Отчет

Лабораторная работа №10

Щанкина Екатерина Викторовна

Содержание

# 1 Цель работы

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

# 2 Задание

Приобрести навыки написания программ с использованием подпрограмм.

Ознакомиться с методами отладки при помощи GDB и его основными возможно- стями.

# 3 Выполнение лабораторной работы

1. Создаем каталог для выполнения лабораторной работы, переходим в него и создаем файл. (рис. 1)

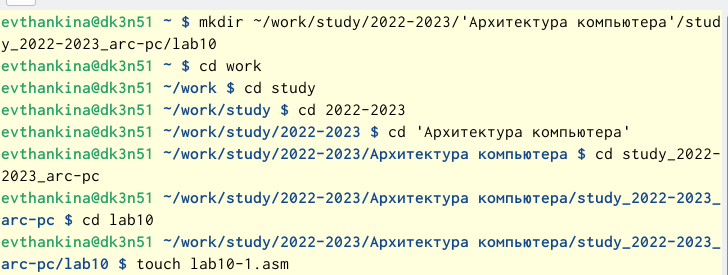


Рис. 1: Каталог

1. Рассмотрим программу вычисления арифметического выражения с помощью подпрограммы \_calcul. (рис. 2) (рис. 3)

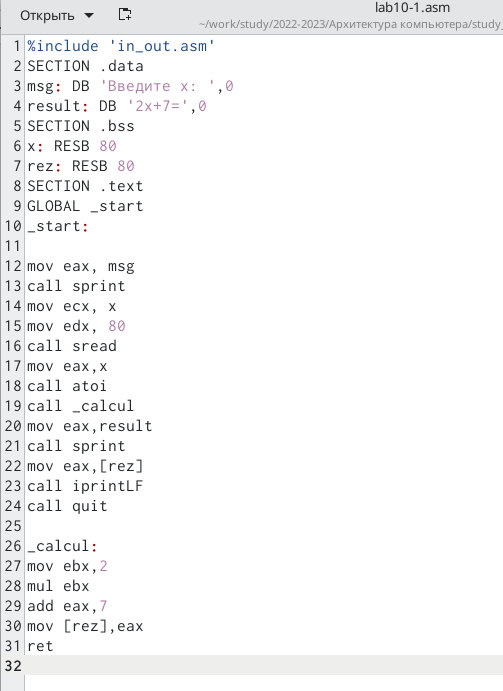


Рис. 2: Текст программы

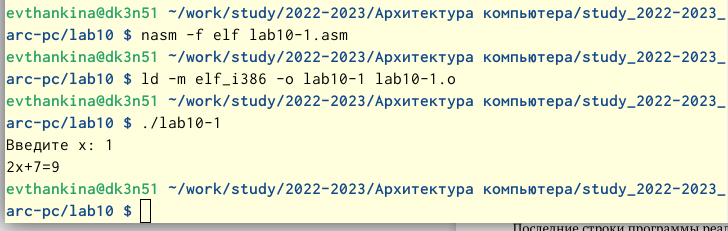


Рис. 3: Проверка программы

1. Изменила программу, добавила подпрограмму g(x)=3x-1 (рис. 4) (рис. 5)

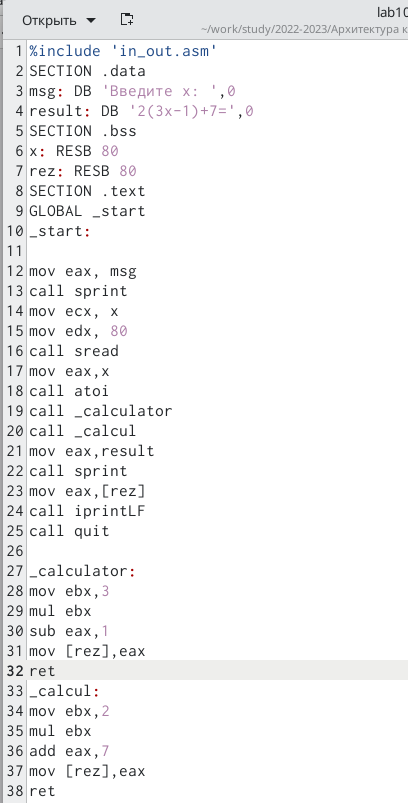


Рис. 4: Текст программы

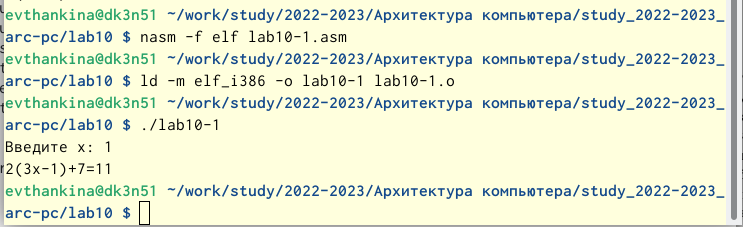


Рис. 5: Проверка программы

Программа работает правильно, ведь если мы подставим x=1, то получим 11.

1. Создала файл lab10-2.asm для Листинга №2. Написала программу печати сообщения Hello world!. (рис. 6)

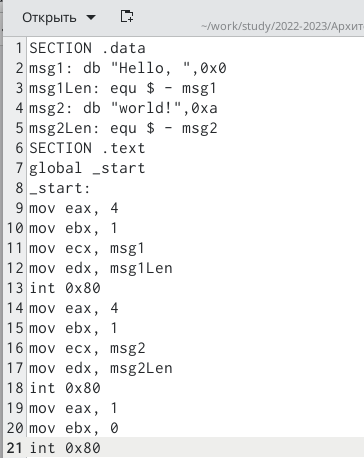


Рис. 6: Текст программы

1. Проверила работу программы. (рис. 7)

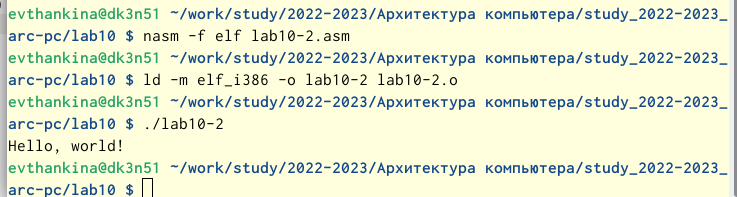


Рис. 7: Проверка программы

1. Загрузила исполняемый файл в отладчик. (рис. 8)

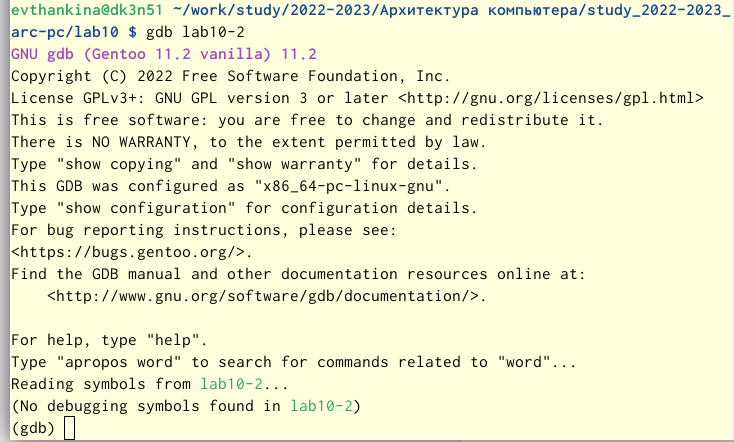


Рис. 8: GDB

Проверим работу программы с помощью запуска в оболочке GDB c помощью команды ‘run’.(рис. 9) Установила брейкпоинт на метку \_start и запустила программу. (рис. 9) Посмотрела дисассимблированный код программы с помощью команды disassemble начиная с метки \_start (рис. 9)



Рис. 9: RUN

Переключилась на отображение команд с помощью команды set disassembly-flavor intel. (рис. 10)

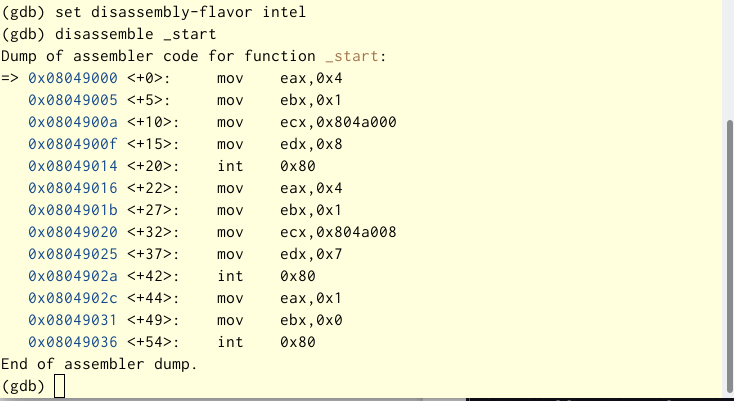


Рис. 10: set disassembly-flavor intel

Включила режим псевдографики.(рис. 11)

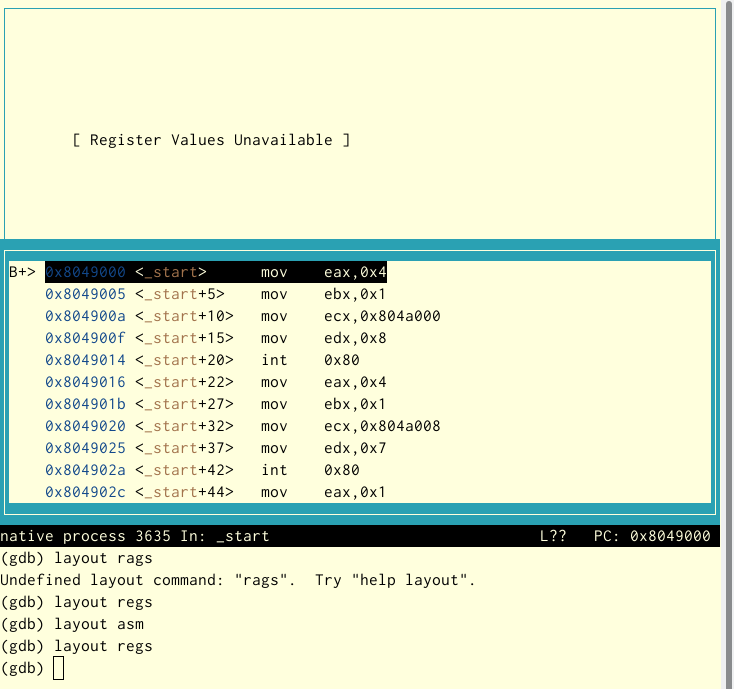


Рис. 11: Псевдографика

В режиме псевдографики у нас есть три окна: - в верхней части названия регистров и их текущие значения. - в средней части результат дисассимилирования программы - в нижней части мы можем вводить команды Различия синтаксиса машинных комнад в режимах ATT и Intel заключаются в том, что в синтаксисе Intel не используются символы $ и %,операнды меняются местами.

1. Проверим точки останова с помощью команды “info breakpoints”. (рис. 12)

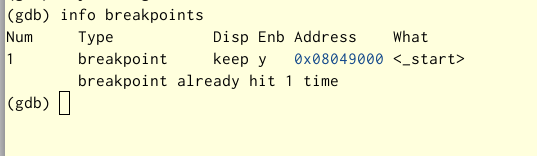


Рис. 12: info breakpoints

Установила точку останова по адресу инструкции (ebx,0x0) (рис. 13)

Рис. 13: Точка останова

Рис. 13: Точка останова

1. Выполнила 5 инструкций с помощью команды stepi и проследила значение регистров. (рис. 14) Мы можем увидеть, что изменяются значения регистров eax, ebx, ecx, edx.

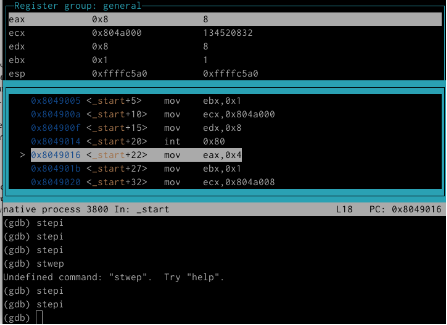


Рис. 14: stepi

Команда ‘x’ позволяется посмотреть содержимое памяти. (рис. 15) (рис. 16)

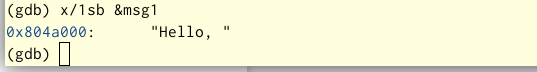


Рис. 15: х

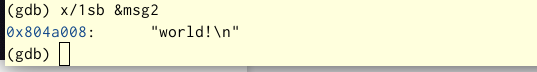


Рис. 16: х

Воспользуемся командой set, которая помогает изменить значение для регистра и ячейки. (рис. 17)

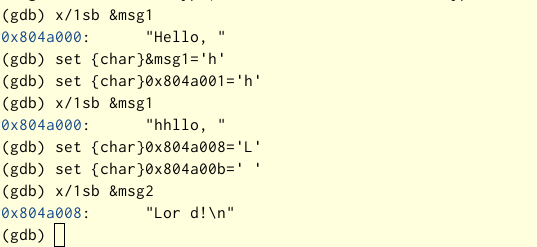


Рис. 17: Изменение значения регистров

С помощью команды ‘set’ изменила значение регистра ebx. (рис. 18)



Рис. 18: Изменение значения регистров

Различие в том, что код символа “2” - “110010” а это есть число 50.

1. Копируем файл из лабораторной работы №9 с новым именем и создаем исполняемый файл. Для загрузки в GDB программы с аргументами нужно использовать ключ -args. (рис. 19)

Чтобы исследовать расположение аргументов командной строки усталовим точку останова и запустим команду. (рис. 19)

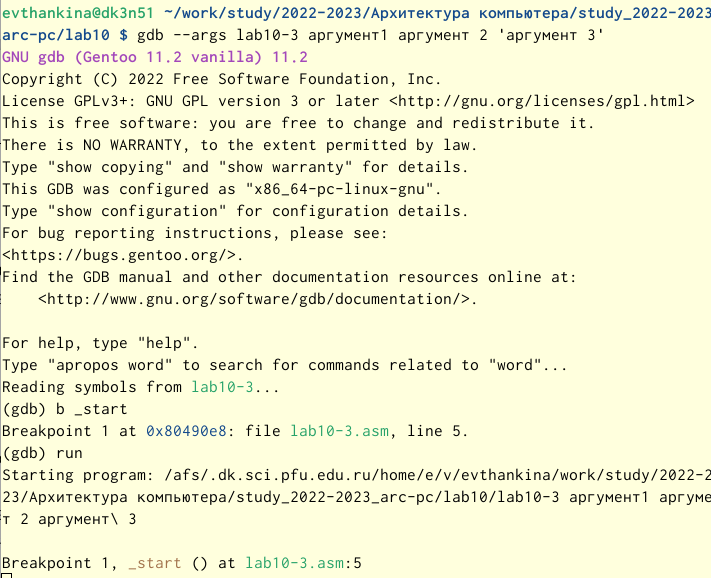


Рис. 19: -args

Посмотрим позиции стека.(рис. 20)

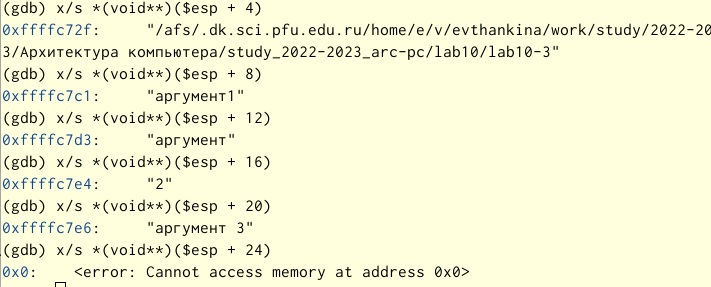


Рис. 20: стек

В данном случае шаг изменения адреса равен 4, т.к. в теле цикла next 4 строки ввода.

# 4 Выполнение самостоятельной работы

1. Скопировала файл из 9 лабораторной работы с помощью команды cp. (рис. 21)

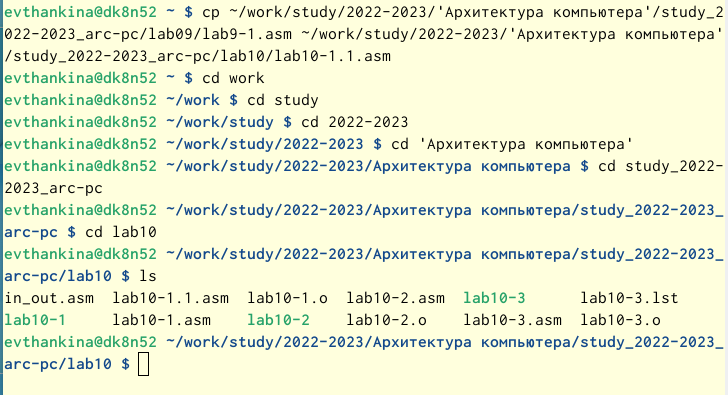


Рис. 21: Копирование

Изменила программу с использованием подпрограмм и запустила ее. (рис. 22) (рис. 23)

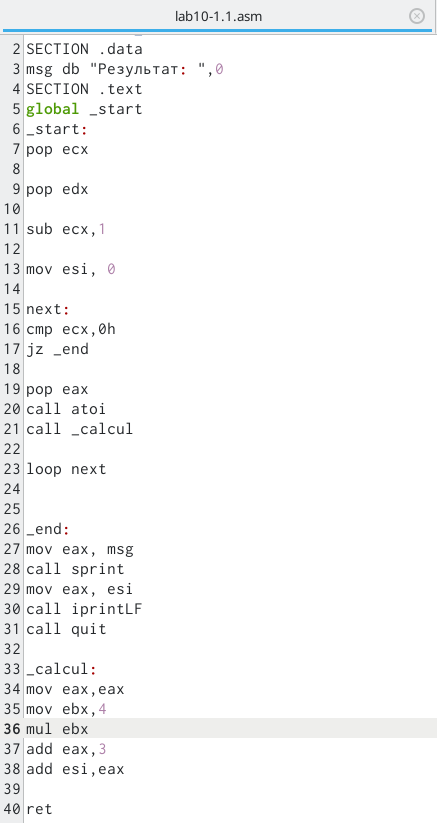


Рис. 22: Текст программы

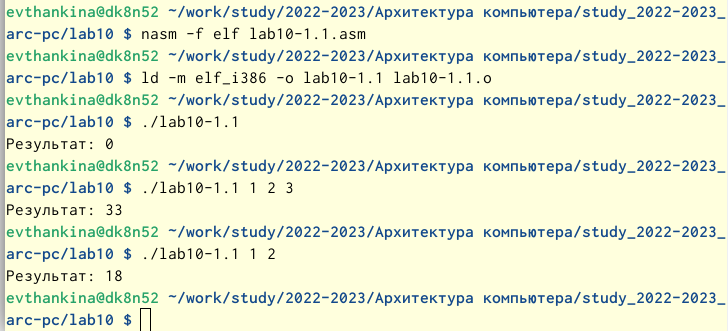


Рис. 23: Проверка

1. Написала программу из листинга и проверила с помощью отладчика. (рис. 24)

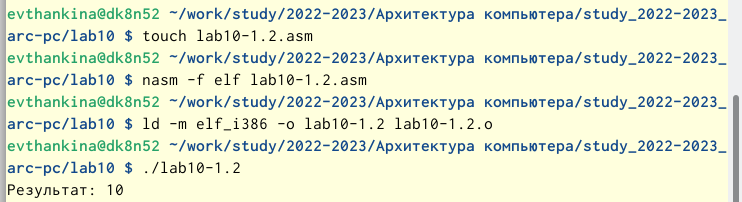


Рис. 24: Проверка файла

Результат программы неверный, для устранения ошибки запускаем отладчик.(рис. 25) Далее установила точку останова. (рис. 25)

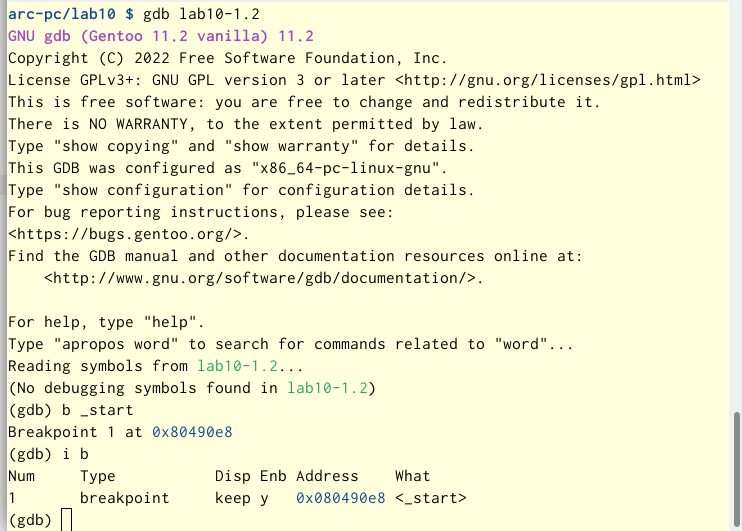


Рис. 25: Отладчик

Запустила код программы с помощью команды run. (рис. 26)

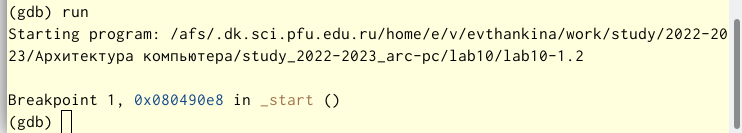


Рис. 26: Запуск кода

Включила режим псевдографики и пошагово прошла все строчки.(рис. 27)

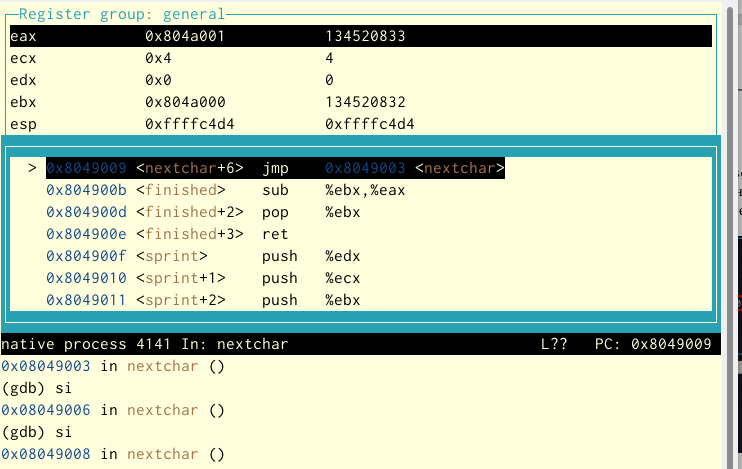


Рис. 27: Поиск ошибки

Мы можем увидеть, что регистр eax должен умножаться на 4, но умножался регистр ebx. А еще число 5 прибавлялось не к произведению, а только к ebx. Исправим ошибки. (рис. 28) (рис. 29)

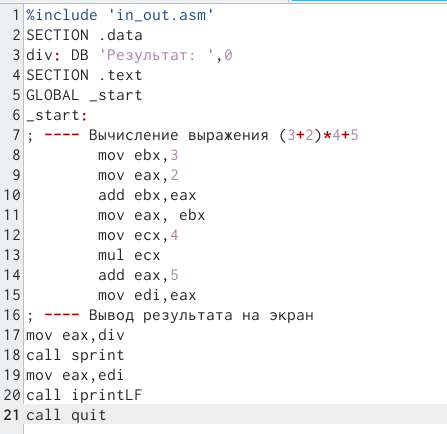


Рис. 28: Исправленный текст программы

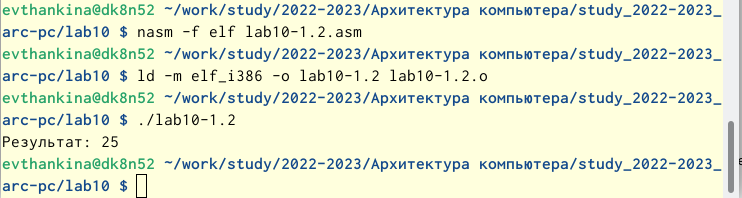


Рис. 29: Проверка программы

Программа работает верно.

# 5 Выводы

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

# Список литературы