Отчёт по лабораторной работе №9

Дисциплина:архитектура компьютера

Логинов Георгий Евгеньевич

Содержание

# 1 Цель работы

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

# 2 Задание

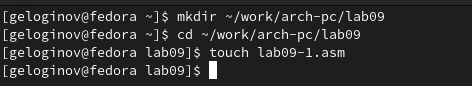
Реализация подпрограмм в NASM Отладка программам с помощью GDB Задание для самостоятельной работы

# 3 Теоретическое введение

Подпрограмма — поименованная или иным образом идентифицированная часть компьютерной программы, содержащая описание определённого набора действий.

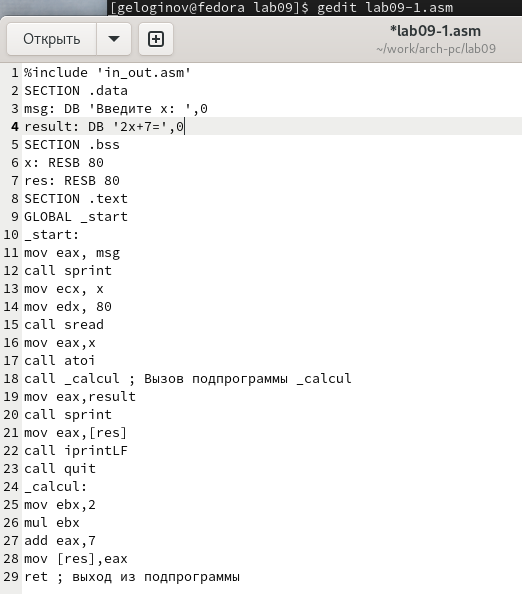
# 4 Выполнение лабораторной работы

##Реализация подпрограмм в NASM Создаю рабочую директорию и файл в ней. (рис. ??).



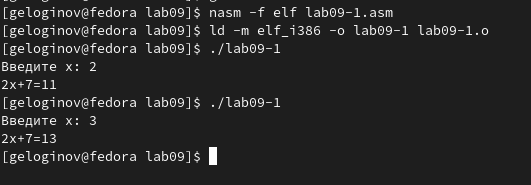
Создание файла и директории

Редактирю файл вводя текст листинга. (рис. ??).



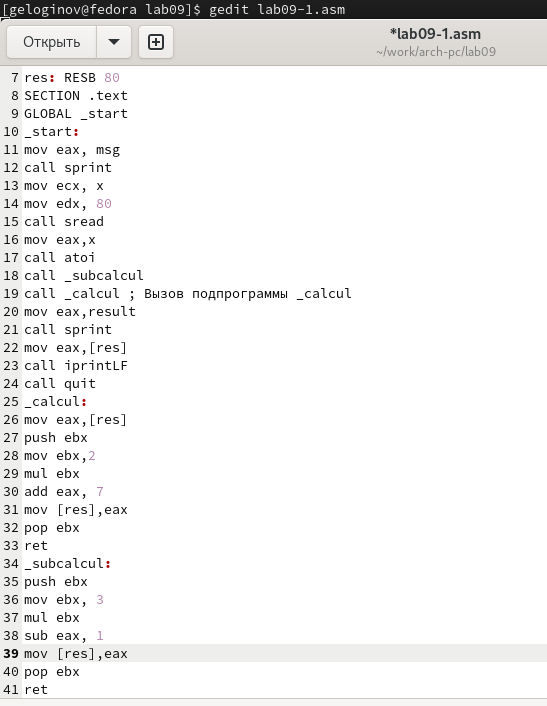
Редактирование файла

Создаю исполняемый файл. Проверяю корректность работы програмы (рис. ??). Программа отработала корректно.



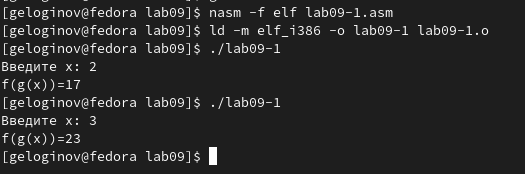
Запуск исполняемого файла

Изменяю текст программы чтобы она считала f(g(x)). (рис. ??).



Редактирование файла

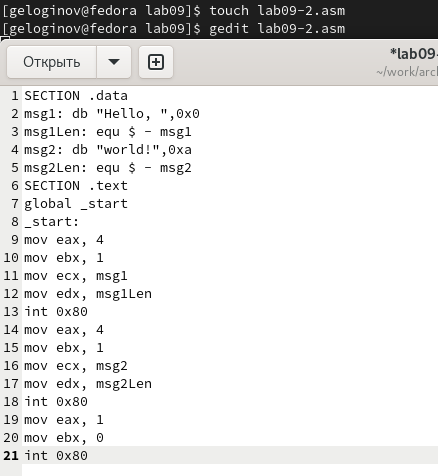
Создаю исполняемый файл. Проверяю корректность работы програмы (рис. ??). Программа отработала корректно.



Запуск исполняемого файла

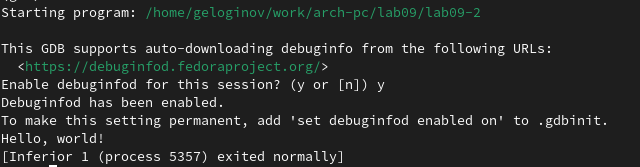
##Отладка программам с помощью GDB

Создаю файл. Ввожу в него текст листинга (рис. ??).



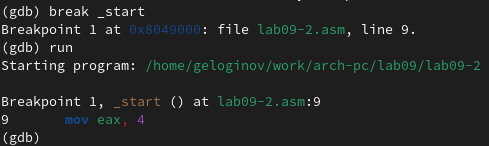
Создание и редактирование файла

Запускаю программу в отладочной оболочке GDB (рис. ??).



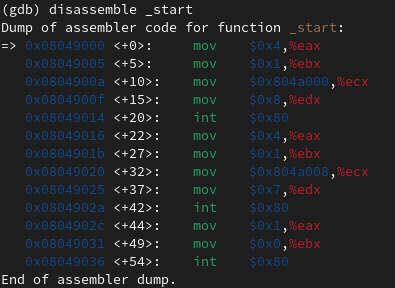
Запуск программы в отладочной оболочке

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



Запуск программы в отладочной оболочке

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



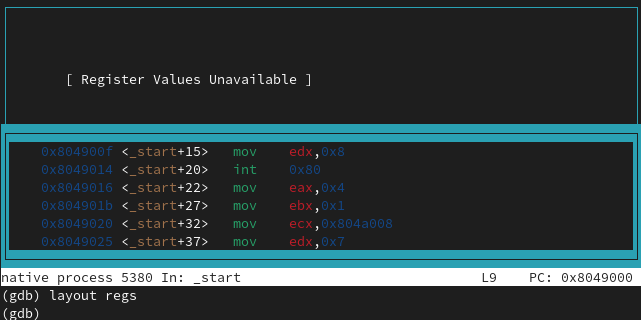
Просмотр дисассимилированного кода программы

Переключаюсь на отображение команд с Intel’овским синтаксисом, введя команду set disassembly-flavor intel (рис. ??). В представлении ATT в виде 16-ричного числа записаны первые аргументы всех комманд, а в представлении intel так записываются адреса вторых аргумантов.



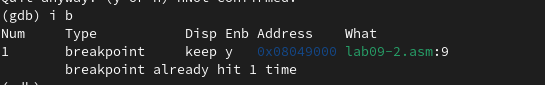
Изменение синтаксиса

Включаю режим псевдографики, с помощью которго отбражается код программы и содержимое регистров(рис. ??).



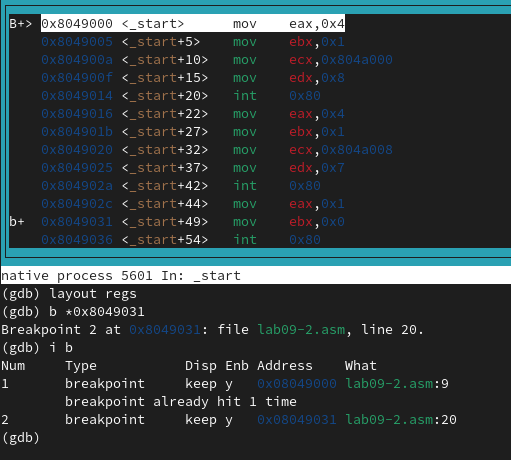
Включение псевдографики

Проверяю наличие точки останова.(рис. ??).



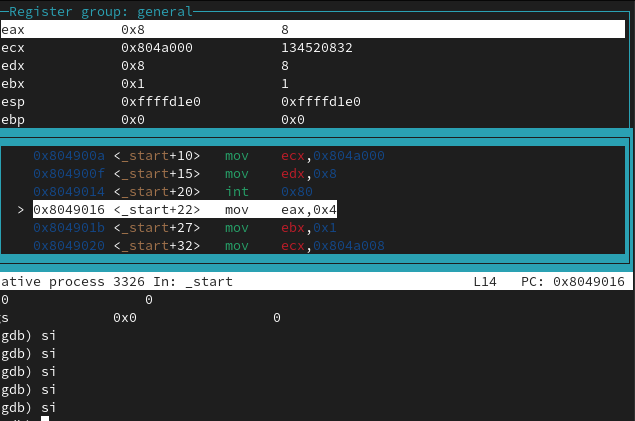
Проверка

Добавляю ещё одну точку остонова и проверяю сколько точек останова еть(рис. ??).



Добавление точки остова и проверка её наличия

Выполняю 5 инструкций с помощью команды si(рис. ??). Изменились значения регистров eax ecx edx ebx



Выполнение некоторых инструкций

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

Просмотр значения переменной

Просмотр значения переменной

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

Просмотр значения переменной

Просмотр значения переменной

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

Изменение символа в переменной

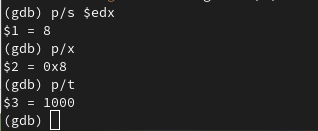
Изменение символа в переменной

Изменяю первый символ переменной msg2 (рис. ??).

Изменение символа в переменной

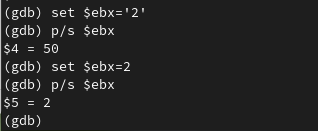
Изменение символа в переменной

Вывожу значение edx в разных форматах: строчном, 16-ричном, двоичном (рис. ??).



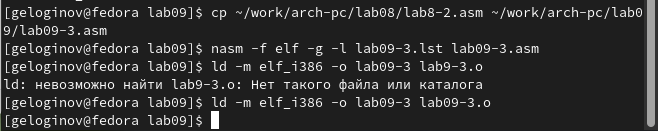
Вывод значения в разных форматах

С помощью команды set изменяю значение регистра ebx(рис. ??). Разница вывода из-за того что в первом случае 2 это символ а во втором число.



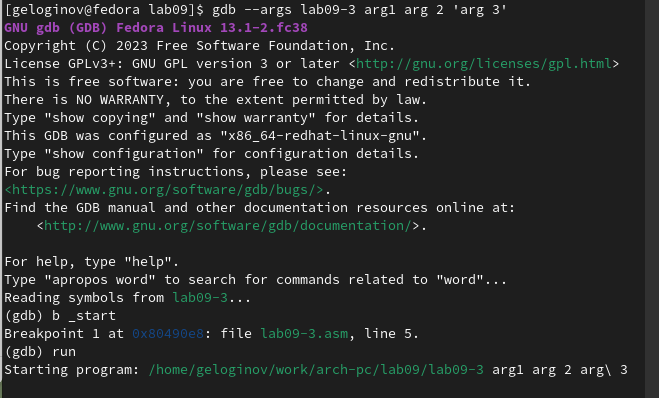
Изменение значения решистра

Скопировал файл lab8-2.asm, созданный при выполнении лабораторной работы No8, с программой выводящей на экран аргументы командной строки (Листинг 8.2) в файл с именем lab09-3.asm:(рис. ??). И создал исполняемый файл.



Копирование файла и создание исполняемого

Запускаю программу в оболочке GDB (рис. ??).



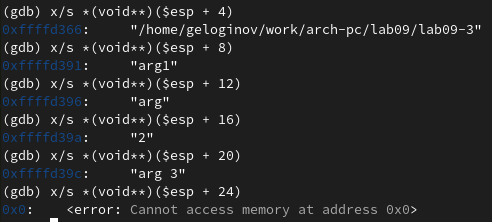
Запуск программы в оболочке отладки

Узнаю количество аргументов (рис. ??).

Количество аргументов 5

Количество аргументов 5

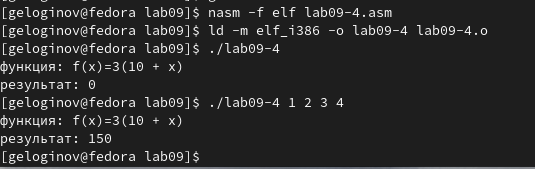
Смотрю все позиции стека (рис. ??). Их адреса распологаются в 4 байтах друг от друга(именно столько заниемает элемент стека)



Просмотр позиций стека

# 5 Задания для самостоятельной работы

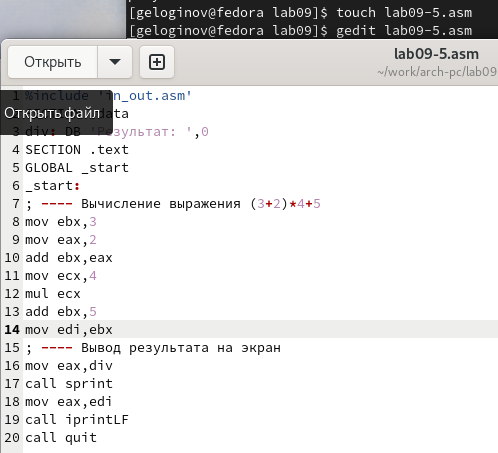
Работа файла из лабораторной работы 8, но с использованием подпрограмм (рис. ??).



Исполнение файла

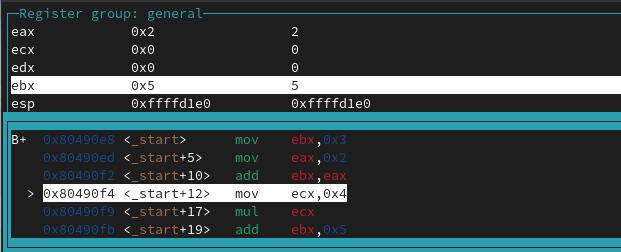
%include 'in\_out.asm'  
  
SECTION .data  
f\_x db "функция: f(x)=3(10 + x)",0h  
msg db 10,13,'результат: ',0h  
  
SECTION .text  
global \_start  
  
\_f:  
push ebx  
add eax, 10  
mov ebx, 3  
mul ebx  
pop ebx   
ret  
  
\_start:  
pop ecx  
pop edx  
sub ecx, 1  
mov esi, 0  
  
next:  
cmp ecx,0h  
jz \_end  
pop eax  
call atoi  
call \_f  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, f\_x  
call sprint  
mov eax, msg  
call sprint  
mov eax, esi  
call iprintLF  
  
call quit

Создаю и редактирую файл (рис. ??).



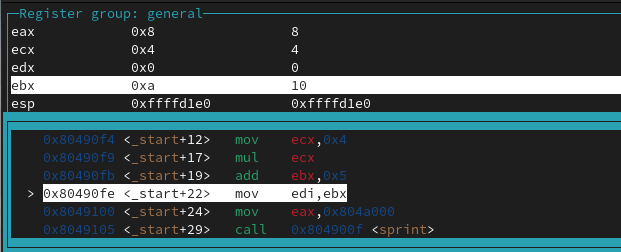
Создание и редактирование файла

Нахожу несостыковки при использовании отладчика(рис. ??).



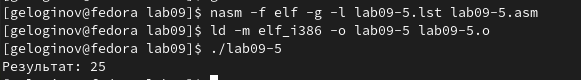
Несостыковка

Нахожу несостыковки при использовании отладчика(рис. ??).



Несостыковка

Раюота исправленного файла(рис. ??).



Исполнение файла

Ошибка была в сторках

add ebx,eax  
mov ecx,4  
mul ecx  
add ebx,5  
mov edi,ebx

Исправленный код

%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

# 6 Выводы

В результате выполнения работы, я научился организовывать код в подпрограммы и познакомился с базовыми функциями отладчика gdb.

# Список литературы

[Лабораторная работа 9](https://esystem.rudn.ru/pluginfile.php/2089551/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%969.%20%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B.%20%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA%20..pdf)