Отчёт по лабораторной работе №9

Архитектура компьютера

Морозова Мария Вячеславовна

Содержание

# 1 Цель работы

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

# 2 Задание

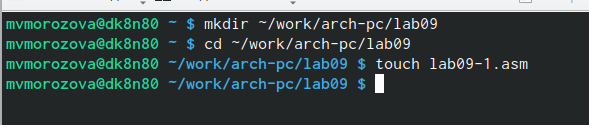
Преобразовать программу из лабораторной работы N8 , реализовав вычисление значения функции f(x) как подпрограмму. С помощью отладчика GDB, анализируя изменения значений регистров, определить ошибку и исправить ее.

# 3 Теоретическое введение

Отладка — это процесс поиска и исправления ошибок в программе. В общем случае его можно разделить на четыре этапа: • обнаружение ошибки; • поиск её местонахождения; • определение причины ошибки; • исправление ошибки. Можно выделить следующие типы ошибок: • синтаксические ошибки — обнаруживаются во время трансляции исходного кода и вызваны нарушением ожидаемой формы или структуры языка; • семантические ошибки — являются логическими и приводят к тому, что программа запускается, отрабатывает, но не даёт желаемого результата; • ошибки в процессе выполнения — не обнаруживаются при трансляции и вызывают пре- рывание выполнения программы (например, это ошибки, связанные с переполнением или делением на ноль). Второй этап — поиск местонахождения ошибки. Некоторые ошибки обнаружить доволь- но трудно. Лучший способ найти место в программе, где находится ошибка, это разбить программу на части и произвести их отладку отдельно друг от друга. Третий этап — выяснение причины ошибки. После определения местонахождения ошибки обычно проще определить причину неправильной работы программы. Последний этап — исправление ошибки. После этого при повторном запуске программы, может обнаружиться следующая ошибка, и процесс отладки начнётся заново.

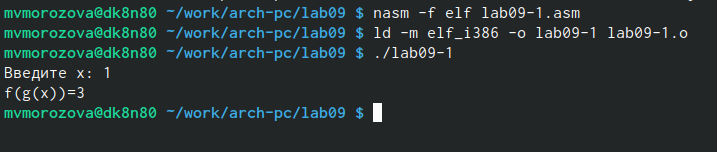
# 4 Выполнение лабораторной работы

Создала каталог для выполнения лабораторной работы No 9, перешла в него и со- здала файл lab09-1.asm: (рис. ??).



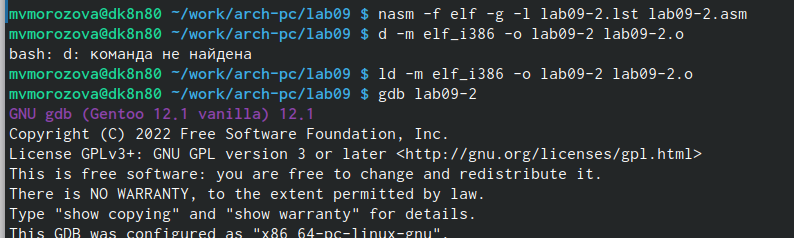
Создание файла

Запустила файл с изменённым текстом листинга 9.1. (рис. ??).



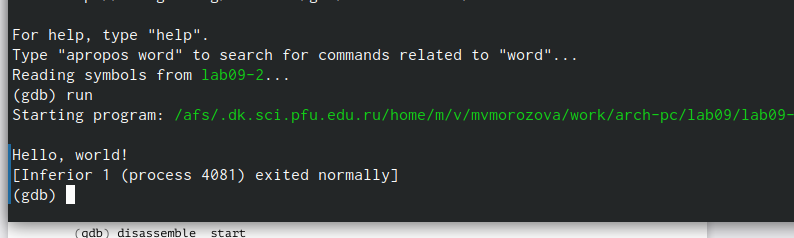
Результат

Провела трансляцию программ с ключом -g, загрузила файл в отладчик. (рис. ??).



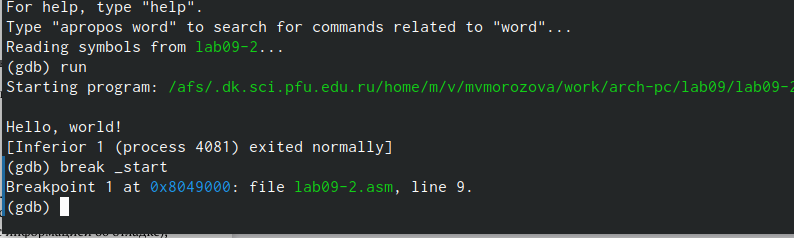
Открываем отладчик

Проверила работу программы, запустив ее в оболочке GDB с помощью команды run: (рис. ??).



Запуск программы

Для более подробного анализа программы установила брейкпоинт на метку \_start, запустила программу. (рис. ??).



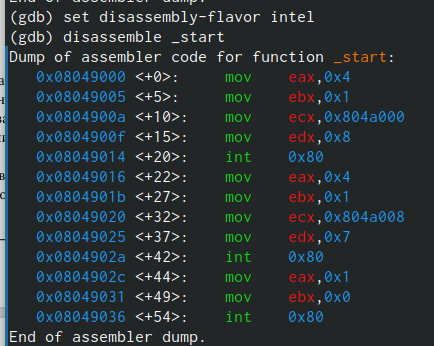
Запуск программы с брейкпоинтом

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



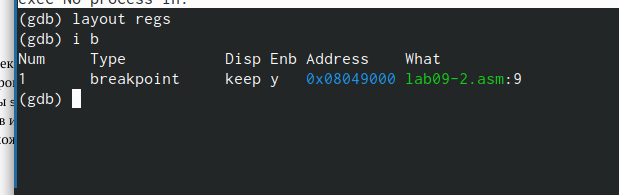
Дисассимилированный код программы

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



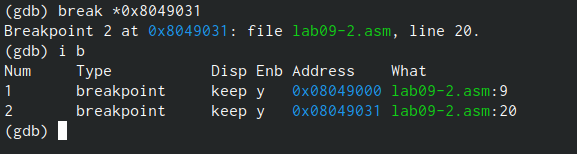
Отображение команд с Intel’овским синтаксисом

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



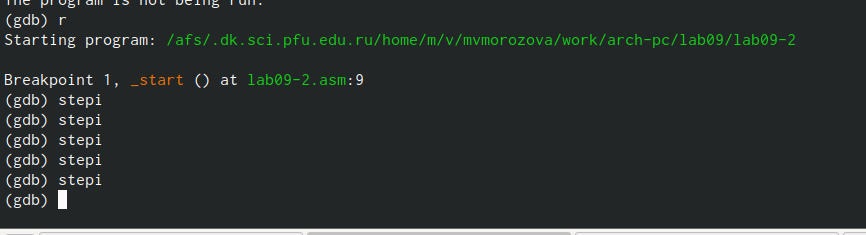
Переход в режим псевдографики

На предыдущих шагах была установлена точка останова по имени метки (\_start). Проверила это с помощью команды info breakpoints (кратко i b): (рис. ??).



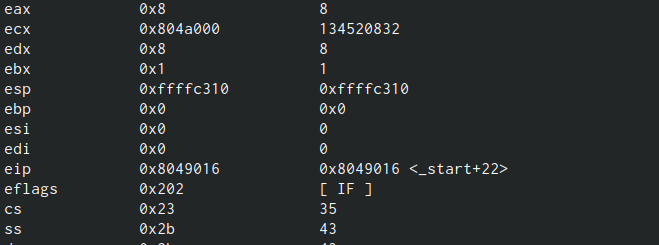
Команда i b

Выполнила 5 инструкций с помощью команды stepi и проследила за изменением значений регистров. (рис. ??).



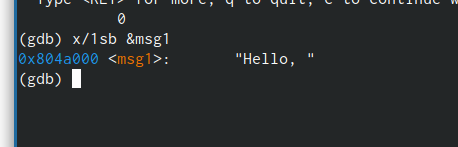
Команда stepi

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



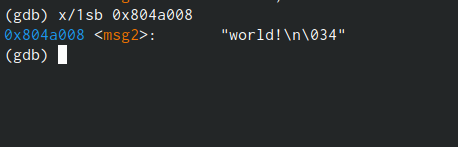
Команда i r

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



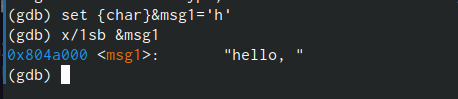
Значение переменной msg1

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



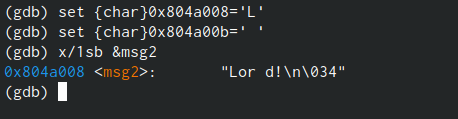
Значение переменной msg2

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



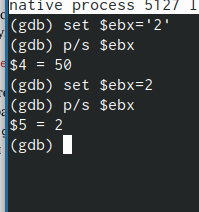
Изменение msg1

Заменила символ во второй переменной msg2. (рис. ??).



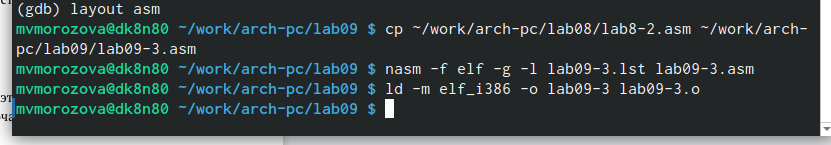
Изменение msg2

С помощью команды set изменила значение регистра ebx:(рис. ??).



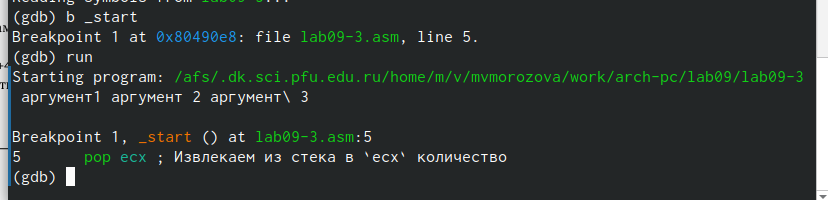
Команда set

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



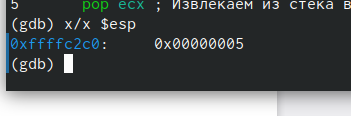
Создание файла lab09-3

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



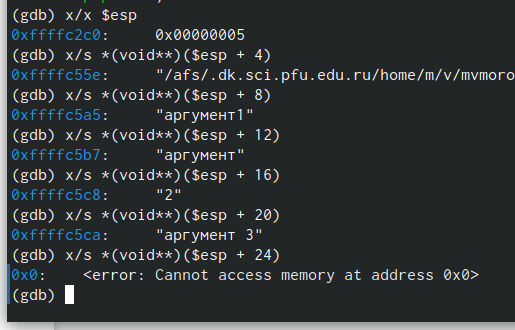
Загрузка файла

Адрес вершины стека хранится в регистре esp и по этому адресу располагается число равное количеству аргументов командной строки (включая имя программы): (рис. ??).



Адрес

Посмотрела остальные позиции стека (рис. ??).



Остальные позиции стека

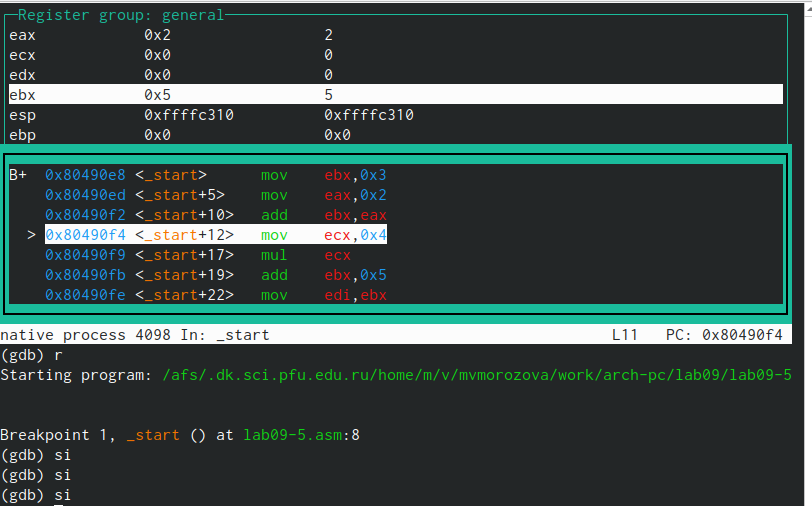
# 5 Выполнение самостоятельной работы

Преобразовала программу из лабораторной работы No8, реализовав вычисление значения функции f(x) как подпрограмму. (рис. ??).



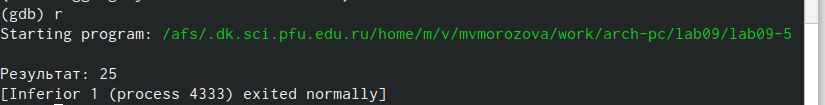
Запуск программы, результат работы программы

С помощью отладчика нашли ошибки в программе. (рис. ??).



Ошибки

Проверила работу программы. (рис. ??).



Результат работы программы

# 6 Выводы

Были приобретены навыки написания программ с использованием подпрограмм.

# 7 Листинги

%include 'in\_out.asm'  
SECTION .data  
f\_x db "функция: 3(x+2)",0h  
msg db 10,13,'результат: ',0h  
SECTION .text  
global \_start  
  
\_f:  
push ebx  
dec eax  
mov ebx, 10  
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 eax,2  
mov ebx,3  
mul ebx  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, f\_x  
call sprint  
mov eax, msg  
call sprint  
mov eax, esi  
call iprintLF  
  
call quit  
  
  
  
%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