Отчёт по лабораторной работе №9

дисциплина: Архитектура компьютера

Попов Даниил Георгиевич

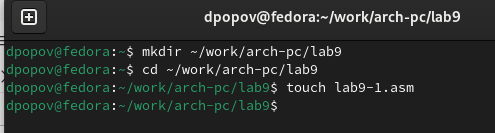
Содержание

# 1)Цель работы

Приобрести навыки написания программ с использованием подпрограмм. Знакомство с методом отладки при помощи GDB и его основными возможностями.

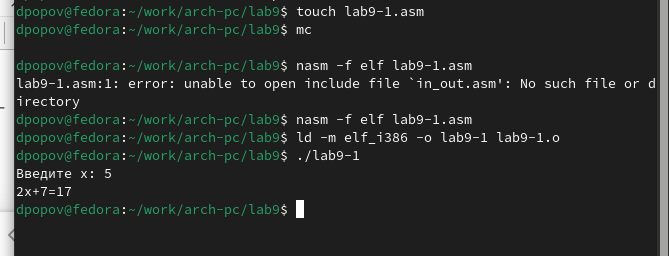
# 2)Выполнение лабораторной работы

## 2.1)Создаем каталог для программ ЛБ9 и создаем в нем файл



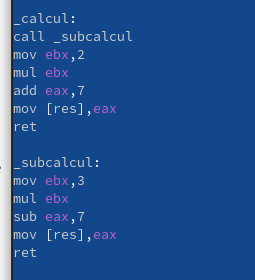
Создаем

## 2.2)Открываем файл в Midnight Commander и заполняем его согласно листингу 1. Создаем исполняемый файл и проверяем

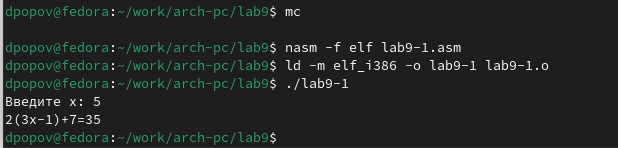


Заполняем, проверяем

## 2.3)Изменяем код в файле и проверяем что изменилось

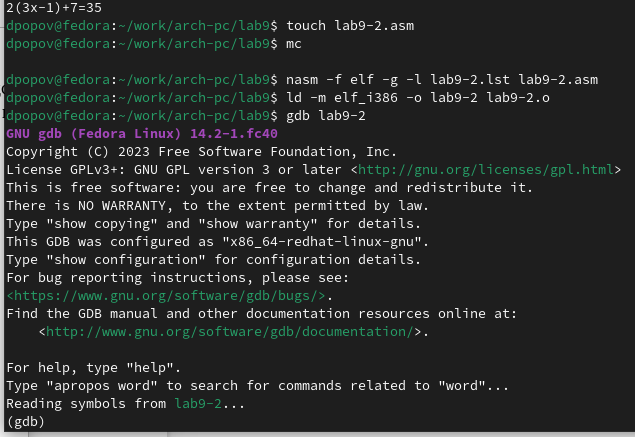


изменяем



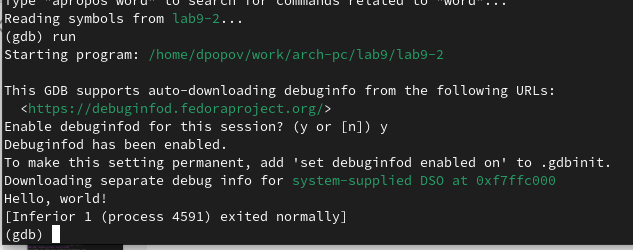
проверяем

## 2.4)Создаем файл, заполняем в соответствии с листингом 2 и получаем исходный файл с использованием отладчика gdb



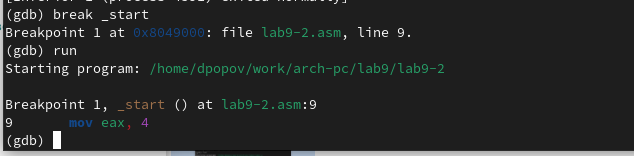
gdb отладчик

## 2.5)Запускаем программу командой run



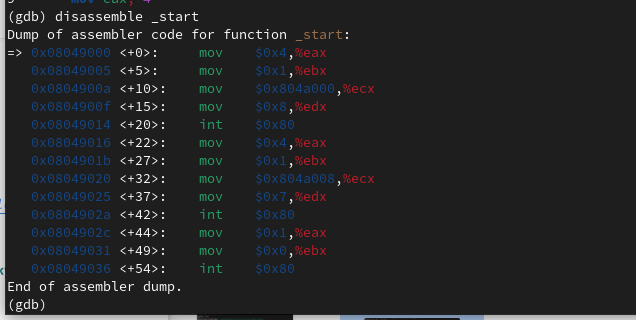
запускаем

## 2.6)Устанавливаем брейкпоинт на метку \_start и запускаем программу



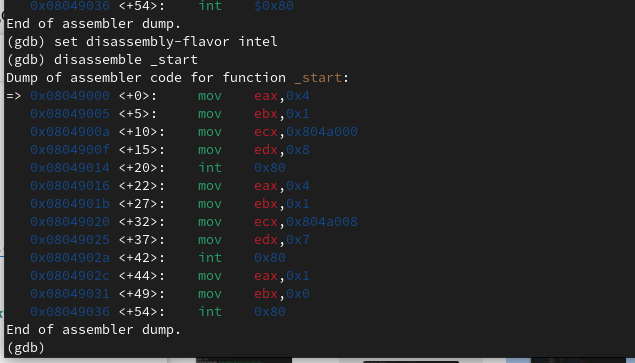
запускаем программу с брейкпоинтом

## 2.7)Смотрим дисассимилированный код программы с помощью команды disassemble, начиная с метку \_start



запускаем

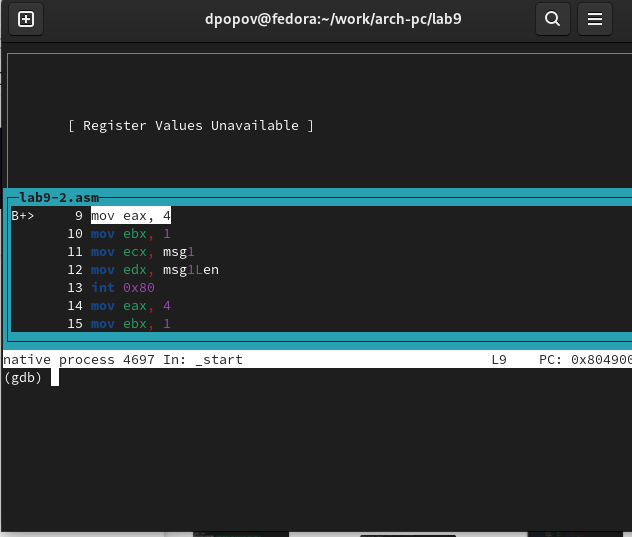
## 2.8)Переключаем на отображение команд с Intel синтаксисом



intel синтаксис

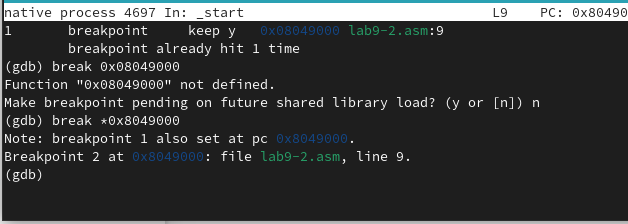
Различия отображения синтаксиса машинных команд в режимах ATT и Intel: 1.Порядок операндов: В ATT синтаксисе порядок операндов обратный, сначала указывается исходный операнд, а затем - результирующий операнд. В Intel син- таксисе порядок обычно прямой, результирующий операнд указывается первым, а исходный - вторым. 2.Разделители: В ATT синтаксисе разделители операндов - запятые. В Intel синтаксисе разделители могут быть запятые или косые черты (/). 3.Префиксы размера операндов: В ATT синтаксисе размер операнда указывается перед операндом с использованием префиксов, таких как “b” (byte), “w” (word), “l” (long) и “q” (quadword). В Intel синтаксисе размер операнда указывается после операнда с использованием суффиксов, таких как “b”, “w”, “d” и “q”. 4.Знак операндов: В ATT синтаксисе операнды с позитивными значениями предваряются символом ““.𝐼𝑛𝑡𝑒𝑙””. 5.Обозначение адресов: В ATT синтаксисе адреса указываются в круглых скоб- ках. В Intel синтаксисе адреса указываются без скобок. 6.Обозначение регистров: В ATT синтаксисе обозначение регистра начинается с символа “%”. В Intel синтаксисе обозначение регистра может начинаться с символа “R” или “E” (например, “%eax” или “RAX”).

## 2.9)Включаем отображение регистров, их значений и результат дисассимилирования программы



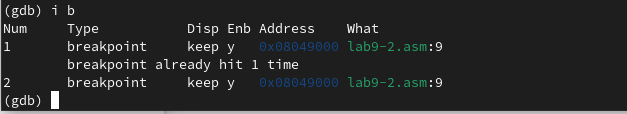
включаем

## 2.10)Проверяем была ли установлена точка остановки и устанавливаем точку остановки последней инструкции



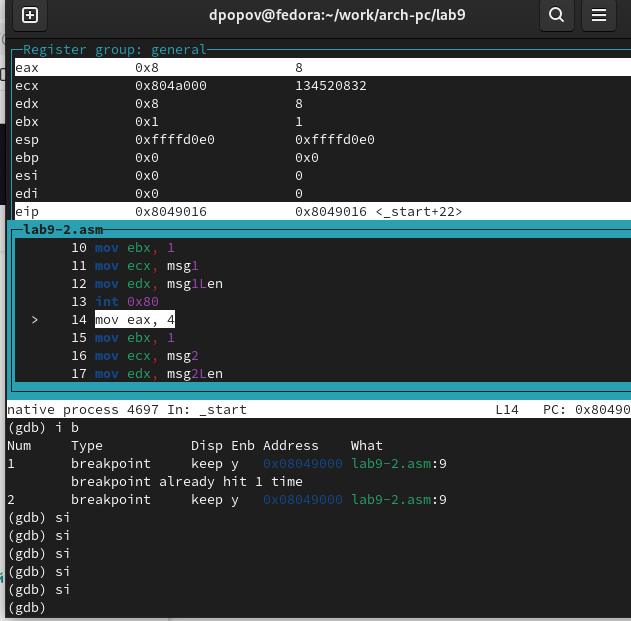
проверяем

## 2.11)Смотрим информацию о всех установленных точках остановки



смотрим

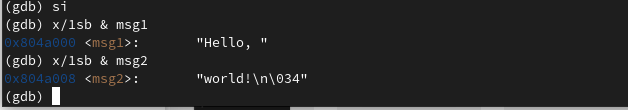
## 2.12)Выполняем 5 команд si



si

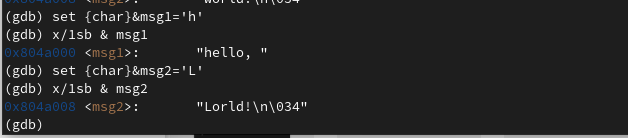
Во время выполнения команд менялись регистры: ebx, ecx, edx, eax, eip

## 2.13)Смотрим значение переменной msg1 и переменной msg2



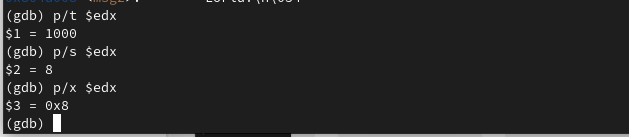
si

## 2.14)Меняем первый символ переменной msg1 и msg2



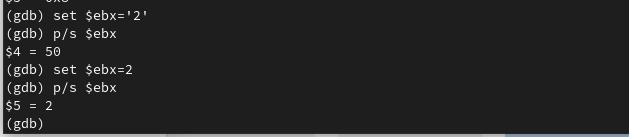
меняем

## 2.15)Смотрим значение регистра edx в разных форматах



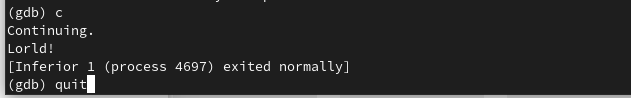
смотрим

## 2.16)Изменяем регистр ebx



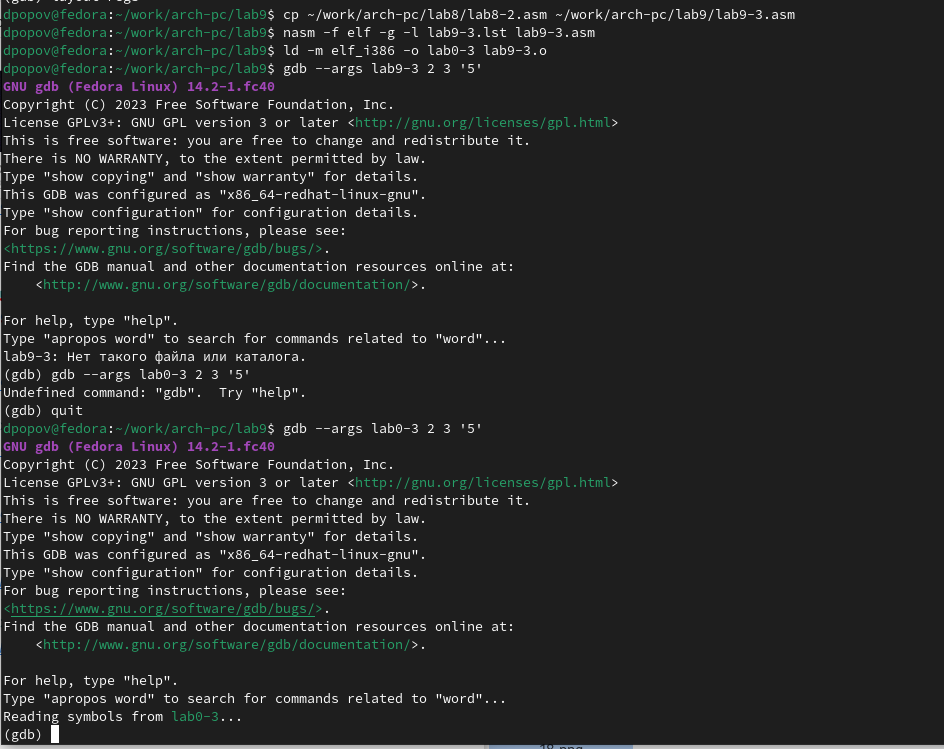
изменяем

## 2.17)Прописываем команды для завершения программы и выхода из GBD



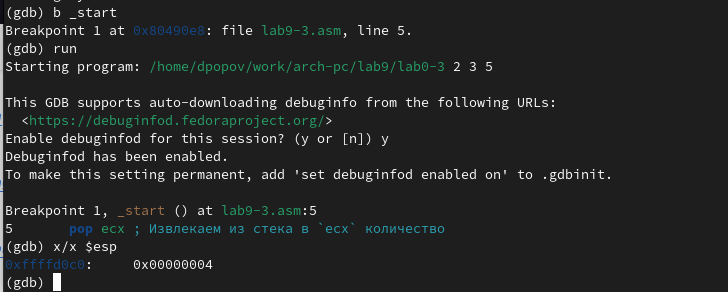
Прописываем

## 2.18)Копируем файл lab8-2.asm в файл с именем lab9-3.asm создаем и запускаем в отладчике файл



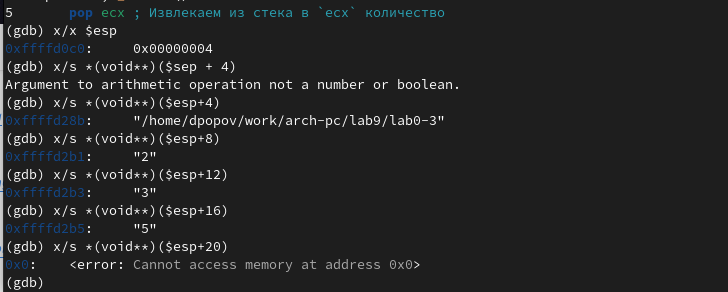
Прописываем

## 2.19)Установим точку останова перед первой инструкцией в программе и запустим ее



Прописываем

## 2.20)Смотрите позиции стека по разным адресам



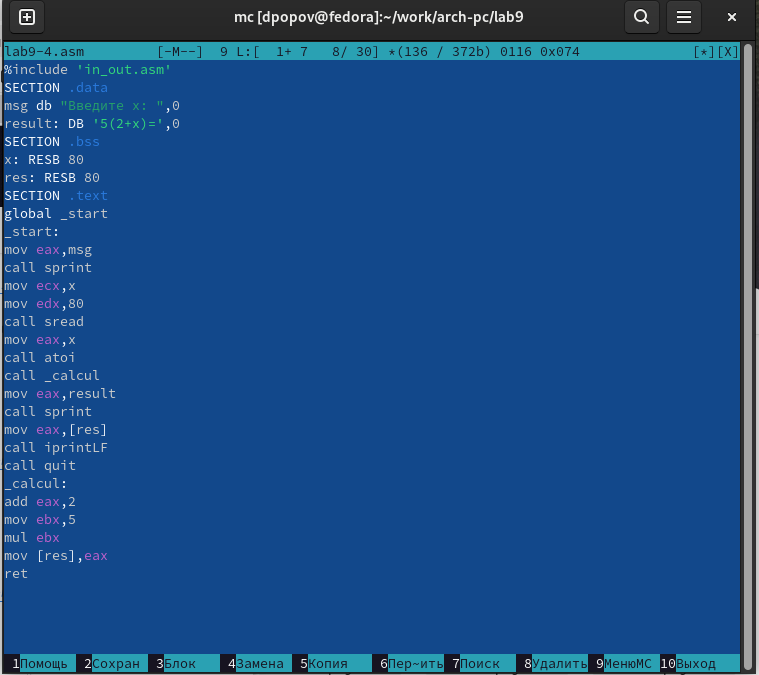
Прописываем

Шаг изменения адреса равен 4 потому что адресная регистрация имеют размерность 32 бита(4байта)

# 3)Задания для самостоятельной работы

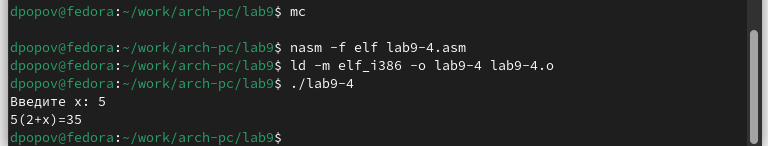
ВАРИАНТ 10 ##Задание 1 ### 3.1.1)Копируем файл lab8-4.asm в файл с именем lab9-3.asm 

### 3.1.2)Открываем файл и пишем код

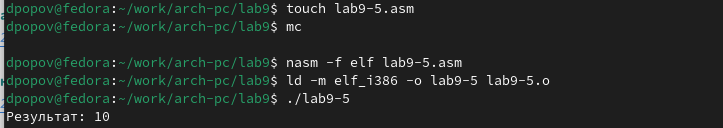
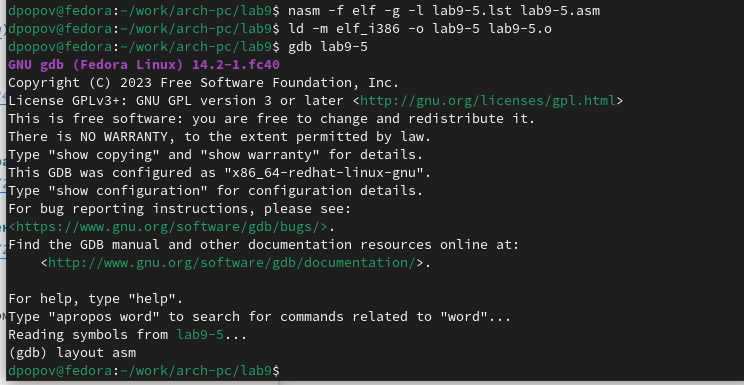
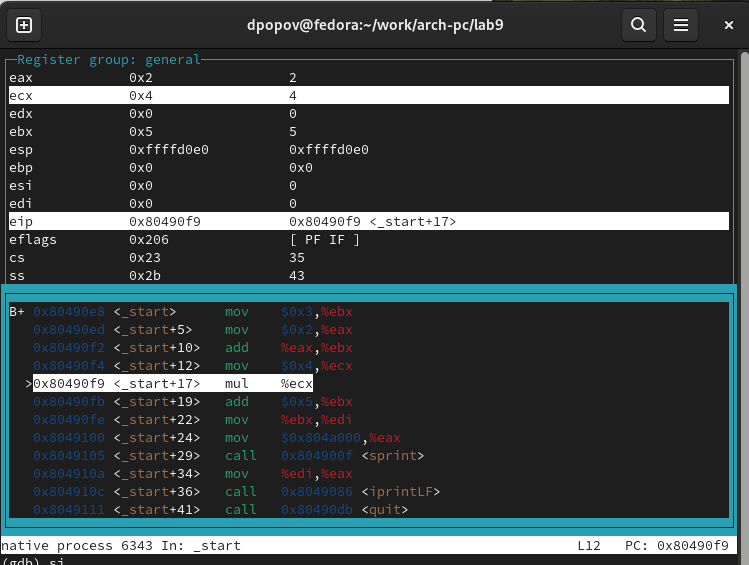
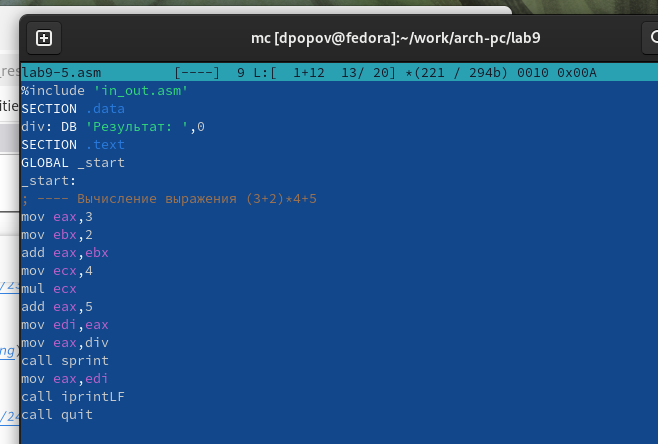
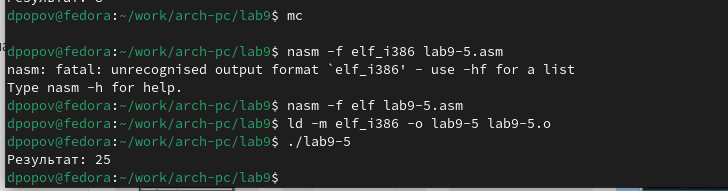


код

### 3.1.3)Проверяем



проверка

##Задание 2 ### 3.2.1)Создаем файл и заполняем файл в соответствии с листингом и проверяем работу  Работает неправильно ### 3.2.2)Создаем исполняем файл и запускаем его в отладке GDB и смотрим на изменение решистров командой si  ### 3.2.3)Ищем ошибку  ### 3.2.4)Изменяем код для корректной работы  ### 3.2.4)Проверяем 

# 4)Выводы

Мы познакомились с методом отладки при помощи GBD и его возможности