Отчёт по лабораторной работе №9

Дисциплина: Архитектура компьютера

Жернаков Данила Иванович

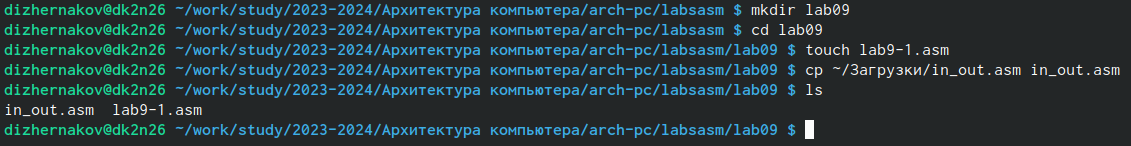
Содержание

# 1 Цель работы

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

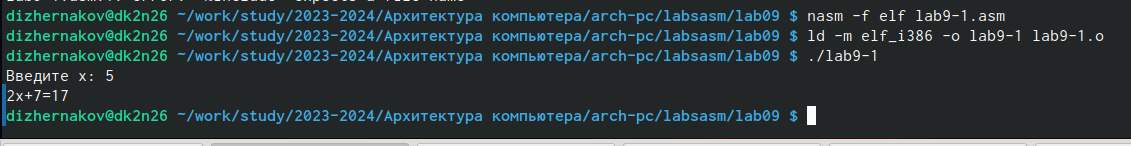
# 2 Выполнение лабораторной работы

Создал каталог для выполнения лабораторной работы №9, перешел в него и создал файл (рис. ??).



Создание каталога, файла

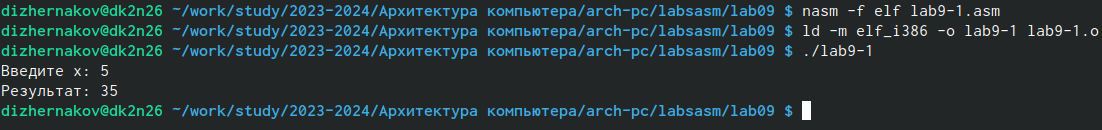
Создал исполняемый файл и проверил его работу (рис. ??).



Работа программы 1

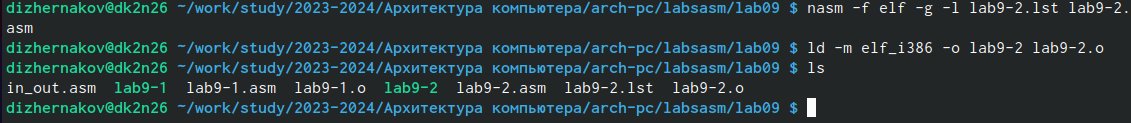
Изменил текст программы, добавив подпрограмму \_subcalcul в подпрограмму \_calcul, для вычисления выражения f(g(x)), где x вводится с клавиатуры, f(x)=2x+7, g(x)=3x-1.

Создал исполняемый файл и проверил его работу (рис. ??).



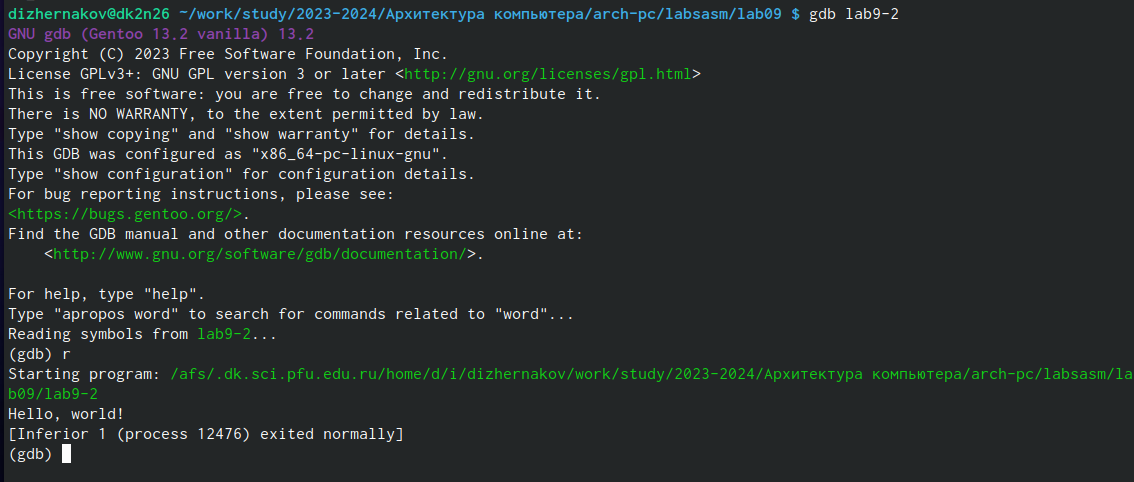
Работа программы

Создал файл lab09-2.asm с текстом программы из листинга 9.2. Получил исполняемый файл (рис. ??).



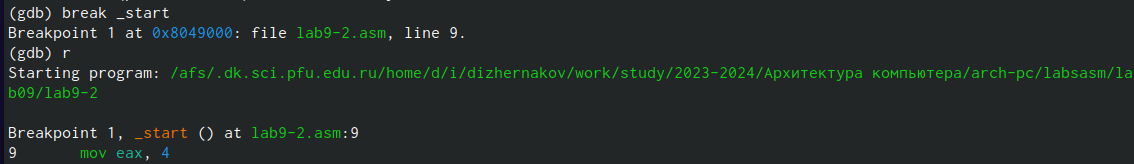
Исполняемый файл

Загрузил исполняемый файл в отладчик gdb (рис. ??).



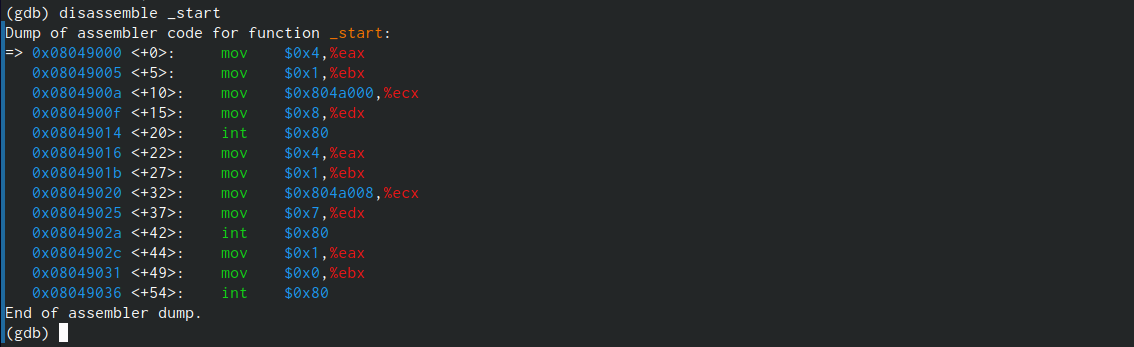
Загрузка файла в отладчик gdb

Для более подробного анализа программы установил брейкпоинт на метку \_start, с которой начинается выполнение любой ассемблерной программы, и запустил её (рис. ??).



Установка брейкпоинта и запуск программы

Посмотрел дисассимилированный код программы с помощью команды disassemble, начиная с метки \_start (рис. ??).



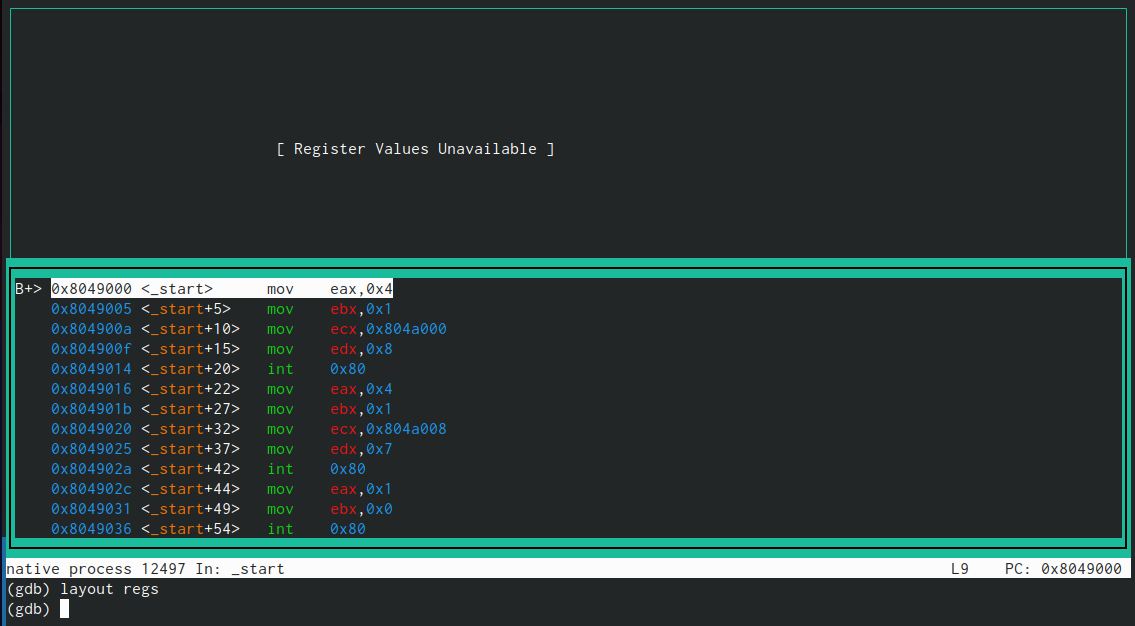
Дисассимилированный код программы

Переключение на отображение команд с Intelовским синтаксисом (рис. ??).



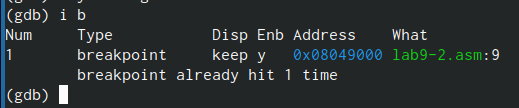
Переключение на отображение команд

Включил режим псевдографики для более удобного анализа программы: layout asm и layout regs. (рис. ??).



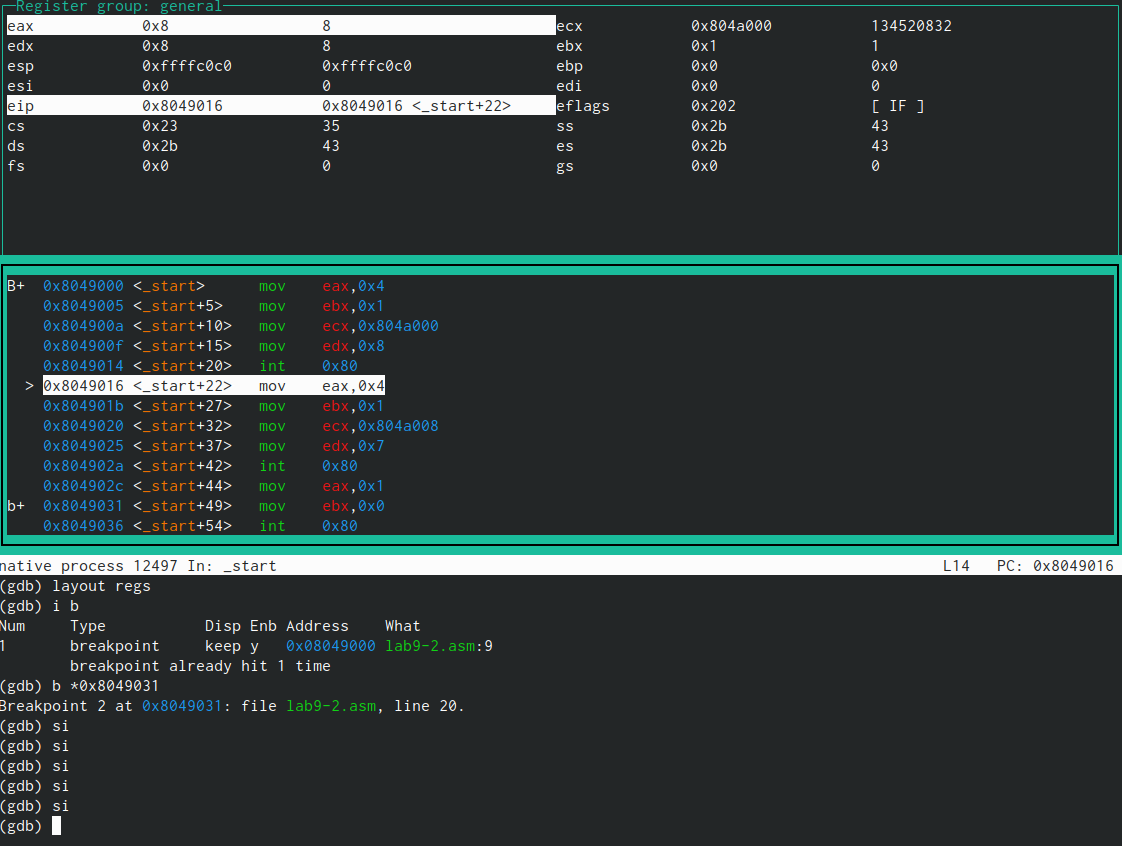
Режим псевдографики

Проверил точку останова по имени метки (рис. ??).



Точка останова

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



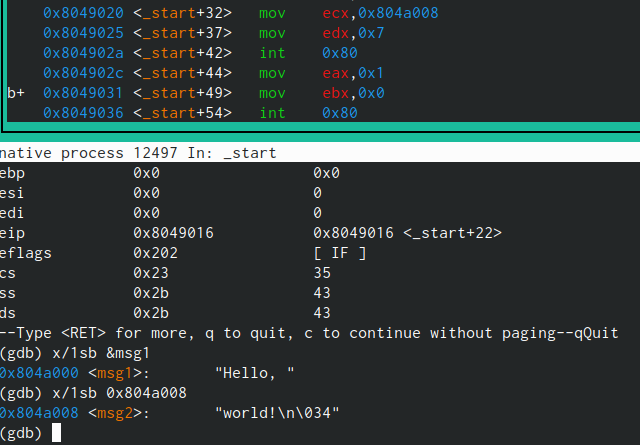
Адрес предпоследней инструкции

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

Значение переменной 1

Значение переменной 1

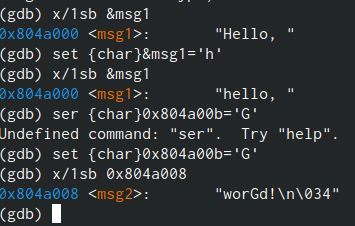
Посмотрел значение переменной msg2 по адресу (рис. ??).



Значение переменной 2

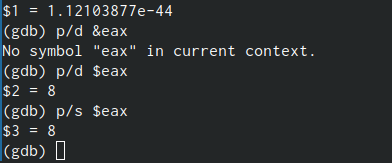
Изменил первый символ переменной msg1.

Заменил некоторый символ во второй переменной msg2 (рис. ??).



Изменение первого символа msg2

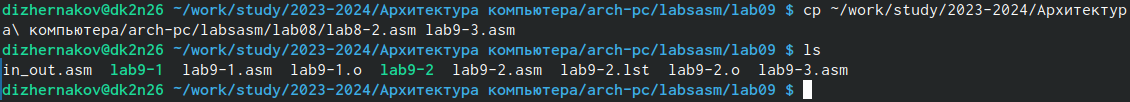
Вывел в различных форматах (в шестнадцатеричном формате, в двоичном формате и в символьном виде) значение регистра eаx (рис. ??).



Вывод значений регистра edx

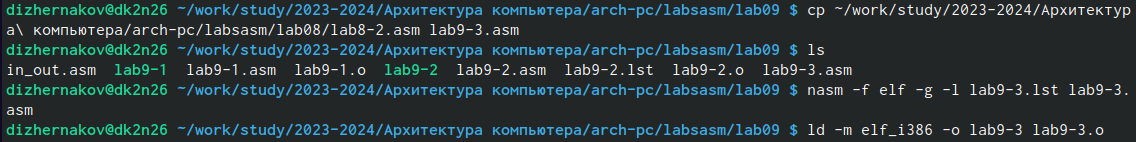
Завершил выполнение программы с помощью команды continue (сокращенно c) или stepi (сокращенно si) и вышла из GDB с помощью команды quit (сокращенно q).

Скопировал файл lab8-2.asm, созданный при выполнении лабораторной работы №8, с программой выводящей на экран аргументы командной строки (Листинг 8.2) в файл с именем lab09-3.asm (рис. ??).



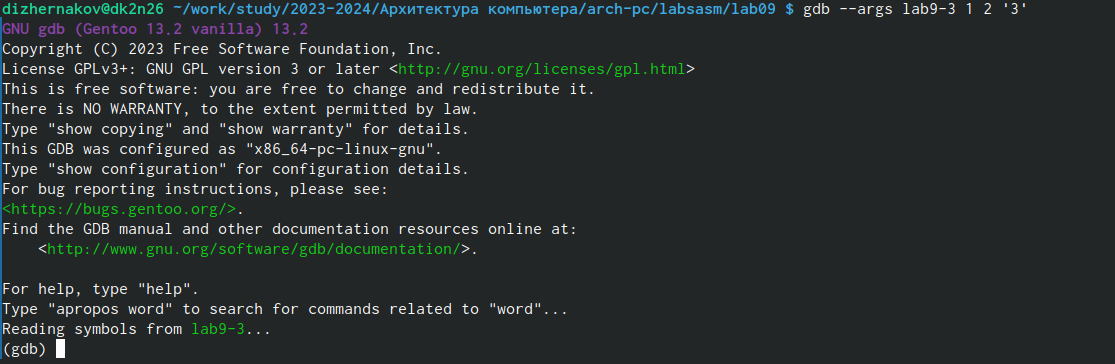
Копирование

Создал исполняемый файл (рис. ??).



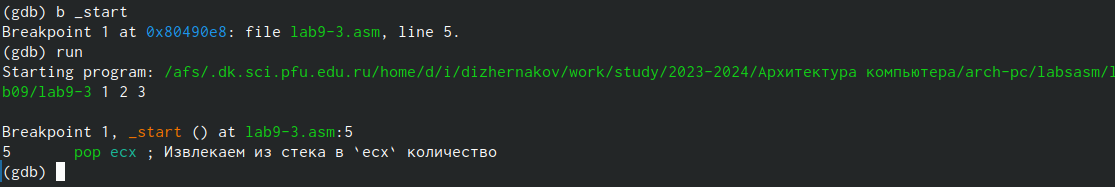
Создание исполняемого файла

Загрузил исполняемый файл в отладчик, указав аргументы (рис. ??).



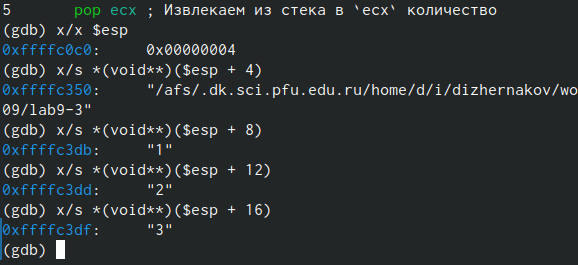
Загрузка в отладчик

Для начала установлю точку останова перед первой инструкцией в программе и запущу ее (рис. ??).



Установка точки останова

Адрес вершины стека храниться в регистре esp, и по этому адресу располагается число, равное количеству аргументов командной строки (включая имя программы) Посмотрел остальные позиции стека (рис. ??).

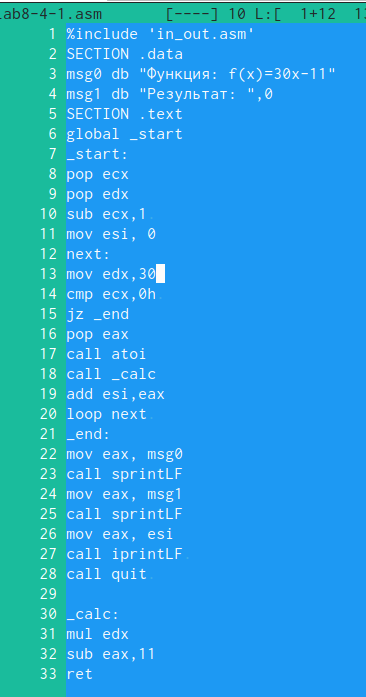


Позиции стека

Шаг изменения адреса равен 4, потому что число аргументов равно 4.

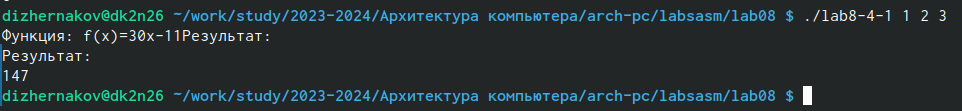
# 3 Самостоятельная работа

Создал файл для самостоятельной работы Преобразовал программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции 𝑓(𝑥) как подпрограмму (рис. ??).

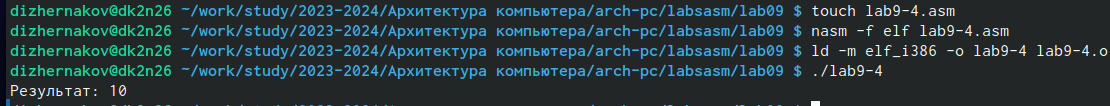


Преобразованная программа

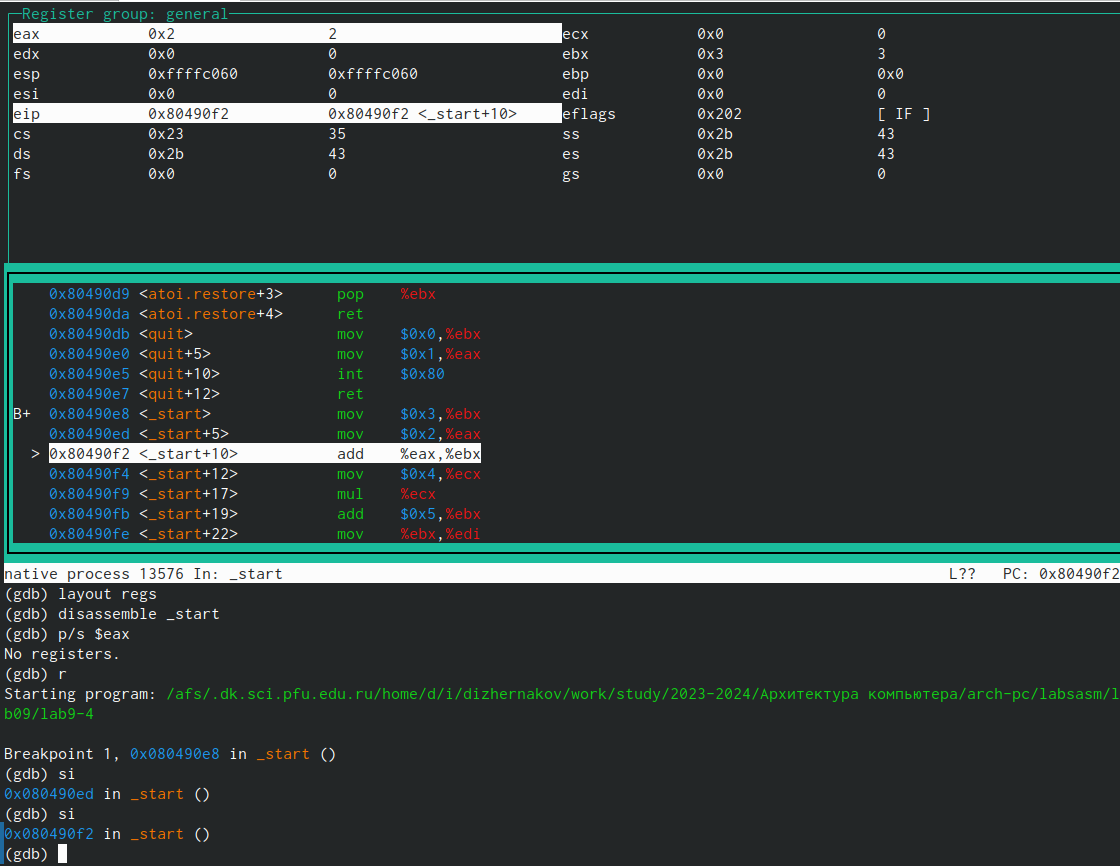
Получил верный ответ (рис. ??).



Результат

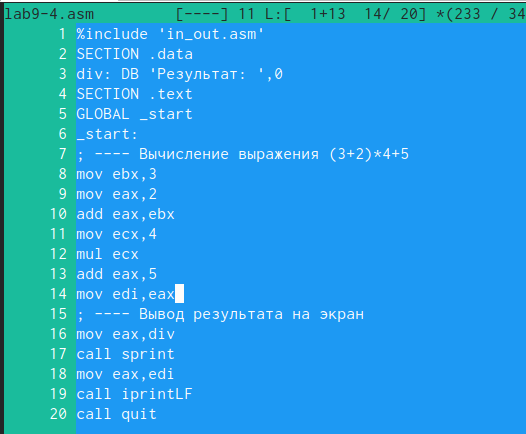
Создал файл для второго задания самостоятельной работы. Ввел программу из листинга 9.3. Попробовал запустить программу. 

Просматрел дисассимилированный код программы, поставил точку останова перед инструкцией \_start и открыл значения регистров на этапе первого сложения (рис. ??).



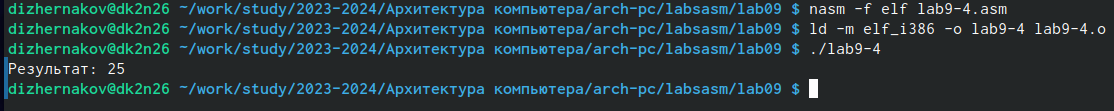
Дисассимилированный код, значения регистров

Регистр ecx со значением 4 умножается не на ebx,сложенным c eax, а только с eax со значением 2. Меняю (рис. ??).



Измененная программа

Получаю верный ответ (рис. ??).



Результат

# 4 Выводы

В ходе работы я приобрел навыки написания программ с использованием подпрограмм, познакомился с методами отладки при помощи GDB и его основными возможностями.

# Список литературы