Отчёт по лабораторной работе №9

Простейший вариант

Бадалов Заури Эльвин оглы

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить работу с подпрограммами и отладчиком gdb.

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

Создаю рабочую директорию и файл. Записываю туда программу из листинга, исправив опечатки. (рис. 3.1).

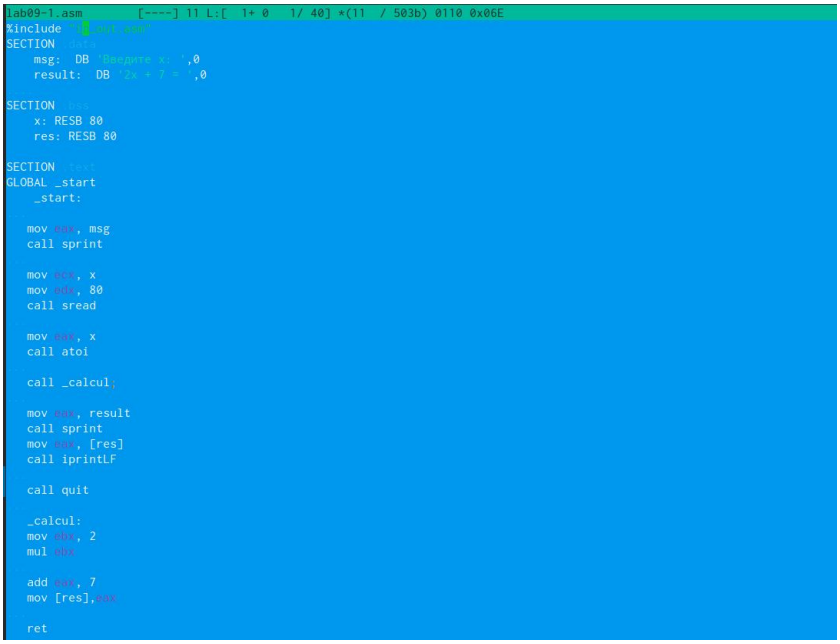

Рис. 3.1: Программа f(g(x))

Проверяю работу программы. (рис. 3.2).

Рис. 3.2: Запуск программы

Вношу текст программы, имитирующей сложную функцию. Функции назову _calcul и _subcalcul (рис. 3.3).

Рис. 3.3: Программа с имитацией сложной функции

Проверяю работу программы. (рис. 3.4).

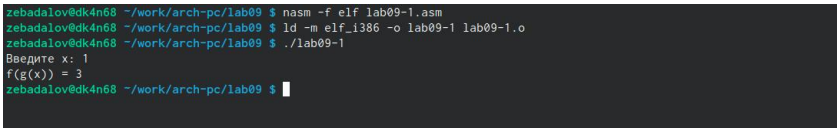

Рис. 3.4: Запуск программы

Создаю файл lab09-2.asm и ввожу в него текст программы. (рис. 3.5).

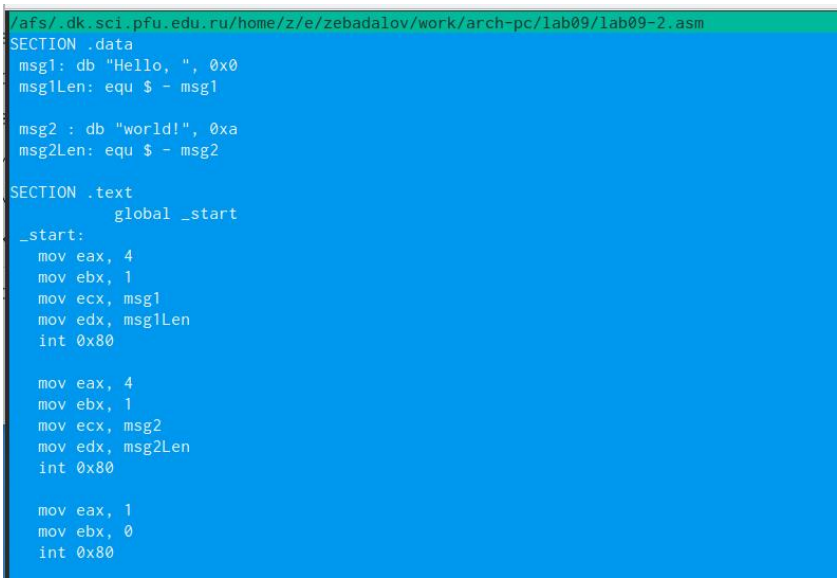

Рис. 3.5: Текст программы файла lab09-2.asm

Транслирую файл с ключом -g. После загружаю исполняемый файл в отладчик gdb. Проверяю его работу запустив ее в отладчике gdb командой run. (рис. 3.6).

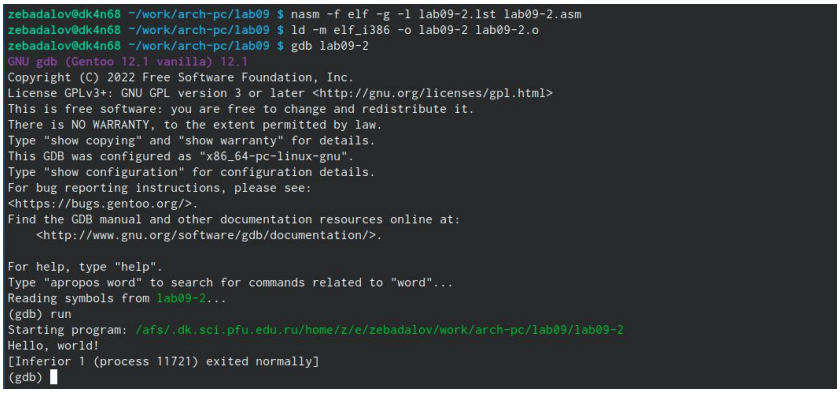

Рис. 3.6: Работа файла в отладчике gdb

Устанавливаю брейкпоинт на метку _start и запускаю ассемблерную программу. (рис. 3.7).

Рис. 3.7: Брейкпоинт на метку _start

Смотрю дисассимилированный код программы, использую команду disassemble, начиная с метки _start. (рис. 3.8).

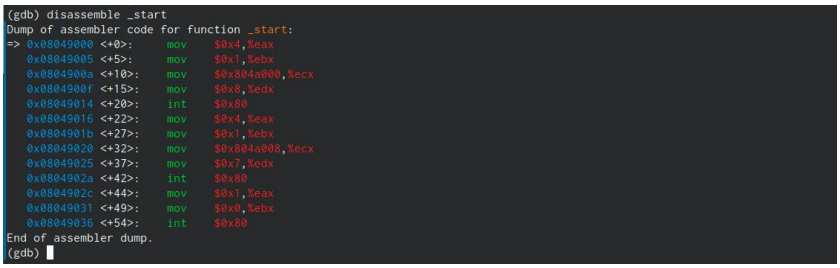

Рис. 3.8: Дисассимилированный код программы

Переключаюсь на отображение команд с Intel’овским синтаксисом. (рис. 3.9).

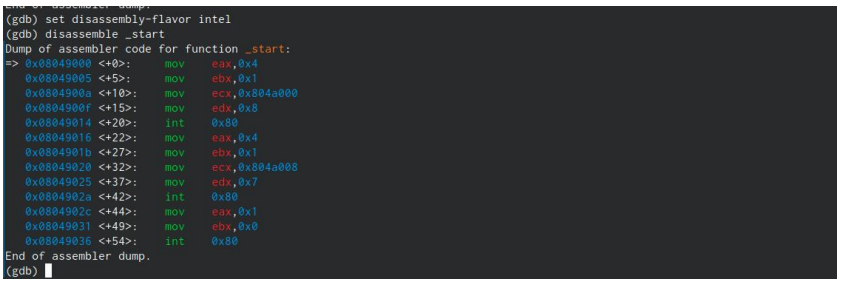

Рис. 3.9: Отображение команд с Intel'овским синтаксисом

В представлении ATT в виде 16-ричного числа записаны первые аргументы всех комманд, а в представлении Intel так записываются адреса вторых аргументов. Включим режим псевдографики, при помощи которого отображается код программы и содержимое регистров. (рис. 3.10).

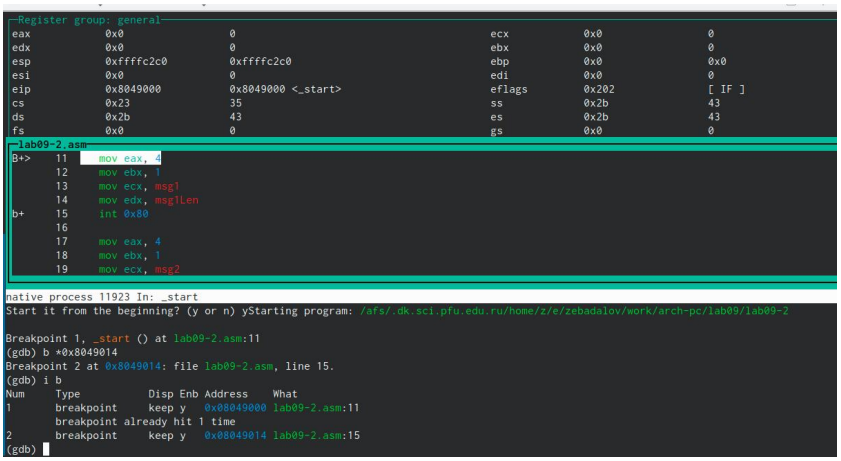

Рис. 3.10: Режим псевдографики

Вывожу значения регистров, используя команду i r. (рис. 3.11).

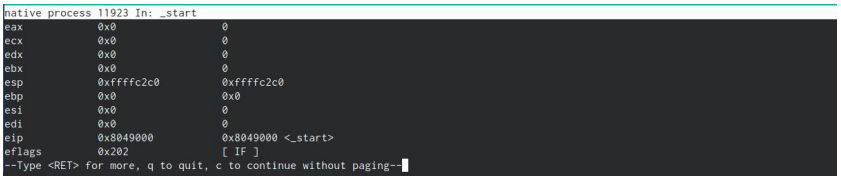

Рис. 3.11: Значения регистров на псевдографике

В отладчике можно вывести текущие значения переменных. Сделать это можно, к примеру, по имени (рис. 3.12) или по адресу (рис. 3.13).

Рис. 3.12: Получение значения переменной по имени

Рис. 3.13: Получение значения переменной по адресу

Так же отладчик даёт менять значения переменных прямо во время выполнения программы. (рис. 3.14).

Рис. 3.14: Замена значение переменной msg1

Заменю первый символ переменной msg2 на символ отступа. (рис. 3.15).

Рис. 3.15: Замена символа переменной msg2

Выведу значение регистра edx в разных форматах: строчном, 16-ричном, двоичном. (рис. 3.16).

Рис. 3.16: Вывод значения edx

Регистрам тоже можно задавать значения. (рис. 3.17).

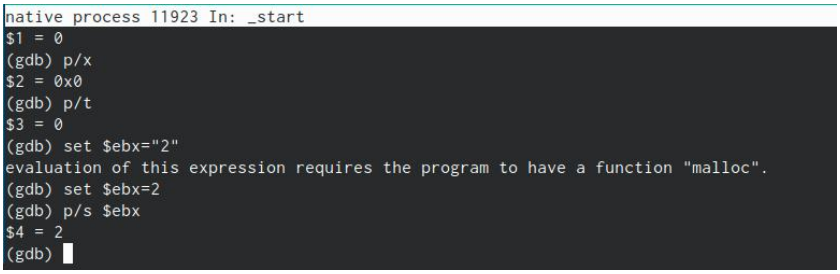

Рис. 3.17: Присвоение значения регистру

Однако при попытке задать строчное значение, происходит ошибка. Завершу работу в gdb командами continue, она закончит выполнение программы, и exit, она завершит сеанс gdb.

Скопирую файл из лабораторной работы №9, переименую и создаю исполняемый файл. Открываю отладчик и задаю аргументы. Создаю точку останова на метке _start и запускаю программу. (рис. 3.18).

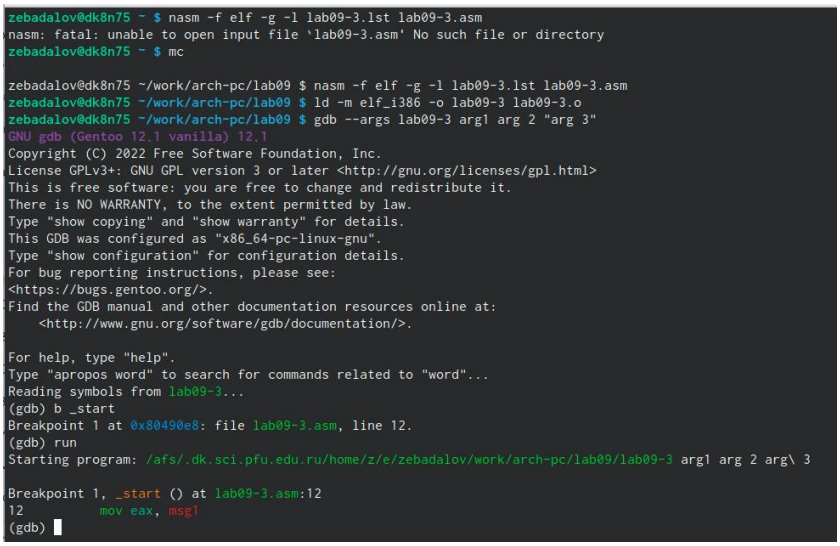

Рис. 3.18: Запуск программы lab09-3.asm посредством gdb

Смотрю на содержимое того, что расположено по адресу, находящемуся в регисре esp. (рис. 3.19).

Рис. 3.19: Содержимое регистра esp

Далее смотрю на все остальные аргументы в стеке. Их адреса располагаются в 4 байтах друг от друга (именно столько занимает элемент стека). (рис. 3.20).

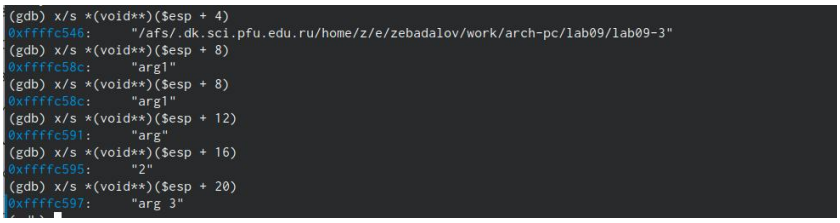

Рис. 3.20: Остальные аргументы в стеке

#Задания самостоятельной работы

Программа из лабораторной работы №8 с использованием подпрограмм. (рис. 3.21).

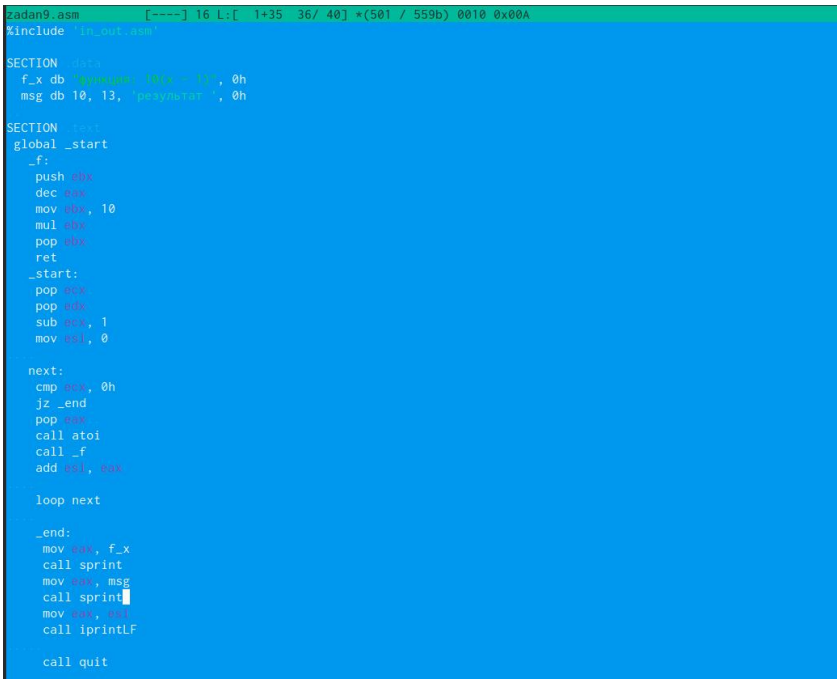

Рис. 3.21: Программа файла zadan9

Проверяю результат выполнения работы программы. (рис. 3.22).

Рис. 3.22: Запуск программы

Редактирую текст программы из листинга 9.3 в новом файле zadan9-1.asm после просмотра регистров для поиска ошибки. (рис. 3.23).

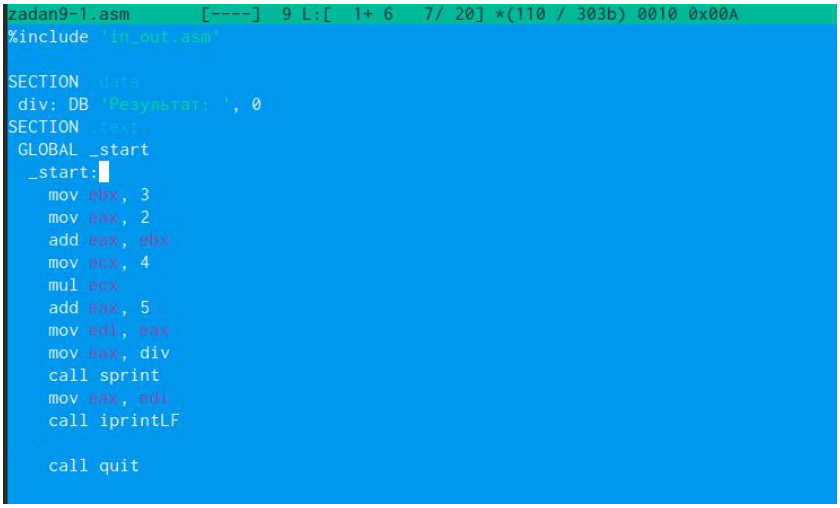

Рис. 3.23: Исправленный текст программы

Проверяю работу программы. (рис. 3.24).

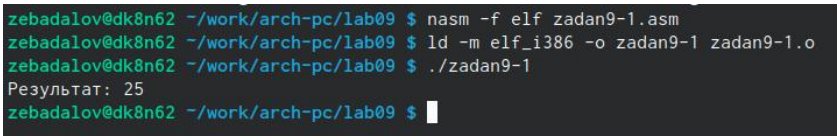

Рис. 3.24: Запуск программы

4 Вывод

Я освоил работу с подпрограммами и отладчиком gdb.