Отчёт по лабораторной работе №9

Дисциплина: Архитектура компьютеров

Лазарев Даниил Михайлович

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

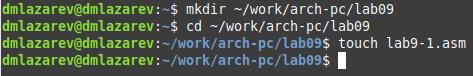
# 2 Теоретическое введение

Отладка — это процесс поиска и исправления ошибок в программе. В общем случае его можно разделить на четыре этапа:

• обнаружение ошибки; • поиск её местонахождения; • определение причины ошибки; • исправление ошибки. Можно выделить следующие типы ошибок: • синтаксические ошибки — обнаруживаются во время трансляции исходного кода и вызваны нарушением ожидаемой формы или структуры языка; • семантические ошибки — являются логическими и приводят к тому, что программа запускается, отрабатывает, но не даёт желаемого результата; • ошибки в процессе выполнения — не обнаруживаются при трансляции и вызывают прерывание выполнения программы (например, это ошибки, связанные с переполнением или делением на ноль).

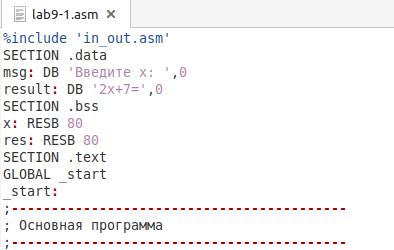
# 3 Выполнение лабораторной работы

Создадим каталог для программ лаб. работы н.9, перейдем в него и создадим файл "lab9-1.asm" (рис. @fig:1)



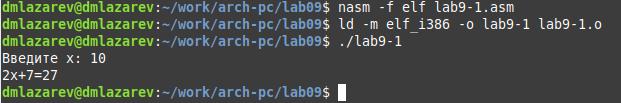
Создание файла в каталоге

Введем в созданный файл текст программы из предложенного нам листинга 9.1 (рис. @fig:2)



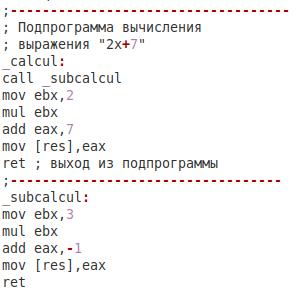
Код программы в файле

Создадим исполняемый файл и запустим его, предварительно скопировав из предыдущей лаб. работы файл "in\_out.asm" для корректной работы (рис. @fig:3)



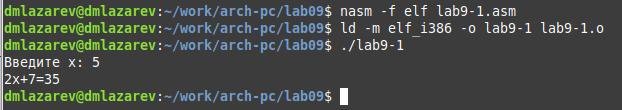
Преобразование в исполняемый файл

Далее дополним код так, чтобы Х проходил еще через одну функцию. (рис. @fig:4)



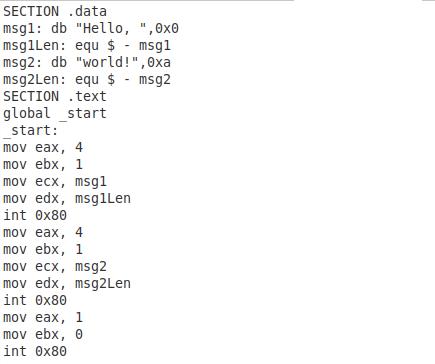
Измененный код программы

Преобразуем в исполняемый файл и проверим правильность выполнения. (рис. @fig:5)



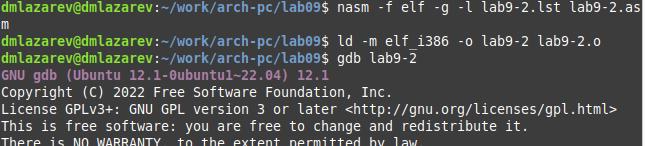
Преобразование измененного файла

Создадим файл "lab9-2.asm" и вставим в него предложенный нам листинг 9.2 (рис. @fig:6)



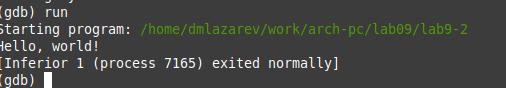
Текст в файле

Преобразуем файл "lab9-2.asm" в исполняемый, добавив в него отладочную информацию. (рис. @fig:7)



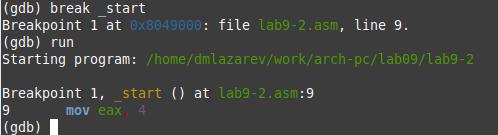
Преобразование файла в исполняемый

Загрузим исполняемый файл в отладчик gdb и запустим программу с помощью команды "run" для проверки ее работоспособности.(рис. @fig:8)



Файл в gdb

Установим брейкпоинт с помощью команды "break" для более подробного анализа и запустим повторно.(рис. @fig:9)



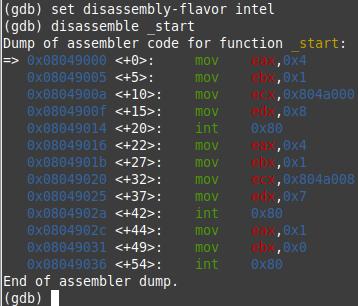
Установка брейкпоинта

Просмотрим код с помощью команды "disassemble".(рис. @fig:10)



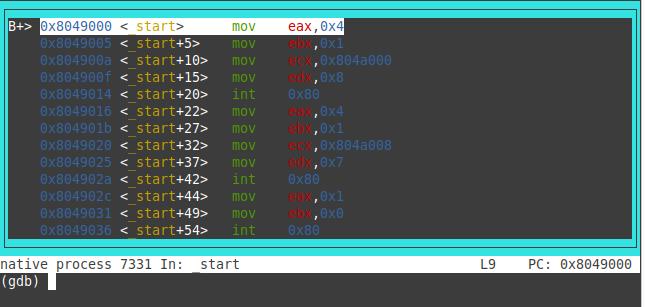
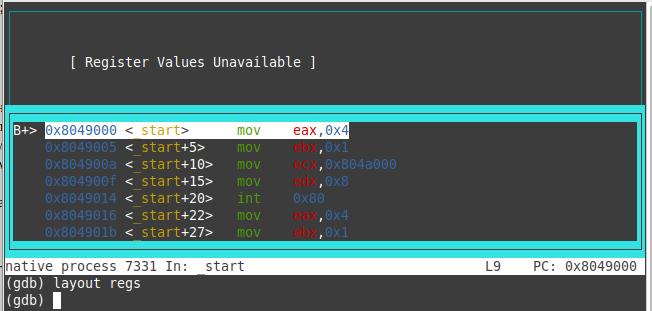
Просмотр кода с помощью disassemble

Включим отображение команд на синтаксисе Интела.(рис. @fig:11)

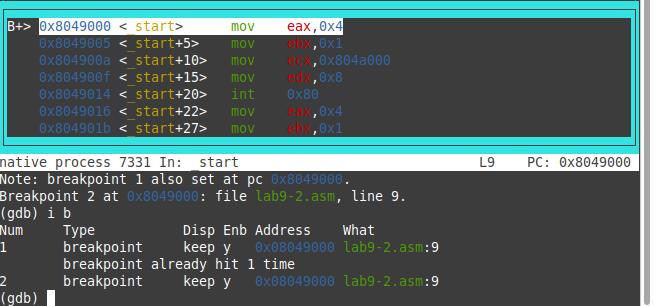


Отображение команд

Включим режим псевдографики для более удобного анализа программы.(рис. @fig:12; рис. @fig:13)

Проверим все установленные брейкпоинты и установим еще одну точку остановки по адресу инструкции.(рис. @fig:14)



Проверка брейкпоинтов

Просмотрим значение переменной msg1 по имени.(рис. @fig:15)

Значение переменной

Значение переменной

Просмотрим значение переменной msg2 по адресу. (рис. @fig:16)

Значение переменной

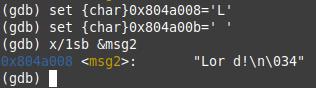
Значение переменной

Изменим первый символ переменной msg1. (рис. @fig:17)

Изменение значения переменной

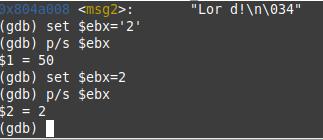
Изменение значения переменной

Так же изменим второй символ во второй переменной msg2 (рис. @fig:18)



Изменение значения переменной

С помощью команды "set" изменим значение регистра ebx. Разница вывода в том, что во втором случае мы задаем число, которое нам необходимо, не зависимо от заложенного в переменную, а в первом первые два символа. (рис. @fig:19)



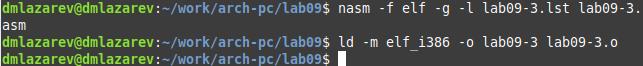
Изменение значения регистра

Скопируем файл "lab8-2.asm" созданный при выполнении предыдущей лабораторной работы и переименуем в "lab9-3.asm" (рис. @fig:20)

Копирование файла

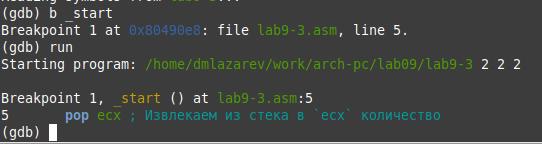
Копирование файла

Преобразуем файл в исполняемый. (рис. @fig:21)



Преобразование файла в исполняемый

Загрузим программу в gdb с использованием ключа --args и указав 3 аргумента. Установим точку остановки перед первой инструкцией в программе и запустим ее. (рис. @fig:22)



Файл в gdb

Введя команду "x/x $esp" увидим то, что аргументов командной строки 4(включая название файла).(рис. @fig:23)

Количество аргументов

Количество аргументов

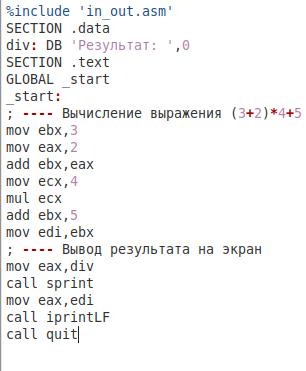
# 4 Выполнение самостоятельной работы

Преобразуем программу из предыдущей лабораторной работы реализовав вычисление значения функции как подпрограмму. (рис. @fig:24)

|  |
| --- |
| Преобразование программы |

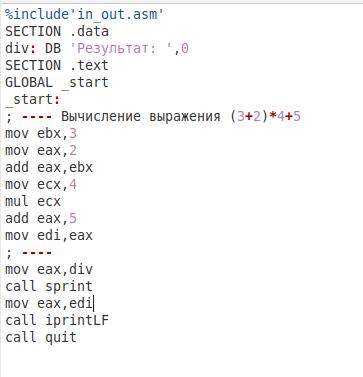
Преобразование программы

Создадим файл “lab9-4.asm” и внесем в него листинг 9.3. С помощью gdb проанализируем и найдем ошибку в коде. (рис. ??)



Код в файле

Исправим ошибку в коде. (рис. ??)



Исправленный файл

# 5 Выводы

В ходе лабораторной работы мы научились написанию программ с использованием подпрограмм, а так же с отладкой программ с помощью GDB.