Отчёта по лабораторной работе №9

Понятие подпрограммы. Отладчик GDB.

Хохлачёва Полина Дмитриевна

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

Создаём новый каталог для лабораторной №9

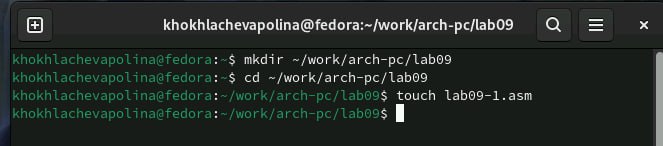


Рис. 1: И в нём новый файл

Открываем файл и заполняем его в соответствие с листингом

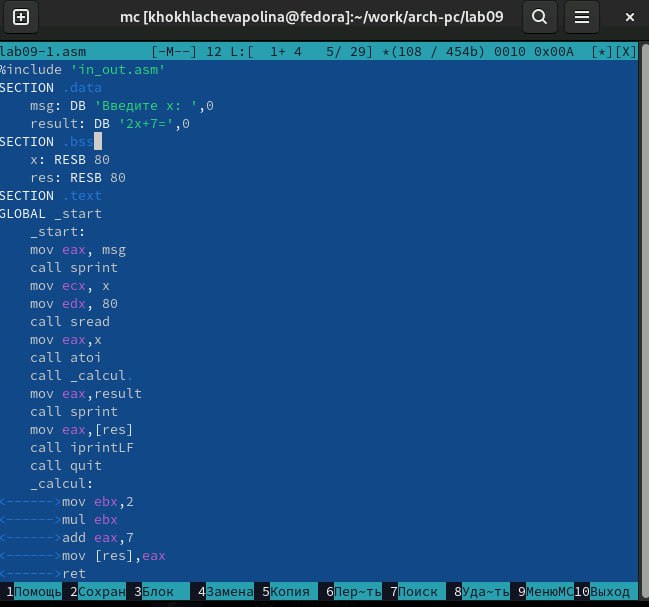


Рис. 2: Заполняем файл

Создаём файл и запускаем его

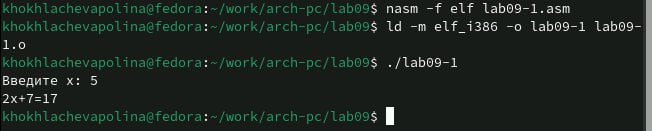


Рис. 3: Запускаем файл и проверяем его работу

Открываем файл для редактирования

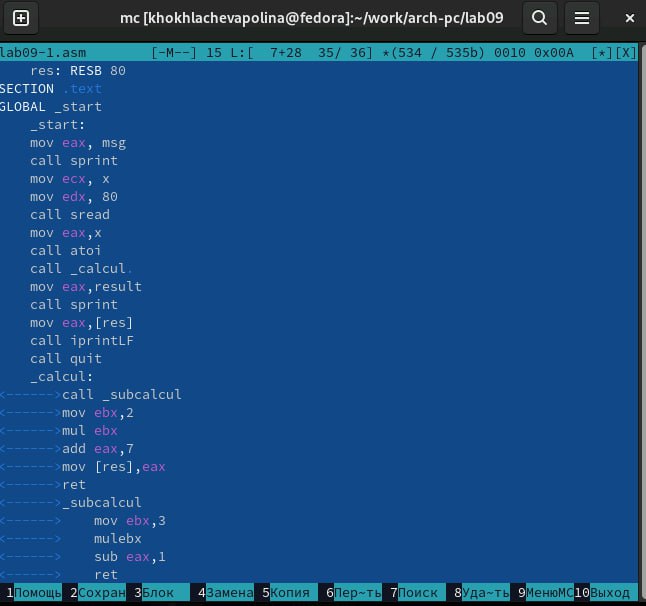


Рис. 4: Редактируем файл

Создаём файл и запускаем его

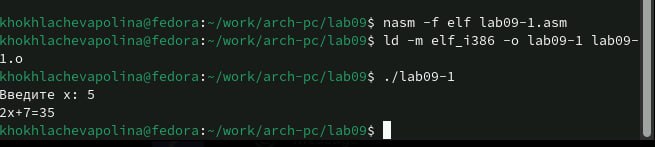


Рис. 5: Запускаем файл и смотрим на его работу

Создаём новый файл в каталоге

Создаём файл

Рис. 6: Создаём файл

Открываем файл и заполняем его в соответствии с листингом

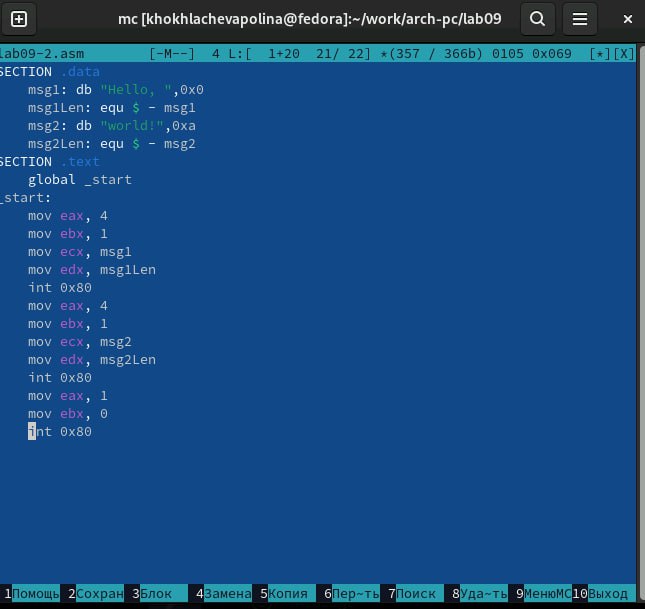


Рис. 7: Заполняем файл

Получаем исходный файл с использованием отладчика gdb

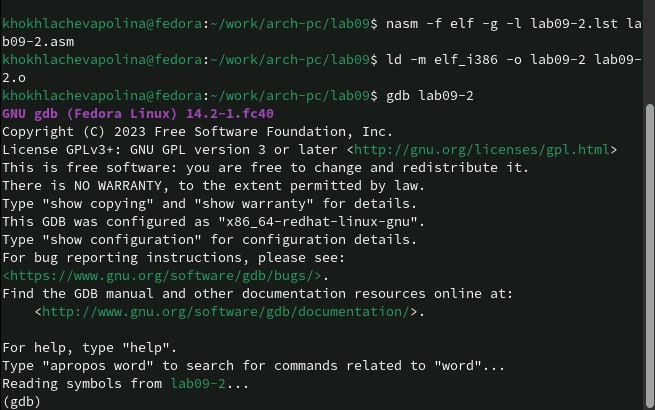


Рис. 8: Загружаем исходный файл

Запускаем команду в отладчике



Рис. 9: Запускаем программу командой run

Устанавливаем брейкпоинт на метку \_start и запускаем программу

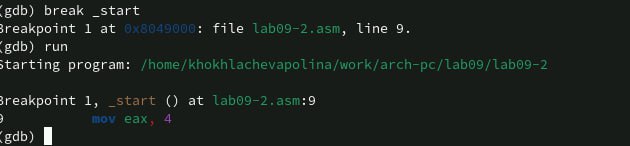


Рис. 10: Запускаем программу с брейкпоином

Смотрим кол программы



Рис. 11: Смотрим код программы

Переключаемся на отображение команд с Intelовским синтаксисом

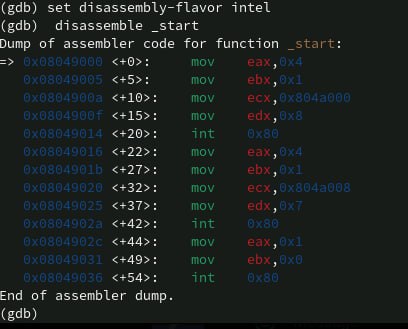


Рис. 12: Переключаемся на синтаксис Inter

Различия отображения синтаксиса машинных команд в режимах ATT и Intel: 1.Порядок операндов: В ATT синтаксисе порядок операндов обратный, сначала указывается исходный операнд, а затем - результирующий операнд. В Intel син- таксисе порядок обычно прямой, результирующий операнд указывается первым, а исходный - вторым. 2.Разделители: В ATT синтаксисе разделители операндов - запятые. В Intel синтаксисе разделители могут быть запятые или косые черты (/). 3.Префиксы размера операндов: В ATT синтаксисе размер операнда указывается перед операндом с использованием префиксов, таких как “b” (byte), “w” (word), “l” (long) и “q” (quadword). В Intel синтаксисе размер операнда указывается после операнда с использованием суффиксов, таких как “b”, “w”, “d” и “q”. 4.Знак операндов: В ATT синтаксисе операнды с позитивными значениями предваряются символом ““.””. 5.Обозначение адресов: В ATT синтаксисе адреса указываются в круглых скоб- ках. В Intel синтаксисе адреса указываются без скобок. 6.Обозначение регистров: В ATT синтаксисе обозначение регистра начинается с символа “%”. В Intel синтаксисе обозначение регистра может начинаться с символа “R” или “E” (например, “%eax” или “RAX”).

Включаем режим псевдографики

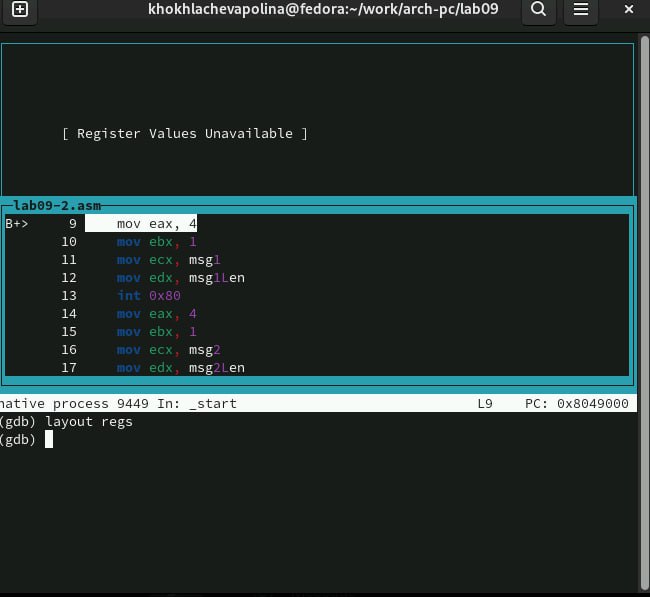


Рис. 13: Включаем отображение регистров, их значений и результат дисассимилирования программы

Проверяем установку точки основы и устанавливаем точку останова предпоследней инструкции

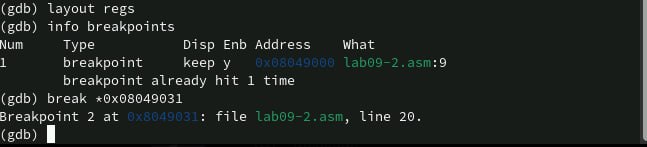


Рис. 14: Используем команду и создаём новую точку останова

Посмотрим информацию о всех установленных точках останова

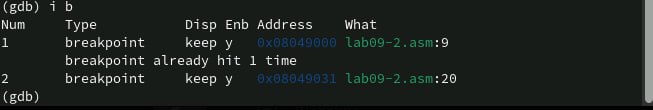


Рис. 15: Смотрим информацию

Выполняем 5 инструкций командой si

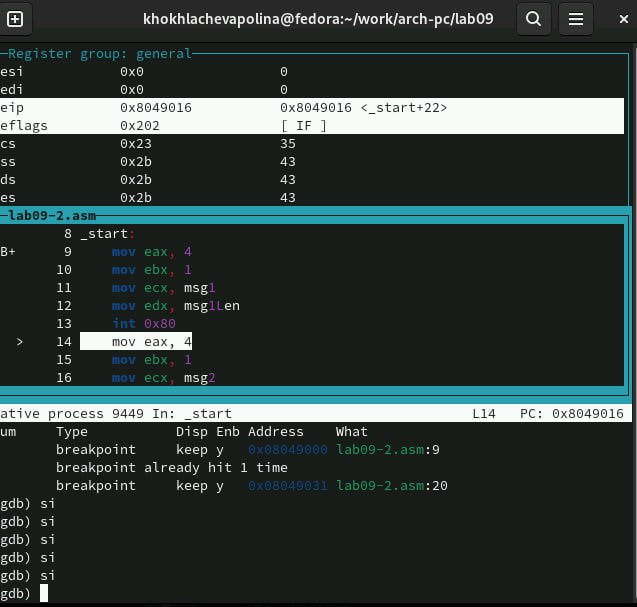


Рис. 16: Отслеживаем регистры

Во время выполнения команд менялись регистры: ebx, ecx, edx,eax, eip.

Смотрим значение переменной msg1 по имени

Смотрим значение переменной

Рис. 17: Смотрим значение переменной

Смотрим значение переменной

Смотрим значение переменной

Рис. 18: Смотрим значение переменной

Изменим первый символ

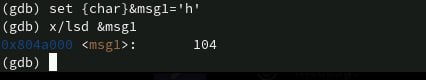


Рис. 19: Меняем символ

Изменим первый символ другой переменной

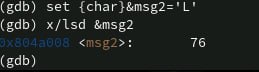


Рис. 20: Меняем символ

Смотрим значение в разных форматах



Рис. 21: Значение регистра

Изменяем регистор ebx

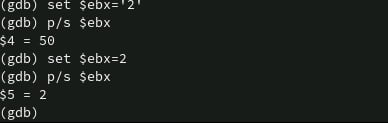


Рис. 22: Изменяем регистор

Выводится разные значения, так как команда без кавычек присваивает регистру вводимое значение.

Прописываем команды для завершения программы

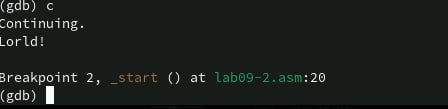


Рис. 23: Прописываем команды

Копируем файл в файл

Копируем

Рис. 24: Копируем

Создаём исполняемый файл и запускаем его

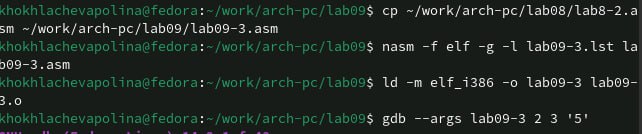


Рис. 25: Создаём и запускаем

Установим точку останова перед первой инструкцией в программе и запустим её

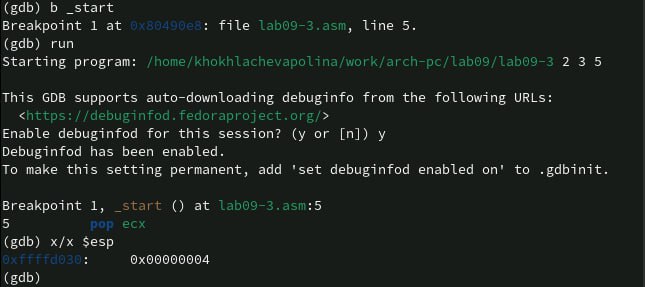


Рис. 26: Устанавливаем точку останова

Смотрим позиции стека по разным адресам

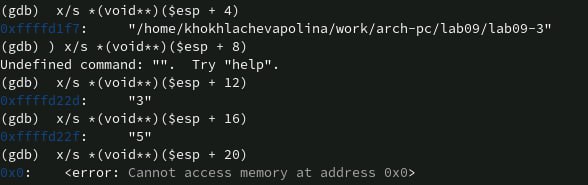


Рис. 27: Изучаем полученные данные

Шаг изменения адреса равен 4 потому что адресные регистры имеют размер- ность 32 бита(4 байта).

# 3 Задание 1

Копируем файл в файл

Копируем файл

Рис. 28: Копируем файл

Открываем файл и меняем его

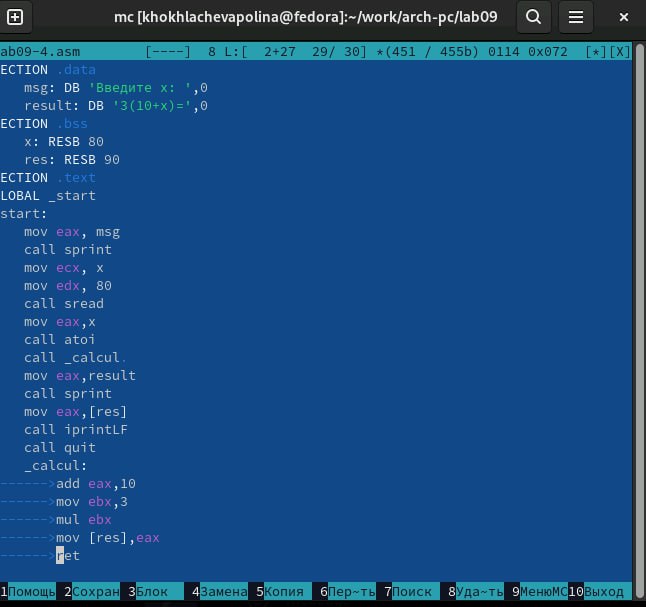


Рис. 29: Изменяем файл

Создаём Исполняемый файл и запускаем его

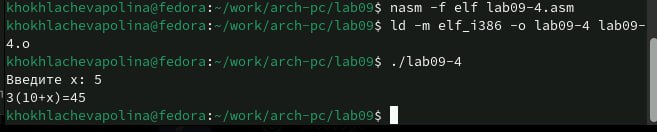


Рис. 30: Проверяем работу

# 4 Задание 2

Создаём новый файл в дирректории

Создаём файл

Рис. 31: Создаём файл

Открываем файл и заполняем его в соответствие с листингом

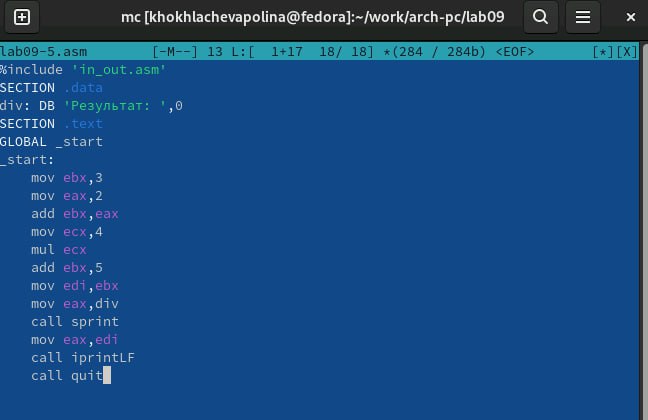


Рис. 32: Изменяем файл

Создаём исполняемый файл и запускаем его

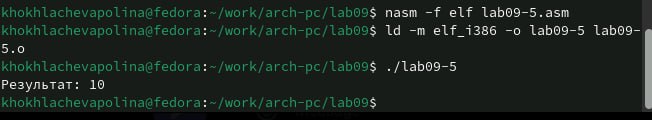


Рис. 33: Смотрим на работу программы

Создаём файл и смотрим на изменение решистров, ищем ошибку

Изменяем программу

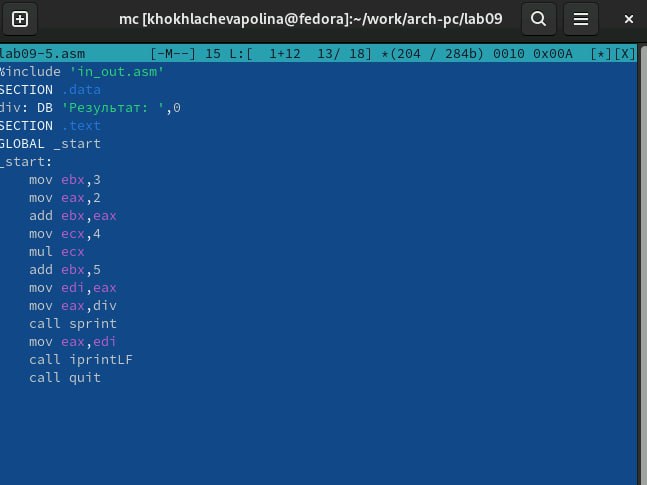


Рис. 34: Меняем файл

Создаём файл и запускаем его

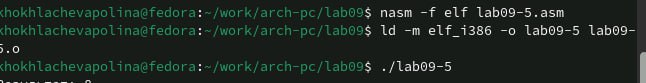


Рис. 35: Проверяем работу

# 5 Выводы

Мы познакомились с методами отладки при помощи GDB и его возможностями.