Отчет по лабораторной работе №10

Дисциплина: Архитектура компьютера

Никулина Ксения Ильинична

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. Познакомиться с методами отладки при помощи GDB и его основными возможностями.

# 3 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. Точки останова — это специально отмеченные места в программе, в которых программа-отладчик приостанавливает выполнение программы и ждёт команд. Подпрограмма — это, как правило, функционально законченный участок кода, который можно многократно вызывать из разных мест программы. В отличие от простых переходов из подпрограмм существует возврат на команду, следующую за вызовом.

# 4 Выполнение лабораторной работы

1. Создала каталог для выполнения лабораторной работы No 10, перешла в него и создала файл lab10-1.asm (рис. 1)

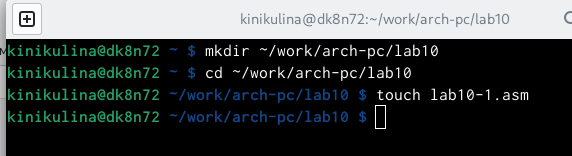


Рис. 1: Содание файла

1. Ввела в файл lab10-1.asm текст программы из листинга 10.1. Создала исполняемый файл и проверила его работу(рис. 2, рис.3 )

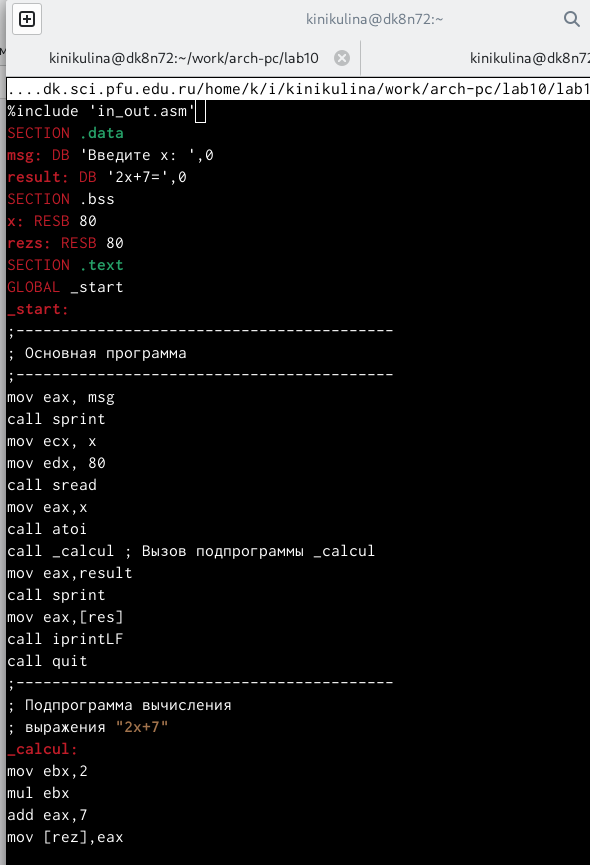


Рис. 2: Текст

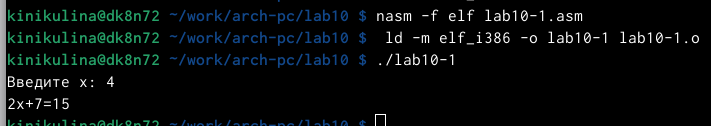


Рис. 3: Проверка работы файла

1. Изменила текст программы, добавив подпрограмму \_subcalcul в подпрограмму \_calcul, для вычисления выражения f(g(x)), где x вводится с клавиатуры, f(x) = 2x + 7, g(x) = 3x − 1.Создала исполняемый файл и проверила его работу (рис. 4, рис.5 )

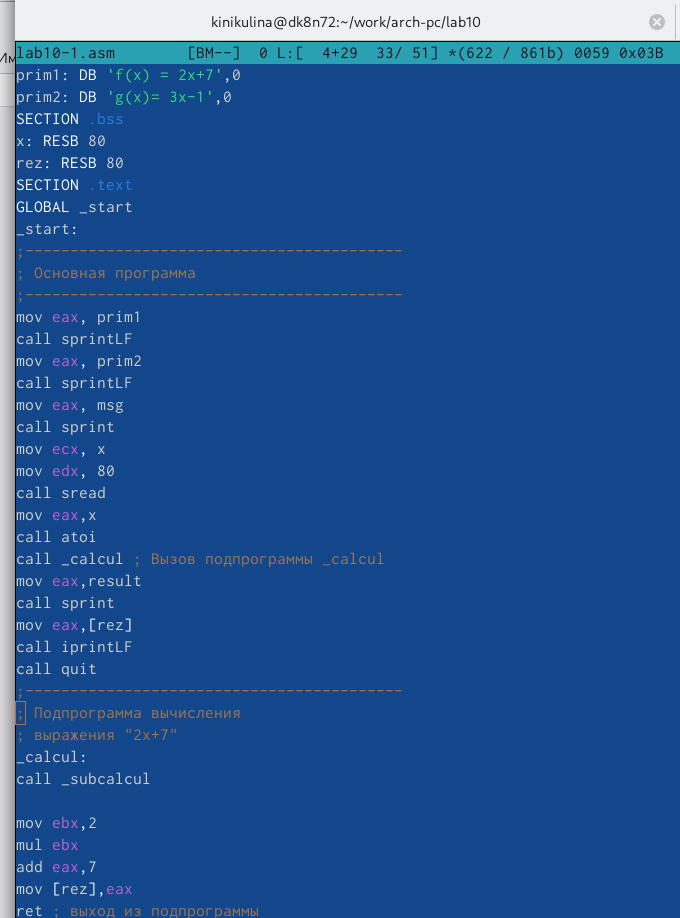


Рис. 4: Текст

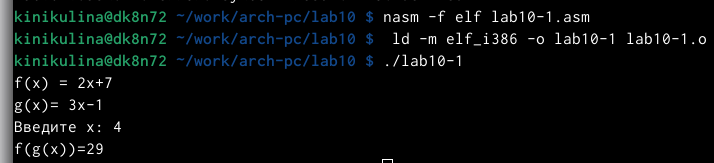


Рис. 5: Проверка работы файла

1. Создала файл lab10-2.asm с текстом программы из Листинга 10.2 (рис. 6, рис.7 )

Рис. 6: Создание файла

Рис. 6: Создание файла

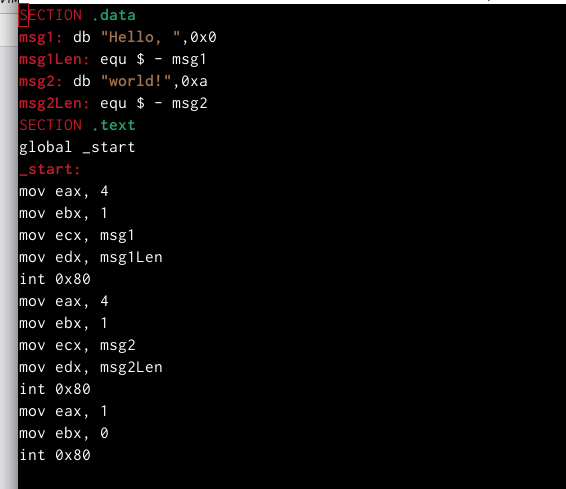


Рис. 7: Текст

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

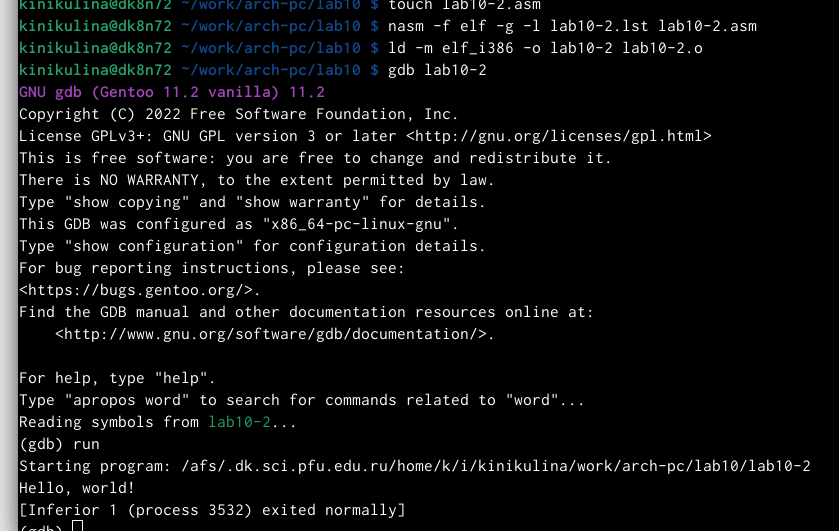


Рис. 8: Загрузка файла

1. Загрузила исполняемый файл в отладчик gdb. Проверила работу программы, запустив ее в оболочке GDB с помощью команды run (рис. 9 )

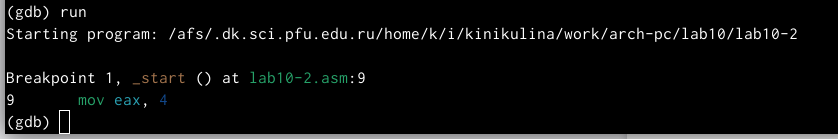


Рис. 9: Проверка работы

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

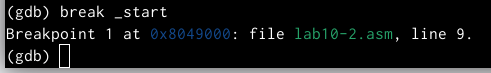


Рис. 10: Запуск

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

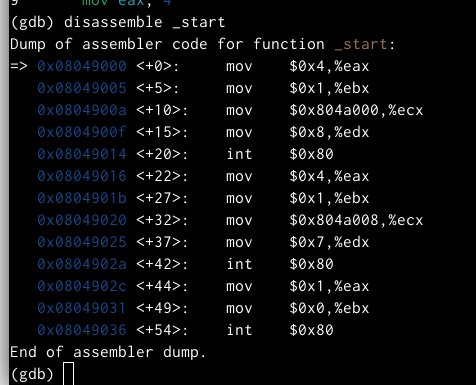


Рис. 11: Код программы

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

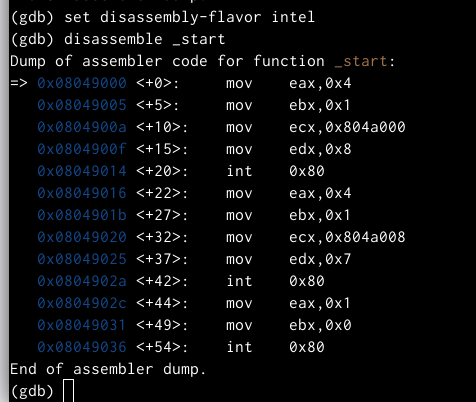


Рис. 12: Команды

1. Включил режим псевдографики для более удобного анализа программы (рис. 13, рис.14 )

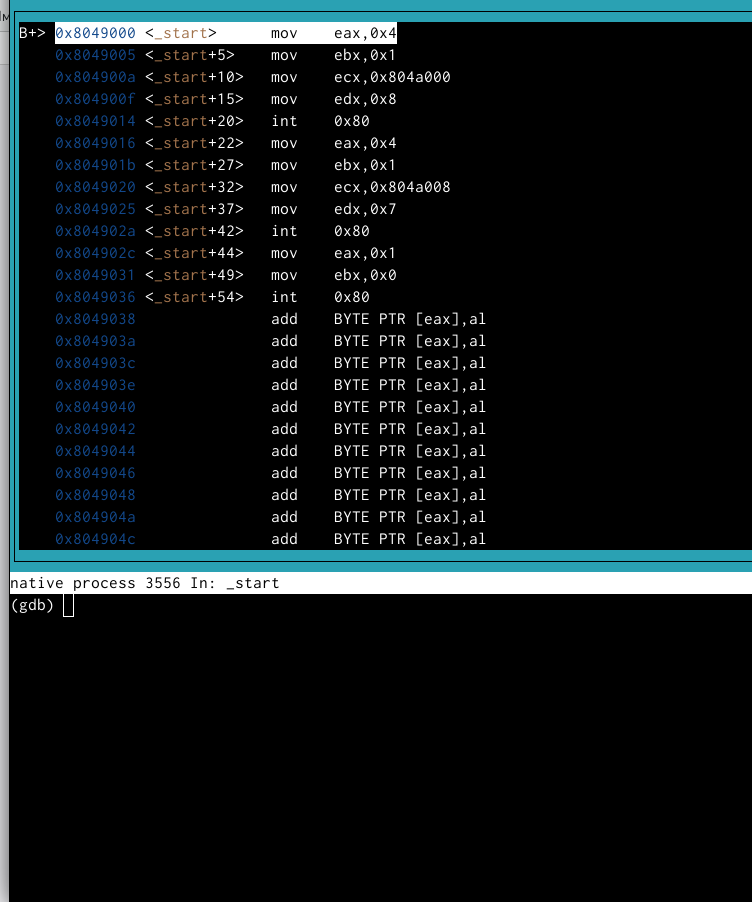


Рис. 13: 1 этап

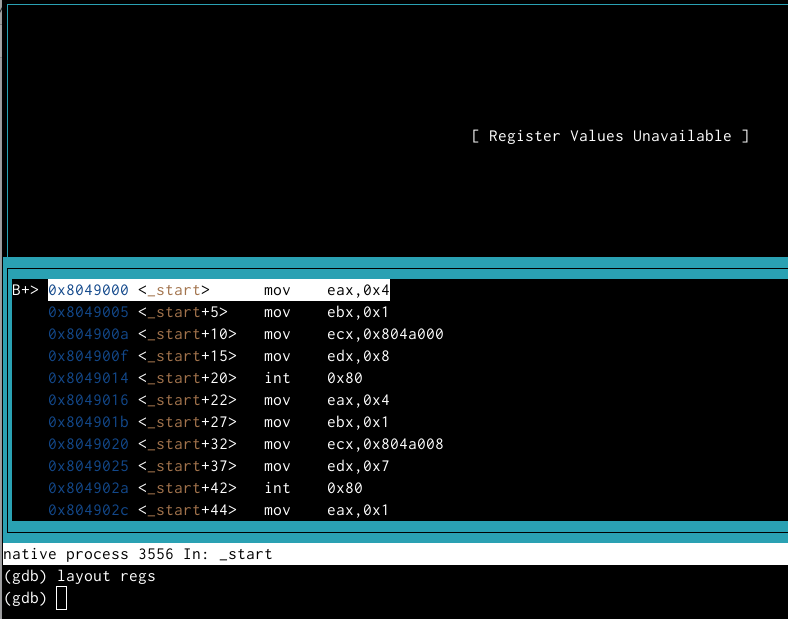


Рис. 14: 2 этап

1. На предыдущих шагах была установлена точка останова по имени метки (\_start). Проверила это с помощью команды info breakpoints (рис. 15 )

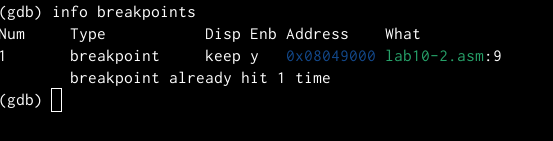


Рис. 15: Проверка

1. Определила адрес предпоследней инструкции (mov ebx,0x0) и установила точку останова.(рис. 16 )

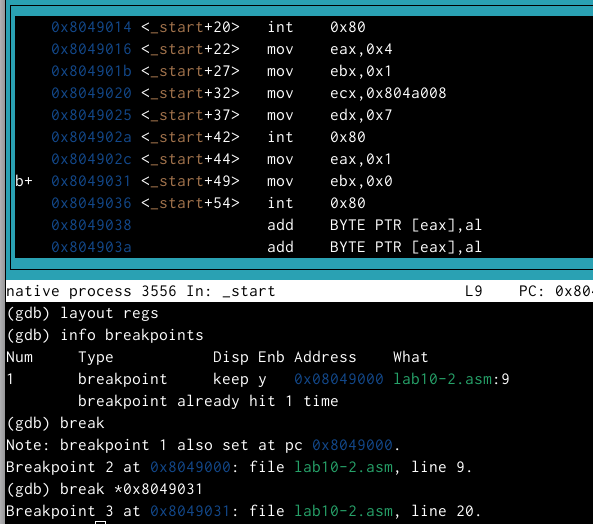


Рис. 16: Выполнение

1. Посмотрела информацию о всех установленных точках останова (рис. 17 )

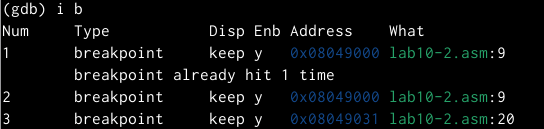


Рис. 17: Информация

1. Посмотрела содержимое регистров также можно с помощью команды info registers (или i r) (рис. 18 )

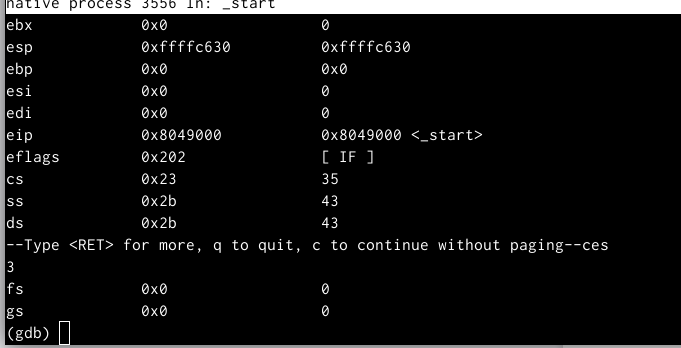


Рис. 18: Содержимое регистров

1. Посмотрела значение переменной msg1 по имени (рис. 19 )

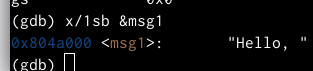


Рис. 19: Значение переменной

1. Изменила первый символ переменной msg1 (рис. 20 )

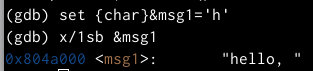


Рис. 20: Замена ‘H’ на ‘h’

1. Заменила символ во второй переменной msg2 (рис. 21 )

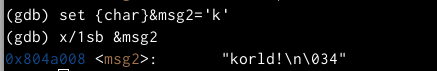


Рис. 21: Замена ‘W’ на ‘k’

1. Вывела в различных форматах (в шестнадцатеричном формате, в двоичном формате и в символьном виде) значение регистра edx (рис. 22 )

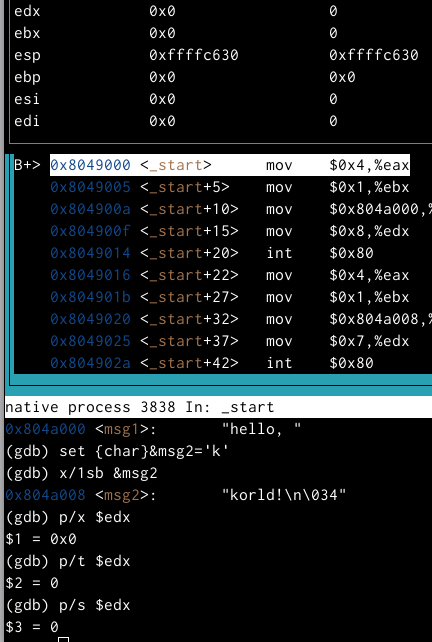


Рис. 22: Значения регистра edx

1. С помощью команды set изменила значение регистра ebx (рис. 23, рис.24 )

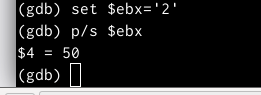


Рис. 23: 1 вариант

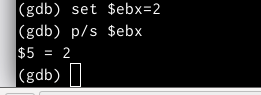


Рис. 24: 2 вариант

1. Скопировала файл lab9-2.asm, созданный при выполнении лабораторной работы No9, с программой выводящей на экран аргументы командной строки (Листинг 9.2) в файл с именем lab10-3.asm (рис. 25 )

Рис. 25: Копирование файла

Рис. 25: Копирование файла

1. Создала исполняемый файл. Для загрузки в gdb программы с аргументами необходимо использовать ключ –args. Загрузила исполняемый файл в отладчик, указав аргументы (рис. 26 )

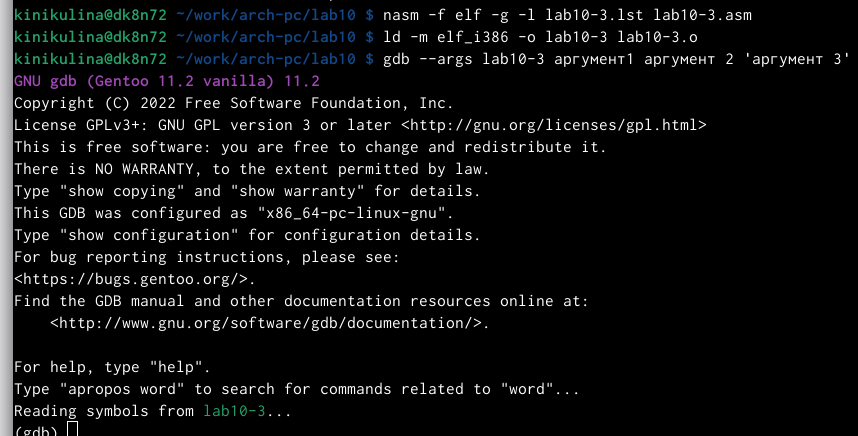


Рис. 26: Значения регистра edx

1. Для начала установила точку останова перед первой инструкцией в программе и запустила ее.(рис. 27 )

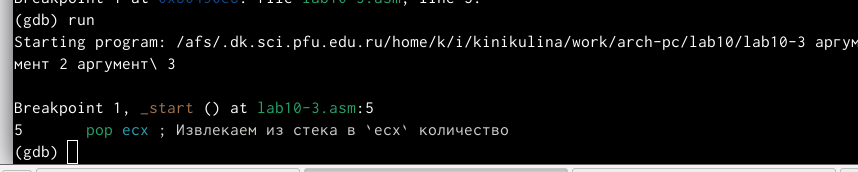


Рис. 27: Запуск

1. Адрес вершины стека храниться в регистре esp и по этому адресу располагается число равное количеству аргументов командной строки (включая имя программы) (рис. 28 )

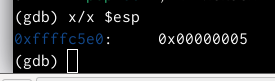


Рис. 28: Количество аргуметов

1. Посмотрела остальные позиции стека (рис. 29 )

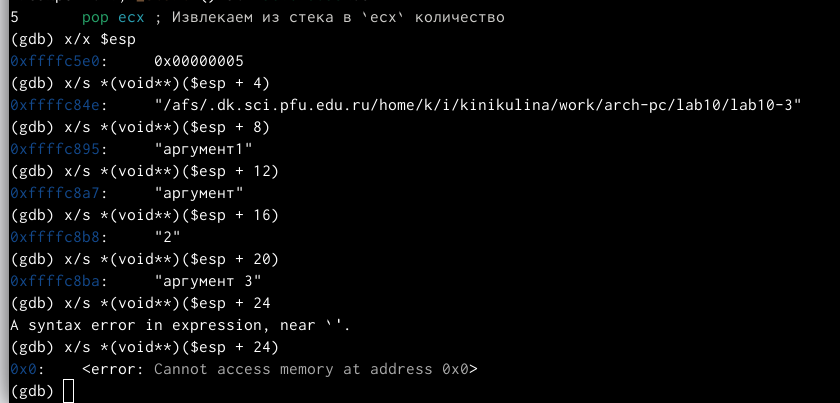


Рис. 29: Остальные позиции

# 5 Самостоятельная работа

1. Преобраpазовала программу из лабораторной работы №9 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции f(x) как подпрограмму. Проверила работу (рис. 30, рис.31 )

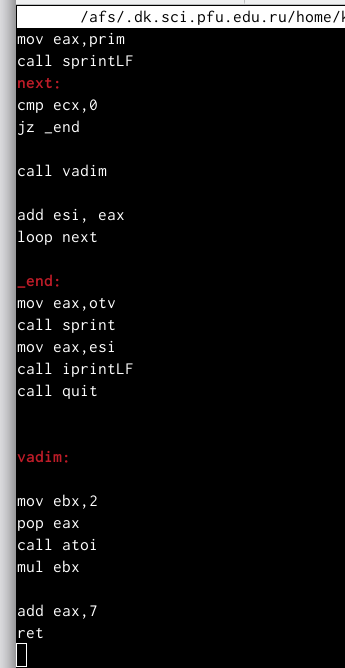


Рис. 30: Текст

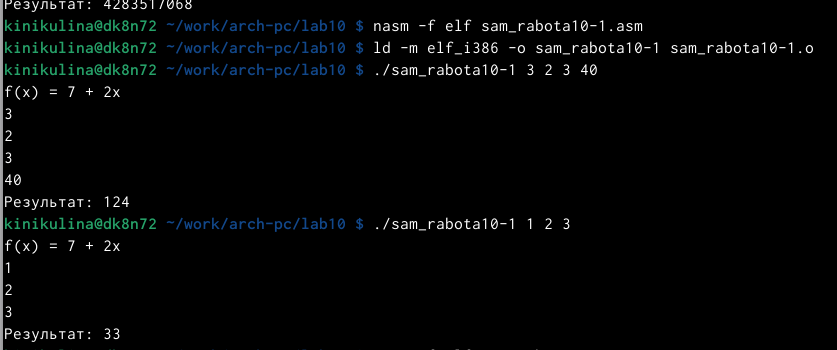


Рис. 31: Проверка работы

1. С помощью отладчика GDB, анализируя изменения значений регистров,определила ошибку и исправила ее (рис. 32, рис.33 )

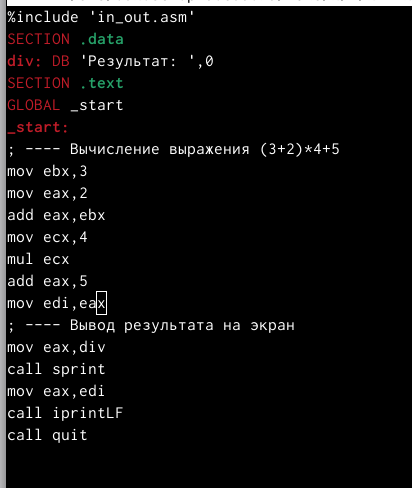


Рис. 32: Текст без ошибок

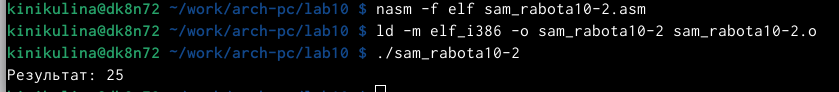


Рис. 33: Проверка работы

# 6 Выводы

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