Лабораторная работа №9

Понятие подпрограммы

Самарханова Полина Тимуровна

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. Реализация подпрограмм в NASM
2. Отладка программ с помощью GDB
3. Добавление точек останова
4. Работа с данными программы в GDB
5. Обработка аргументов командной строки в GDB
6. Задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация подпрограмм в NASM

Я создала каталог для выполнения работы №9 (рис. ??).

Создание каталога

Создание каталога

После перещла в созданную директорию (рис. ??).

Перемещение по директории

Перемещение по директории

Создала файл lab09-1.asm в новом каталоге (рис. ??).

Создание файла

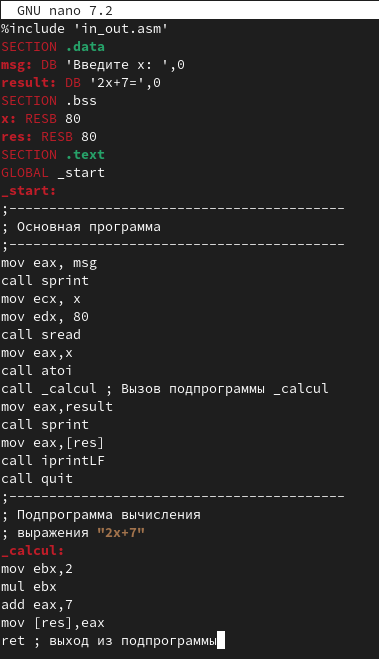
Создание файла

Далее я скопировала файл in\_out.asm в созданный каталог(рис. ??).

Копирование файла

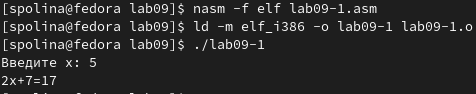
Копирование файла

После чего я открыла файл в GNU nano и переписала код программы из листинга 9.1 (рис. ??).



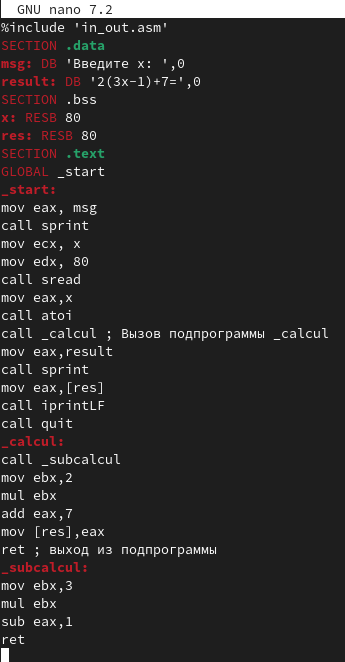
Редактирование файла

Далее создала объектный файл программы и после компановки запустила его (рис. ??).



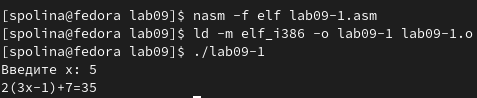
Запуск программы

Я изменила текст файла,добавив подпрограмму sub\_calcul в подпрограмму \_calcul (рис. ??).



Редактирование файла

После я запустила исполняемый файл (рис. ??).



Запуск программы

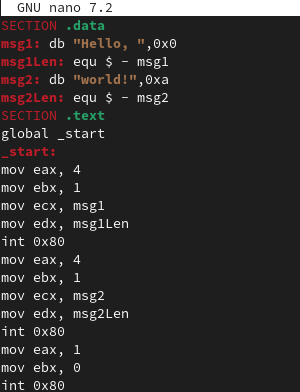
## 3.2 Отладка программ с помощью GDB

Далее я создала файл lab09-2.asm, используя команду touch (рис. ??).

Создание файла

Создание файла

Записала код программы из листинга 9.2,который выводит сообщение Hello world (рис. ??).



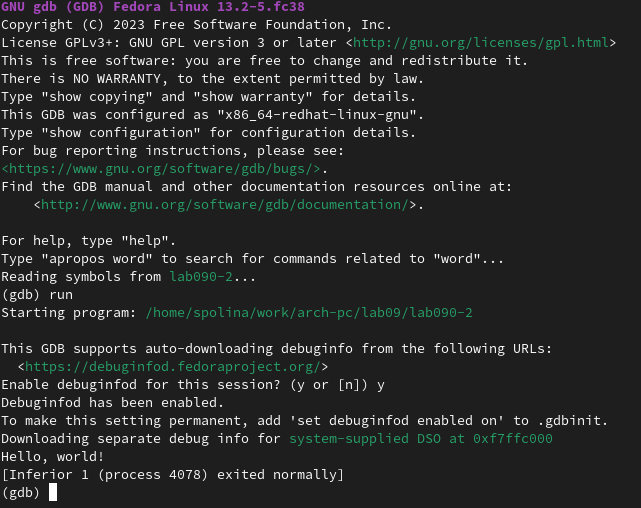
Редактирование файла

Получила исполняемый файл. Для работы с GDB провела трансляцию программ с ключом “-g” и загрузила исполняемый файл в отладчик (рис. ??).

Запуск исполняемого файла

Запуск исполняемого файла

После я проверила работу программы в оболоке GDB с помощью команды run (рис. ??).



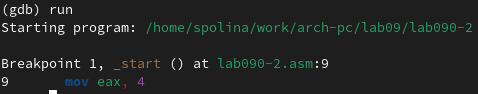
Запуск программы в отладчике

Установила брейкпоинт на метку \_start,с которой начинается выполнение ассемблерной программы (рис. ??).

Установка брейкпоинта

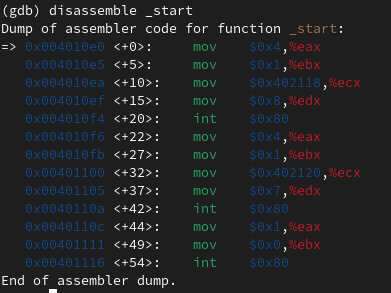
Установка брейкпоинта

Далее запустила её (рис. ??).



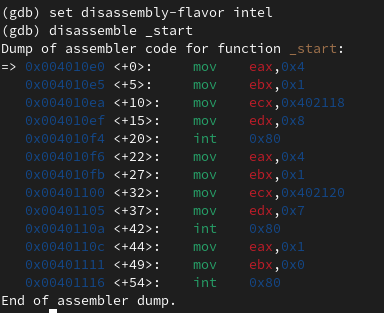
Запуск

С помощью команды “disassemble \_start” я просматрела дисассимилированный код программы (рис. ??).



Диссассимилированный код программы

После я переключилась на отображение команд с Intel’овским синтаксисом, введя команду “set disassembly-flavor intel” (рис. ??).

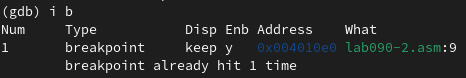


Отображение с Intel’овским синтаксисом

Основное различие заключается в том,что в режиме Intel пишется сначала сама команда,а потом её машинный код, в то время как в режиме ATT идет сначала машинный код,а только потом сама команда.

## 3.3 Добавление точек останова

Я проверила наличие точки останова с помощью команды info breakpoints (рис. ??).



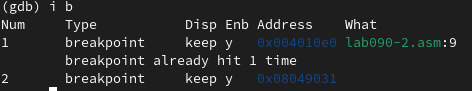
Точка останова

Устанавила ещё одну точку останова по адресу инструкции (рис. ??).

Установка точки останова

Установка точки останова

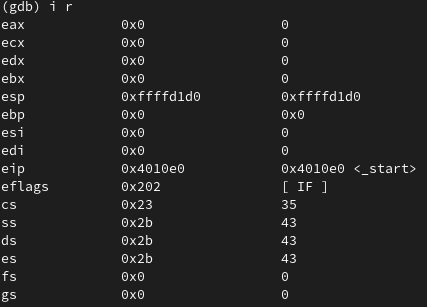
Просматрела информацию (рис. ??).



Точки останова

## 3.4 Работа с данными программы в GDB

Далее я просмотрела содержимое регистров с помощью команды info register (рис. ??).



info register

Узнала значение переменной msg1 по имени (рис. ??).

Значение переменной по имени

Значение переменной по имени

После просмотрела значение переменной msg2 по адресу,который можно определить по дизассемблированной инструкции (рис. ??).

Значение переменной по адресу

Значение переменной по адресу

Изменила первый символ переменной msg1 (рис. ??).

Изменение переменной

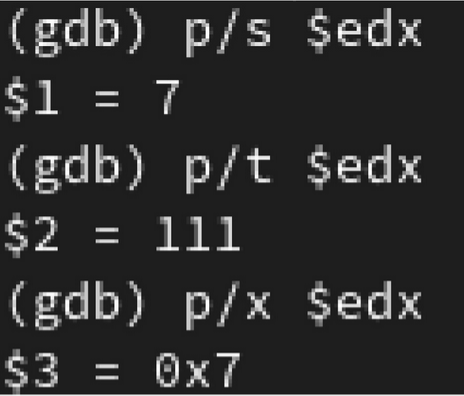
Изменение переменной

Также поменяла первый символ переменной msg2 (рис. ??).

Изменение второй переменной

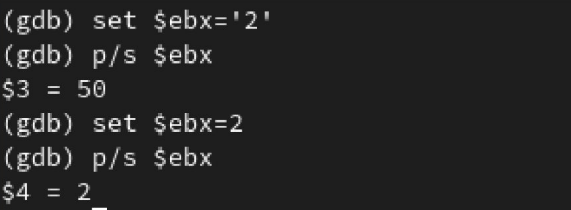
Изменение второй переменной

Далее я вывела значение регистра edx (рис. ??).



Изменение значений в разные форматы

С помощью команды set я изменила значение регистра ebx (рис. ??).



Изменение значений ebx

Значение регистра отличаются,так как в первом случае мы выводим код символа 2,который в десятичной системе счисления равен 50,а во втором случае выводится число 2,представленное в этой же системе.

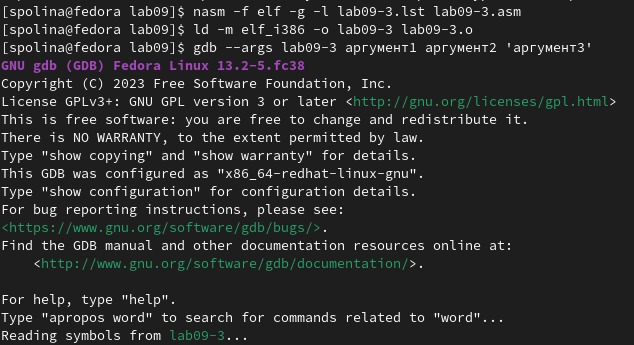
## 3.5 Обработка аргументов командной строки в GDB

Я скопировала файл lab8-2.asm в файл с именем lab09-3.asm (рис. ??).

Копирование файла

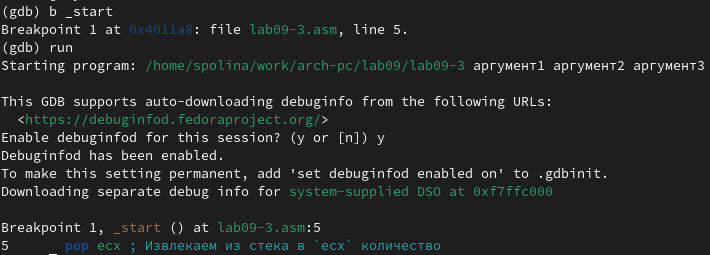
Копирование файла

Создала исполняемый файл,использую ключ –args для загрузки программы в GDB. Загрузила исполняемый файл,указав аргументы (рис. ??).



Создание файла

Устанавила точку останова перед первой инструкцией в программе и запустила её (рис. ??).



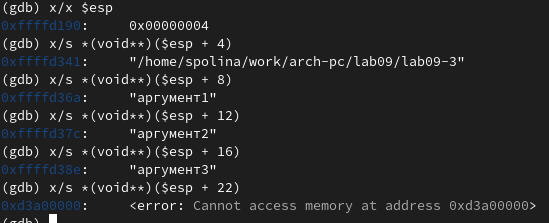
Запуск программы с точкой останова

Просмотрела адрес вершины стека, который хранится в регистре esp (рис. ??).

Регистр esp

Регистр esp

Ввела другие позиции стека(рис. ??).



Позиции стека

Количество аргументов командной строки 4,следовательно и шаг равен четырем.

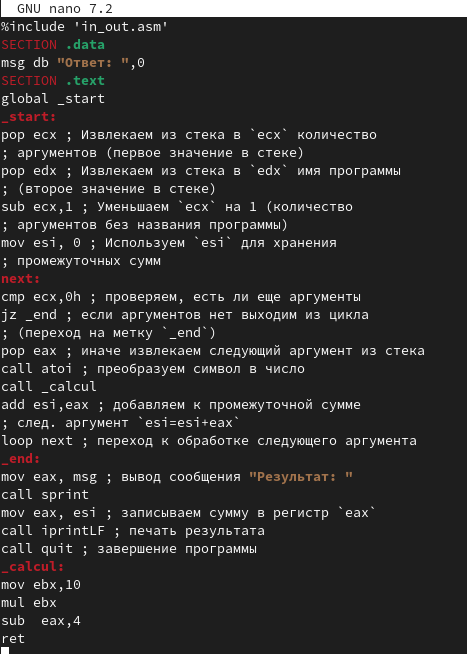
## 3.6 Задание для самостоятельной работы

Далее я создала файл для первого самостоятельного задания lab09-4.asm (рис. ??).

Создание файла

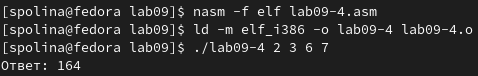
Создание файла

После чего отредактировала код программы lab8-4.asm,добавив подпрограмму,которая вычисляет значения функции f(x) (рис. ??).



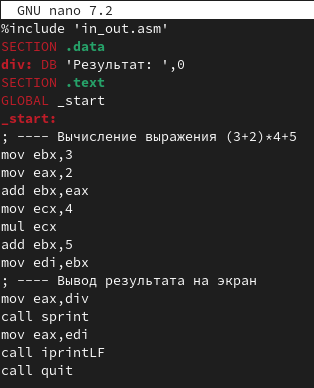
Редактирование файла

Далее я создала исполняемый файл и ввела аргументы (рис. ??).



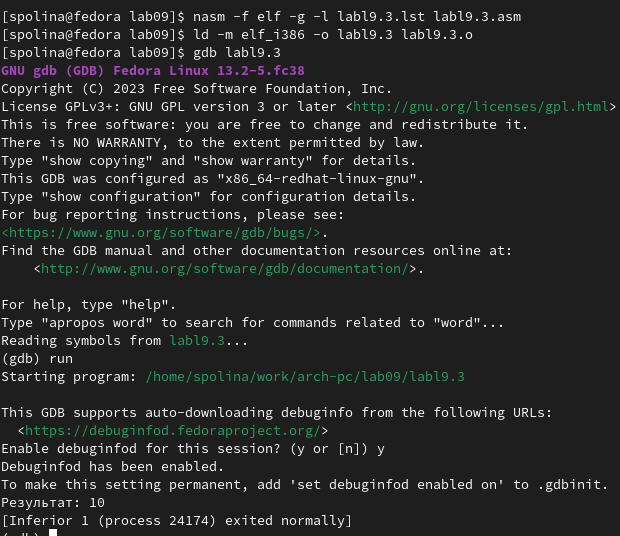
Запуск программы

Создала файл и ввела код из листинга 9.3 (рис. ??).



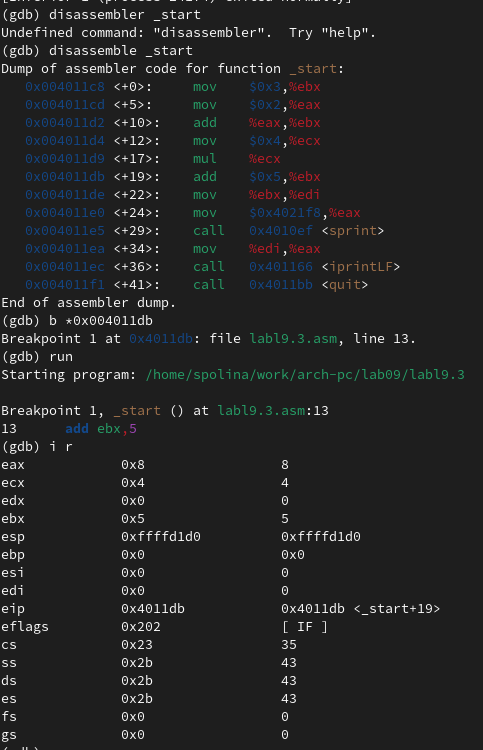
Редактирование файла

Далее я открыла файл в отладчике GDB и запускаю программу (рис. ??).



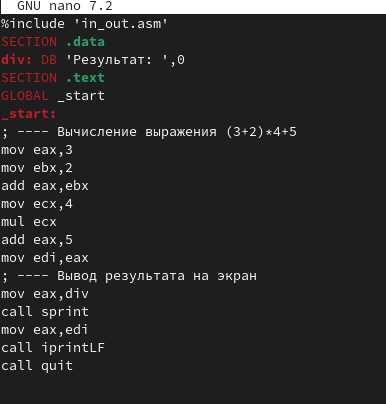
Запуск программы в отладчике

Просмотрев дисассимилированный код программы, поставила точку останова перед прибавлением 5 и открыла значения регистров на данном этапе (рис. ??).



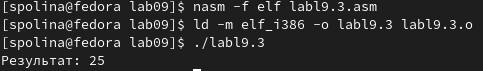
Действия в отладчике

Как можно увидеть, регистр ecx со значением 4 умножается не на ebx,сложенным c eax, а только с eax со значением 2. Значит нужно поменять значения регистров(например присвоить eax значение 3 и просто прибавит 2. После изменений программа будет выглядить следующим образом:(рис. ??).



Измененная программа

Далее я запустила программу (рис. ??).



Запуск программы

# 4 Выводы

В данной работе я приобрела навыки написания программ с подпрограммами и познакомилась с методами отладки при помощи GDB.

# Список литературы

Лабораторная работа №9