Лабораторная работа номер 10

Сафин Андрей Алексеевич

Содержание

# 1 Цель работы

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

# 2 Задание

Написать и отладить ряд программ с использованием подпрограмм.

# 3 Ход лабораторной работы

1. Создан файл ~/work/arch-pc/lab10/lab10-1.asm с программой, вычисляющей f(x)=2x+7 с использованием подпрограммы (рис. 1). Ее работа проверена (рис. 2). В текст добавлена подпрограмма (рис. 3), вычисляющая g(x)=3x-1, и ссылка на неё вставлена в lab10-1.asm так, что вычисляется f(g(x)) (рис. **¿fig:004?**).

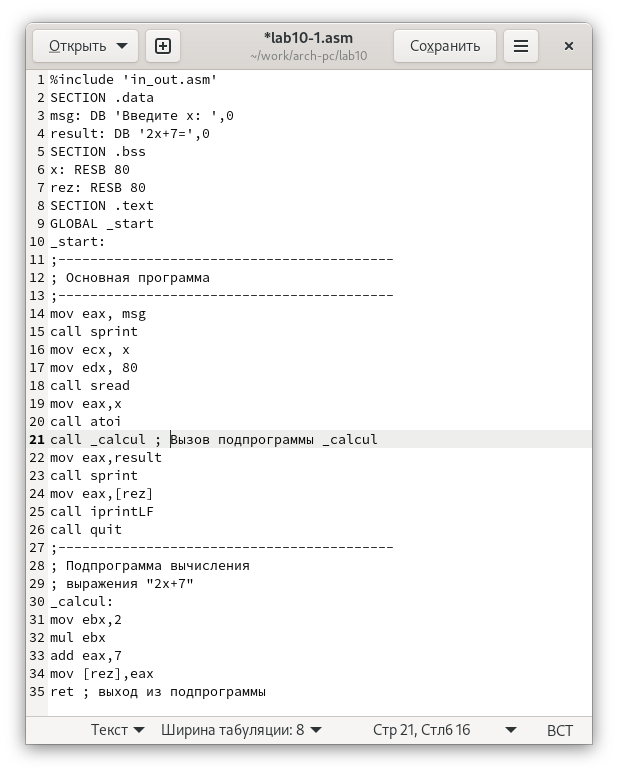


Рис. 1: Текст lab10-1.asm

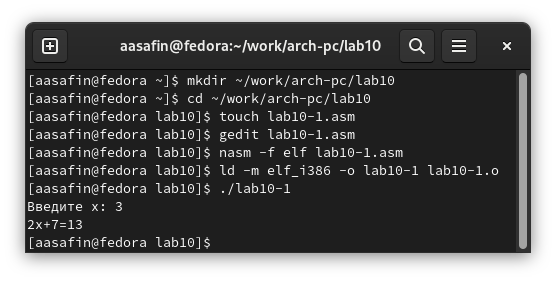


Рис. 2: Работа lab10-1.asm

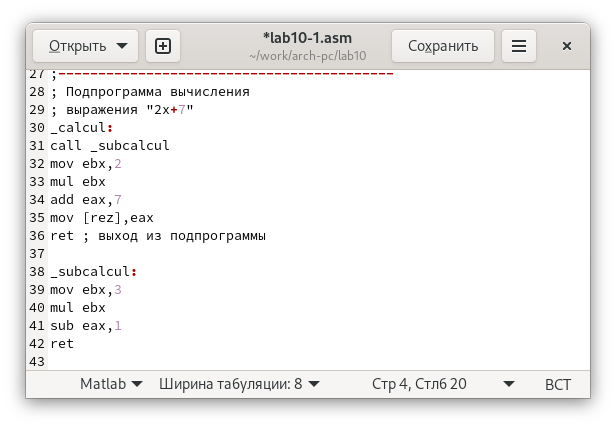


Рис. 3: Новый текст lab10-1.asm

Повторное выполнение lab10-1.asm

1. Создан файл lab10-2.asm с текстом программы из листинга 10.2 (рис. 4), печатающей Hello world. Файл оттранслирован, скомпанован и загружен в отладчик gdb (рис. 5). Проверена её работа с помощью run (рис. 6). Установлен брейкпоинт на \_start (рис. 7). С этой же метки программа дизассемблирована сначала в синтаксисе ATT (рис. 8), а затем в intel (рис. 9). Из наблюдаемых отличий, в ATT ставится $ перед численными операндами и адресам, и % перед регитсрами.

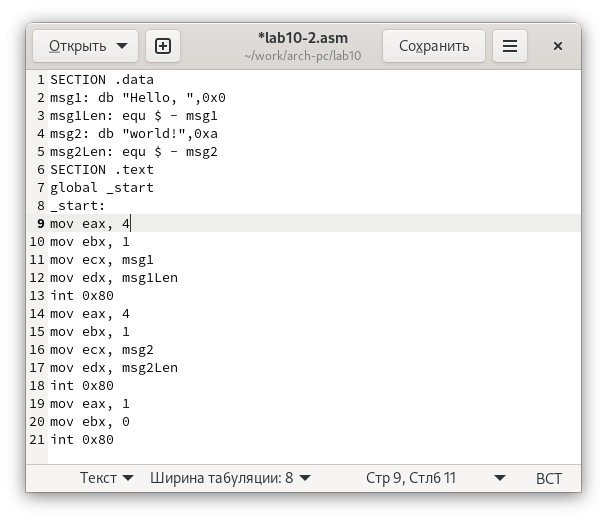


Рис. 4: Текст lab10-2.asm

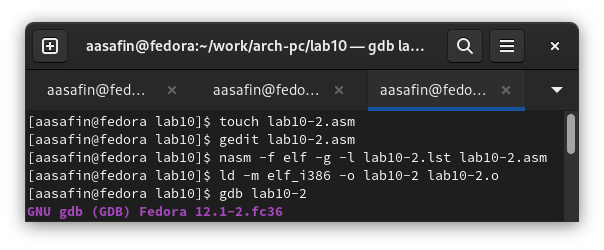


Рис. 5: Использование gdb к lab10-2

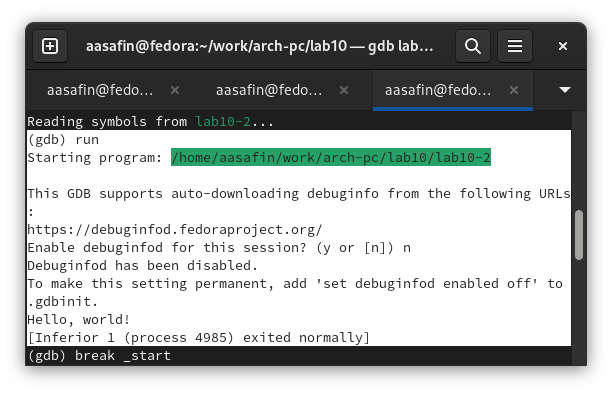


Рис. 6: Использование run

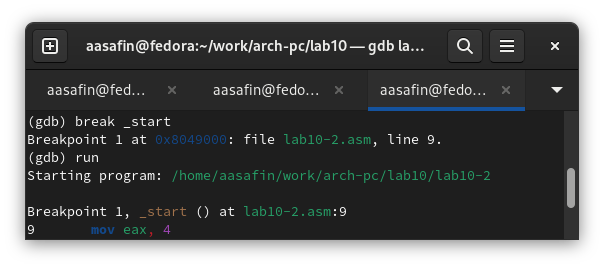


Рис. 7: Создание брейкпоинта



Рис. 8: Дизассемблированный код ATT

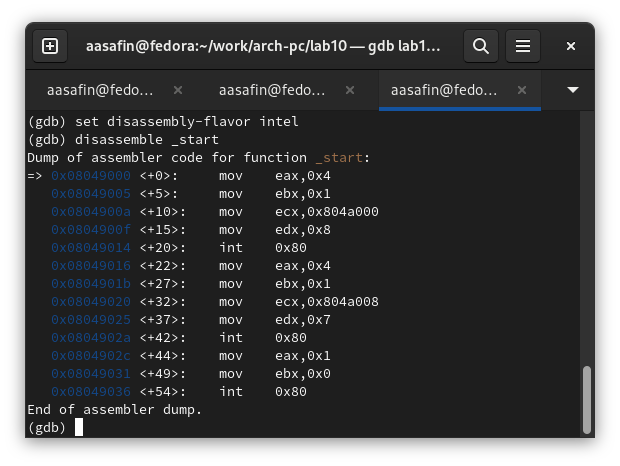


Рис. 9: Дизассемблированный код intel

Включен режим псевдографики, просмотрена информация по точкам останова, создана ещё одна. (рис. 10). С помощью stepi выполнено пять инструкций (рис. 11-15). Изменялись значения регистров eax, ebx, ecx и edx.

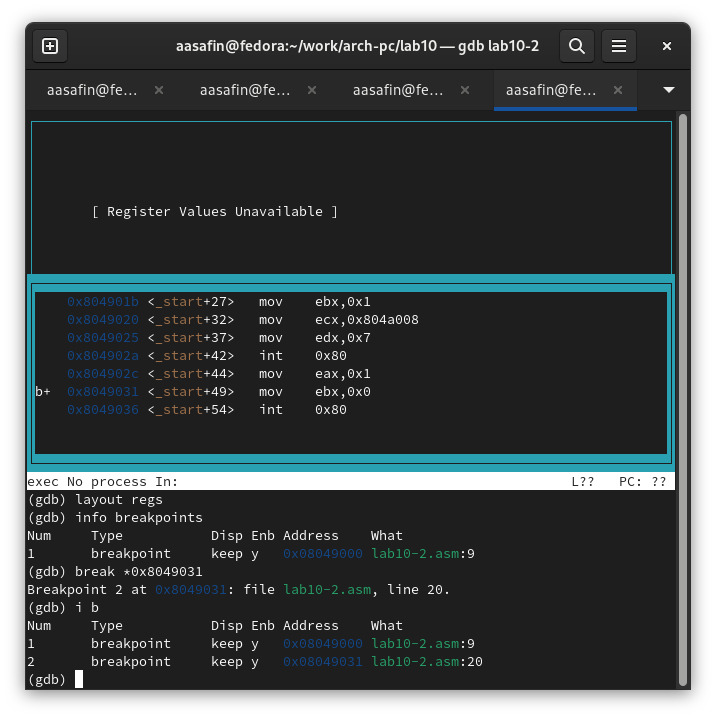


Рис. 10: Рассмотрение и создание брейкпоинтов в режиме всевдографики

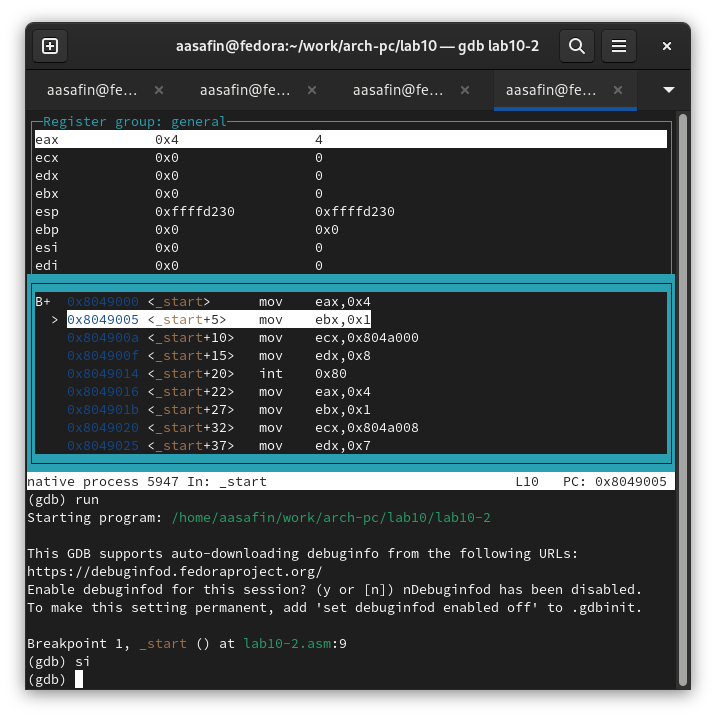


Рис. 11: Рассмотрение инструкций с помощью stepi (1)

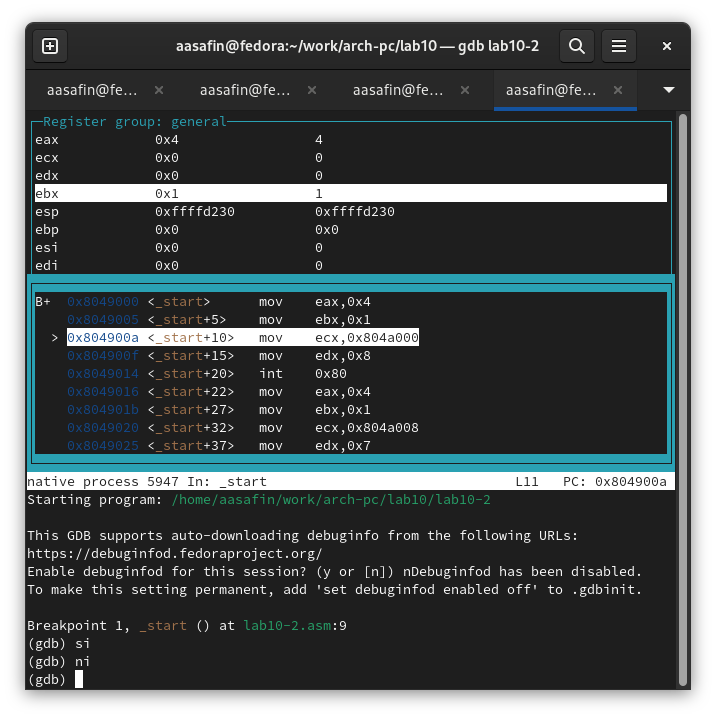


Рис. 12: Рассмотрение инструкций с помощью stepi (2)

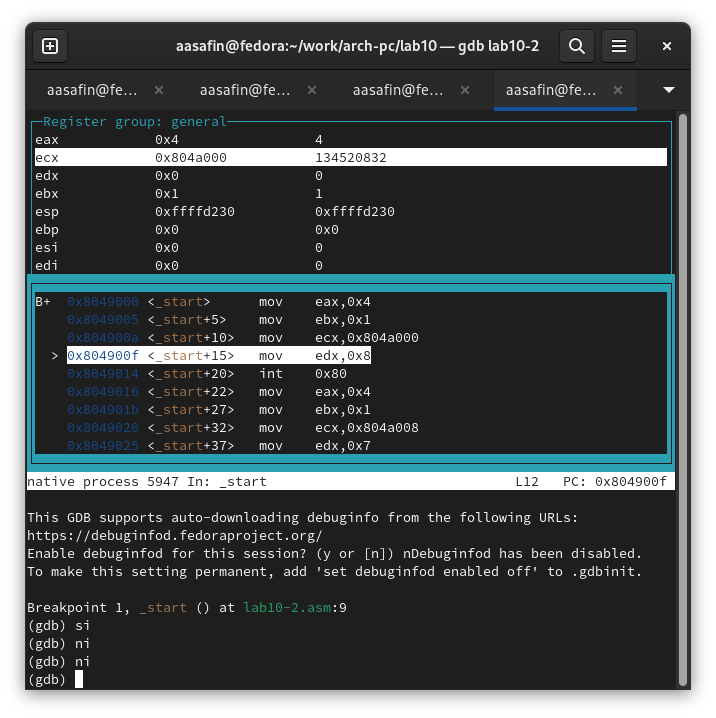


Рис. 13: Рассмотрение инструкций с помощью stepi (3)

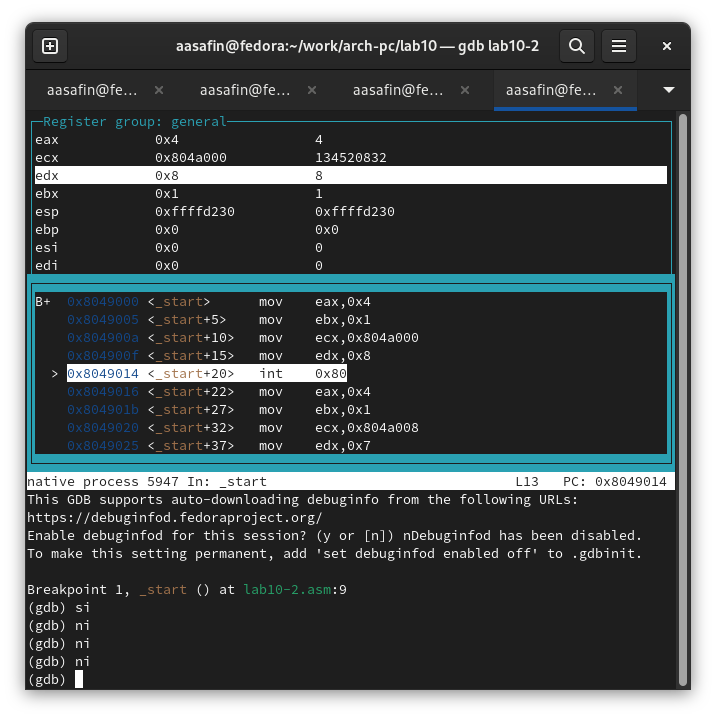


Рис. 14: Рассмотрение инструкций с помощью stepi (4)

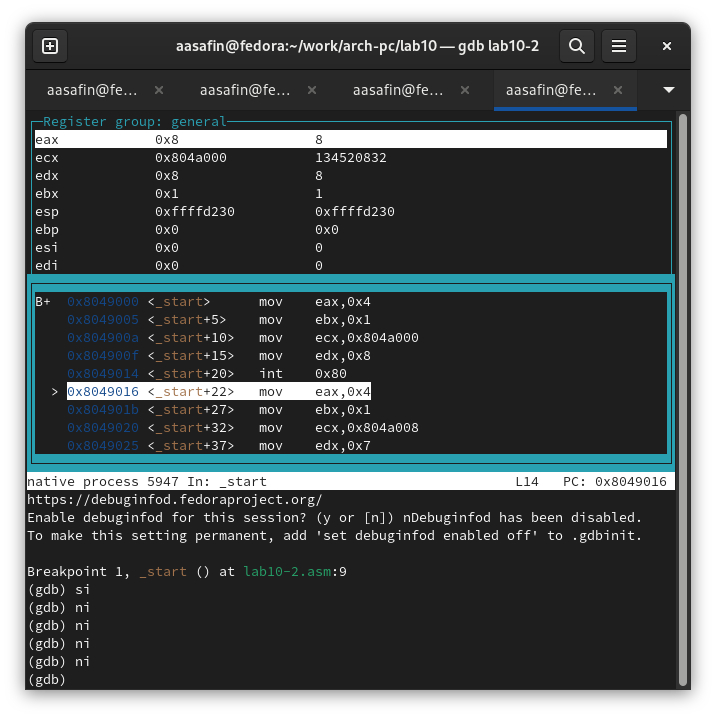


Рис. 15: Рассмотрение инструкций с помощью stepi (5)

Значениу msg1 просмотрено (рис. 16), изменено (рис. 17), а затем просмотрено и изменено значени msg2 ((рис. 18). Выведены в различных форматах значение edx (рис. 19). C помощью команды set изменено значение ebx сначала на строчную двойку, а затем на численную. Поскольку в обоих случаях выводится численное значение двойки, вывод отличется (рис. 20). После выполнение программы было заверешер с помощью quit.

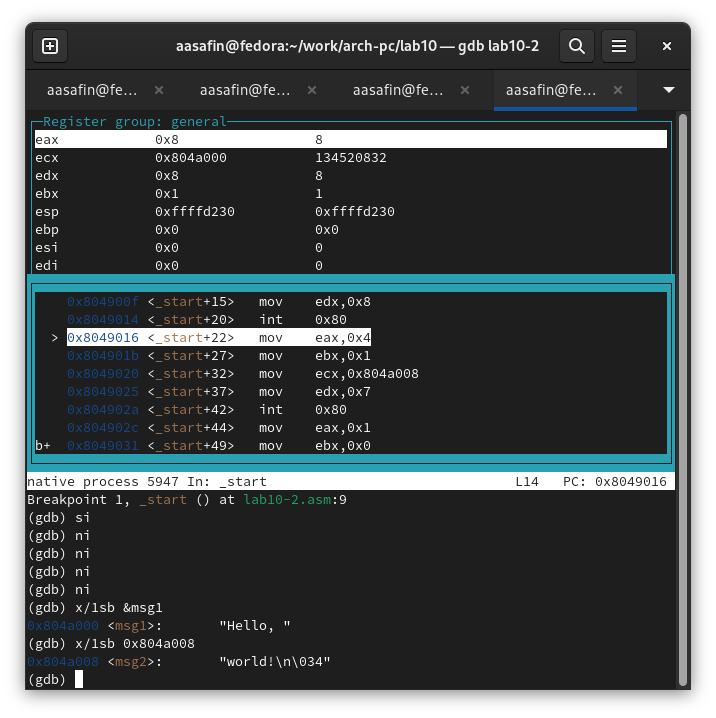


Рис. 16: Значение msg1

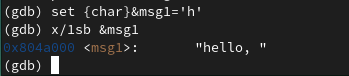


Рис. 17: Новое значение msg1

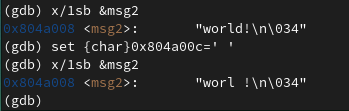


Рис. 18: Изменение значение msg2

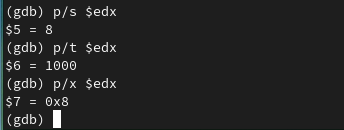


Рис. 19: Значение edx в разных форматах

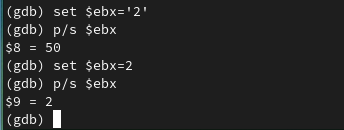


Рис. 20: Установка и выведение зачений ebx

1. Скопирован lab9-2.asm в lab10-3.asm, а затем к итоговой программе применен gdb при введении многих аргументов с помощью ключа –args (рис. 21). Установлена точка останова на \_start (рис. 22). Выведено число аргументов, хранящееся в esp (рис. 23). Выведены значения в остальных позициях стека (рис. 24). Адресация сдвигается на четыре, так как на элемент стека выведено по 4 байта.

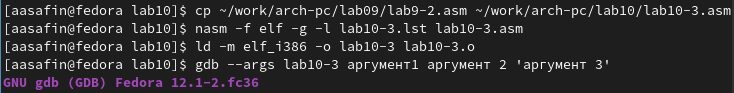


Рис. 21: Использование gdb к lab10-3.asm

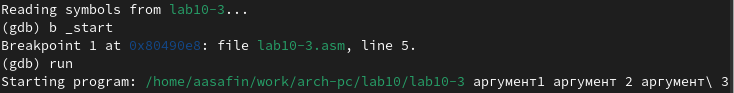


Рис. 22: Установление точки останова на \_start

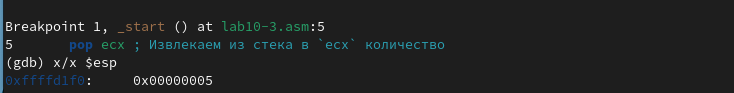


Рис. 23: Выведение значения esp

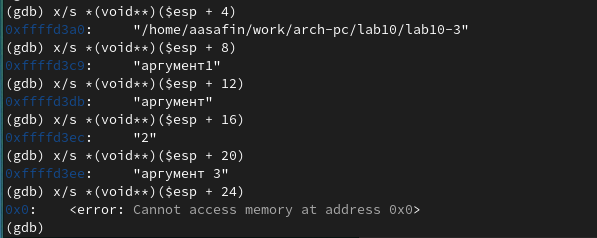


Рис. 24: Выведение значений стека

# 4 Самостоятельная работа

1. Изменена программа из лабораторной работы 9 так, что f(x) вычисляется в подпрограмме (рис. 25, 26). Результат полностью соответствует таковому из лабораторной работы 9.



Рис. 25: Текст новой программы sr.asm

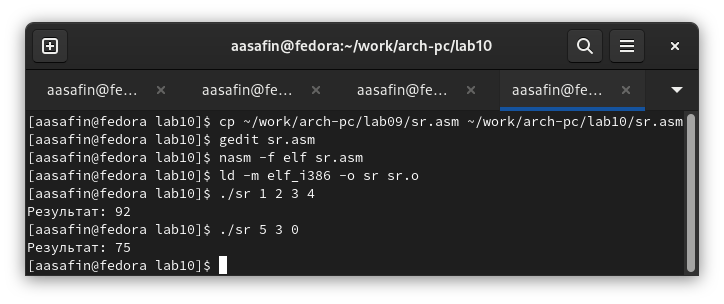


Рис. 26: Выполнение sr

1. Создан файл sr2.asm (рис. 27), в который введена программа из листинга 10.3, вычисляющая 4(3+2)+5 с ошибкой (рис. 28). С помощью отладчика рассмотрены изменения в регистрах (рис. 29-32). По ним видно, что ошибка возникает из-за сохранения результата суммы в ebx и продолжения работы с этим регистром несмотря на то, что умножение выполняется с eax. Ошибка исправлена (рис. 33). Программа выполняется верно (рис. 34).

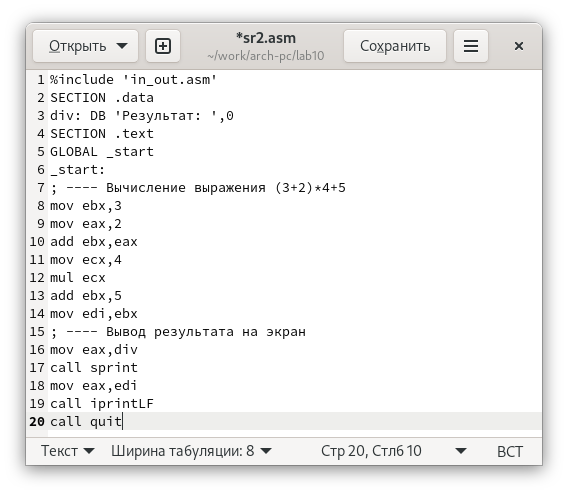


Рис. 27: Текст новой программы sr2.asm

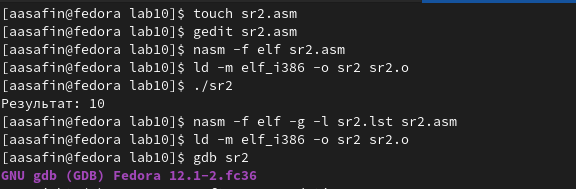


Рис. 28: Выполнение sr2

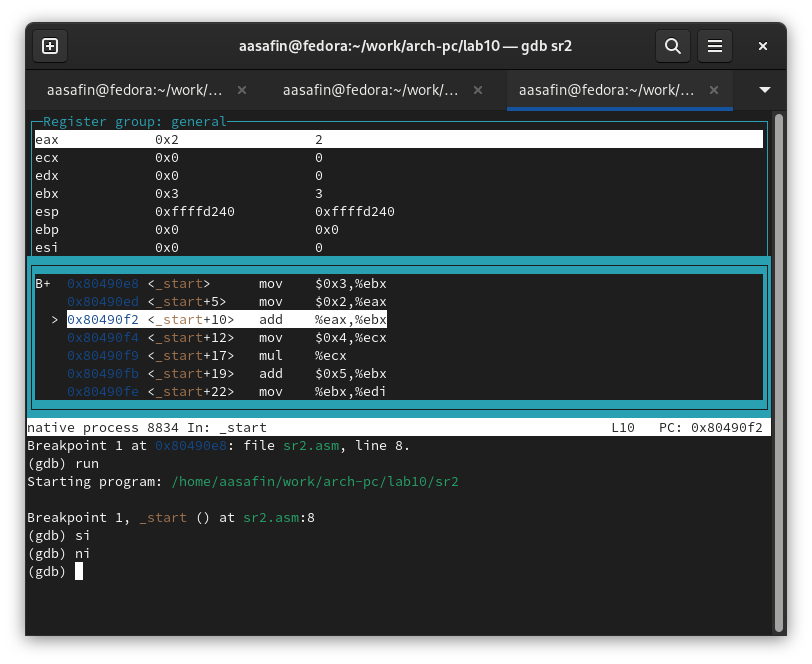


Рис. 29: Рассмотрение работы регистров в GDB (1)

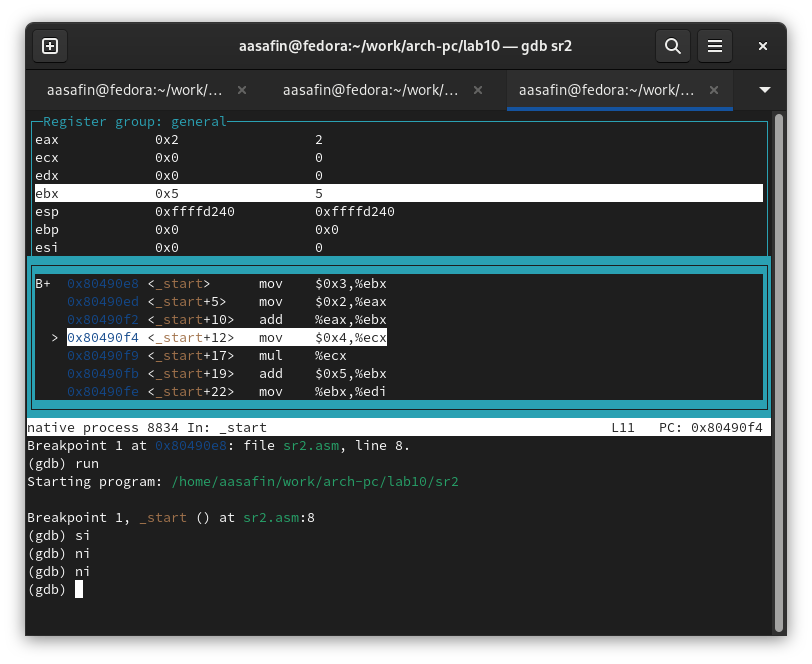


Рис. 30: Рассмотрение работы регистров в GDB (2)

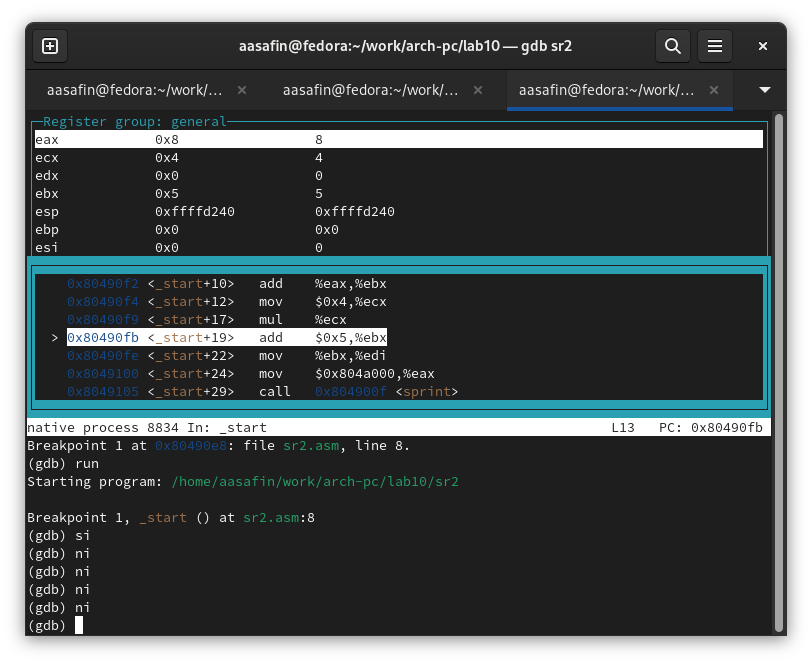


Рис. 31: Рассмотрение работы регистров в GDB (3)

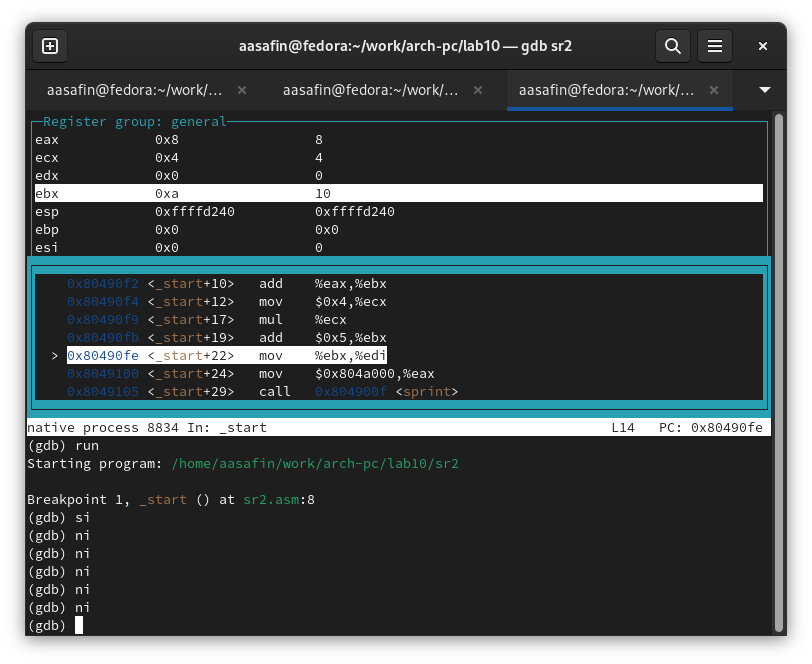


Рис. 32: Рассмотрение работы регистров в GDB (4)

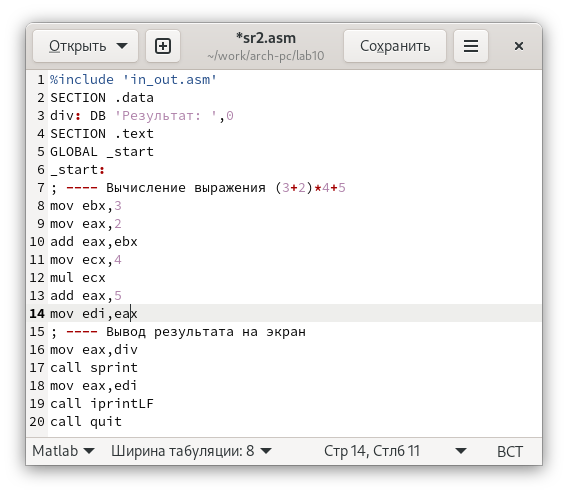


Рис. 33: Текст измененной программы sr2.asm

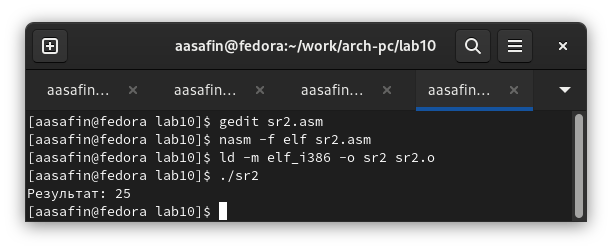


Рис. 34: Работа sr2

# 5 Выводы

Все программы с подпрограммами составлены. Задания по работе с отладчиком выполнены. Навык работы приобретен.

# Список литературы