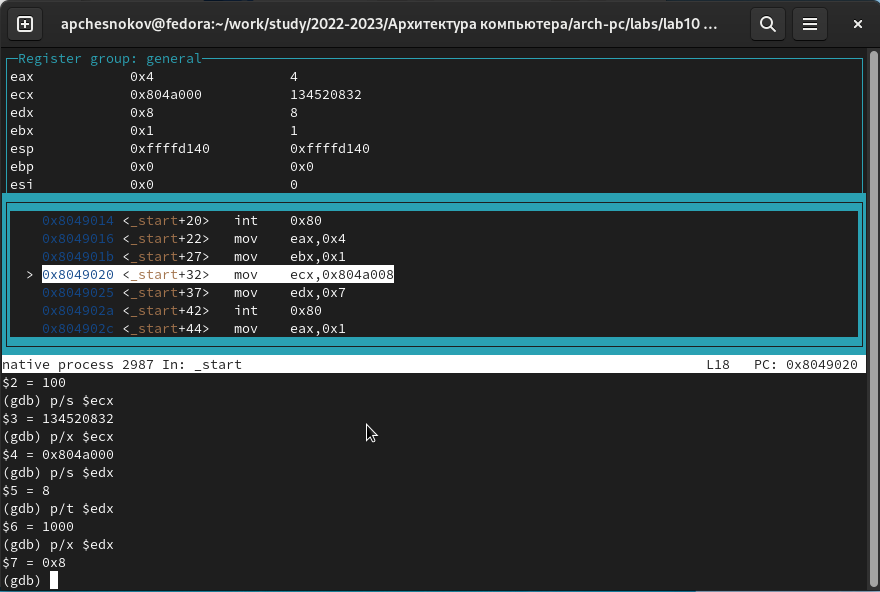


Рис. 13: вывод значения регистра

С помощью команды set изменим значение регистра ebx:(рис. 14)

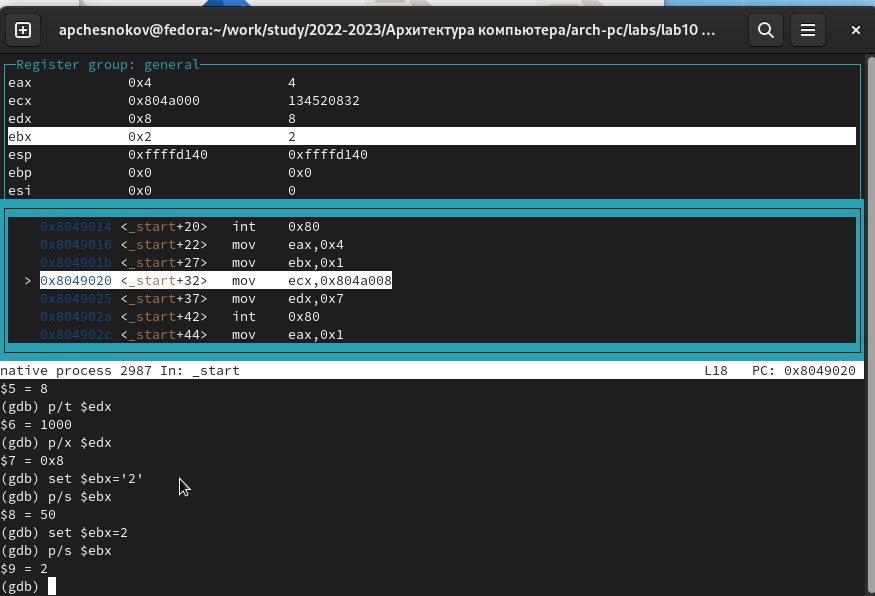


Рис. 14: вывод значения регистра

1. Скопируем файл lab9-2.asm. Создадим исполняемый файл. Загрузим исполняемый файл в отладчик, указав аргументы.

Для начала установим точку остановки перед первой инструкцией в программе и запустим ее.

Как видно, число аргументов равно 5 – это имя программы lab10-3 и непосредственно аргументы: аргумент1, аргумент, 2 и ‘аргумент 3’.

Посмотрим остальные позиции стека (рис. 15)

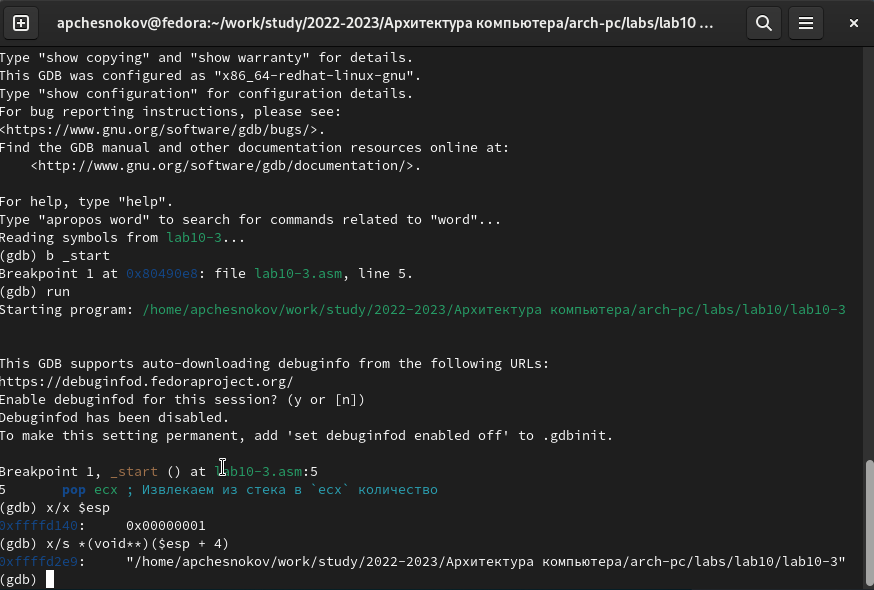


Рис. 15: вывод значения регистра

шаг равен размеру переменной - 4 байтам., поэтому шаг изменения адреса равен 4 ([esp+4], [esp+8], [esp+12]

# 3 Самостоятельная работы

1. Преобразуем программу из самостоятельного задания 1, лабораторной работы №9 , реализовав вычисление значения функции f(x) как подпрограмму. (рис. 16 17)

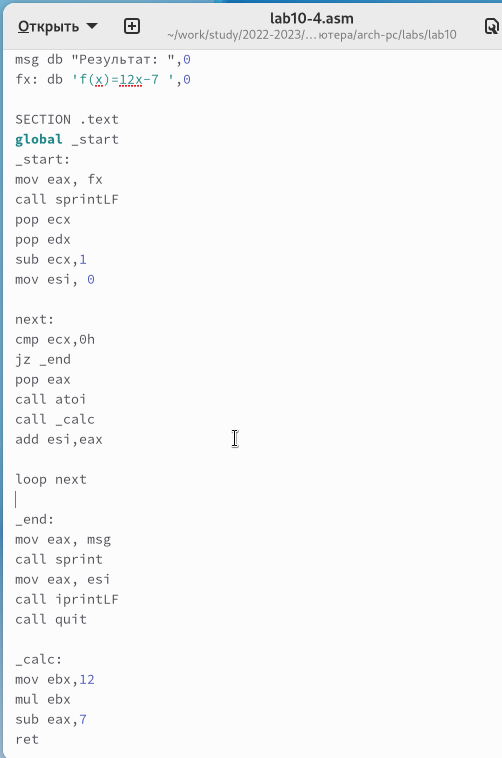


Рис. 16: Файл lab10-4.asm

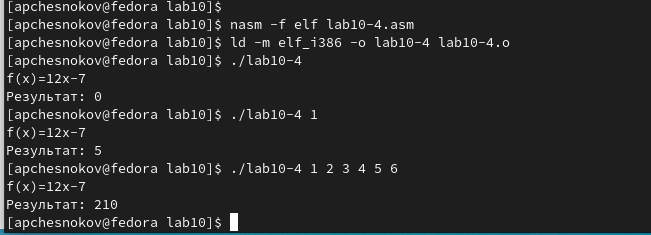


Рис. 17: Работа программы lab10-4.asm

1. Скопируем из листинга программу, неверно рабочую. Проверим её с помощью отладчика GDB, определим ошибку и исправим ее.(рис. 18 19 20 21)

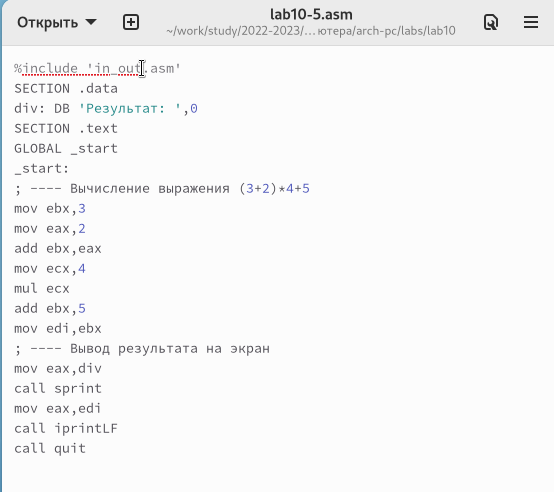


Рис. 18: код с ошибкой

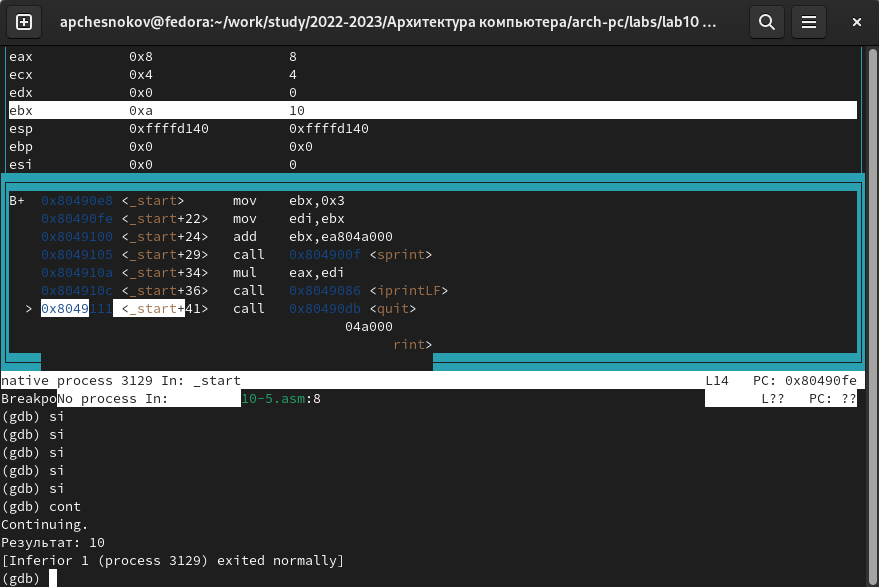


Рис. 19: отладка

Перепутан порядок аргументов у инструкции add. Так же по окончании работы в edi отправляется ebx вместо eax. Исправим програму и запустим её. Убедимся в работоспособности

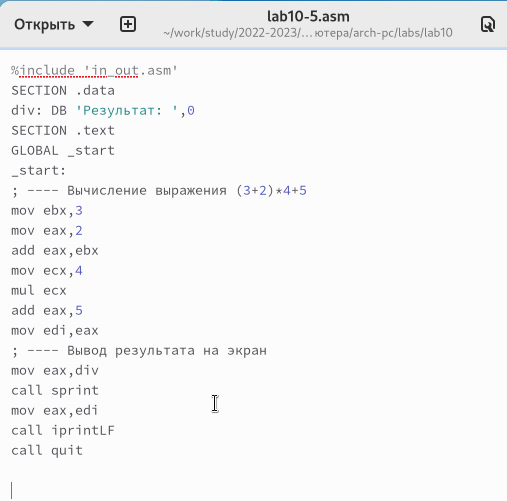


Рис. 20: код исправлен

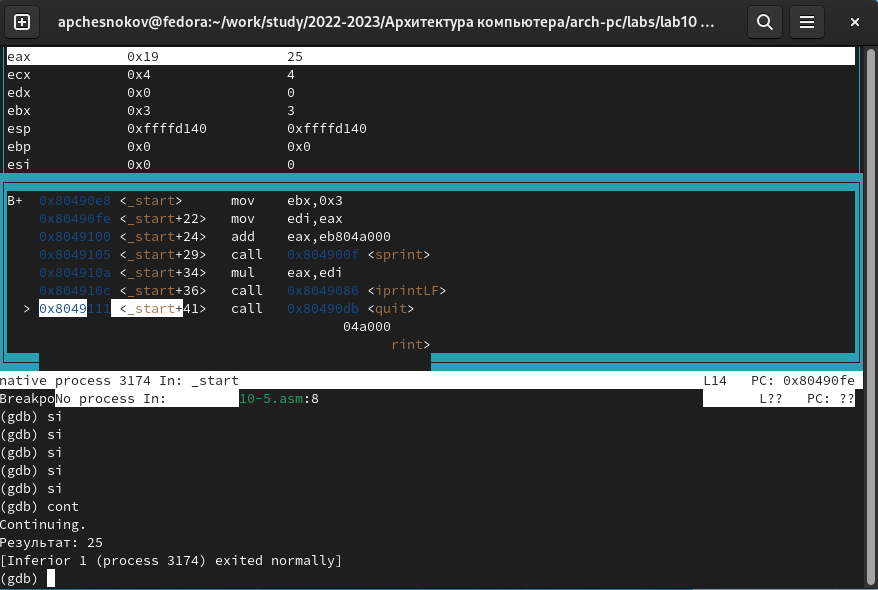


Рис. 21: проверка работы

# 4 Выводы

Приобрели навыки написания программ с использованием подпрограмм. Ознакомились с методами отладки при помощи GDB и его основными возможностями. https://github.com/Sinabon2004/study\_2022-2023\_arh-pc