Отчет по лаборатной работе №9

Дисциплина: архитектура компьютера

Желобицкая П.А.

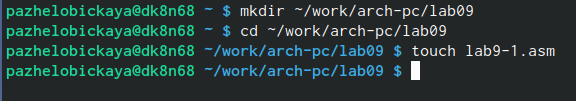
Содержание

# 1 Цель работы

Приобрести навыки написания программ с использованием подпрограмм. Познакомиться с методами откладки при помощи GDB и его основными возможностями.

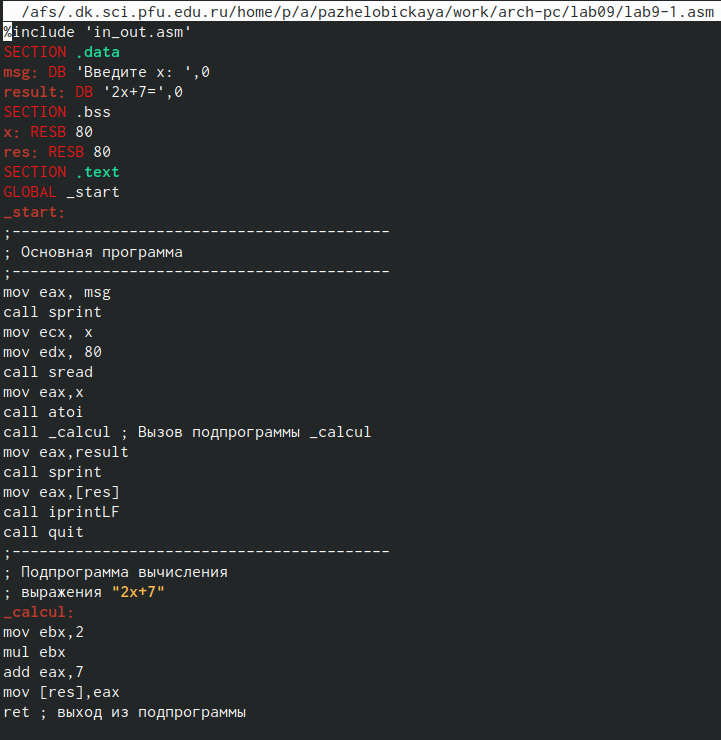
# 2 Выполнение лабораторной работы

Создаю каталог для выполнения лабораторной работы №9, перехожу в него и создаю файл lab9-1.asm (рис. ??).



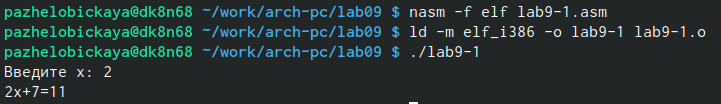
Создание каталога и файла

Ввожу в файл lab9-1.asm текст программы из листинга 9.1 (рис. ??).



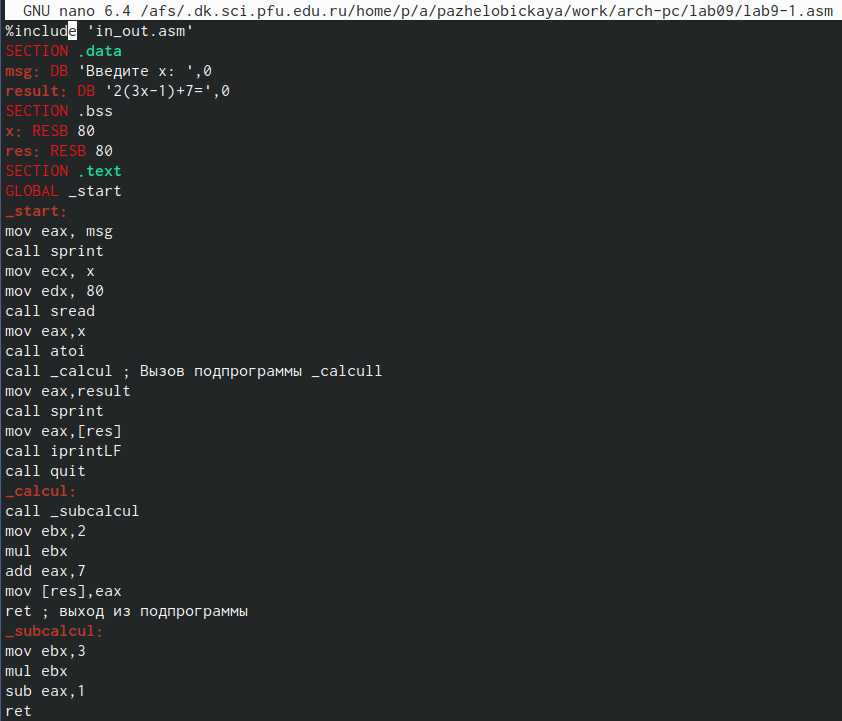
Ввод текста программы

Создаю исполняемый файл, запускаю его и проверяю работу (рис. ??).



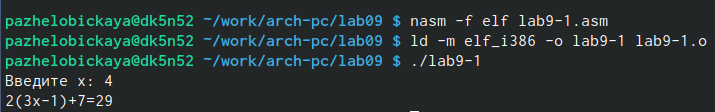
Запуск и проверка файла

Вношу изменения в текст программы (рис. ??).



Изменения текста программы

Создаю исполняемый файл, запускаю его и проверяю работу (рис. ??).



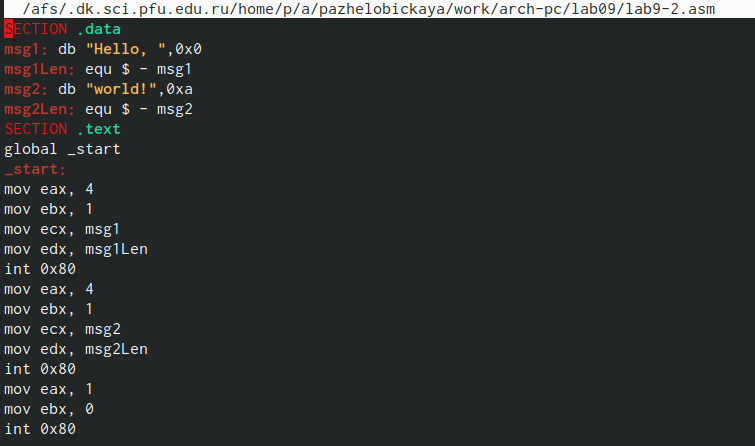
Запуск и проверка файла

Создаю файл lab9-2.asm (рис. ??).

Создание файла

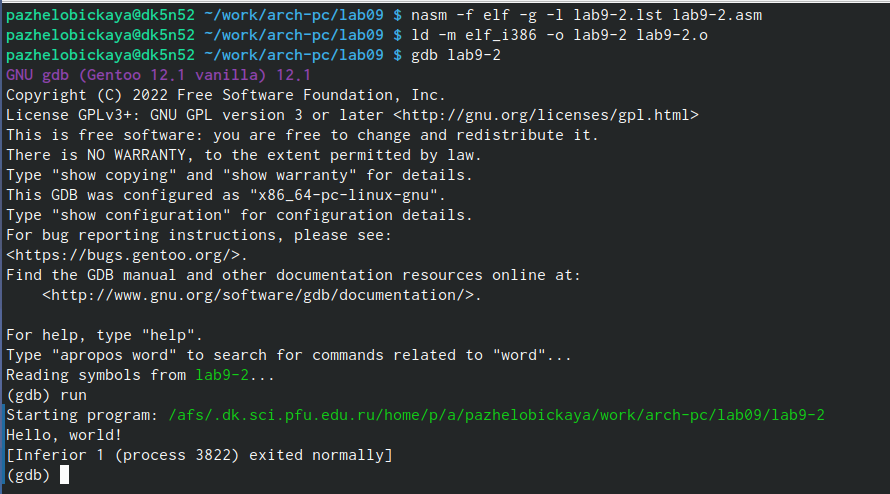
Создание файла

Ввожу в файл текст программы из листинга 9.2 (рис. ??).



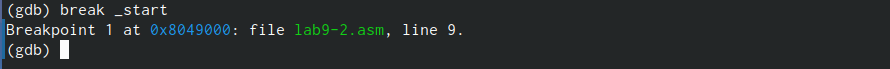
Ввод программы

Создаю исполняемый файл, загружаю его в откладчик gdb. Проверяю работу программы, запуская ее в оболочке GDB с помощью команды run (рис. ??).



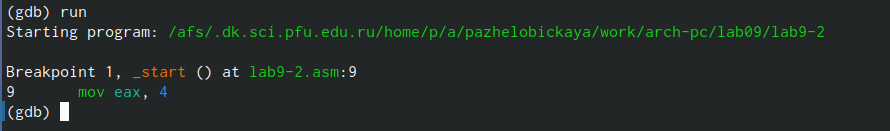
Загрузка в откладчик gdb и запуск программы в оболочке GDB

Устанавливаю брейкпоинт на метку \_start (рис. ??).



Установка брейкпоинта

Запускаю эту программу (рис. ??).



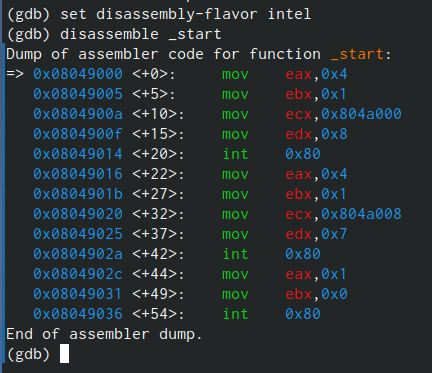
Запуск программы

С помощью команды disassemble смотрю дисассимилированный код программы (рис. ??).



Просмотр дисассимилированного кода программы

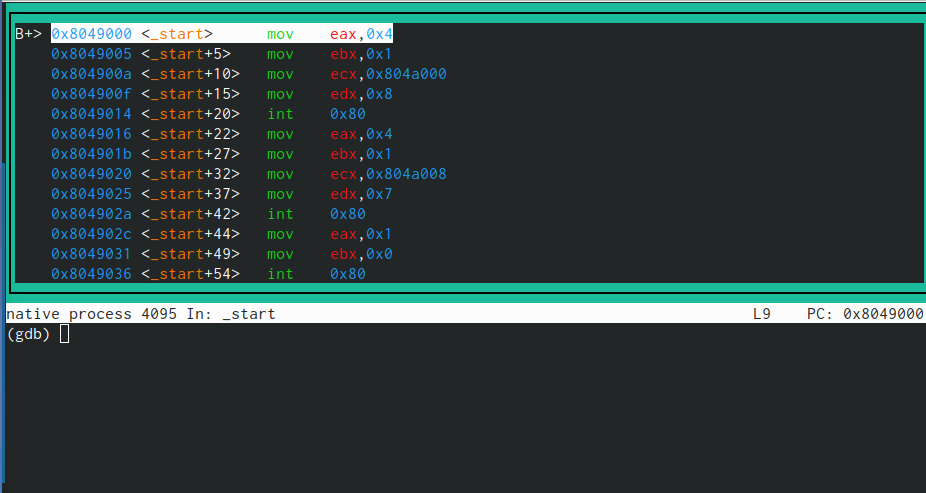
Переключаюсь на отображение команд с Intel’овским синтаксисом, с помощью команды disassemble (рис. ??).



Переключение на отображение команд с Intel’овским синтаксисом

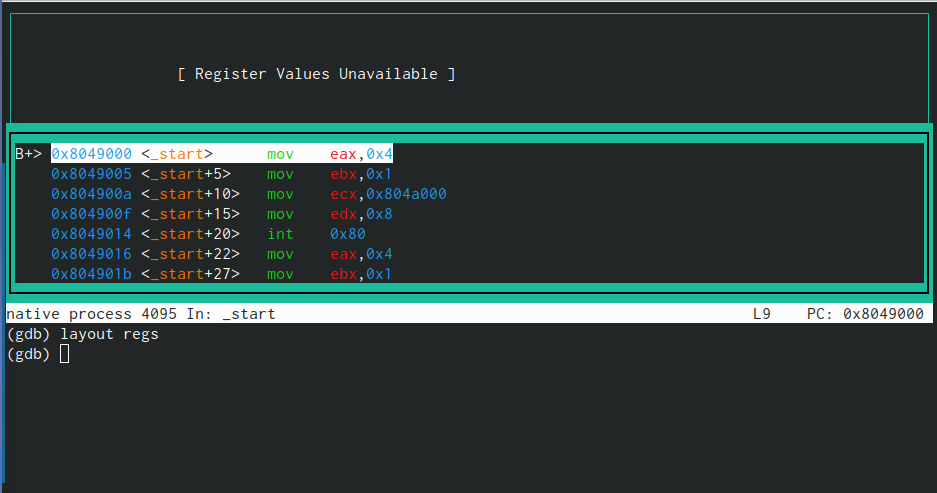
Основное различие заключается в том, что в режиме Intel пишется сначала сама команда, а потом уже ее машинный код, когда в режиме ATT идет сначала машинный код и уже потом сама команда.

Включаю режим псевдографики (рис. ??).



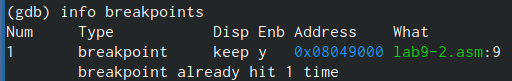
Режим псевдографики

Команда layout regs (рис. ??).



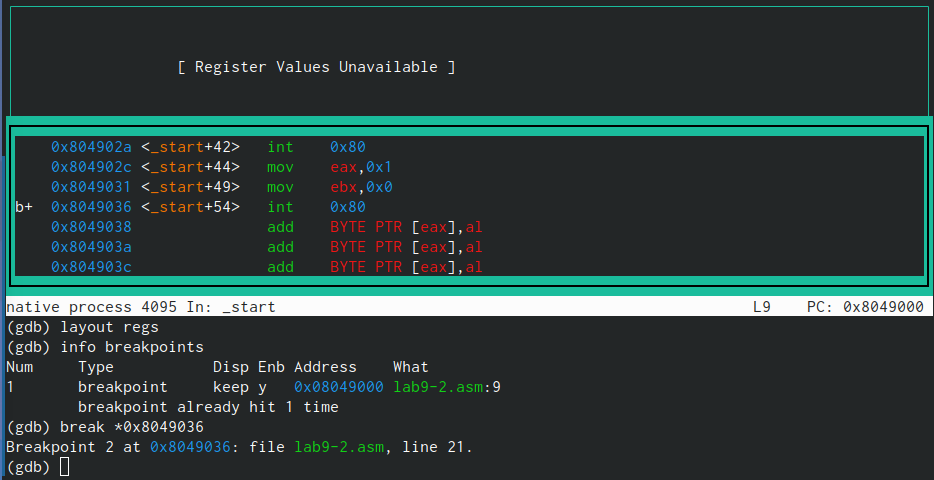
Режим псевдографики

Проверяю точку останова с помощью команды info breakpoints (рис. ??).



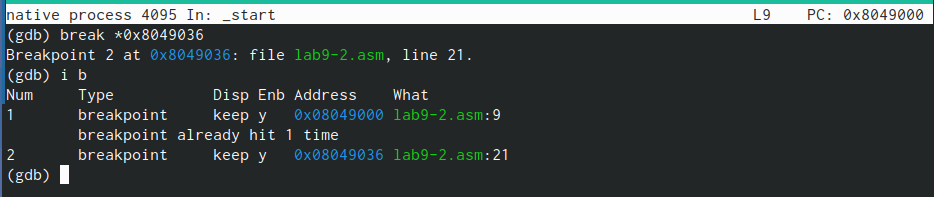
Проверка точки останова

Определяю адрес последней инструкции (mov ebx,0x0) и устанавливаю точку останова (рис. ??).



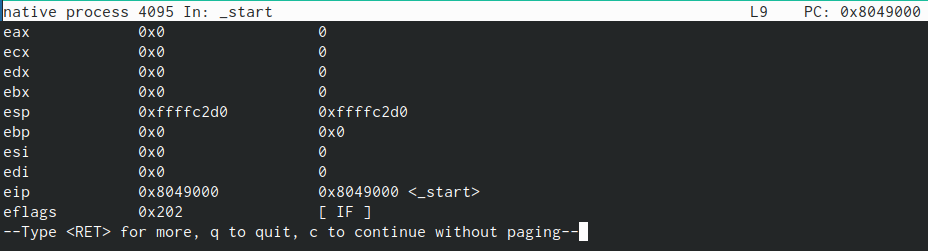
Адрес предпоследней инструкции

Просматриваю информацию о всех установленных точках останова (рис. ??).



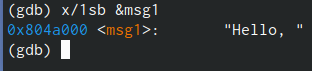
Информация о всех точках останова

Просматриваю содержимое регистров с помощью команды info registers (рис. ??).



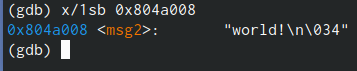
Содержимое регистров

Просматриваю значение переменой msg1 по имени (рис. ??).



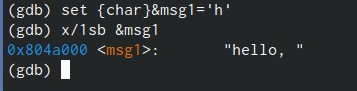
Значение переменной msg1 по имени

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



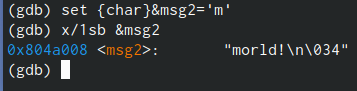
Значение переменной msg2 по адресу

Изменяю первый символ переменной msg1 (рис. ??).



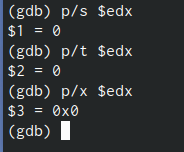
Изменение первого символа переменной msg1

Заменяю первый символ во второй переменной msg2 (рис. ??).



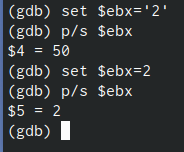
Замена второго символа в переменной msg2

Вывожу в различных форматах (в шестнадцатеричном, в двоичном форматах и в символьном виде) значение регистра ebx (рис. ??).



Выведение значений регистра ebx

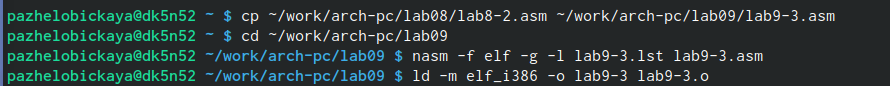
Изменяю значение регистра ebx с помощью команды set (рис. ??).



Изменение значений регистра ebx

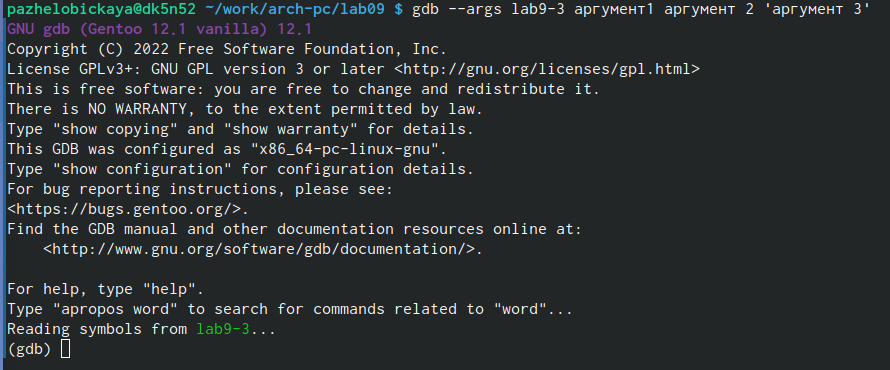
Завершаю выполнение программы с поощью команды continue или stepi и выхожу из GDB с помощью команды quit.

Копирую файл lab8-2.asm в файл lab9-3.asm и создаю исполняемый файл (рис. ??).



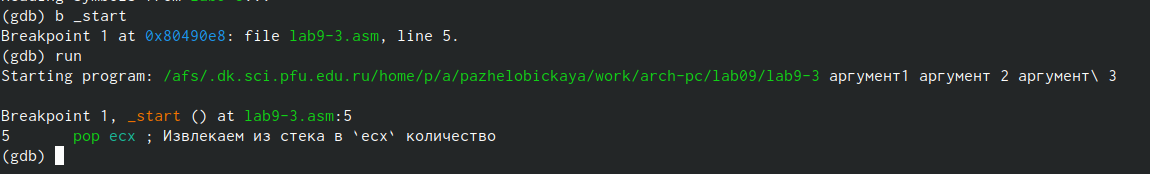
Копирование и создание исполняемого файла

Загружаю исполняемый файл в откладчик, указывая аргументы (рис. ??).



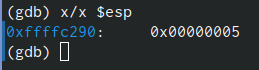
Загрузка исполняемого файла в откладчик

Устанавливаю точку останова перед первой инструкцией в программе и запускаю ее (рис. ??).



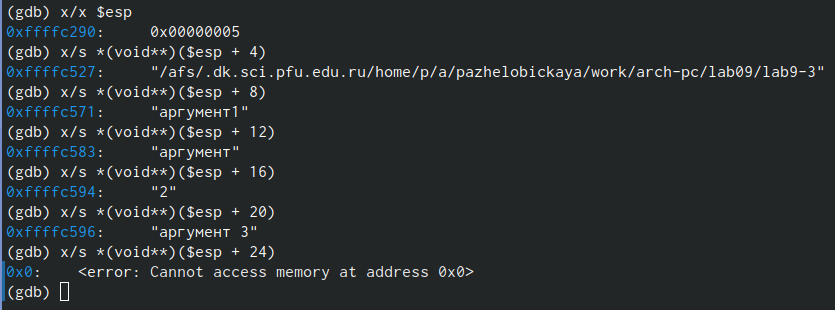
Запуск программы

Адрес вершины стека храниться в регистре esp и по этому адресу располагается число равное количеству аргументов командной строки (рис. ??).



Адрес вершины стека

Просматриваю остальные позиции стека (рис. ??).

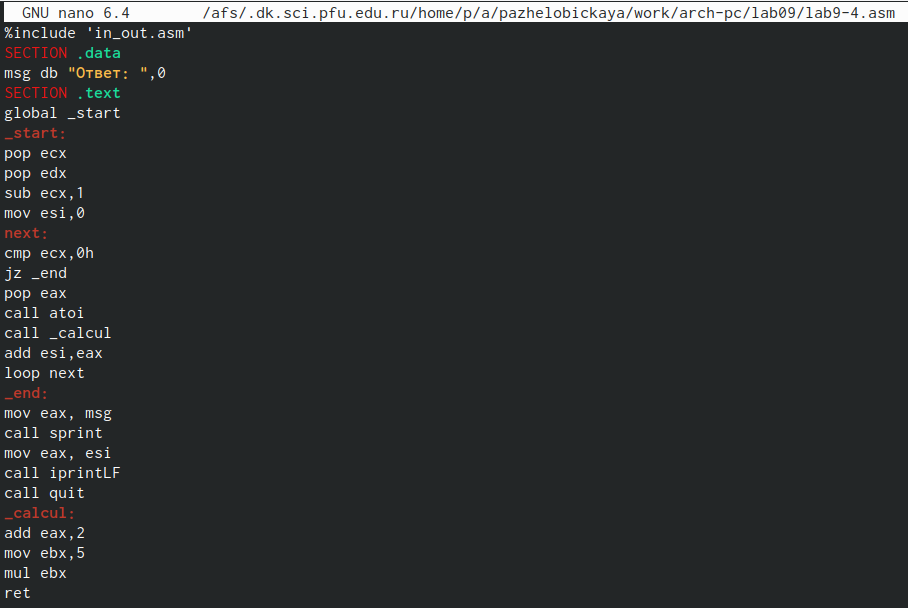


Остальные позиции стека

Количество аргументов командной строки 4, значит и шаг равен 4.

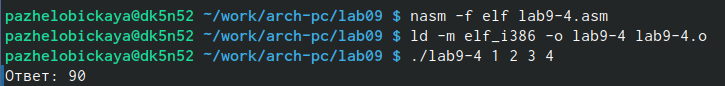
# 3 Выполнение заданий самостоятельной работы

Преобразую программу из лабораторной работы №8, добавляю подпрограмму, которая вычисляет знаения функции f(x) (рис. ??).



Редактирование файла

Создаю исполняемый файл и ввожу аргументы. Получаю верный ответ (рис. ??).



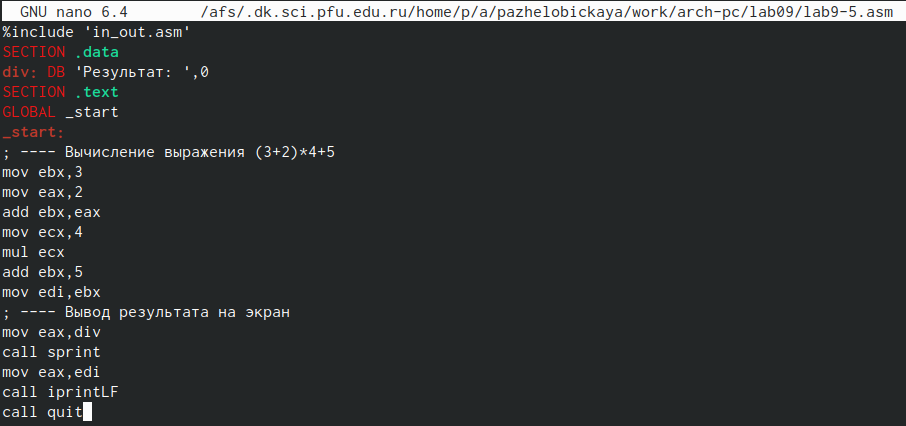
Создание исполняемого файла и проверка его

Создаю файл lab9-5.asm (рис. ??).

Создание файла

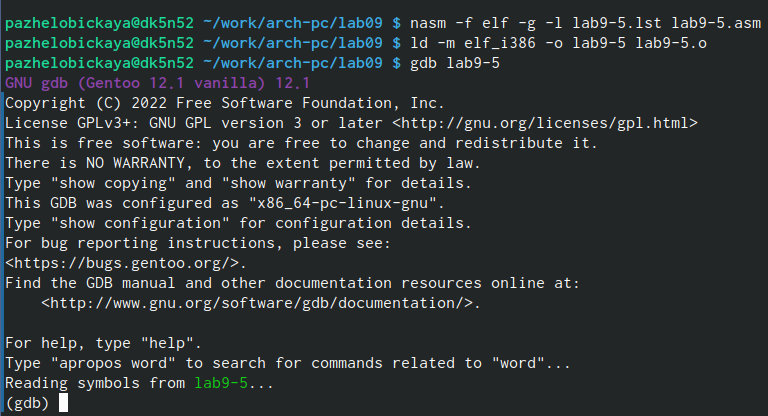
Создание файла

Ввожу в этот файл текст программы из листинга 9.3 (рис. ??).



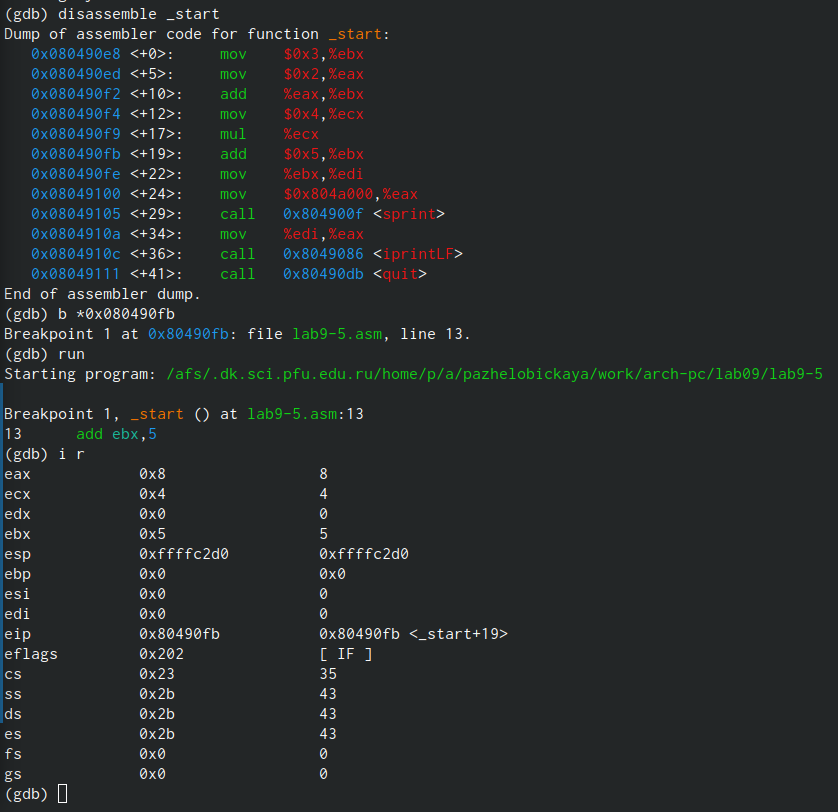
Ввод текста программы

Открываю файл в откладчике GDB и запускаю программу (рис. ??).



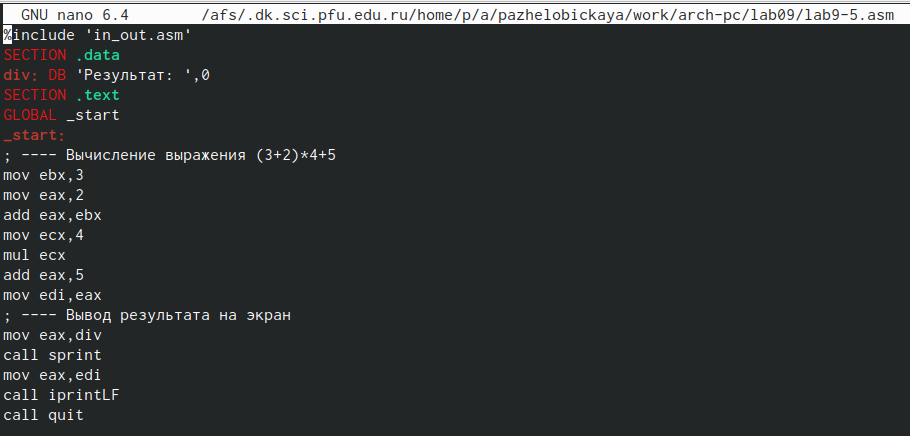
Запуск программы в откладчике

Просматриваю дисассимилированный код программы, ставлю точку останова перед прибавлением 5 и открываю значения регистров на данном этапе (рис. ??).



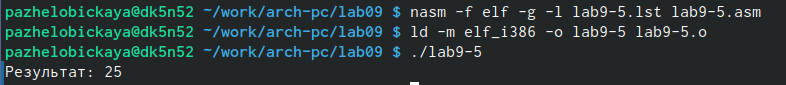
Действия в откладчике

Регистр ecx со значением 4 умножается не на ebx, сложенным с eax, а только с eax со значением 2. Следовательно, нужно поменять значения регистров (рис. ??).



Изменения в программе

Запускаю программу и получаю верный ответ (рис. ??).



Запуск и проверка программы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. ??).



Название рисунка

# 4 Выводы

я приобрела навыки написания программ с использованием подпрограмм. Ознакомилась с методами откладки при помощи GDB и его основными возможностями.

# Список литературы