Лабораторная работа №9

Отчет

Устинова Виктория Вадимовна

Содержание

# 1 Цель работы

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

# 2 Задание

Выполнить отчет по лабораторной работе №9, исправить ошибки программы и написать свою.

# 3 Выполнение лабораторной работы

**Реализация подпрограмм в NASM**

Cоздаем каталог для 9 лабораторной работы(рис. 1).

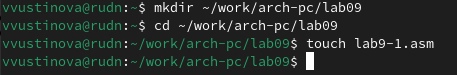


Рис. 1: Переходим в каталог и создаем там файл lab9-1.asm

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 9.1(рис. 2).

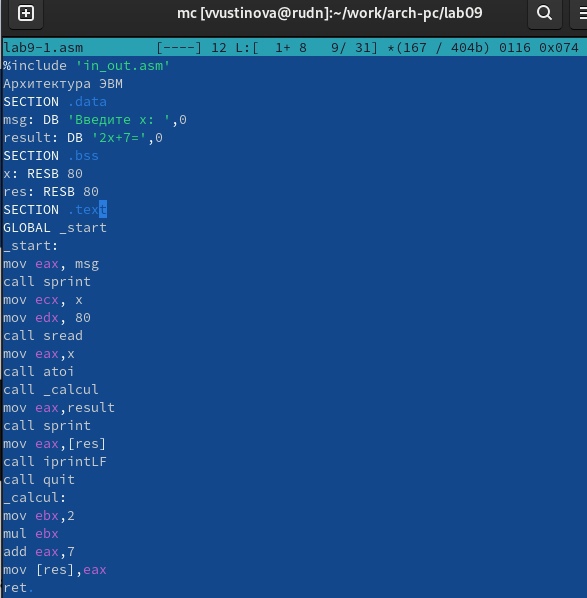


Рис. 2: Заполняем данный файл

Запускаем файл(рис. 3).



Рис. 3: Смотрим как работает файл

Снова открываем файл для редактирования и изменяем его, добавив подпрограмму(рис. 4).

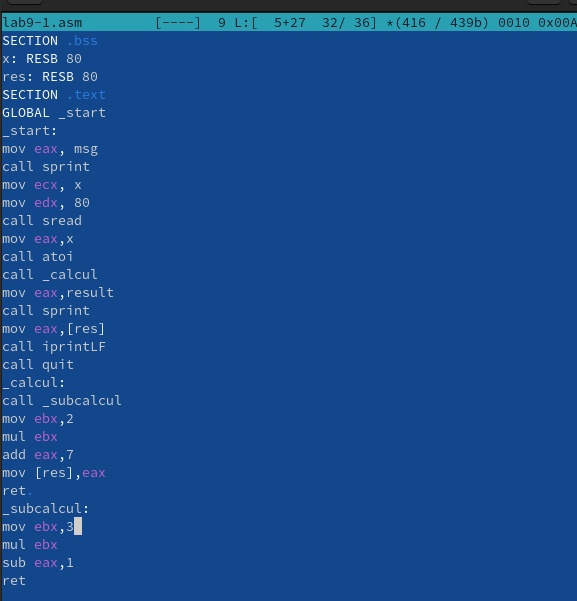


Рис. 4: Редактируем файл

Запускаем файл(рис. 5).

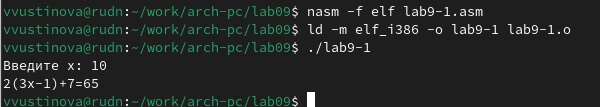


Рис. 5: Смотрим как работает файл

**Отладка программам с помощью GDB**

Создаем файл Lab9-2.asm(рис. 6).

Используем команду touch

Рис. 6: Используем команду touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 9.2(рис. 7).

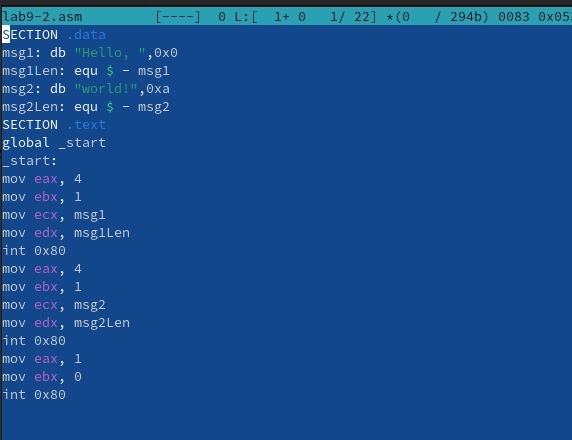


Рис. 7: Заполняем наш файл

Получаем исходный файл с использованием отладчика gdb(рис. 8).

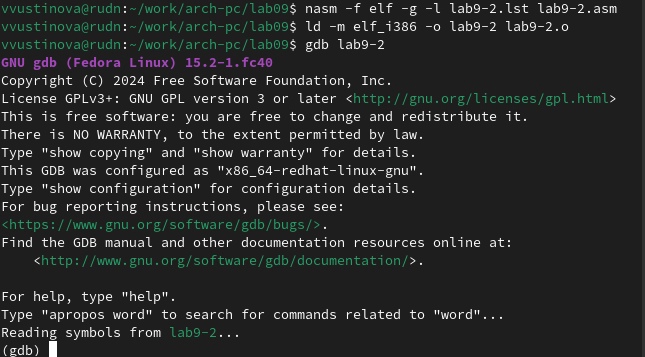


Рис. 8: Выгружаем файл в отладчик

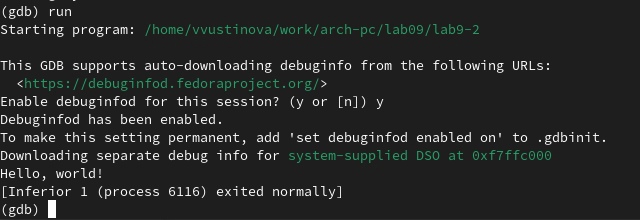


Рис. 9: Запускаем программу командой run в отладчике

Устанавливаем брейкпоинт на метку \_start и запускаем программу(рис. 10).

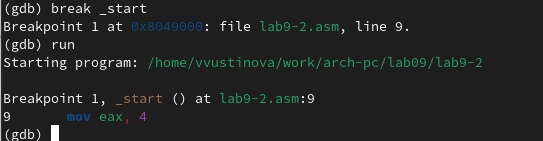


Рис. 10: Заупскаем файл

Смотрим дисассимилированный код программы с помощью команды disassemble, начиная с метки \_start(рис. 11).

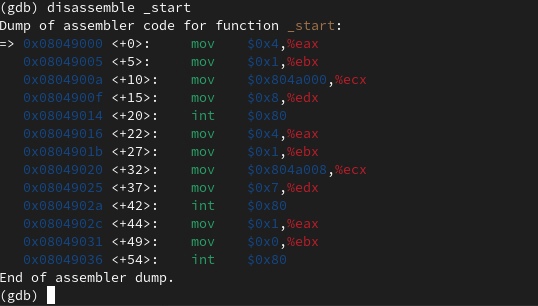


Рис. 11: Смотрим дисассимилированный код

Переключаемся на отображение команд с Intel’овским синтаксисом(рис. 12).

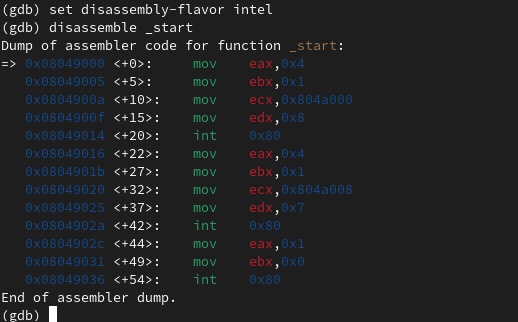


Рис. 12: Переключаемся на синтаксис Intel

Различия отображения синтаксиса машинных команд в режимах ATT и Intel: 1. В ATT операнды указываются в формате сперва источник потом назначение, а в intel наоборот. 2. В ATT перед регистрами ставится %, а перед значениями $. В Intel префиксы отсутствуют. 3. В ATT синтаксисе адреса указываются в круглых скобках. В Intel синтаксисе адреса указываются без скобок. 4. В ATT синтаксисе разделители операндов - запятые. В Intel синтаксисе разделители могут быть запятые или косые черты /.

Переходим в режим псевдографики(рис. 13).

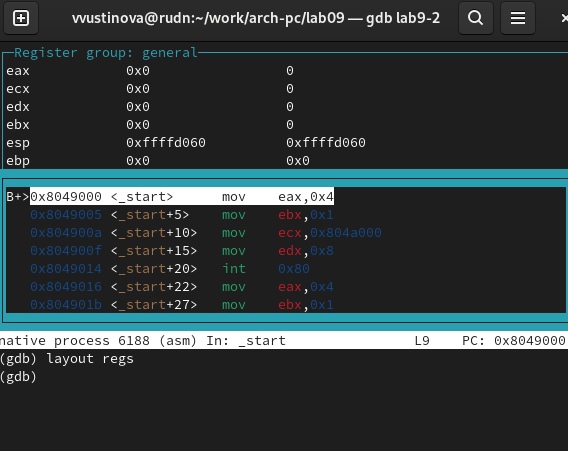


Рис. 13: Открываются три окна

**Добавление точек останова**

Проверяем была ли установлена точка останова и устанавливаем точку останова(рис. 14).

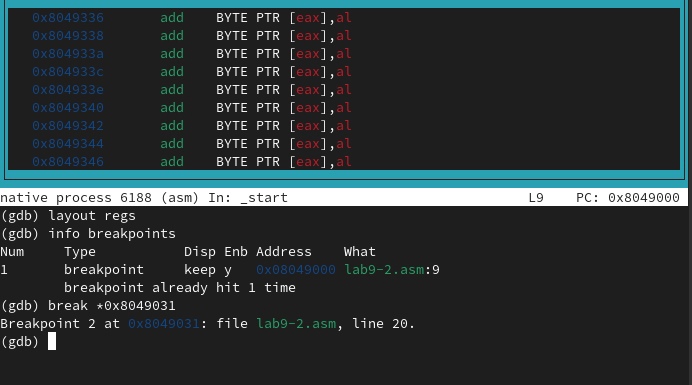


Рис. 14: ТИспользуем команду info breakpoints, создаем новую точку

Посмотрим информацию о всех установленных точках останова(рис. 15).

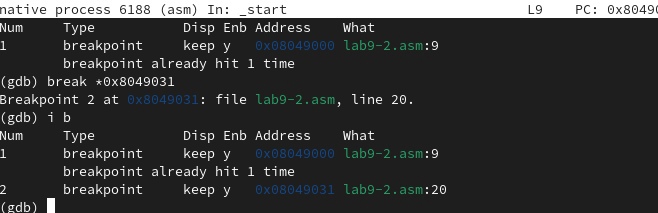


Рис. 15: Просматриваем информацию

**Работа с данными программы в GDB**

Посматриваем содержимое регистров с помощью команды info registers(рис. 16).

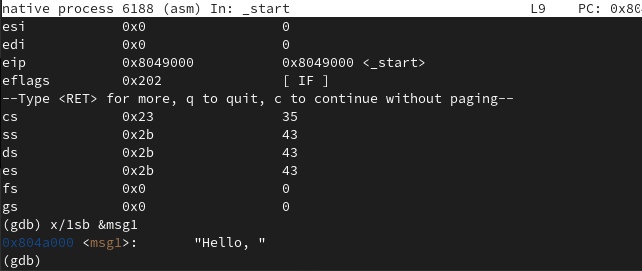


Рис. 16: Cмотрим значения регистров и переменной msg1

Поменялись регистры ebx, ecx, edx, eax, eip.

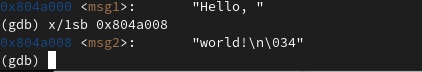


Рис. 17: Смотрим значение переменной msg2

Изменим первый символ переменной msg1(рис. 18).

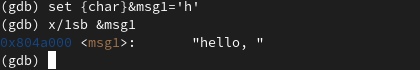


Рис. 18: Меняем символ

Изменим первый символ переменной msg2(рис. 19).

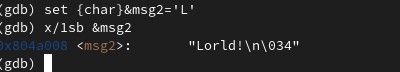


Рис. 19: Меняем символ

Смотрим значение регистра edx в разных форматах(рис. 20).

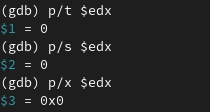


Рис. 20: Смотрим значение регистра

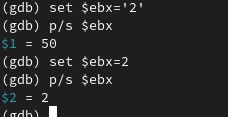


Рис. 21: Изменяем регистр ebx командой set

Выводятся разные значения, потому что команда без кавычек присваивает регистру это значение.

**Обработка аргументов командной строки в GDB**

Копируем файл lab8-2.asm в файл с именем lab09-3.asm и запускаем его в отладчике(рис. 22).

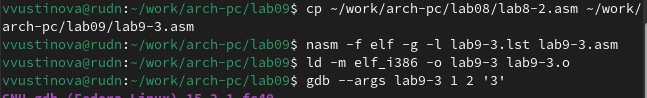


Рис. 22: Копируем фпйл и запускаем его

Установим точку останова перед первой инструкцией в программе и запустим ее(рис. 23).

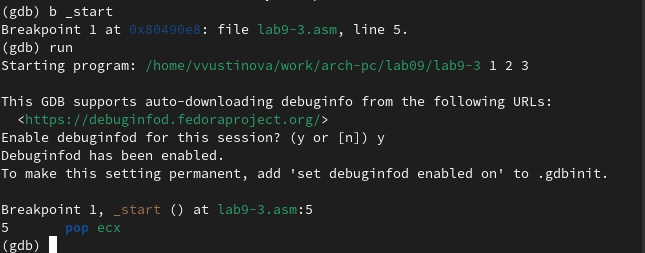


Рис. 23: Устанавливаем точку останова

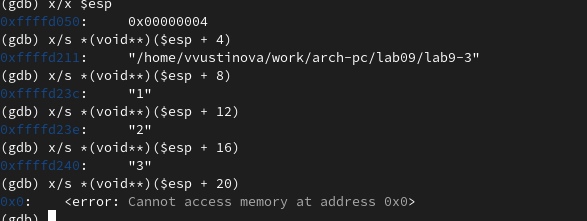


Рис. 24: Смотрим позиции стека по разным адресам

Шаг изменения адреса равен 4 потому что адресные регистры имеют размерность 32 бита

**Задание для самостоятельной работы**

1 задание

Переходим в нужный каталог и копируем оттуда файл lab8-4.asm в lab9-4.asm(рис. 25).

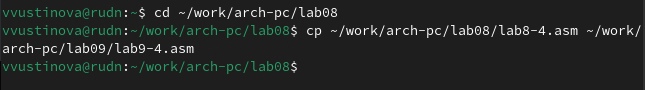


Рис. 25: Копируем файл

Открываем файл в Midnight Commander и меняем его, создавая подпрограмму(рис. 26).

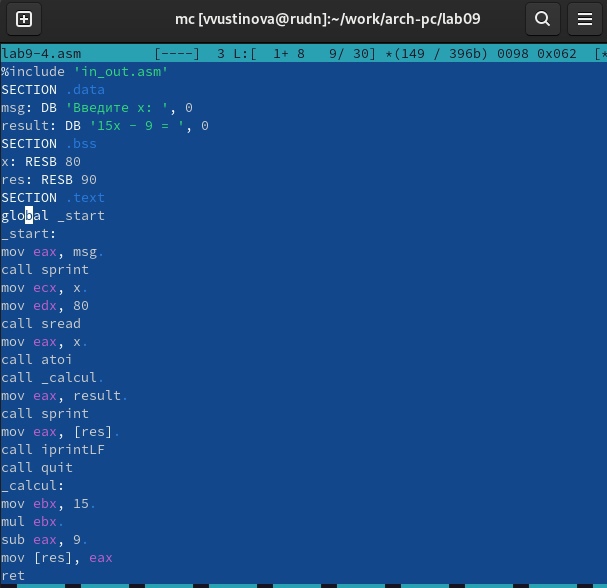


Рис. 26: Изменяем файл

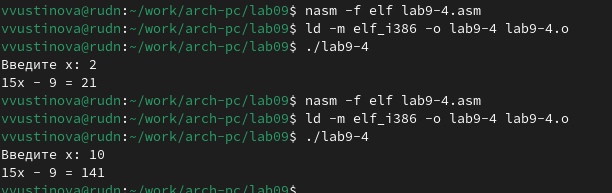


Рис. 27: Запускаем файл

2 задание

Создаем новый файл lab9-5.asm и открываем его в Midnight Commander и заполняем его в соответствии с листингом 9.3(рис. 28).

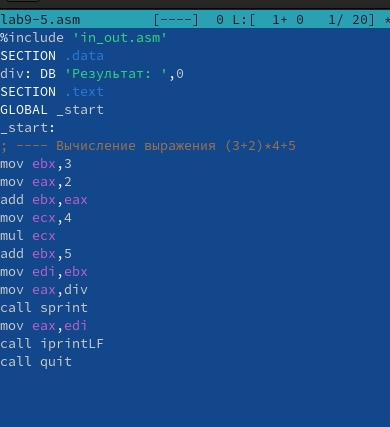


Рис. 28: Изменяем файл

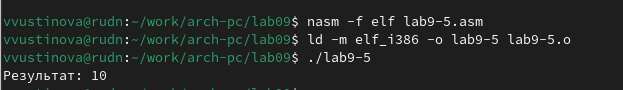


Рис. 29: Запускаем файл

Создаем файл и запускаем его в откладчике, смотрим на изменение регистров(рис. 30).

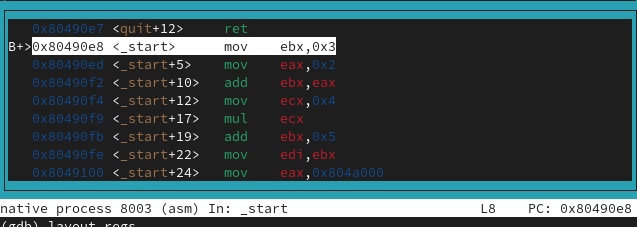


Рис. 30: Ищем ошибку

Изменяем программу(рис. 31).

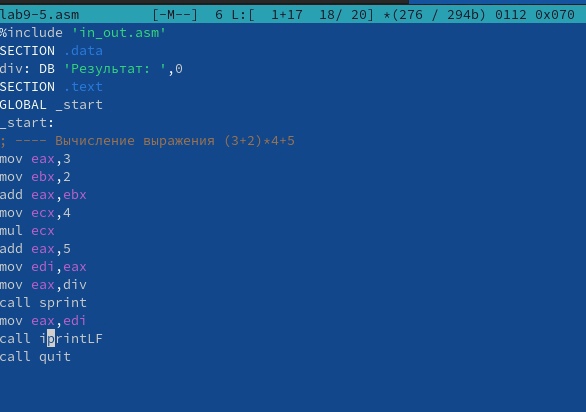


Рис. 31: Меняем файл

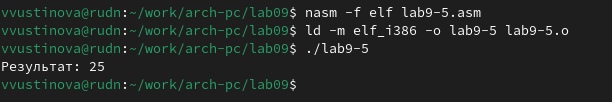


Рис. 32: Запускаем файл

# 4 Выводы

Мы приобрели навыки написания программ с использованием подпрограмм. Ознакомились с методами отладки при помощи GDB и его основными возможностями.