Лабораторная работа №9

Архитектура компьютера

Башиянц Александра Кареновна

Содержание

# 1 Цель работы

Цель работы — приобретести навыки написания программ с использованием подпрограмм, познакомиться с методами отладки при помощи GDB и его основными возможностями.

# 2 Задание

В этой лабораторной работе необходимо изучить работу циклов и обратку аргументов командной строки.

Необходимо научиться:

* Изучить использование подпрограмм;
* Изучить создание точек останова;
* Узнать работу с данными программы в GDB;
* Изучить как обрабатывать аргументы командной строки в GDB

Выполняя это задание, мы получим практический опыт с работой подпрограмм и методами отладки при помощи GDB

# 3 Выполнение лабораторной работы

## 3.1 Реализация подпрограмм в NASM

Создадим директорию для 9 лабораторной работы и создадим файл lab09-1.asm (рис. 1).

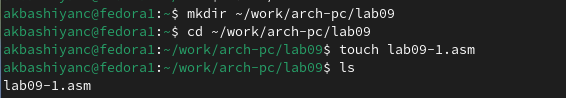


Рис. 1: Создание директории

Скопируем файл in\_out.asm из lab06 с помощью mc (рис. 2).

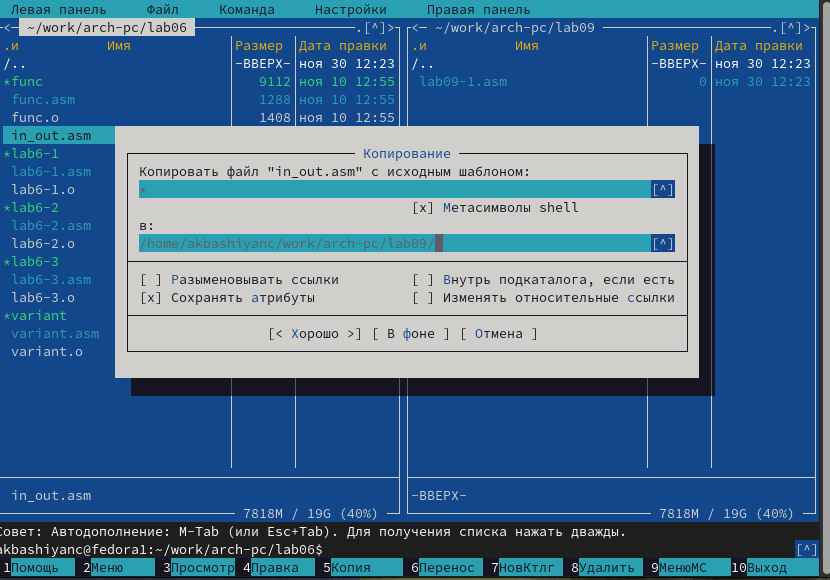


Рис. 2: Копирование in\_out.asm

Введем код в lab09-1.asm и создадим исполняемый файл и запустим его (рис. 3 и 4).

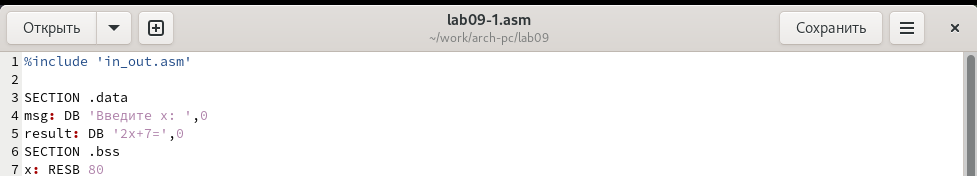


Рис. 3: Ввод кода

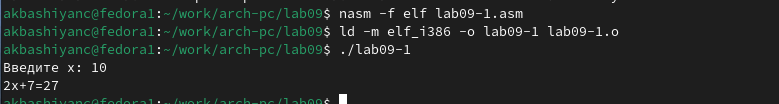


Рис. 4: Запуск файла

## 3.2 Отладка программам с помощью GDB

Создадим файл lab09-2.asm (рис. 5).

Рис. 5: Создание файла

Рис. 5: Создание файла

Введем код в lab09-2.asm (рис. 6).

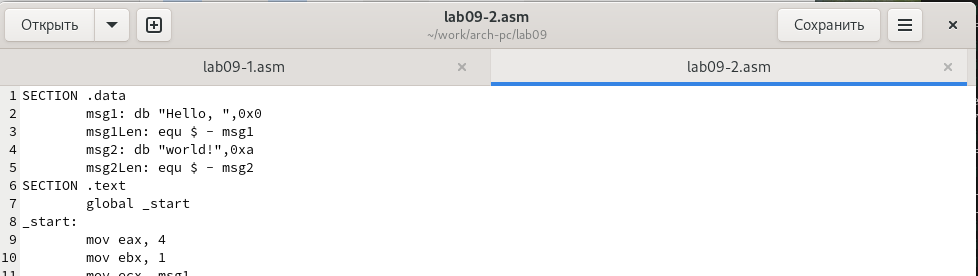


Рис. 6: Ввод кода

Создадим исполняемый файл (рис. 7). Для работы с GDB в исполняемый файл необходимо добавить отладочную информацию, для этого трансляцию программ необходимо проводить с ключом ‘-g’.

Рис. 7: Создание исполняемого файла

Рис. 7: Создание исполняемого файла

Загрузим исполняемый файл в отладчик gdb (рис. 8).

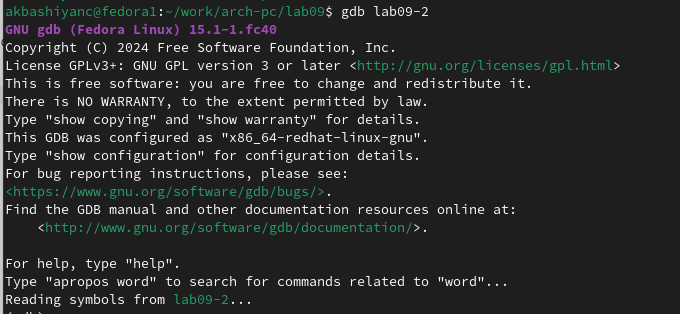


Рис. 8: Загрука файла в gdb

Запустим файл (рис. 9).

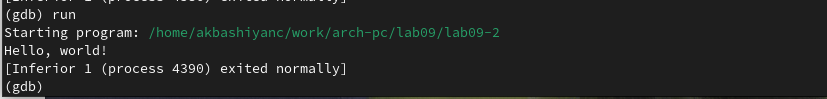


Рис. 9: Запуск файла в gdb

Для более подробного анализа программы установим брейкпоинт на метку \_start, с которой начинается выполнение любой ассемблерной программы, и запустим её (рис. 10).

Рис. 10: Установка брейкпоинта

Рис. 10: Установка брейкпоинта

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

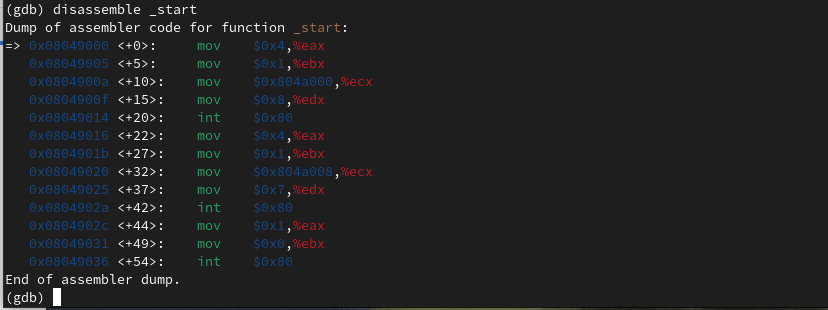


Рис. 11: disassemble

Переключимся на отображение команд с Intel’овским синтаксисом, введя команду set disassembly-flavor intel (рис. 12).

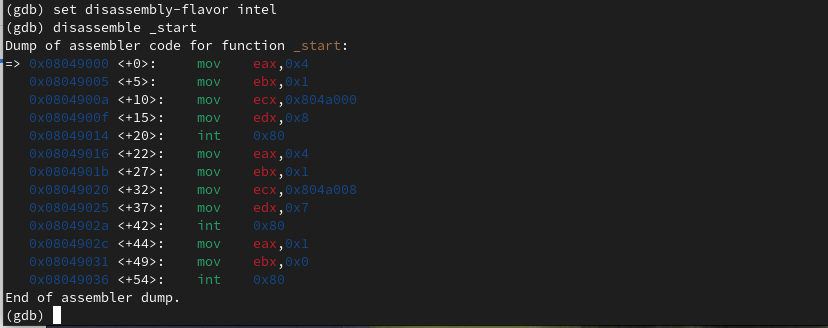


Рис. 12: Переключение на синтаксис Intel

Включим режим псевдографики для более удобного анализа программы (рис. 13).

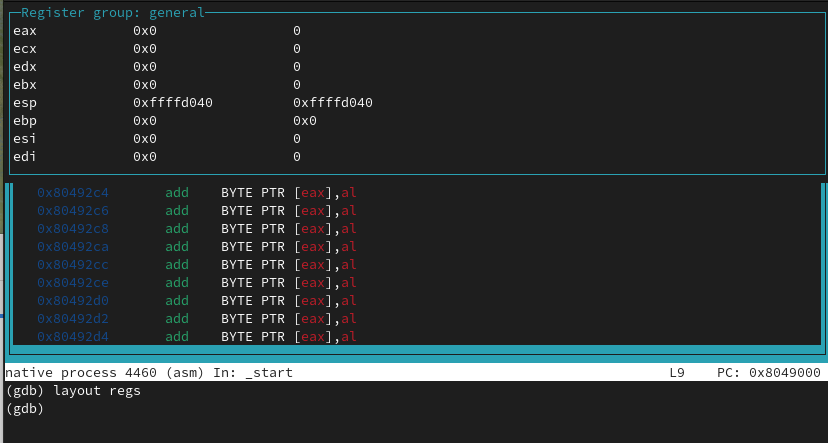


Рис. 13: Включение режим псевдографики

### 3.2.1 Добавление точек останова

На предыдущих шагах была установлена точка останова по имени метки (\_start). Проверим это с помощью команды info breakpoints (рис. 14).

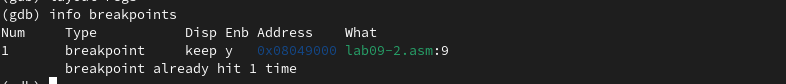


Рис. 14: Включение режим псевдографики

Установим еще одну точку останова по адресу инструкции (рис. 15).

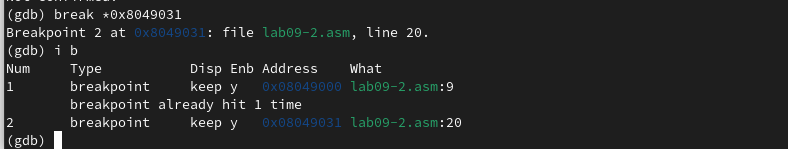


Рис. 15: Установка точки останова

### 3.2.2 Работа с данными программы в GDB

Посмотрим содержимое регистров также можно с помощью команды info registers (рис. 16).

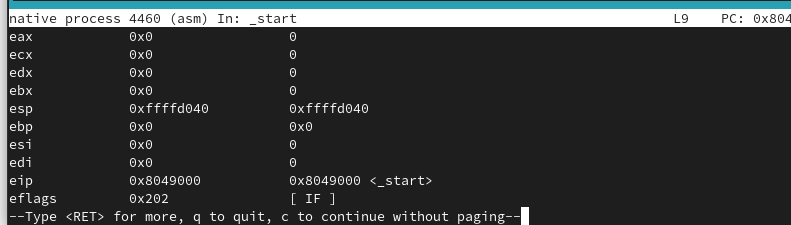


Рис. 16: Содержимое регистров

Посмотрим значение переменной msg1 по имени (рис. 17).

Рис. 17: Содержимое регистров

Рис. 17: Содержимое регистров

Посмотрим значение переменной msg2 по имени (рис. 18).

Рис. 18: Содержимое регистров

Рис. 18: Содержимое регистров

Изменим значение для регистра или ячейки памяти можно с помощью команды set, задав ей в качестве аргумента имя регистра или адрес (рис. 19).

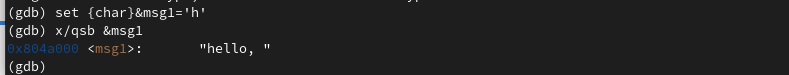


Рис. 19: Изменение значения регистра msg1

Изменим 1 символ в msg2 (рис. 20).



Рис. 20: Изменение значения регистра msg2

С помощью команды set изменим значение регистра ebx (рис. 21).



Рис. 21: Изменение значения регистра ebx

Завершим выполнение программы с помощью команды continue (сокращенно c) и выйдем из GDB с помощью команды quit (сокращенно q) (рис. 22).



Рис. 22: Выход из GDB

### 3.2.3 Обработка аргументов командной строки в GDB

Скопируем файл lab8-2.asm, созданный при выполнении лабораторной работы №8, назовем его lab9-3.asm (рис. 23).

Рис. 23: Копирование lab8-2.asm

Рис. 23: Копирование lab8-2.asm

Создадим исполняемый файл (рис. 24).

Рис. 24: Создание исполняемого файла

Рис. 24: Создание исполняемого файла

Загрузим исполняемый файл в отладчик gdb (рис. 25).

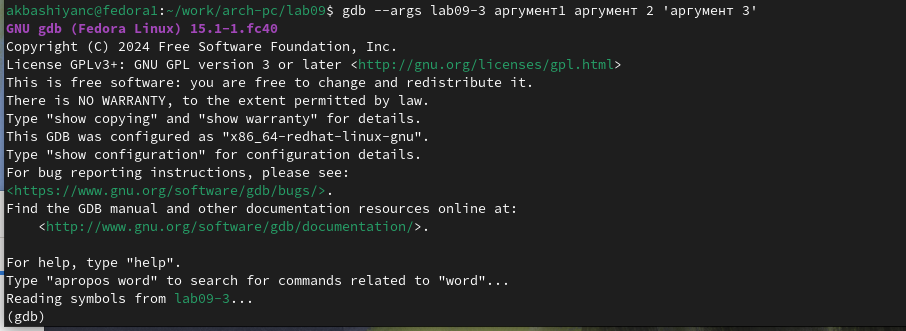


Рис. 25: Загрука файла в gdb

Для начала установим точку останова перед первой инструкцией в программе и запустим ее (рис. 26).

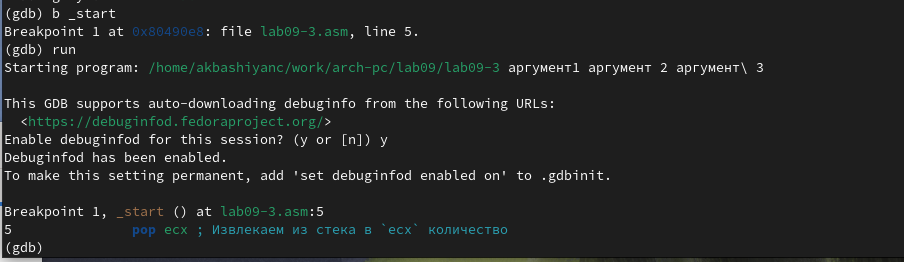


Рис. 26: Загрука файла в gdb

Адрес вершины стека храниться в регистре esp и по этому адресу располагается число равное количеству аргументов командной строки (включая имя программы) (рис. 27).

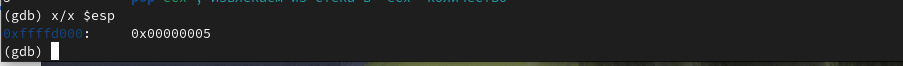


Рис. 27: Вершина стека

Посмотрим остальные позиции стека (рис. 28).

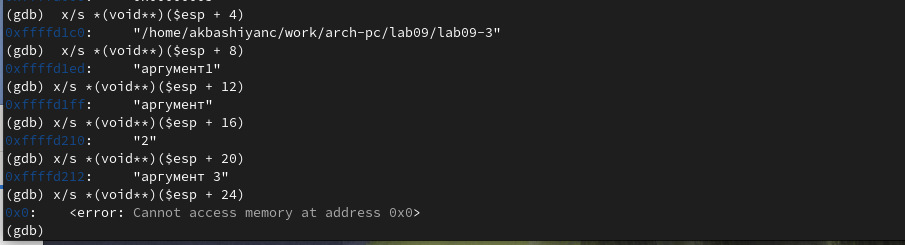


Рис. 28: Остальные позиции стека

## 3.3 Задание для самостоятельной работы

### 3.3.1 Задание 1

Скопируем файл ex1.asm из Лабораторной работы №8 (рис. 29).

Рис. 29: Создание файла

Рис. 29: Создание файла

Изменим код для нахождения суммы функции f(x)=5\*x+17 с помощью подпрограммы (рис. 30).

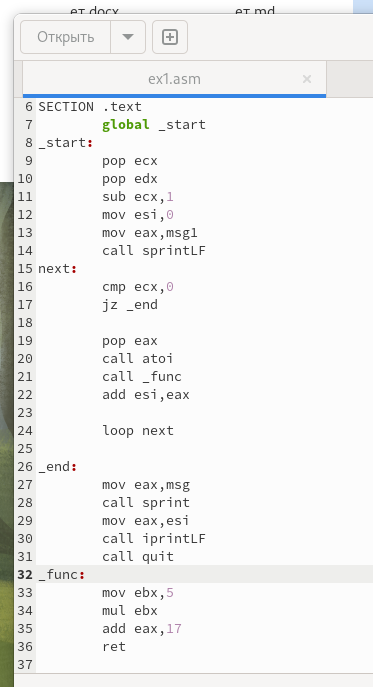


Рис. 30: Изменение кода

Создадим исполняемый файл и запустим его (рис. [-fig. 31).

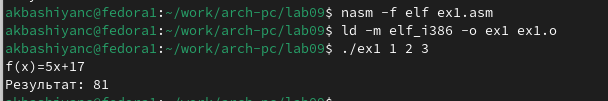


Рис. 31: Запуск файла

### 3.3.2 Задание 2

Создадим файл ex2.asm (рис. 32).

Рис. 32: Создание файла

Рис. 32: Создание файла

Введем код в ex2.asm (рис. 33).

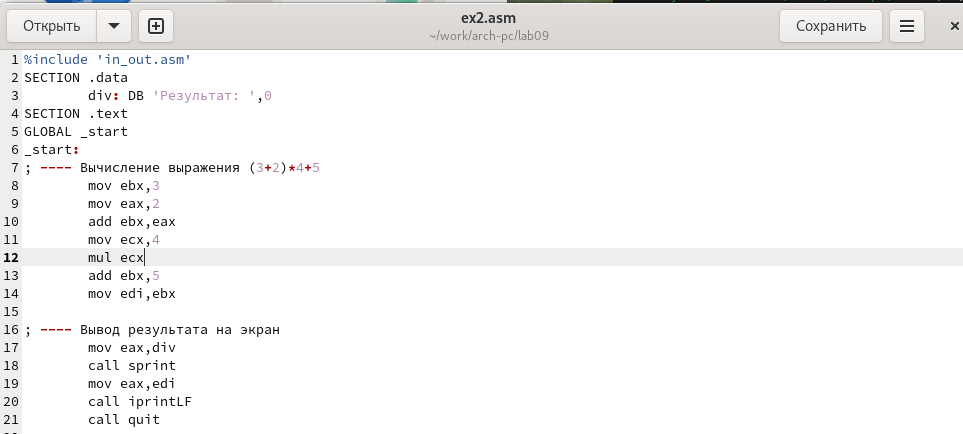


Рис. 33: Ввод кода

Создадим исполняемый файл и запустим файл, как обычно (рис. 34).

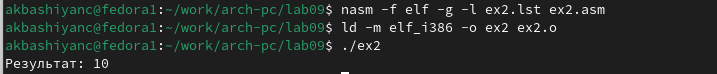


Рис. 34: Создание исполняемого файла

Заметим, что ответ неверный. Выводится 10, хотя должно быть 25.

Загрузим исполняемый файл в отладчик gdb (рис. 35).

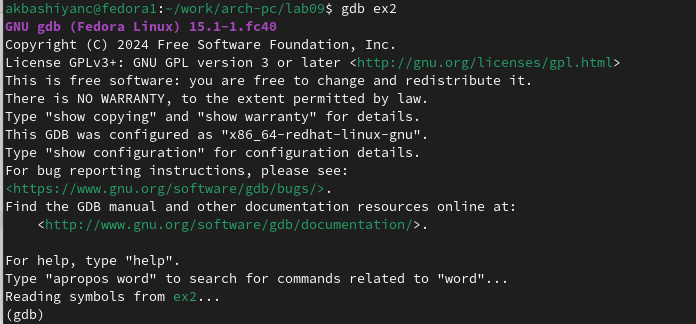


Рис. 35: Загрука файла в gdb

Для начала установим точку останова перед первой инструкцией в программе и запустим ее (рис. 36).

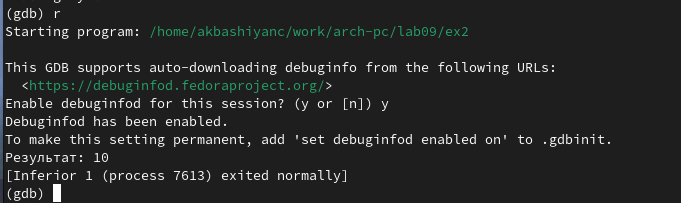


Рис. 36: Запуск файла в gdb

Откроем регистры (рис. 37).

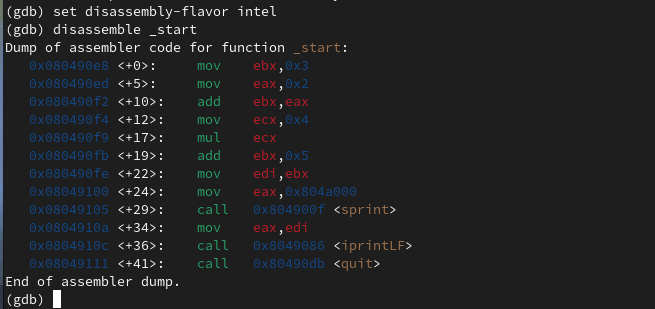


Рис. 37: Регистры

Заметим, что некоторые регистры стояит не на своих местах. Исправим это (рис. 38).

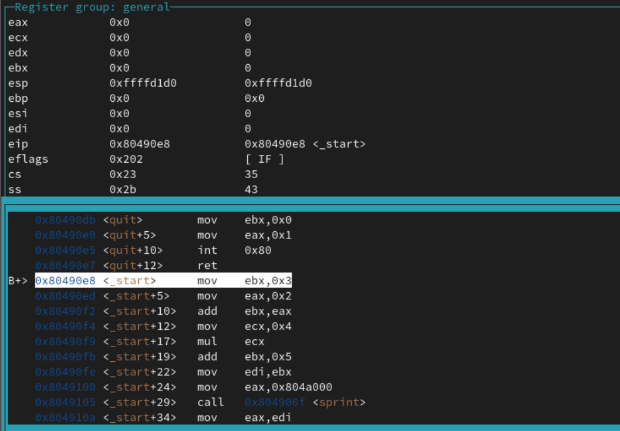


Рис. 38: Исправление регистров

Выйдем из gdb, сформируем заново исполняемый файл из запустим файл заново (рис. 38).

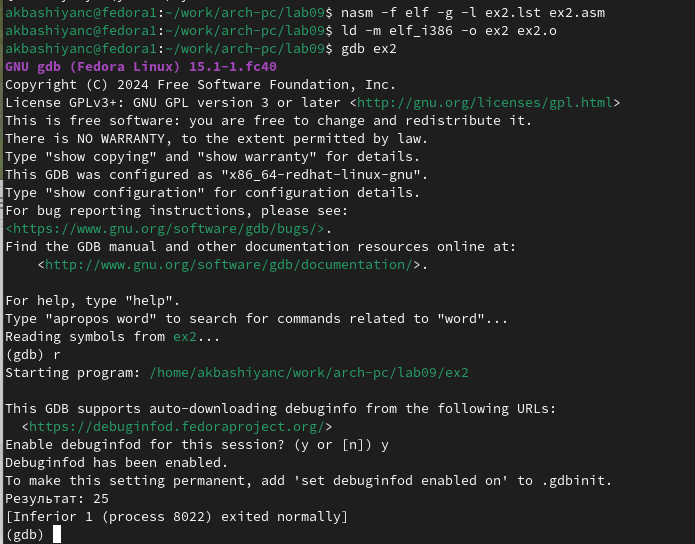


Рис. 39: Запуск исправленного файла

Теперь все считается корректно.

# 4 Выводы

В ходе выполнения работы были получены навыки работы с подпрограммами и методами отладки при помощи GDB.