## 0.1 Front matter

title: “Лабараторная работа № 10” subtitle: Дисциплина: Архитектура компьютера” author: “Никулина Ксения Ильинична”

## 0.2 Generic otions

lang: ru-RU toc-title: “Содержание”

## 0.3 Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## 0.4 Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs: english ## Fonts mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9 ## Biblatex biblatex: true biblio-style: “gost-numeric” biblatexoptions: - parentracker=true - backend=biber - hyperref=auto - language=auto - autolang=other\* - citestyle=gost-numeric ## Pandoc-crossref LaTeX customization figureTitle: “Рис.” listingTitle: “Листинг” lofTitle: “Список иллюстраций” lotTitle: “Список таблиц” lolTitle: “Листинги” ## Misc options indent: true header-includes: -

# keep figures where there are in the text

## # keep figures where there are in the text

# 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)

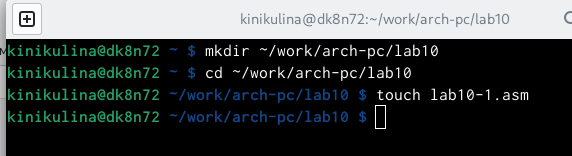


Figure 1: Содание файла

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

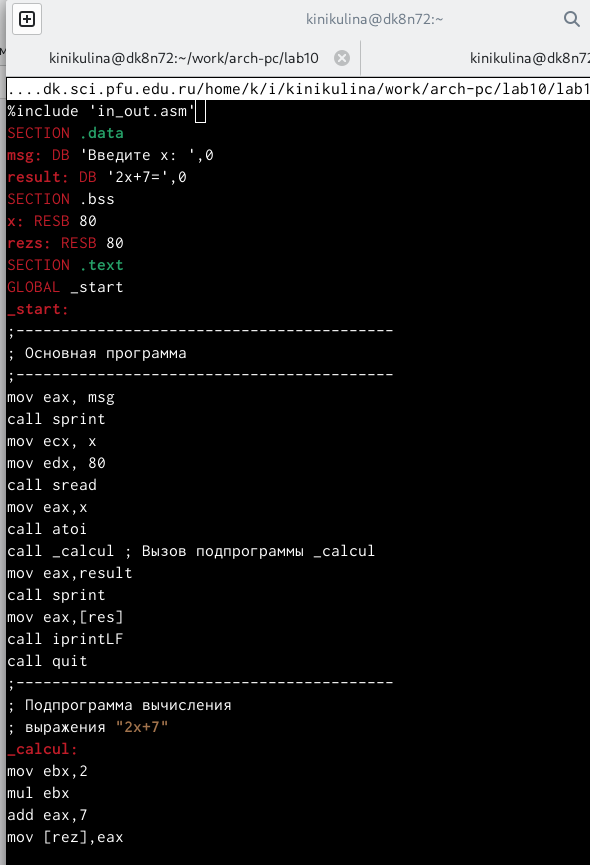


Figure 2: Текст

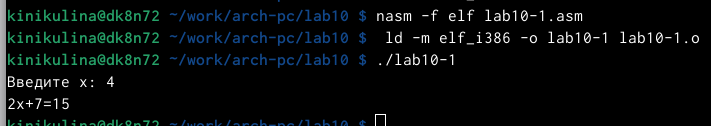


Figure 3: Проверка работы файла

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 )

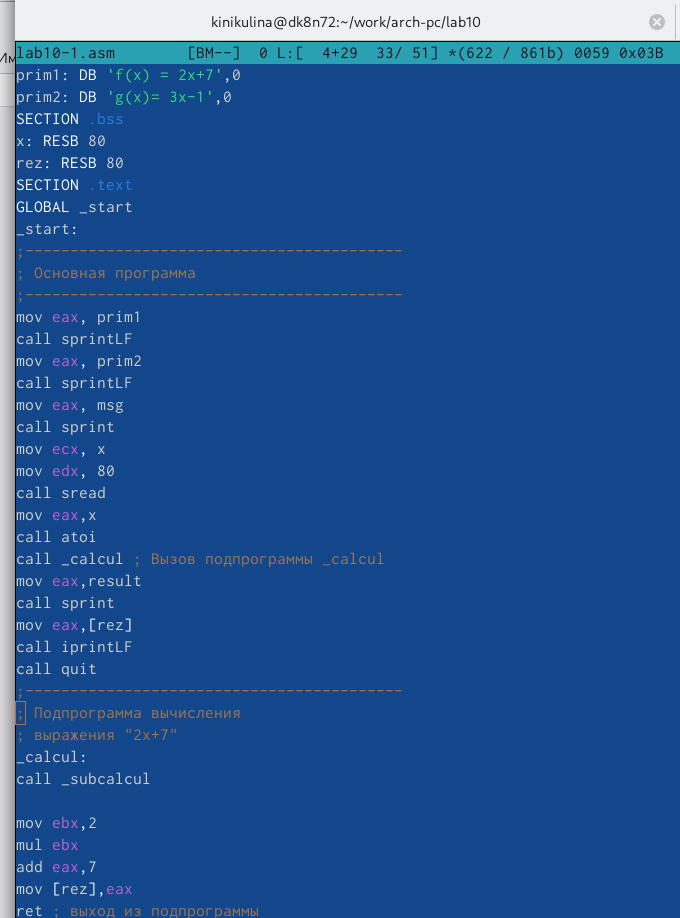


Figure 4: Текст

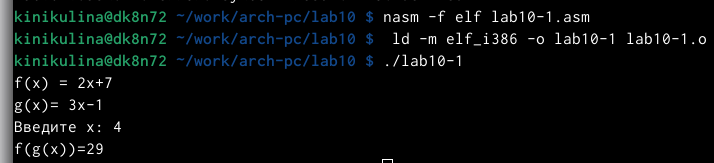


Figure 5: Проверка работы файла

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

Figure 6: Создание файла

Figure 6: Создание файла

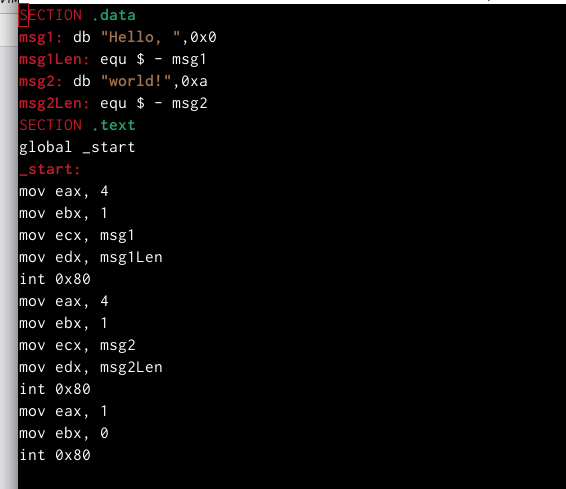


Figure 7: Текст

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

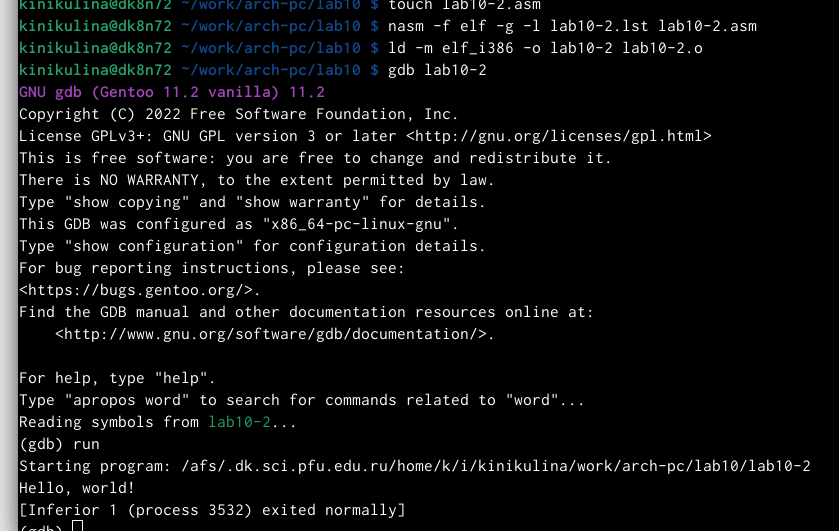


Figure 8: Загрузка файла

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

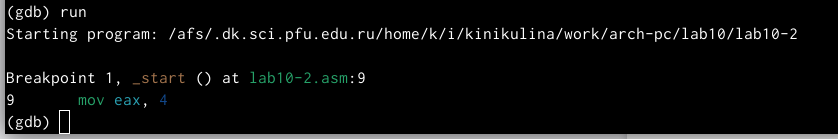


Figure 9: Проверка работы

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

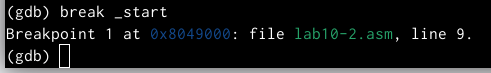


Figure 10: Запуск

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

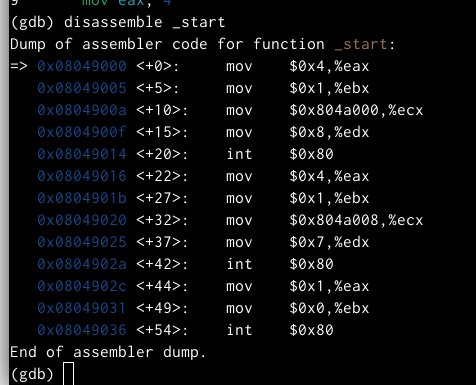


Figure 11: Код программы

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

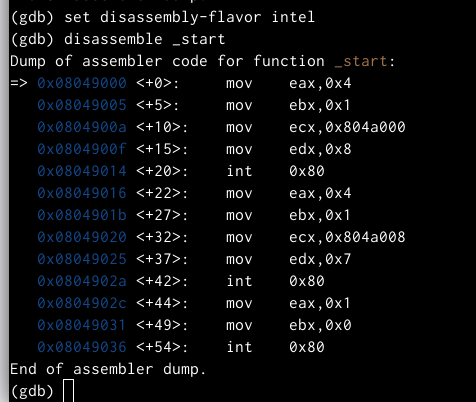


Figure 12: Команды

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

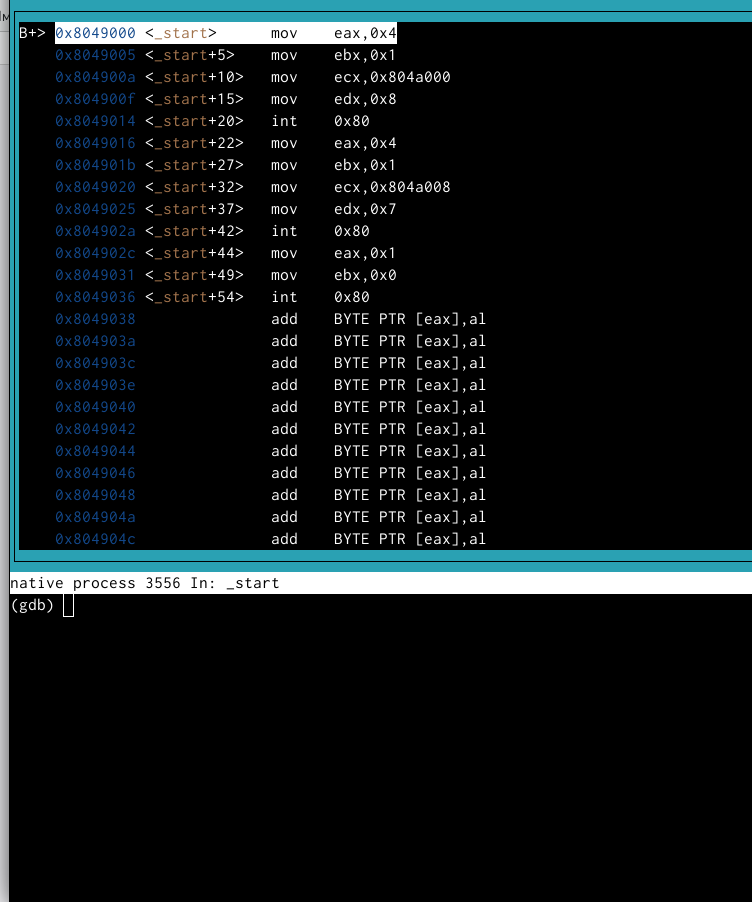


Figure 13: 1 этап

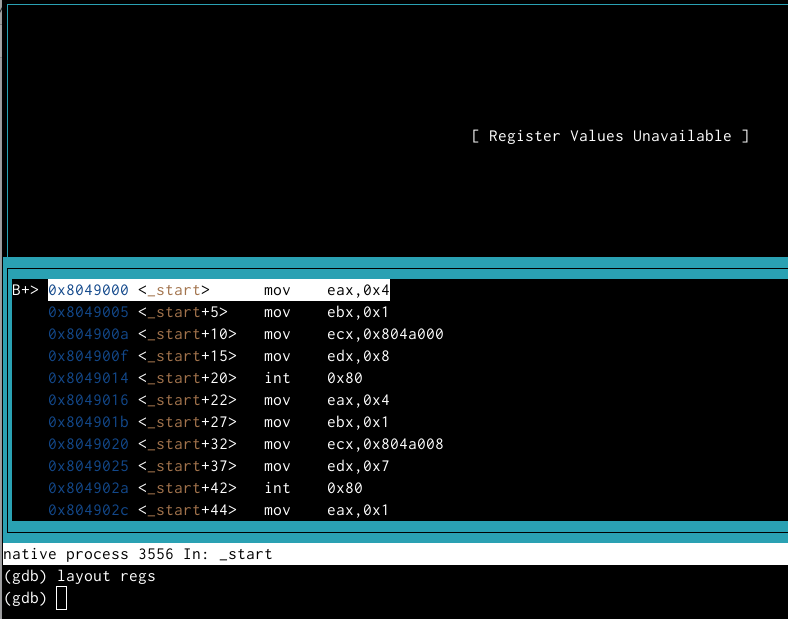


Figure 14: 2 этап

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

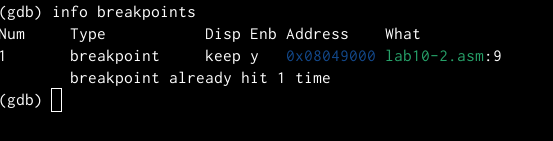


Figure 15: Проверка

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

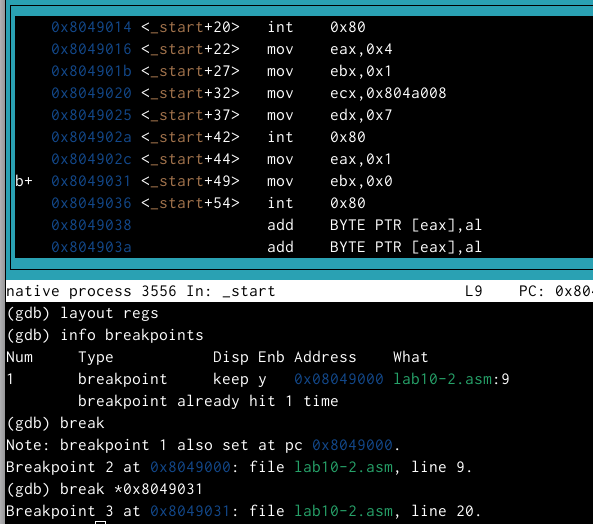


Figure 16: Выполнение

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

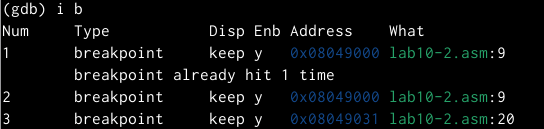


Figure 17: Информация

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

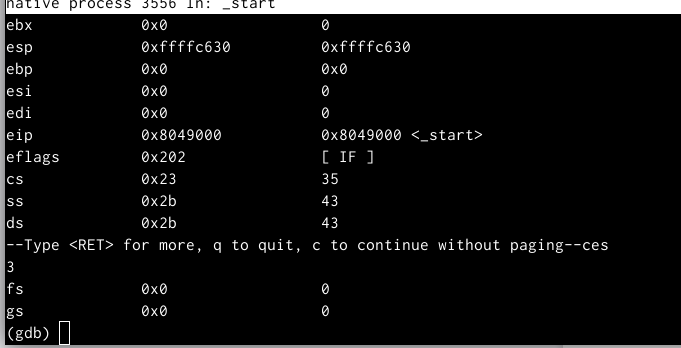


Figure 18: Содержимое регистров

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

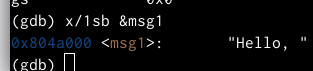


Figure 19: Значение переменной

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

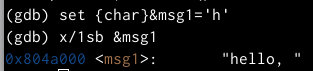


Figure 20: Замена ‘H’ на ‘h’

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

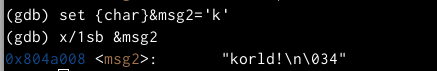


Figure 21: Замена ‘W’ на ‘k’

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

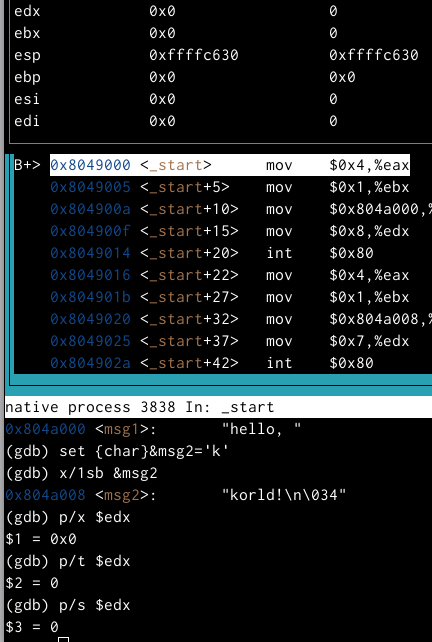


Figure 22: Значения регистра edx

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

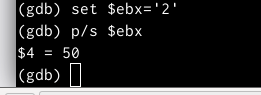


Figure 23: 1 вариант

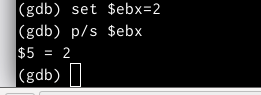


Figure 24: 2 вариант

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 )

Figure 25: Копирование файла

Figure 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 )

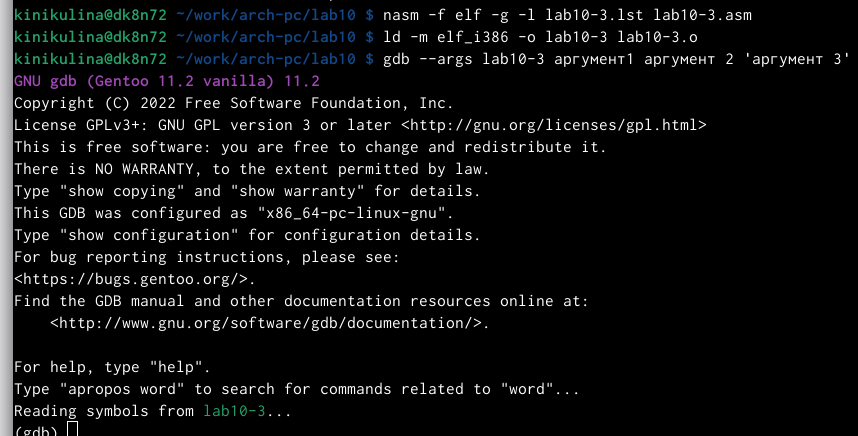


Figure 26: Значения регистра edx

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

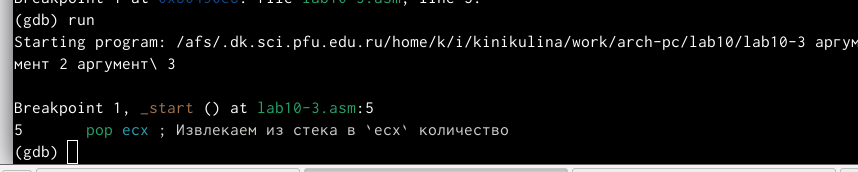


Figure 27: Запуск

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

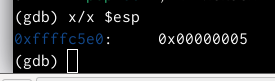


Figure 28: Количество аргуметов

1. Посмотрела остальные позиции стека (рис. 29 )

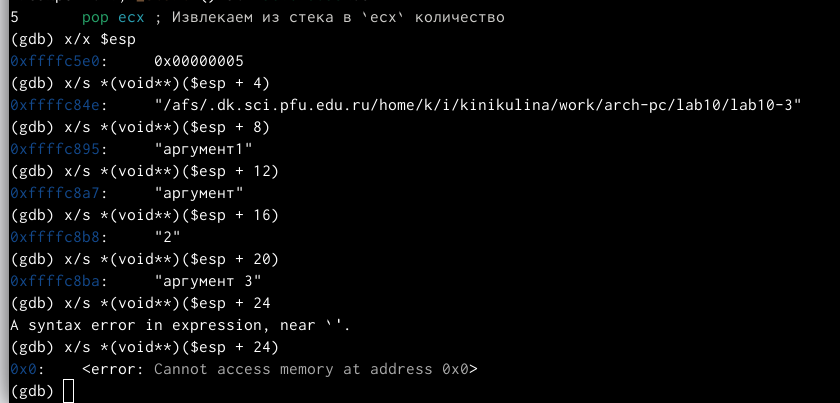


Figure 29: Остальные позиции

# 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 )

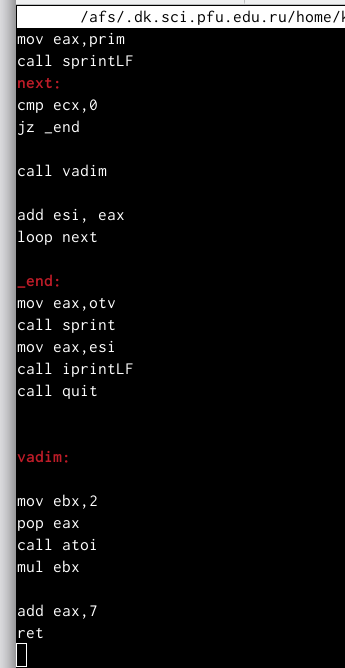


Figure 30: Текст

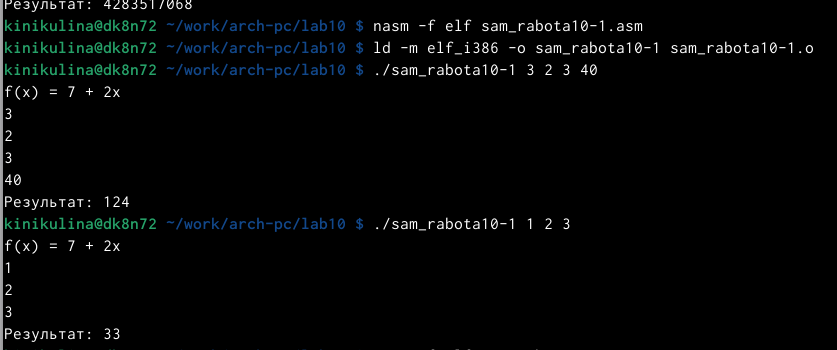


Figure 31: Проверка работы

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

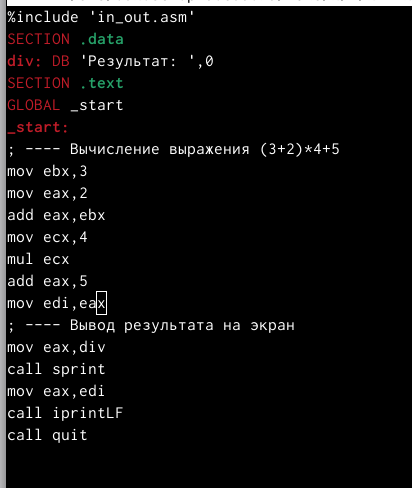


Figure 32: Текст без ошибок

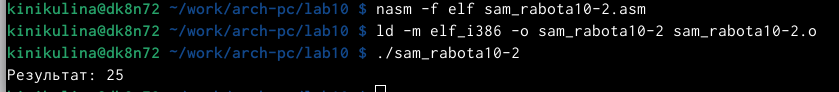


Figure 33: Проверка работы

# 6 Выводы

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