Отчет по лабораторной работе №9

Архитектура компьютеров и операционные системы

Никита Сергеевич Кокшаров

Содержание

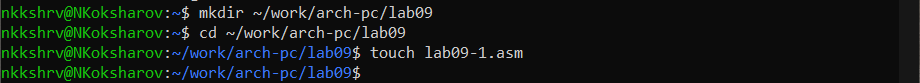
# 1 Цель работы

Целью работы является приобретение навыков написания программ с использование подпрограмм, ознакомление с методами отладки при помощи GDB и его основными возможностями.

# 2 Выполнение лабораторной работы

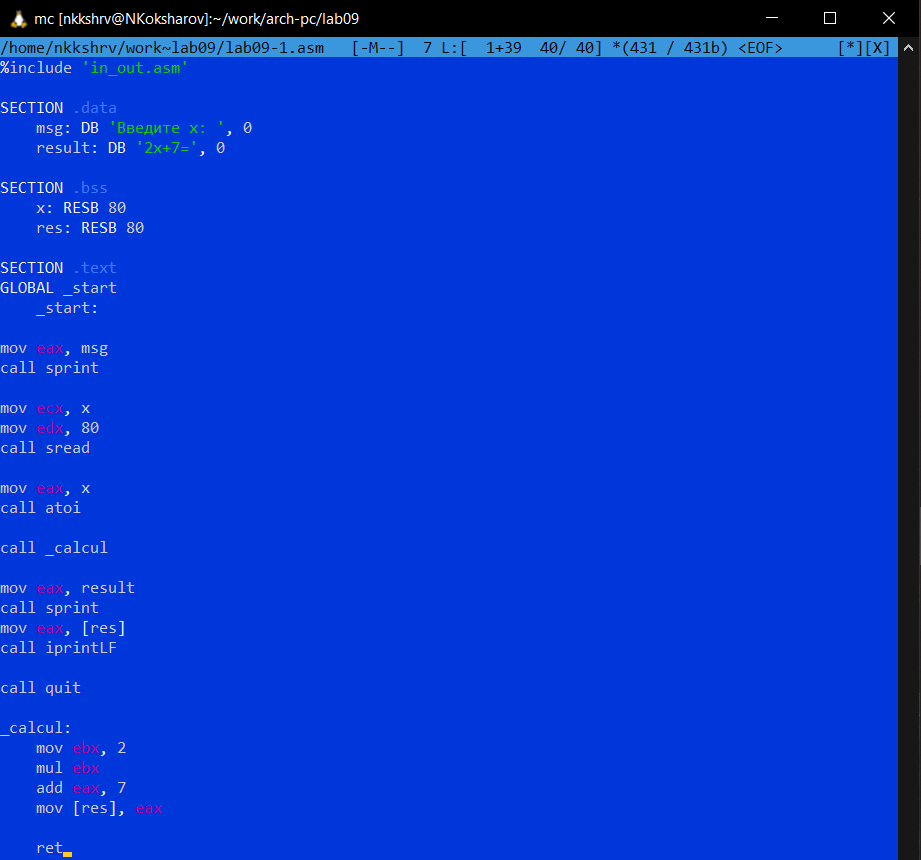
## 2.1 Реализация подпрограмм в NASM

Создаю файл lab09-1.asm в новой директории lab09 (рис. ??).



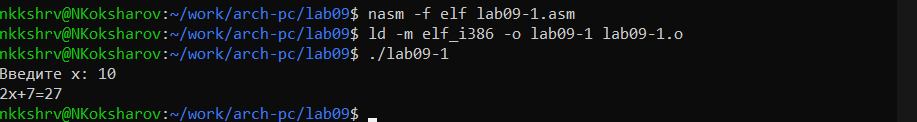
Создание lab09-1.asm

Пишу код программы из листинга 9.1 (рис. ??)



Код в lab09-1.asm

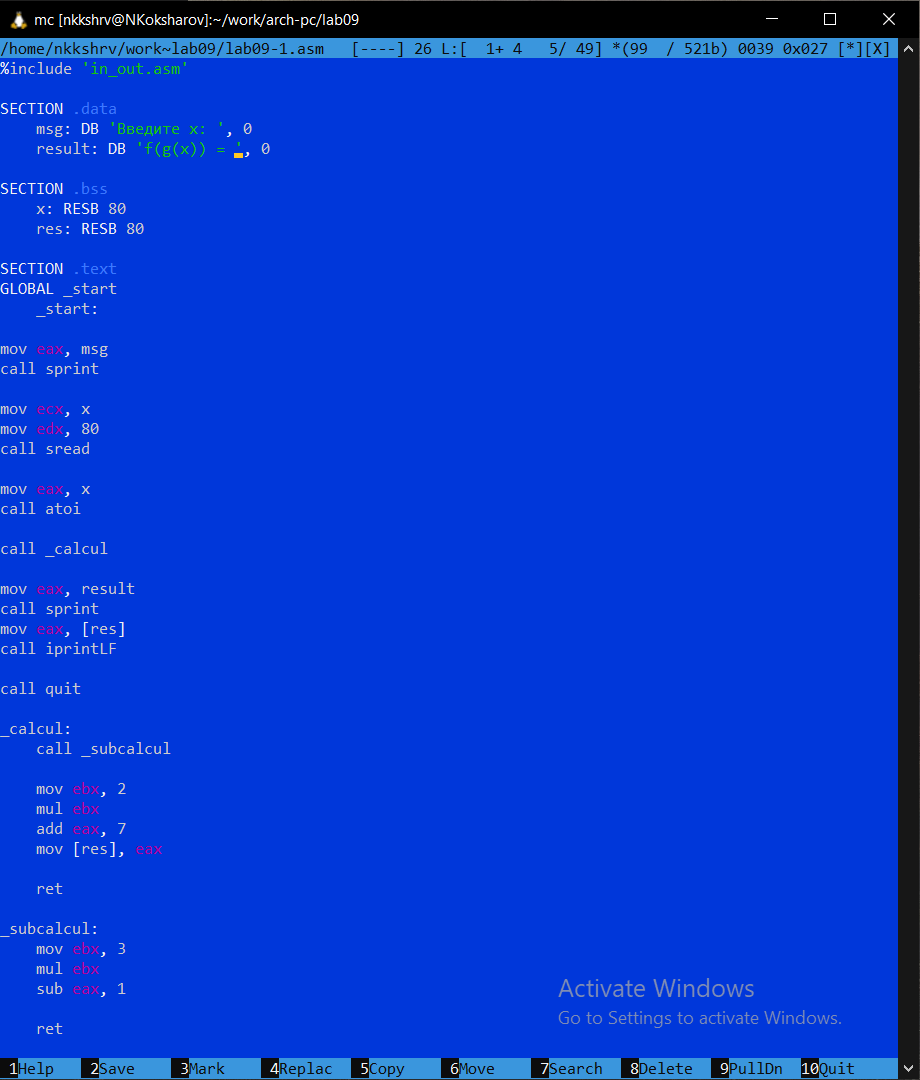
Создаю исполняемый файл и запускаю его (рис. ??)



Запуск lab09-1

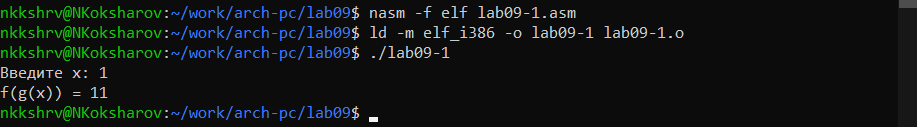
Ответ верный.

Изменяю код программы, добавляя подпрограмму \_subcalcul, вычисляя значение , где , (рис. ??)



Изменение кода в lab09-1.asm

Создаю исполняемый файл и запускаю его (рис. ??)

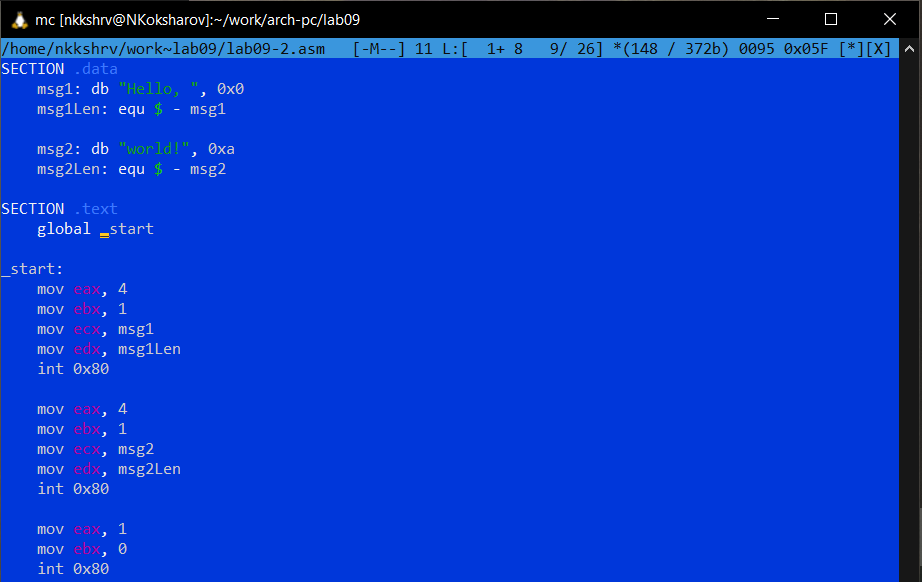


Запуск измененного lab09-1

Ответ верный.

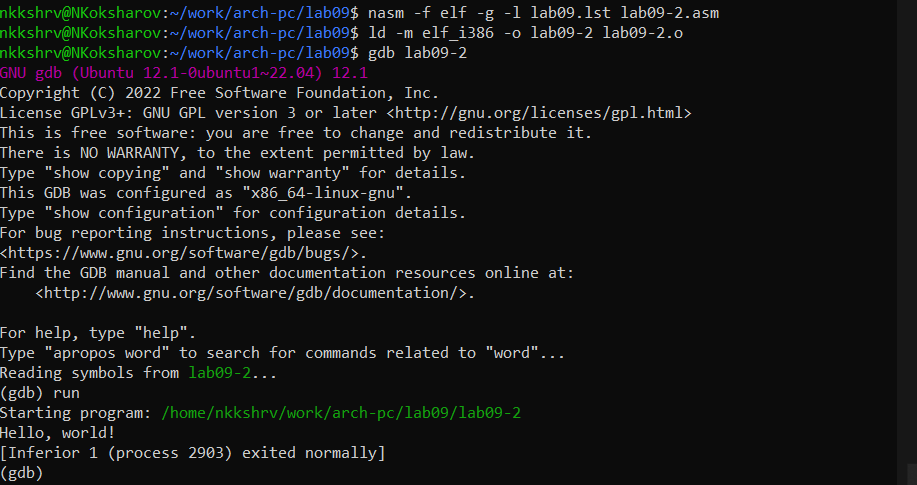
## 2.2 Отладка программам с помощью GDB

Создаю lab09-2.asm, пишу туда код из листинга 9.2 (рис. ??)



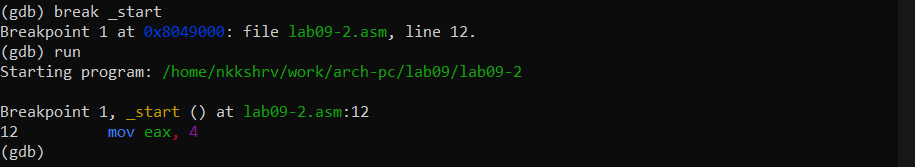
Код в lab09-2.asm

Создаю исполняемый файл и запускаю его c помощью отладчика (рис. ??)



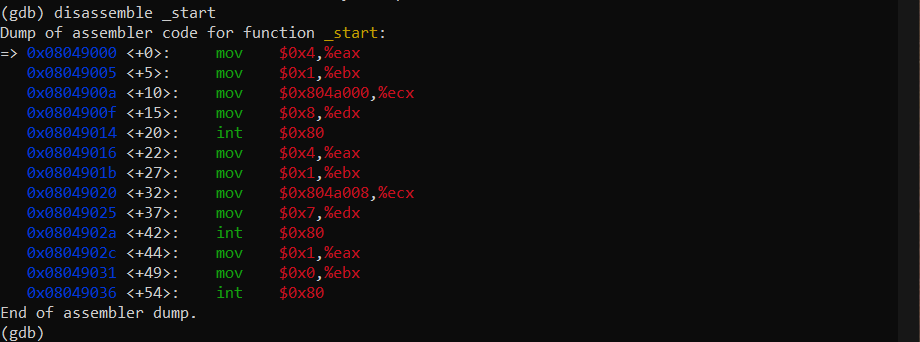
Запуск lab09-2 с помощью отладчика

Ставлю точку останова на метку \_start и запускаю программу(рис. ??)



Запуск lab09-2 с брейкпоинтом

Изучаю дизассемблированный код от отметки \_start (рис. ??)



Дизассемблированный код от отметки \_start

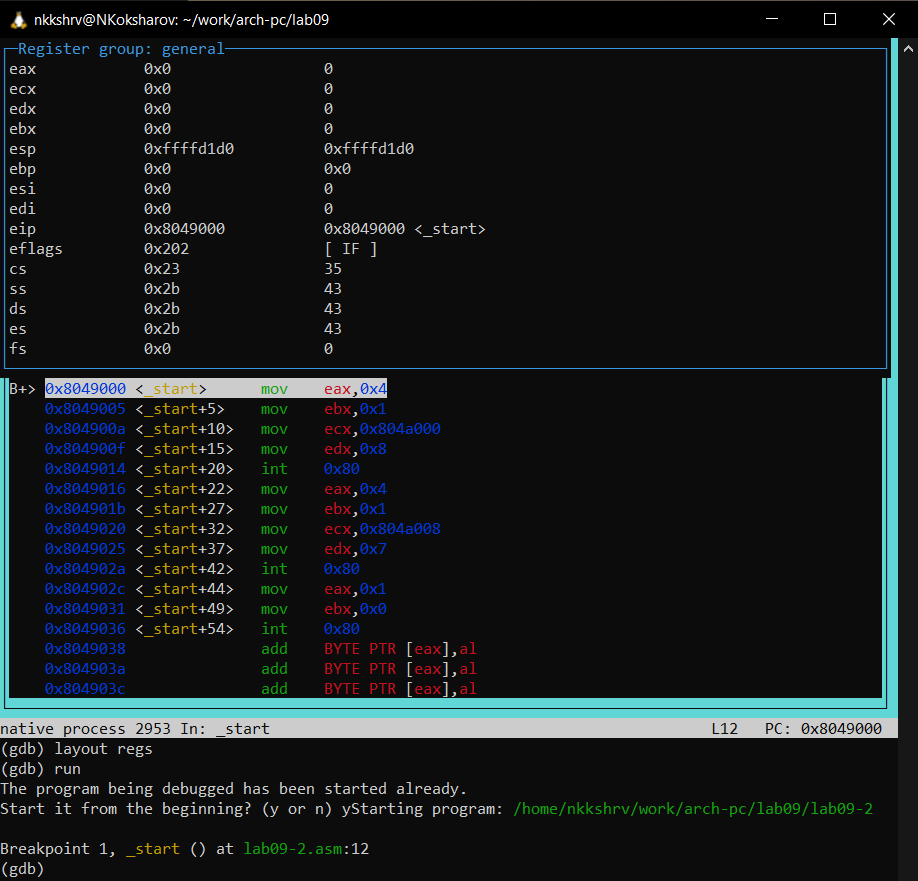
Переключаюсь на отображение команд с синтаксисом Intel (рис. ??)



Дизассемблированный код от отметки \_start (Intel-синтаксис)

Intel-синтаксис более читабелен в отличие от ATT-синтаксиса, поскольку порядок указания регистров идентичен тому, как он указан в коде.

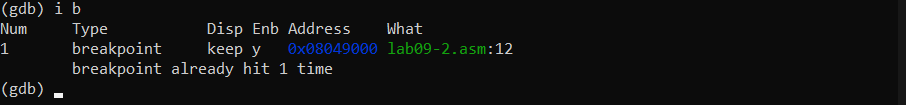
Включаю режим псевдографики (рис. ??)



Режим псевдографики

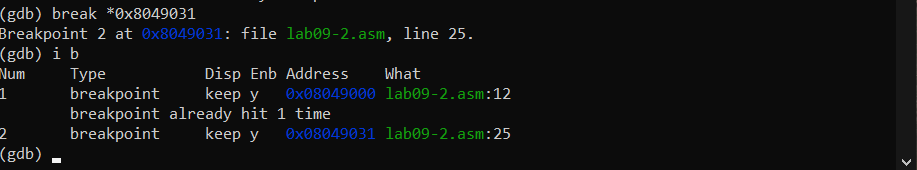
### 2.2.1 Добавление точек останова

С помощью команды info breakpoints (i b) узнаю, где установлены точки останова (рис. ??)



Точки останова при отладке lab09-2

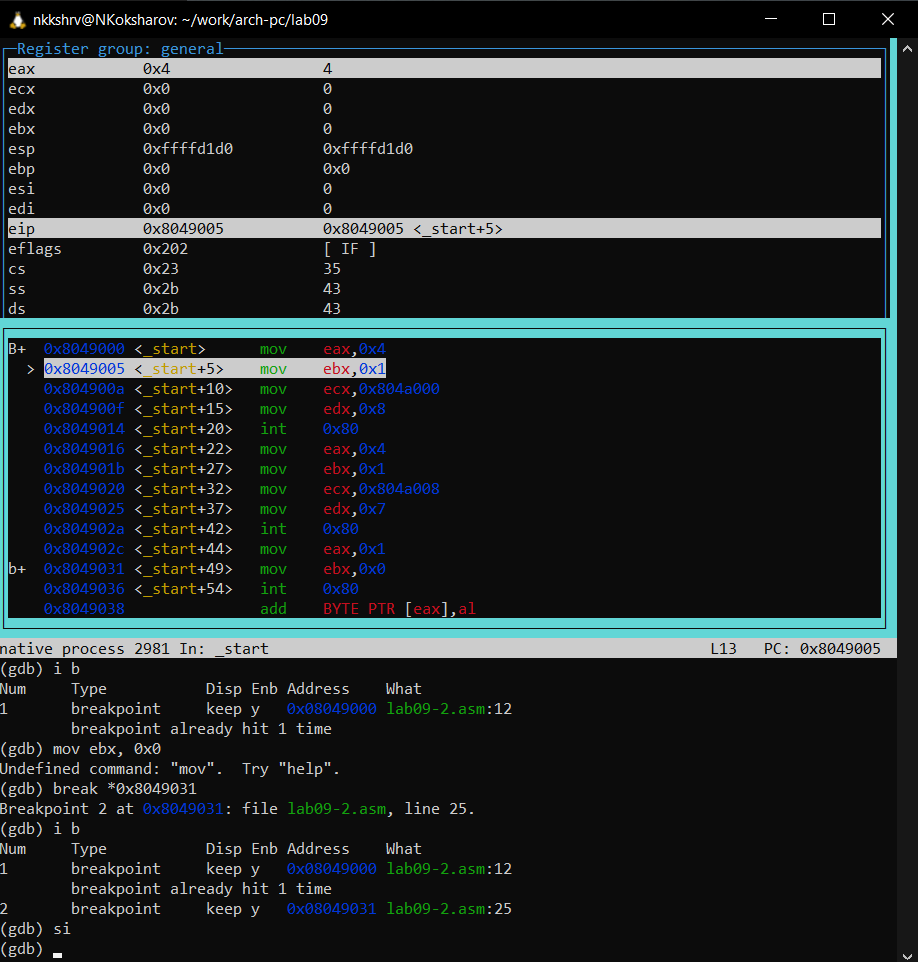
Ставлю точку останова на предпоследней инструкции (рис. ??)



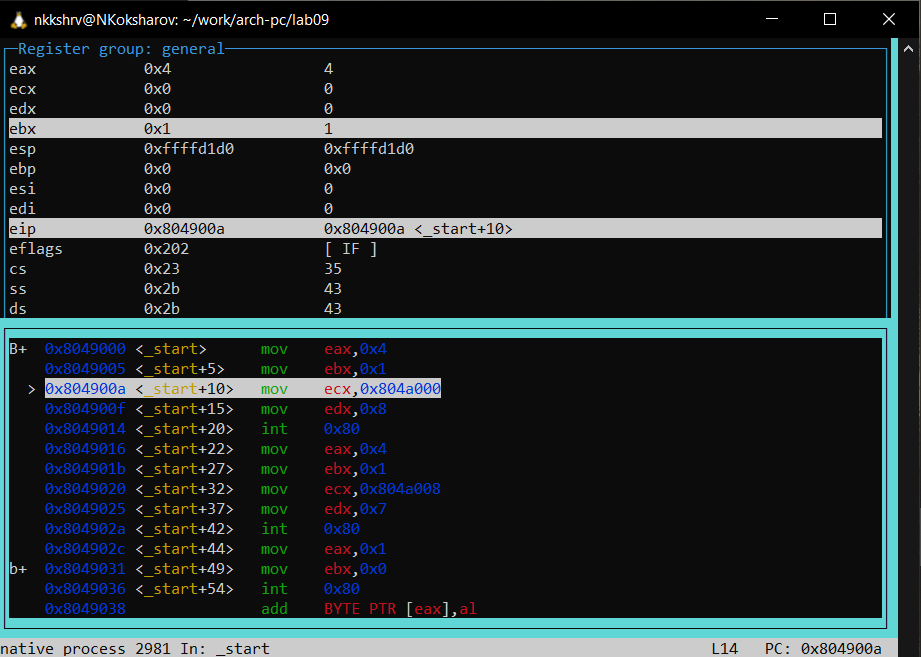
Установка второго брейкпоинта

### 2.2.2 Работа с данными программы в GDB

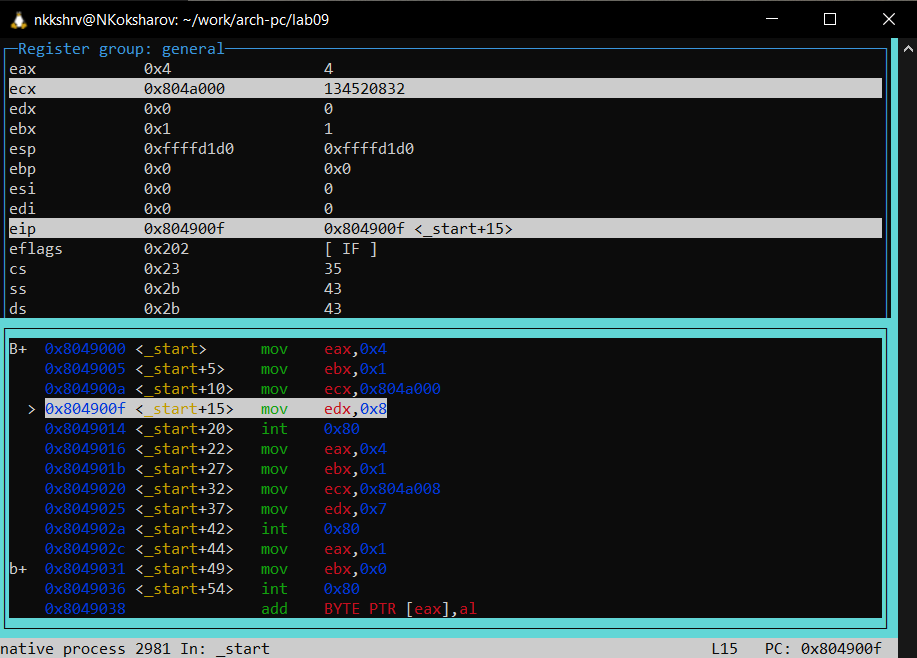
Выполняю 5 инструкций с помощью stepi. Изменяются значения регистров eax, ebx, ecx, edx (рис. ?? - ??)



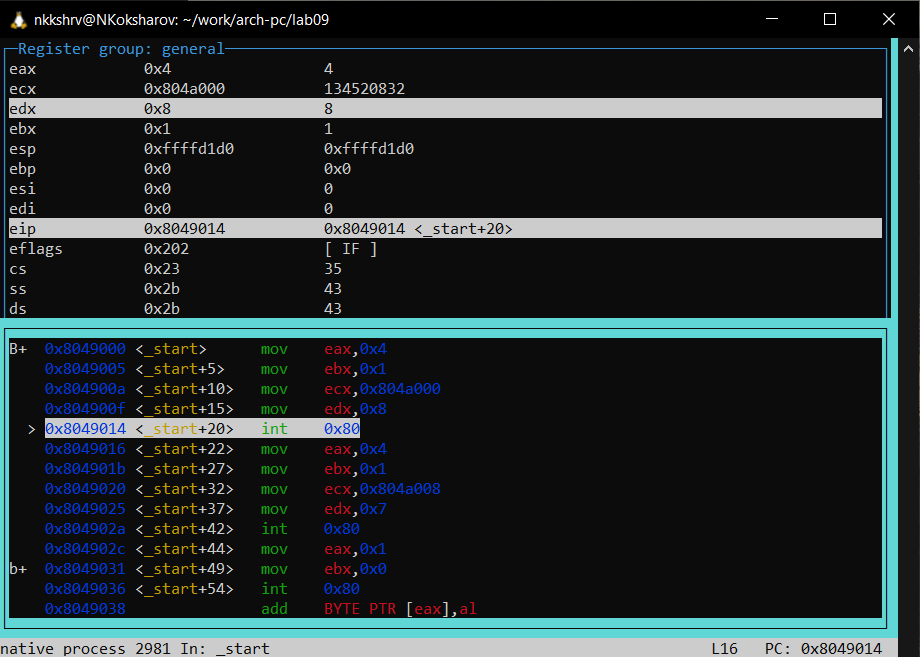
Изменение регистра eax



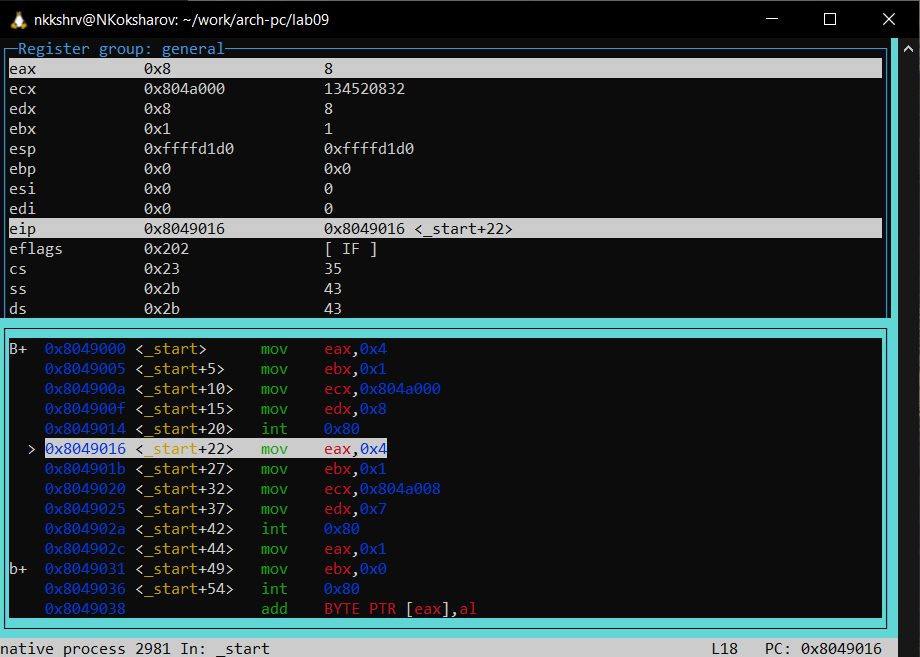
Изменение регистра ebx



Изменение регистра ecx

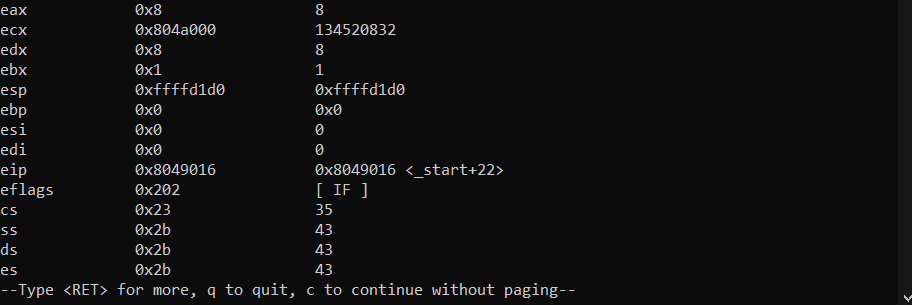


Изменение регистра edx



Изменение регистра eax

Узнаю содержимое регистров (рис. ??)



Содержимое регистров

Узнаю содержимое msg1 по имени, msg2 по имени и адресу (рис. ??)



Содержимое msg1, msg2

Меняю первый символ переменной msg1 (рис. ??)



Изменение msg1

Меняю символ переменной msg2 (рис. ??)



Изменение msg2

Вывожу значения регистра edx в шестнадцатеричном, двоичном и символьном видах (рис. ??)



Значения регистра edx

Меняю значение регистра ebx (рис. ??)

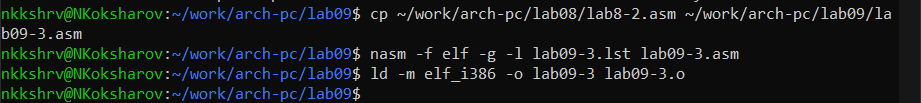


Изменение значения регистра ebx

Разница вывода комант p/s $ebx заключается в том, что сначала регистру присвоена строка '2', а потом число . Эти символы соответствуют разным кодам в UTF-8 (запрашивается значение именно в символьном виде).

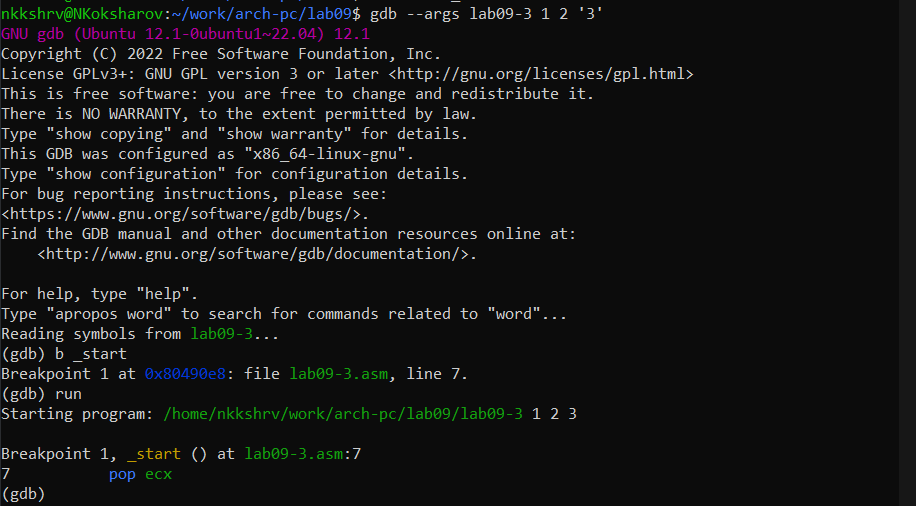
### 2.2.3 Обработка аргументов командной строки в GDB

Копирую файл lab8-2.asm, создаю исполняемый файл (рис. ??)



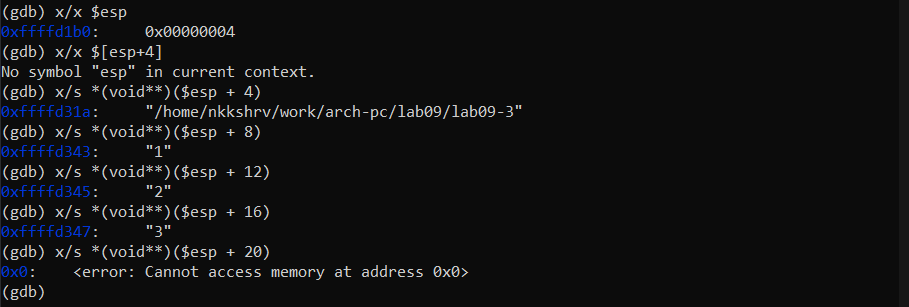
Создание lab09-3

Запускаю отладку lab09-3, ставлю брейкпоинт перед первой инструкцией (рис. ??)



Отладка lab09-3

Обращаюсь к первым пяти позициям стека, где хранятся число аргументов командной строки, имя программы, и три аргумента (рис. ??)

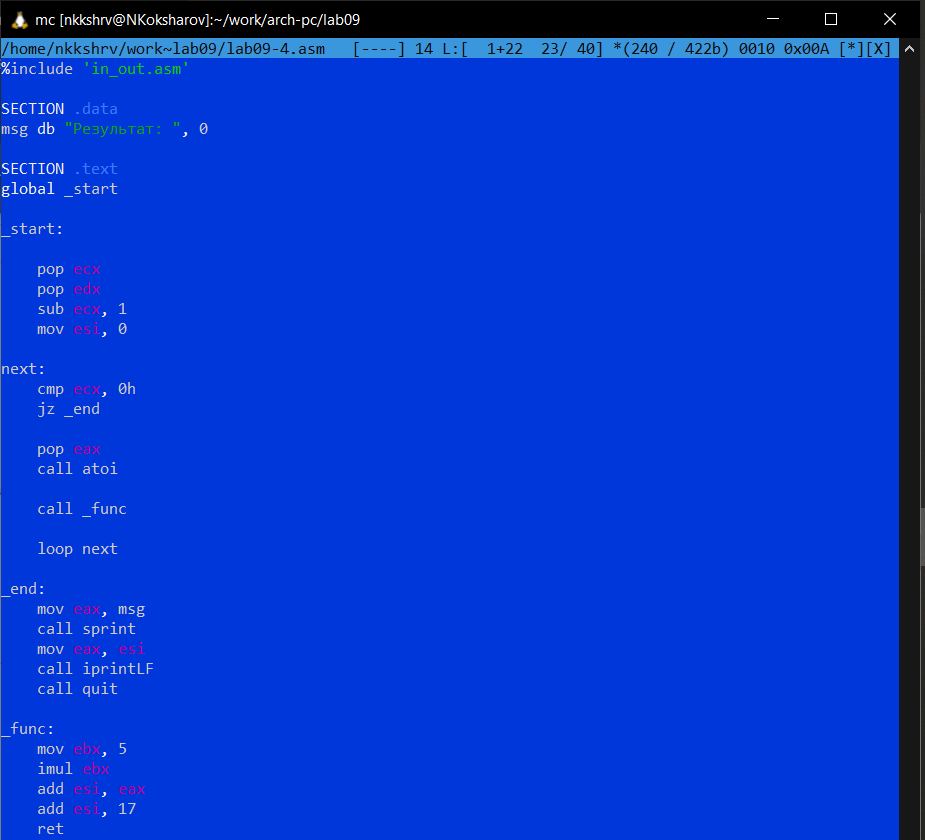


Обращение к позициям стека

Шаг равен изменения адреса равен 4, потому что каждая ячейка занимает 4 байта в памяти.

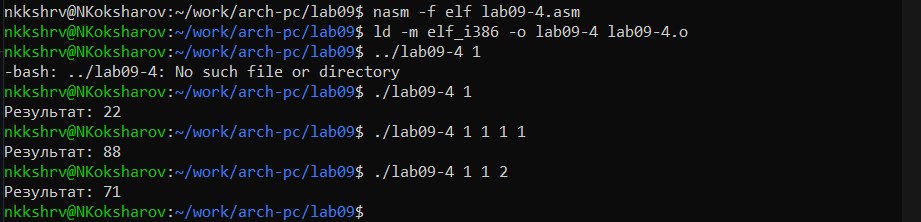
## 2.3 Задание для самостоятельной работы

Преобразовываю код программы из лабораторной работы №8, реализуя вычисление значения функции как подпрограмму (рис. ??)



Код в lab09-4.asm

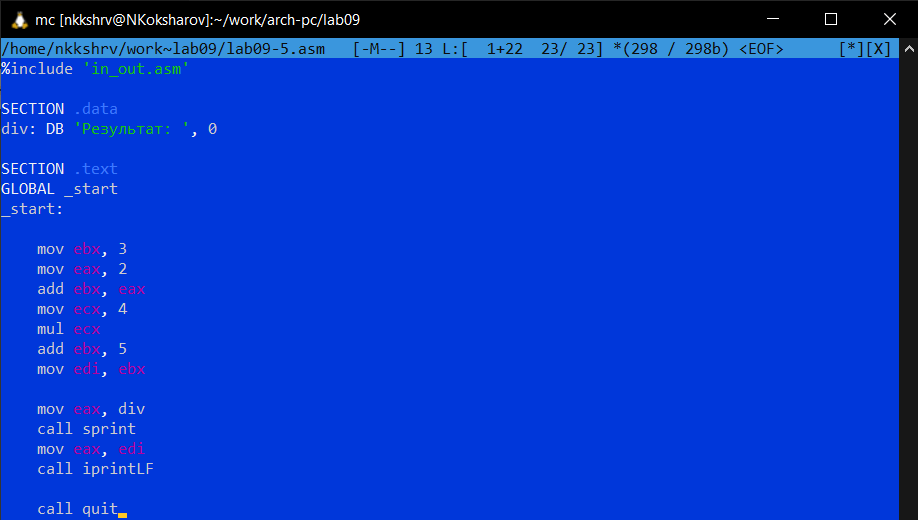
Создаю исполняемый файл и запускаю его (рис. ??)



Запуск lab09-4

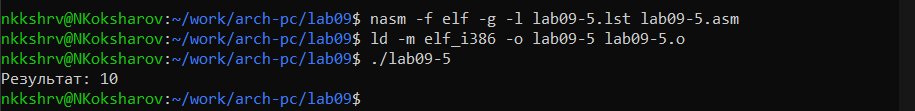
Ответ для разного набора аргументов верный.

Создаю файл lab09-5.asm и пишу в нем код программы из листинга 9.3 (рис. ??)



Код в lab09-5.asm

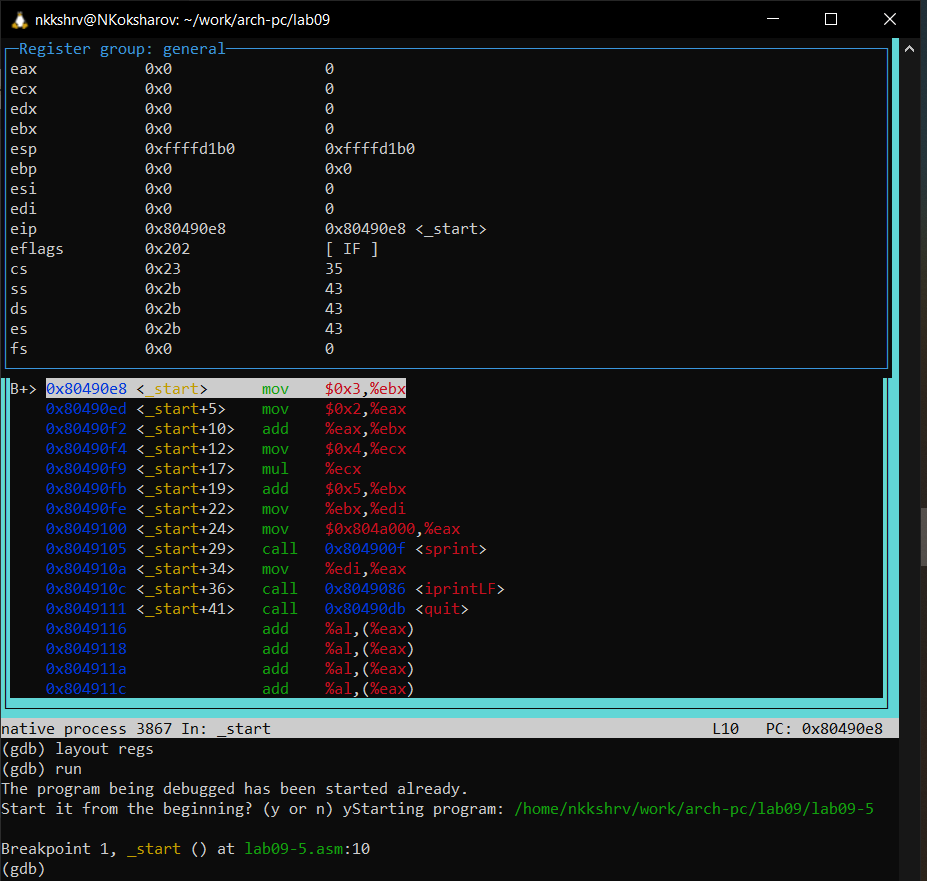
Создаю исполняемый файл и запускаю его (рис. ??)



Запуск lab09-5

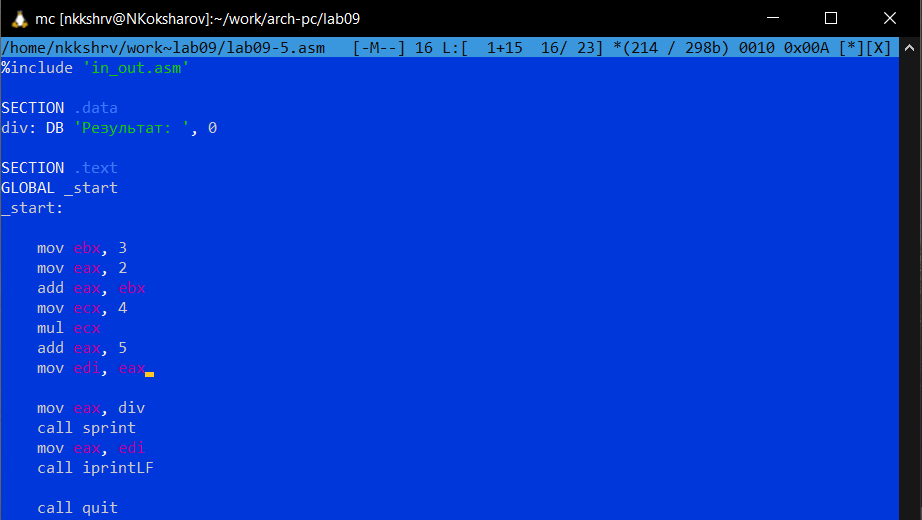
Результат действительно неверный:

C помощью отладчика пытаюсь найти ошибку в коде (рис. ??)



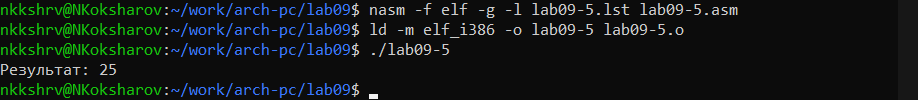
Отладка lab09-5

Нашедши ошибку, исправляю код (рис. ??)



Исправленный код в lab09-5.asm

Создаю исполняемый файл и запускаю его (рис. ??)



Запуск исправленного lab09-5

Ответ верный.

# 3 Выводы

При выполнении лабораторной работы я приобрел навыки написания программ с использование подпрограмм и ознакомился с возможностями отладки с помощью GDB.