Лабораторная работа №9.

Понятие подпрограммы. Отладчик GDB.

Митрофанов Тимур Александрович

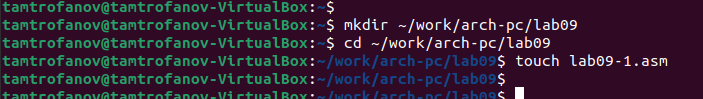
Содержание

# 1 Цель работы

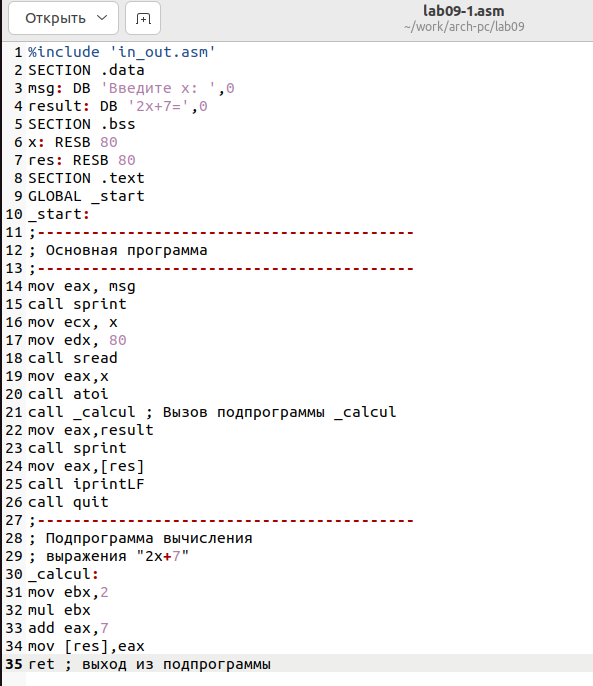
Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

# 2 Выполнение лабораторной работы

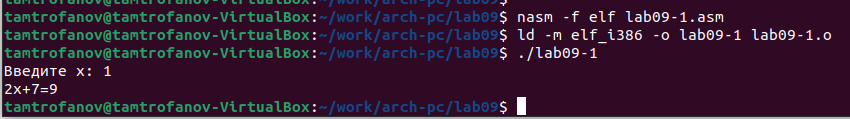
Создал подкаталог *lab09*. В нём создал файл touch *lab09-1.asm* (рис. ??). В созданный файл ввёл текст листинга 9.1(рис. ??). Скомпелировал файл и проверил его работу (рис. ??).



Создание подкаталога и файла

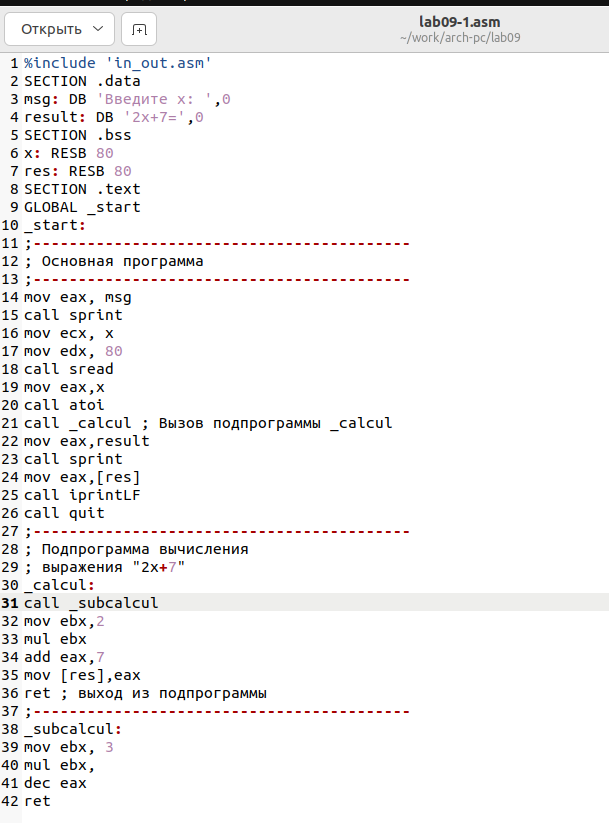


Содержимое файла *lab09-1.asm*

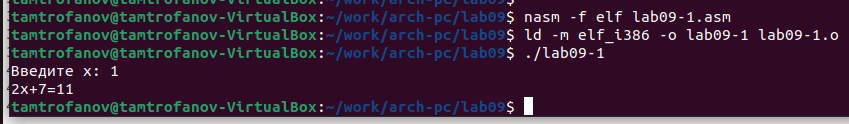


Создание исполняемого файла *lab09-1* и его запуск

В соответствии с заданием создал подпрограмму которая отвечает за вычисления (рис. ??). Скомпелировал файл и проверил его работу (рис. ??).



Изменённое содержимое файла *lab09-1.asm*

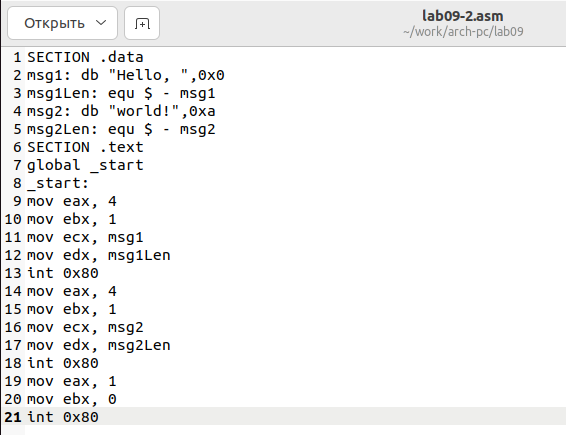


Создание исполняемого файла и его запуск

Создал файл *lab09-2.asm* (рис. ??). В созданный файл ввёл текст листинга 9.2 (рис. ??). Для работы с GDB при создании исполняемого файла добавил ключ ***-g*** (рис. ??).

Создание файла lab09-2.asm

Создание файла *lab09-2.asm*



Содержимое файла *lab09-2.asm*

Создание исполняемого файла lab09-2

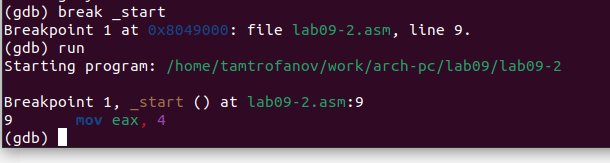
Создание исполняемого файла *lab09-2*

Загрузил исполняемый файл в отладчик gdb и запустил его в нём (рис. ??).



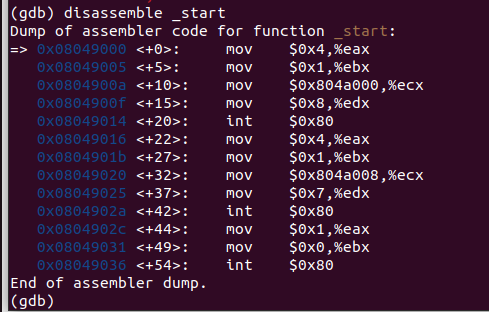
Загрузка исполняемого файла в отладчик gdb и его запуск

Установил брейкпоинт на метку \_start и запустил программу (рис. ??).



Установка брейкпоинта и запуск программы

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



Ввод команды \*\*\*disassemble \_start\*\*\*

Переключил на отображение команд с Intel’овским синтаксисом, введя команду ***set disassembly-flavor intel*** (рис. ??).

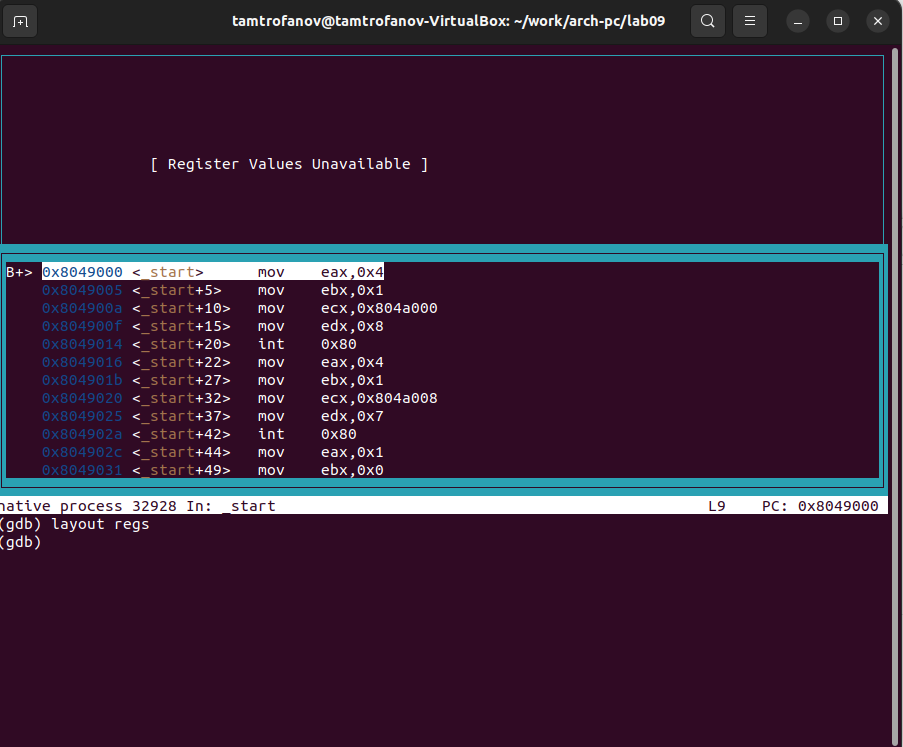


ереключение на отображение команд с Intel’овским синтаксисом

***Перечил различия отображения синтаксиса машинных команд в режимах ATT и Intel.***

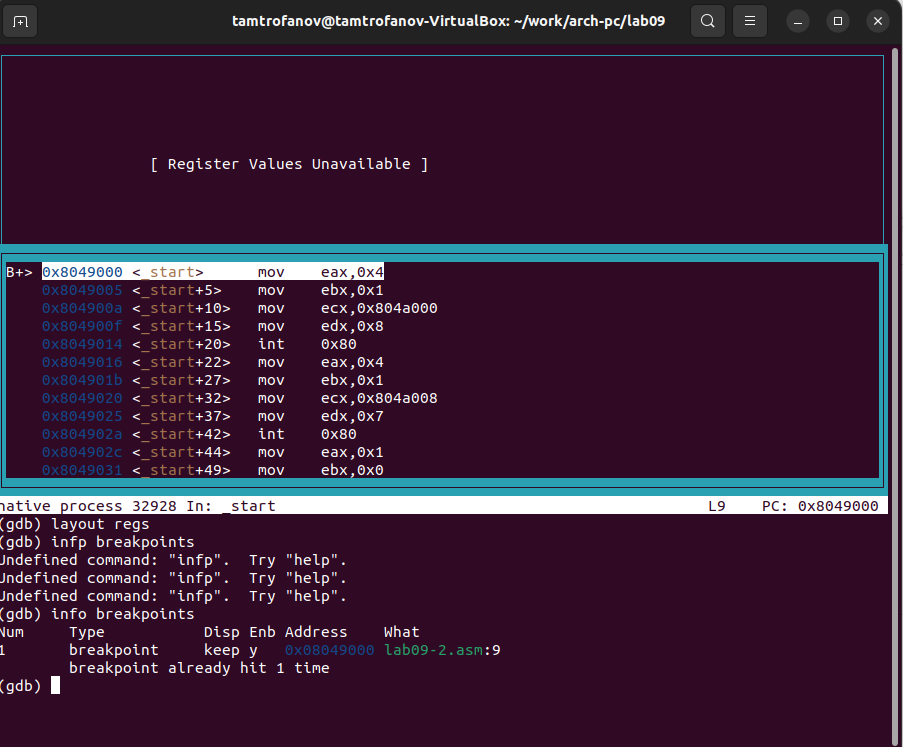
* В ATT имена регистров начинаются с символа %, а имена опрерандов с $, в то время как в intel используется привычный нам синтаксис.

Включил режим псевдографики для более удобного анализа программы (рис. ??).

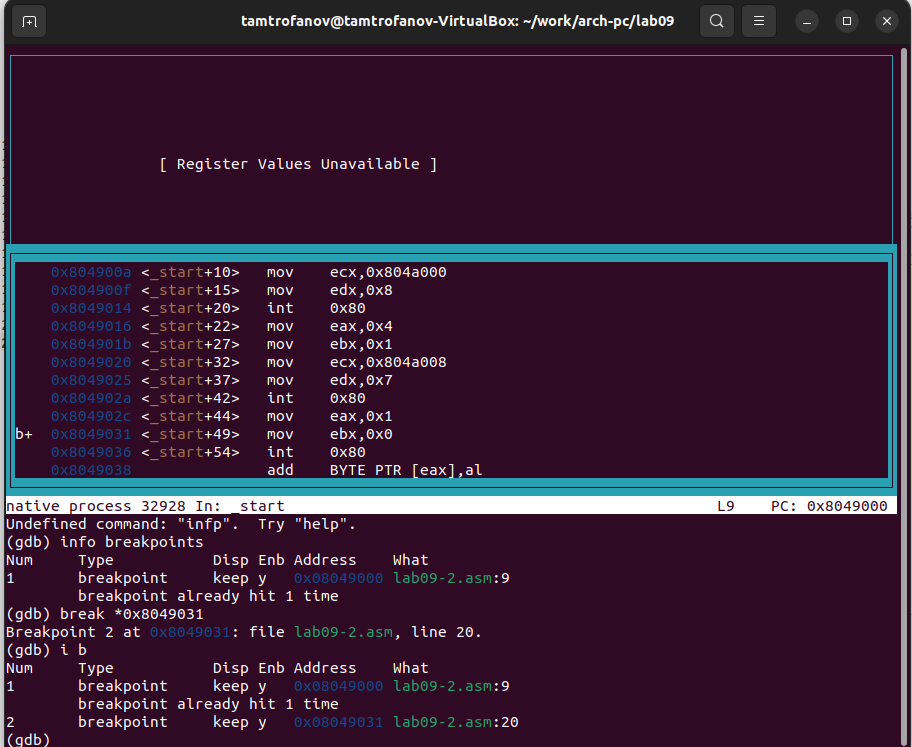


Режим псевдографики

Получил данные о брейкпоинтах (рис. ??). Добавил ещё один брейкпоинт и проверил его наличие (рис. ??).

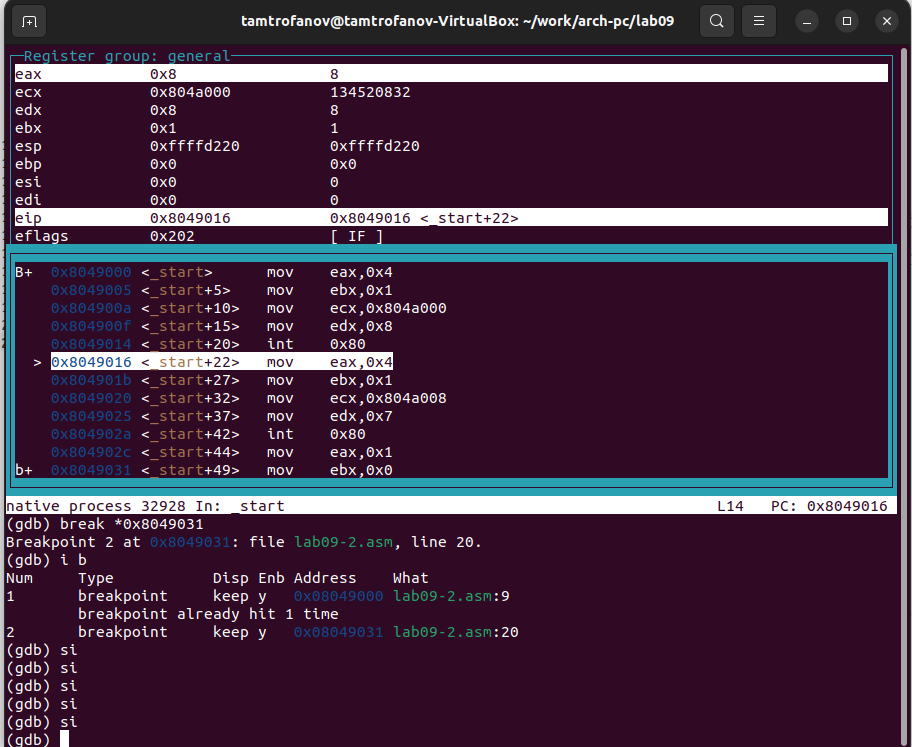


Режим псевдографики



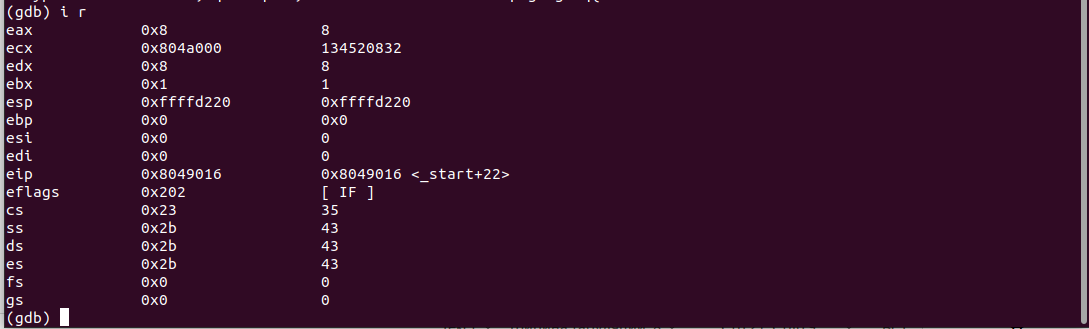
Работа с брейкпоинтами

Выполнил 5 инструкций с помощью команды stepi (рис. ??).



Выполнение инструкций

Посмотрел содержимое регистров при помощи команды info registers (рис. ??).



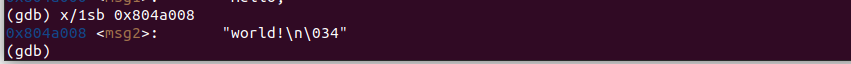
Посмотр содержимого регистров

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



Посмотр значения переменной msg1 по имени

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



Посмотр значения переменной msg2 по адресу

Измененил первыый символ переменной msg1 на ***h*** (рис. ??).



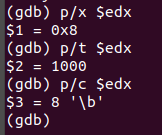
Изменение значения переменной msg1

Измененил первыый символ переменной msg2 на ***Q*** (рис. ??).



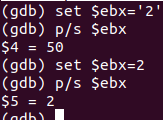
Изменение значения переменной msg2

Выведел в различных форматах значение регистра edx (рис. ??).



различные форматы значений регистра edx

С помощью команды ***set*** измените значение регистра *ebx* (рис. ??).

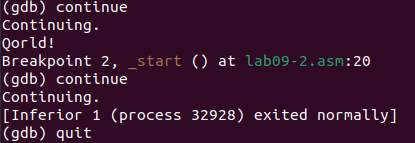


Изменение значения регистра *ebx* при помощи set

***Объясните разницу вывода команд p/s $ebx.***

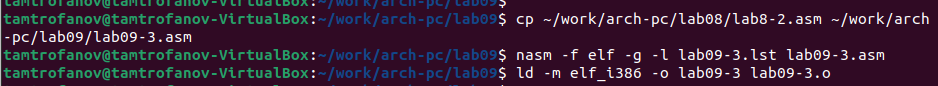
* В первом случае мы переводим символ в строковой вид.

Завершил выполнение программы с помощью команды continue и выйшел из GDB с помощью команды quit (рис. ??).



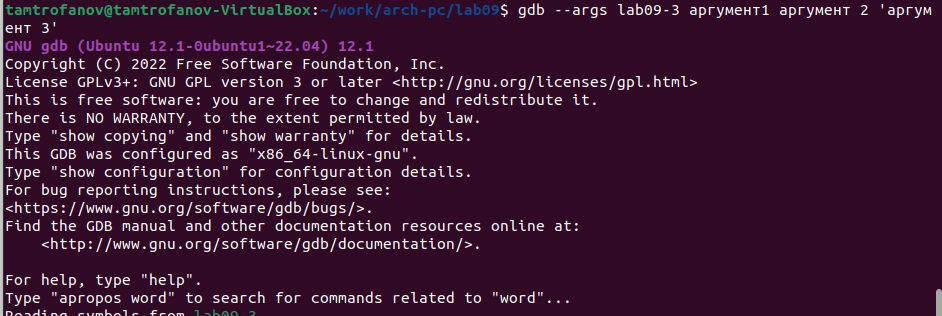
Завершение работы с GDB

Скопировал файл *lab8-2.asm* в файл с именем *lab09-3.asm* и создал исполняемый файл (рис. ??).



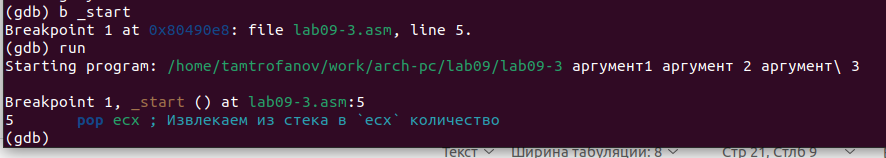
Создание файла *lab09-3.asm* и преобразование его в исполняемый файл

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



Загрузка файла *lab09-3.asm* в отладчик

установил точку остановки перед первой инструкцией в программе и запустил ее (рис. ??).



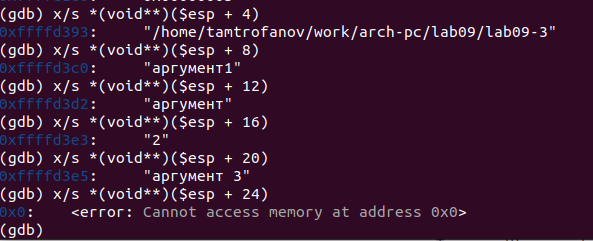
Установка брейкпойнта и запуск программы

Проверил сколько в стеке хранится переменных (рис. ??).

Проверка стека

Проверка стека

Посмотрел остальные позиции стека (рис. ??).



Проверка переменных в стеке

***Объясните, почему шаг изменения адреса равен 4 ([esp+4], [esp+8], [esp+12] и т.д.)***

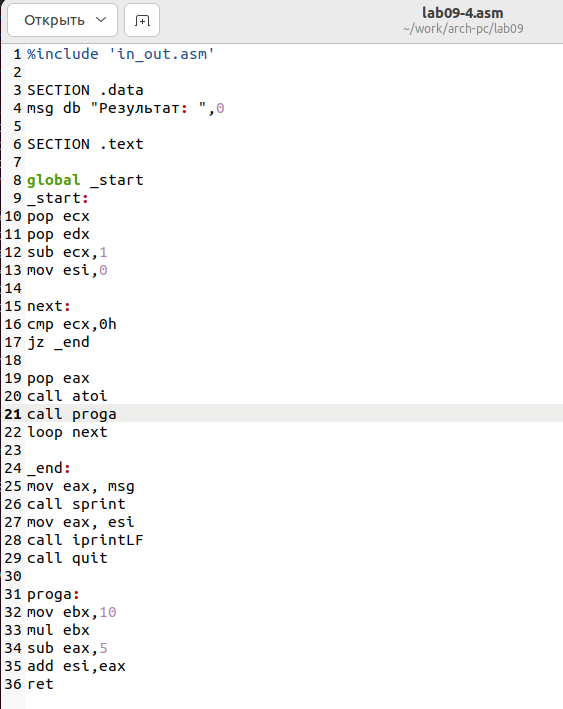
* Так как занчение колличества аргументов равно 4

# 3 Задание для самостоятельной работы

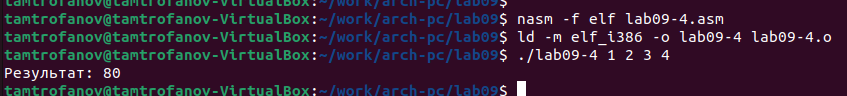
Создал файл *lab09-4.asm* (рис. ??). Перенёс программу из лабораторной работы № 8 в файл и изменил в соответствии с заданием (рис. ??). Создал исполнительный файл и проверил его работу (рис. ??).

Созлание файла lab09-4.asm

Созлание файла *lab09-4.asm*



Содержимое файла *lab09-4.asm*



Создание файла *lab09-4* и проверка его работы

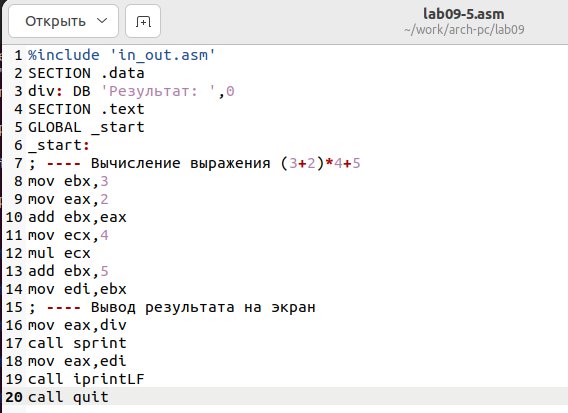
***КОД ПРОГРАММЫЙ***

%include 'in\_out.asm'  
   
SECTION .data  
msg db "Результат: ",0  
   
SECTION .text  
  
global \_start  
\_start:  
pop ecx   
pop edx   
sub ecx,1   
mov esi,0  
   
next:  
cmp ecx,0h   
jz \_end   
   
pop eax   
call atoi  
call proga  
loop next  
   
\_end:   
mov eax, msg   
call sprint  
mov eax, esi   
call iprintLF   
call quit  
  
proga:  
mov ebx,10  
mul ebx  
sub eax,5  
add esi,eax  
ret

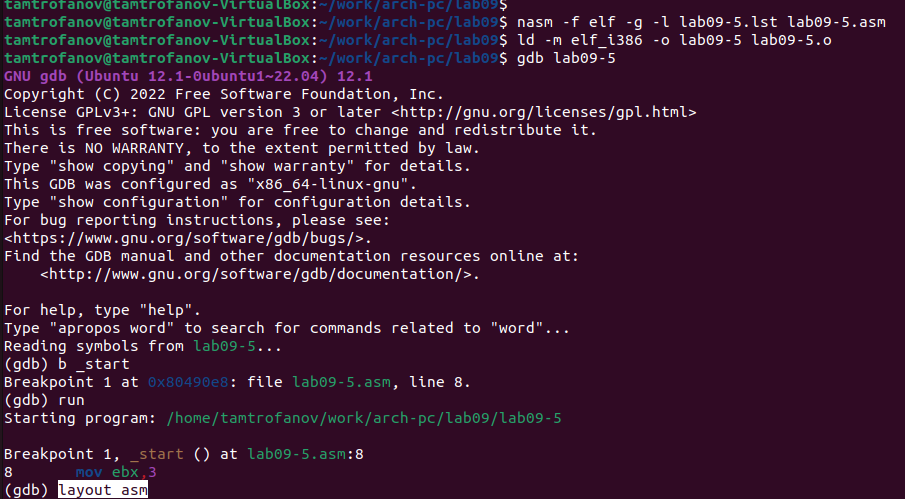
Создал файл *lab09-5.asm* (рис. ??). Внёс уод из листинга 9.3 в файл (рис. ??). Создал исполнительный файл с применением отладчика, загрузил в него этот файл, подготовил программу для работы в отладчике (поставил брейкпойт, подключил граф. интр.) (рис. ??).

Созлание файла lab09-5.asm

Созлание файла *lab09-5.asm*

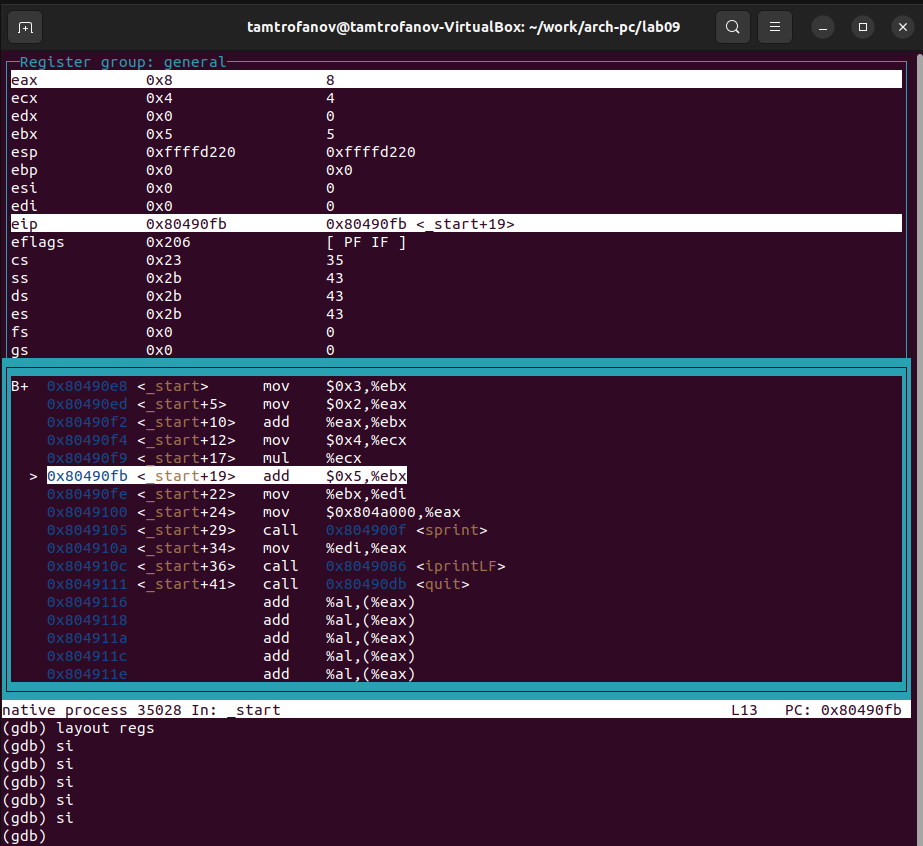


Содержимое файла *lab09-5.asm*



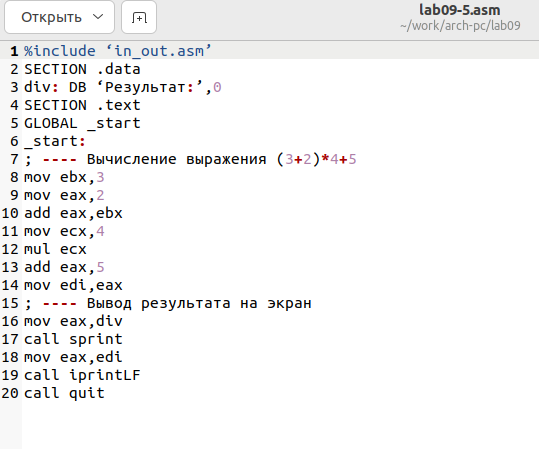
Создание файла *lab09-5* и его первоначальная отладка в GDB

Пошагово выполня программу нашел ошибку в использовании неправильного регистра при умножении (рис. ??).

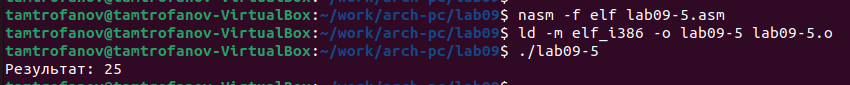


Пошаговое выполнение программы

Внёс необходимые изменения в файл *lab09-5.asm* (рис. ??). Создал исполнительный файл и проверил его работу (рис. ??).



Содержимое файла *lab09-5.asm*



Создание файла *lab09-5* и проверка его работы

***КОД ПРОГРАММЫЙ***

%include 'in\_out.asm'  
SECTION .data  
div: DB 'Результат: ',0  
SECTION .text  
GLOBAL \_start  
\_start:  
; ---- Вычисление выражения (3+2)\*4+5  
mov ebx,3  
mov eax,2  
add eax,ebx  
mov ecx,4  
mul ecx  
add eax,5  
mov edi,eax  
; ---- Вывод результата на экран  
mov eax,div  
call sprint  
mov eax,edi  
call iprintLF  
call quit

# 4 Выводы

Сегодня я приобрёл навыки написания программ с использованием подпрограмм. Познакомился с методами отладки при помощи GDB и его основными возможностями.