Отчёт по лабораторной работе №9

дисциплина: архитектура компьютеров

Ведьмина Александра Сергеевна

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. Изучить понятие процесса отладки
2. Познакомиться с отладчиков gdb
3. Изучить листинг с подпрограммой
4. Сделать задания для самостоятельной работы

# 3 Теоретическое введение

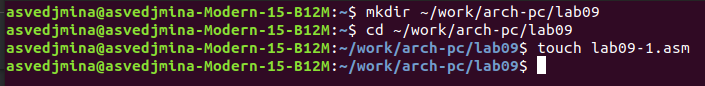
Отладка — это процесс поиска и исправления ошибок в программе. Этапы отладки: 1. обнаружение ошибки 2. поиск её местонахождения 3. определение причины ошибки 4. исправление ошибки

Отладчики позволяют управлять ходом выполнения программы, контролировать и изменять данные. Точки останова устанавливаются в отладчике на время сеанса работы с кодом программы, т.е. они сохраняются до выхода из программы-отладчика или до смены отлаживаемой программы. GDB (GNU Debugger — отладчик проекта GNU, который работает на многих UNIX-подобных системах и умеет производить отладку многих языков программирования. Если есть файл с исходным текстом программы, а в исполняемый файл включена информация о номерах строк исходного кода, то программу можно отлаживать, работая в отладчике непосредственно с её исходным текстом.

Подпрограмма — это, как правило, функционально законченный участок кода, который можно многократно вызывать из разных мест программы. Применяется, если есть одинаковые участки кода. Для вызова подпрограммы из основной программы используется инструкция call, которая заносит адрес следующей инструкции в стек и загружает в регистр eip адрес соответствующей подпрограммы, осуществляя таким образом переход. Затем начинается выполнение подпрограммы, которая, в свою очередь, также может содержать подпрограммы.

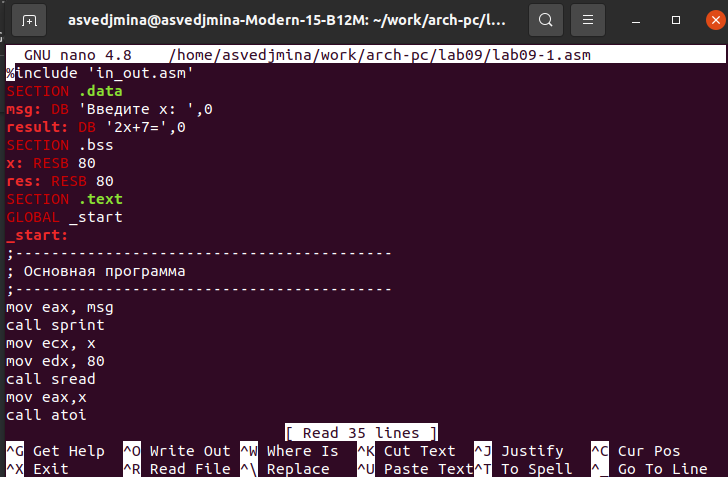
# 4 Выполнение лабораторной работы

Создаю каталог lab09, перехожу в него и создаю там файл lab09-1.asm.



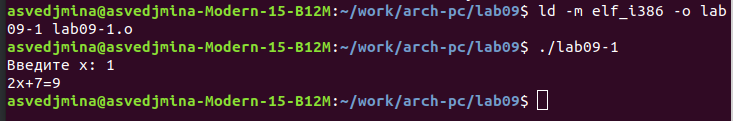
Создание файла lab09-1.asm

Ввожу в данный файл текст программы, использующую вызов подпрограммы.



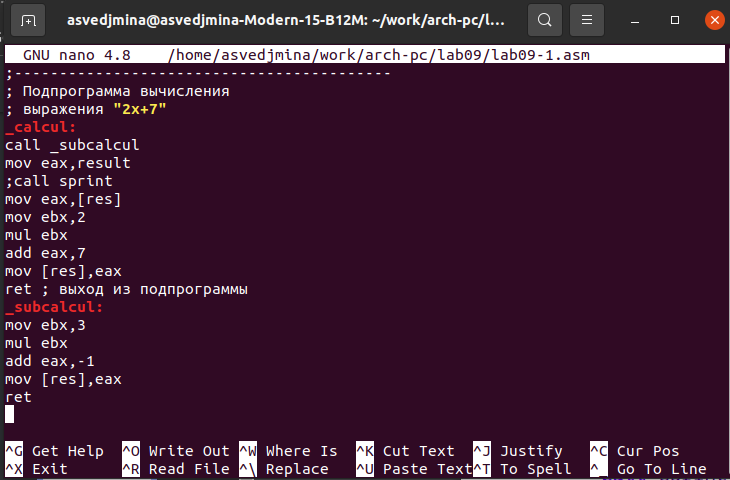
Ввод программы в lab09-1.asm

Создаю исполняемый файл и запускаю его.



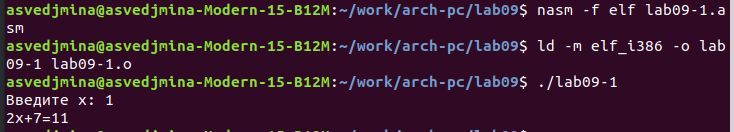
Запуск lab09-1

Затем изменяю текст программы, добавив подпрограмму \_subcalcul в подпрограмму \_calcul, чтобы \_subcalcul вычисляла выражение 3\*х-1.



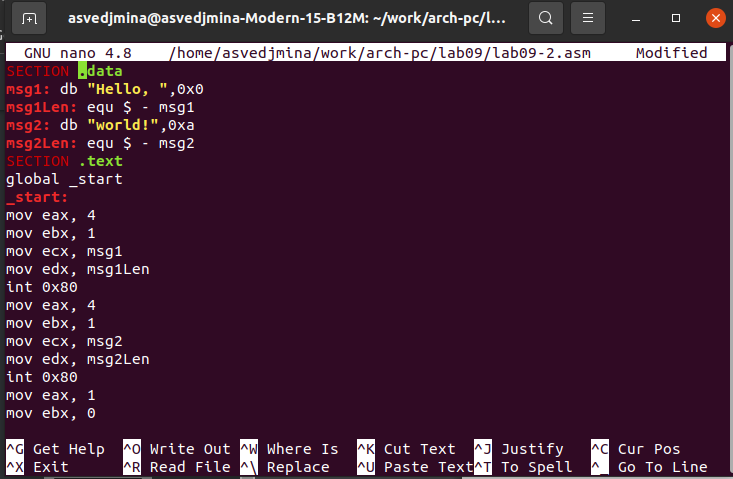
Изменение программы в lab09-1.asm

Создаю исполняемый файл и запускаю его.



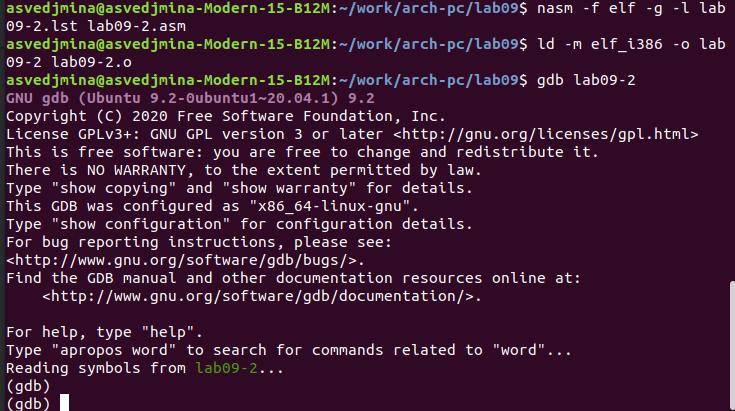
Запуск файла lab09-1

Создаю файл lab09-2.asm, ввожу в него текст программы с выводом сообщения Hello world!



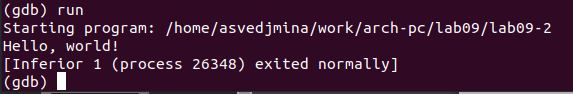
Ввод программы в lab09-2.asm

Получаю исполняемый файл, необходимый для работы с gdb, и загружаю его в отладчик.



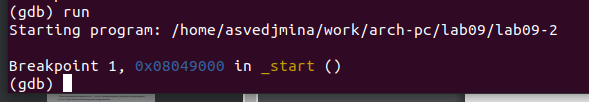
Загрузка lab09-2 в gdb

Запускаю программу в отладчике.



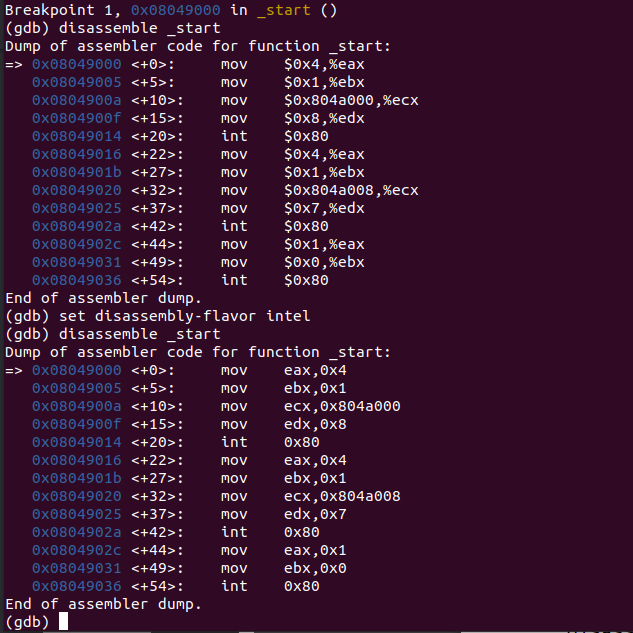
Запуск lab09-2

Устанавливаю брейкпоинт на метку \_start, после чего вновь запускаю программу.



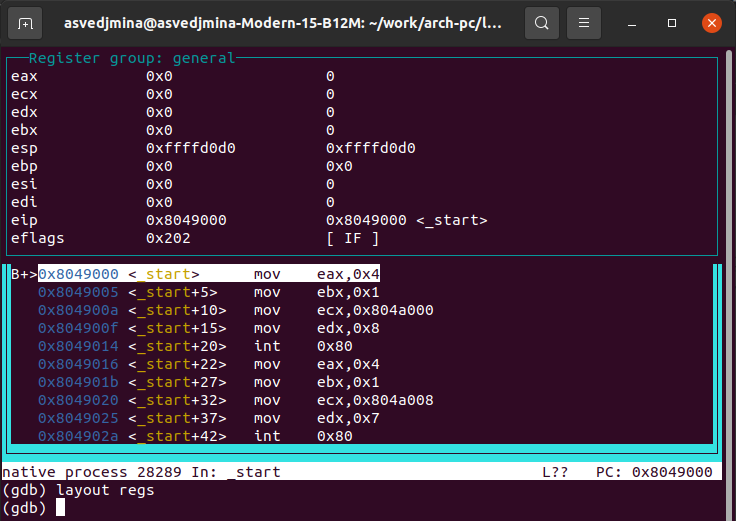
Повторный запуск lab09-2

Открываю дисассимилированный код программы с помощью команды disassemble, начиная с метки \_start, а затем переключаюсь на отображение с Intel’овским синтаксисом. В машинном коде используются $ и % для отображения регистров.



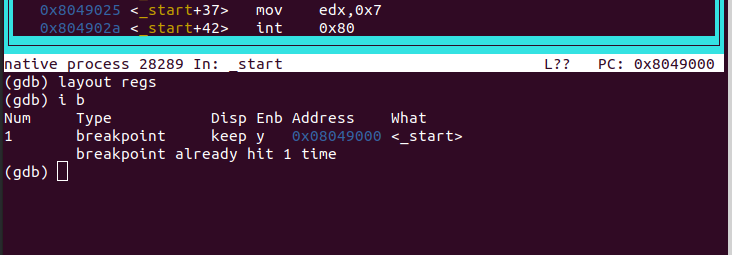
Открытие дисассимилированного кода программы

Далее включаю режим псевдографики.



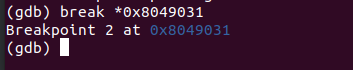
Включение режима псевдографики

Вывожу информацию о всех точках останова.



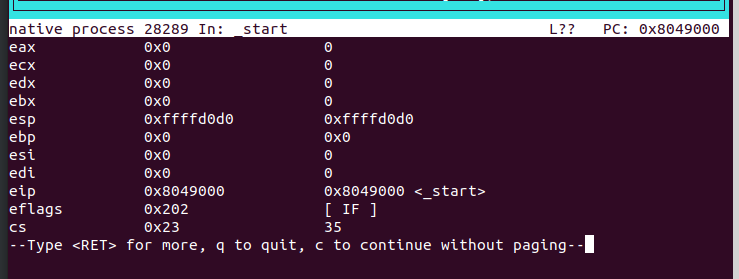
Вывод точек останова

Устанавливаю вторую точку останова по адресу предпоследней инструкции.



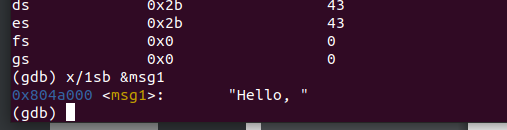
Установка второй точки останова

С помощью команды info registers, вывожу текущие значения регистров.



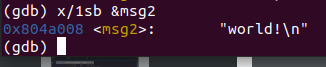
Вывод текущих значений регистров

Затем смотрю значение переменной msg1.



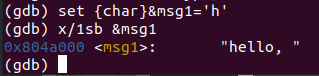
Значение msg1

Смотрю значение переменной msg2.



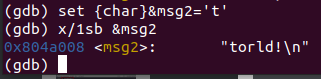
Значение msg2

Изменяю первый символ переменной msg1 с помощью команды set.



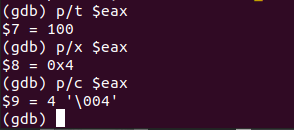
Изменение символа в msg1

Заменяю также символ и в переменной msg2.



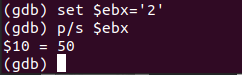
Изменение символа в msg2

Вывожу значение регистра eax в шестнадцатеричном формате, в двоичном формате и в символьном виде.



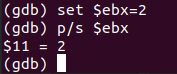
Вывод регистра в разных форматах

С помощью команды set изменяю значние регистра ebx.



Изменение значения ebx

Вновь изеняю значение этого регистра.



Повторное изменение значения ebx

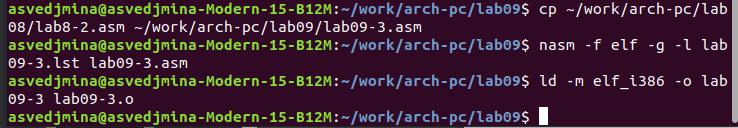
Разница в том, что в первый раз я записала в ebx символ, а во второй раз - число.

Завершаю выполнение программы с помощью команды continue и выхожу из gdb, используя команду quit. После этого копирую файл lab8-2.asm в lab09-3.asm.

Копирование lab8-2.asm

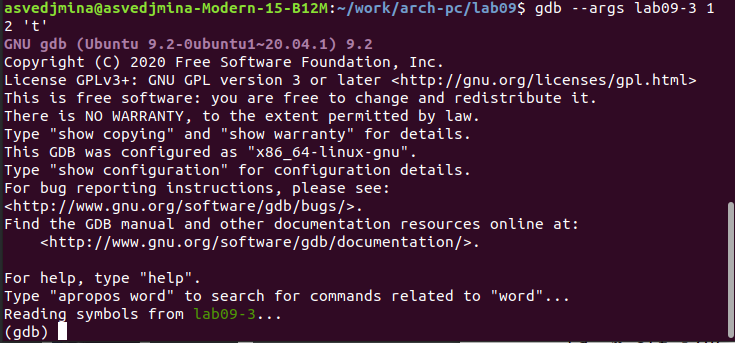
Копирование lab8-2.asm

Создаю исполяемый файл.



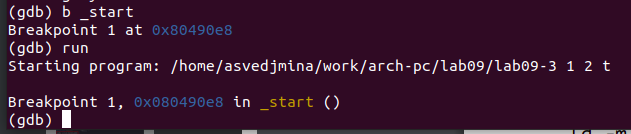
Создание исполняемого файла lab09-3

Передаю файл в gdb с тремя аргументами.



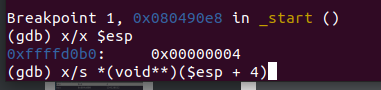
Передача файла в gdb

Ставлю точку останова перед первой инструкцией и запускаю программу.



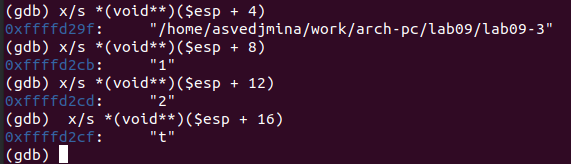
Установка точки останова

По адресу вершины стека узнаю количество аргументов, переданное программе (включая имя программы).



Вывод адреса стека

Затем смотрю отдельные позиции стека.



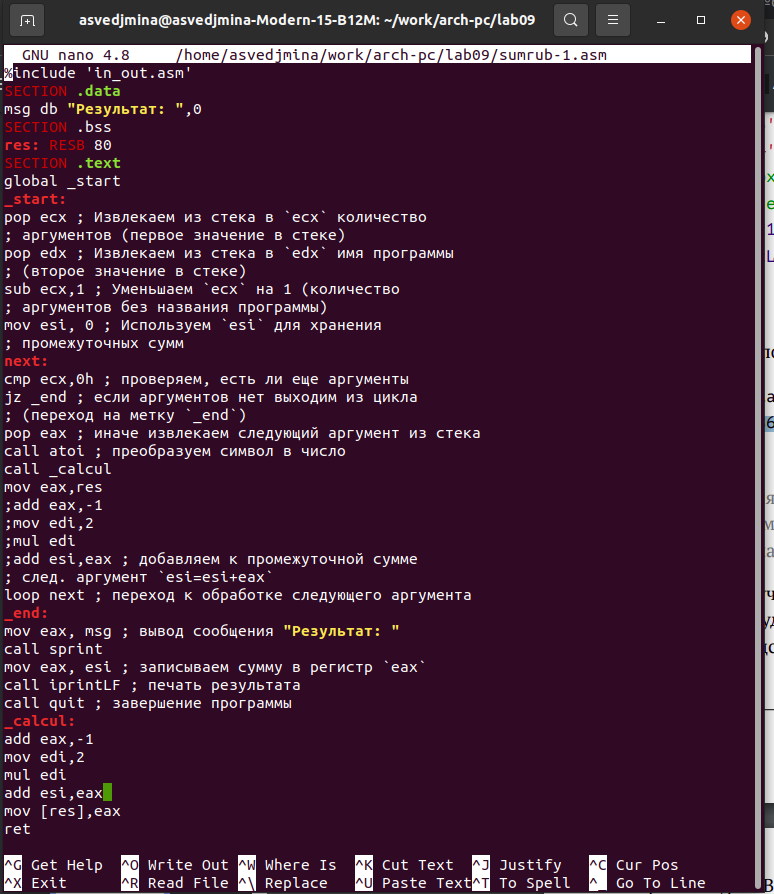
Вывод отдельных позиций стека

Шаг адреса равен 4, так как размер переменных составляет 4 байта.

#Выполнение заданий для самостоятельной работы

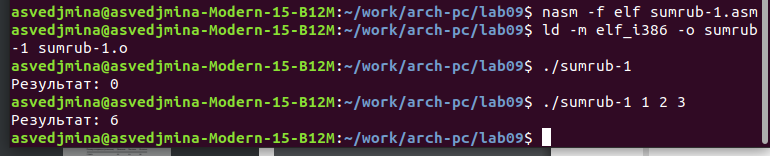
1. Преобразуйте программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции 𝑓(𝑥) как подпрограмму.

Для выполнение задания создаю файл sumrub-1.asm, копирую в него текст программы из файла c заданием 1 из 8 лабораторной работы. После этого реализую вычисление функции как подпрограмму.



Программа в sumrub-1.asm

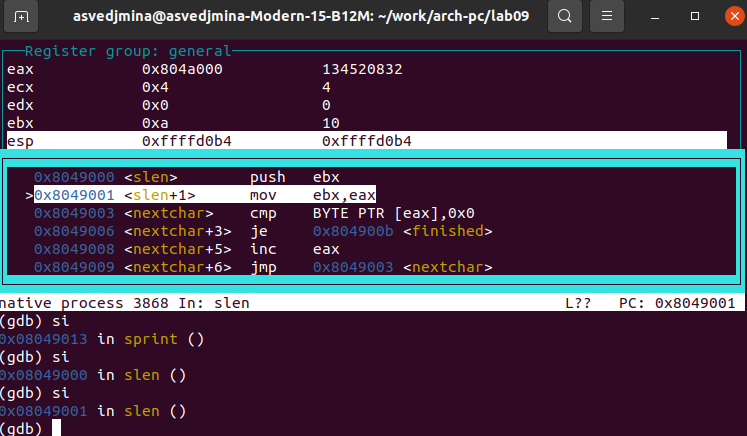
Создаю исполняемый файл и запускаю его.



Запуск sumrub-1

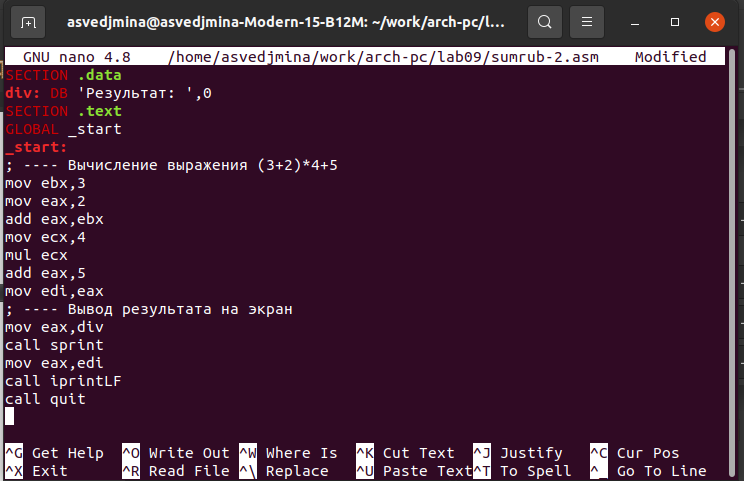
1. В листинге 9.3 приведена программа вычисления выражения (3 + 2) ∗ 4 + 5. При запуске данная программа дает неверный результат. Проверьте это. С помощью отладчика GDB, анализируя изменения значений регистров, определите ошибку и исправьте ее.

Для выполнения задания создаю файл sumrub-2.asm. Затем с поомощью gdb изучаю значения регистров.



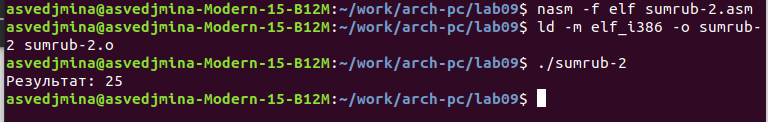
Анализ значений регистров

Узнаю, что в тексте программы регистры перепутаны местами и исправляю ошибку.



Исправление ошибки в программе

Для проверки создаю исполняемый файл и запускаю его.



Запуск sumrub-2

# 5 Выводы

В ходе лабораторной работы я научилась использовать подпрограммы и работать с gdb.