Отчёт по лабораторной работе № 9

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить работу с подпрограммами и отладчиком gdb.

2 Выполнение лабораторной работы

Создаю необходимый каталог и файлы (рис. 2.1).

```
[evmusatova@fedora ~]$ mkdir ~/work/arch-pc/lab09
[evmusatova@fedora ~]$ cd ~/work/arch-pc/lab09
[evmusatova@fedora lab09]$ touch lab09-1.asm
[evmusatova@fedora lab09]$
```

Рис. 2.1: Создание файлов и каталога

Ввожу в файл lab09-1.asm текст программы из листинга 9.1. Создаю исполняемый файл и проверяю его работу (рис. 2.2).

```
[evmusatova@fedora lab09]$ nasm -f elf lab09-1.asm
[evmusatova@fedora lab09]$ ld -m elf_i386 lab09-1.o -o lab09-1
[evmusatova@fedora lab09]$ ./lab09-1
Введите x: 5
2x+7=17
[evmusatova@fedora lab09]$
```

Рис. 2.2: Проверка программы

Добавляю подпрограмму _subcalcul в подпрограмму _calcul, для вычисления выражения ☒(☒(☒)), (рис. 2.3).

```
37 imul ebx
38
39 add eax, 7
40
41 pop eax
42 ret
43
44 _subcalcul:
45 push eax
46 mov ebx, 3
47 imul ebx
48 dec ebx
49 mov [res],eax
50
51 pop eax
52 ret
```

Рис. 2.3: Изменение программы

Создаю исполняемый файл и проверяю работу программы (рис. 2.4).

```
[evmusatova@fedora lab09]$ nasm -f elf lab09-1.asm
[evmusatova@fedora lab09]$ ld -m elf_i386 lab09-1.o -o lab09-1
[evmusatova@fedora lab09]$ ./lab09-1
Введите x: 1
f(g(x))=3
[evmusatova@fedora lab09]$
```

Рис. 2.4: Проверка

Создаю файл lab09-2.asm с текстом программы из Листинга 9.2. Затем получаю исполняемый файл и загружаю его в отладчик gdb (рис. 2.5).

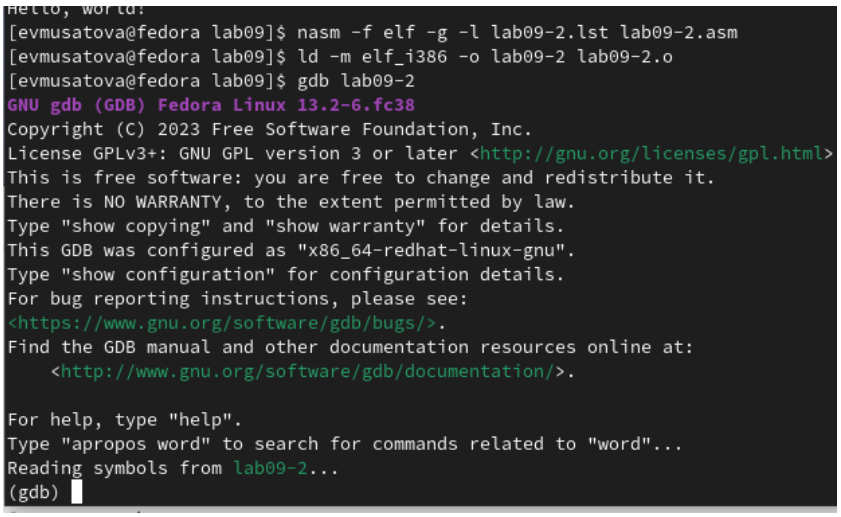

Рис. 2.5: Загрузка в отладчик

Запускакю программу в оболочке GDB с помощью команды run (рис. 2.6).

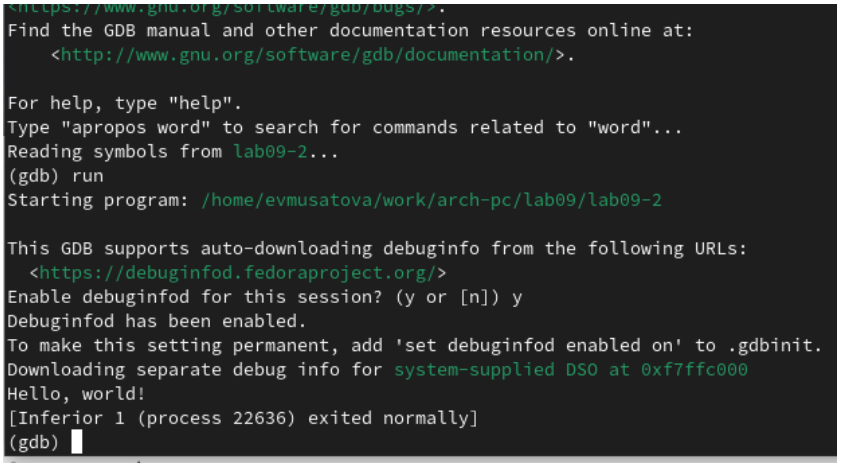

Рис. 2.6: Запуск программы

Для более подробного анализа программы устанавливаю брейкпоинт на метку _start, с которой начинается выполнение любой ассемблерной программы, и запускаю её (рис. 2.7).

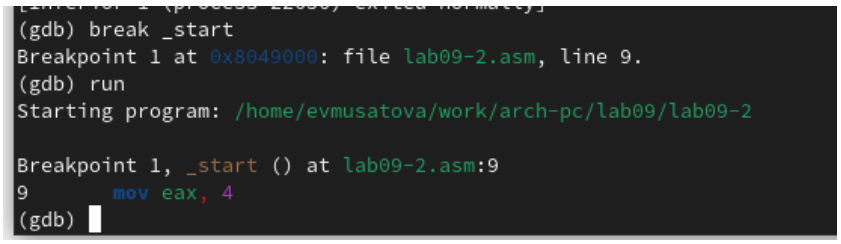

Рис. 2.7: Установка брейкпоинта

Смотрю дисассимилированный код программы с помощью команды disassemble (рис. 2.8).

Рис. 2.8: Просмотр кода

Переключаюсь на отображение команд с Intel'овским синтаксисом (рис. 2.9). В представлении ATT в виде 16-ричного числа записаны первые аргументы всех комманд, а в представлении intel так записываются адреса вторых аргумантов.

Рис. 2.9: Переключение

Включаю режим псевдографики (рис. 2.10).

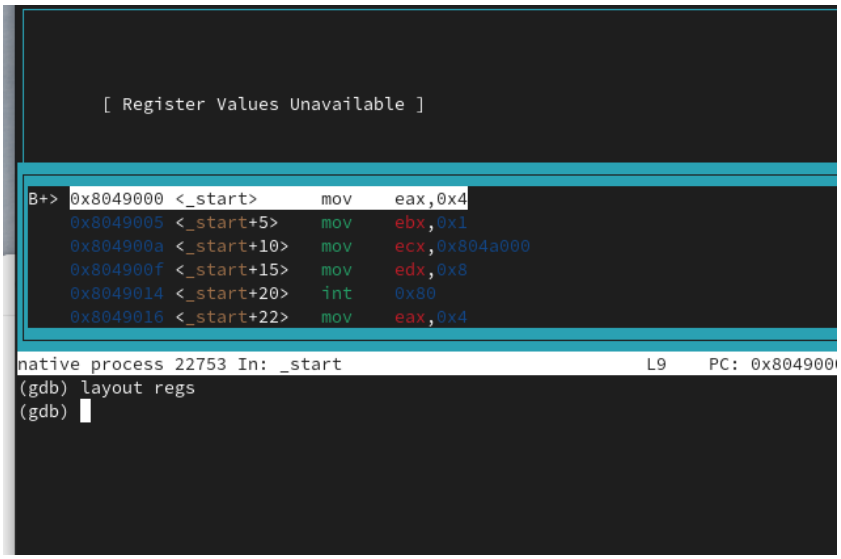

Рис. 2.10: Псевдографика

Проверяю что была установлена точка основа. Затем устанавливаю еще одну точку основа по адресу инструкции (рис. 2.11).

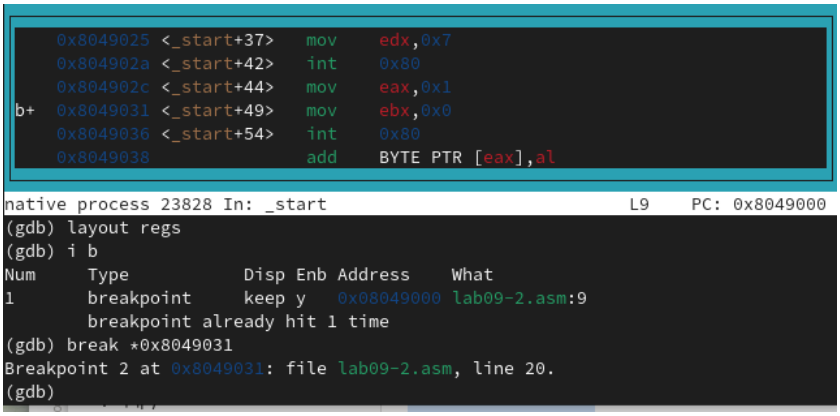

Рис. 2.11: Проверка

Смотрю информацию о всех установленных точках основа (рис. 2.12).

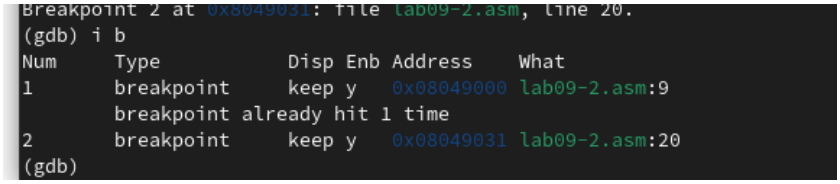

Рис. 2.12: проверка

Выполняю 5 инструкций (рис. 2.13).

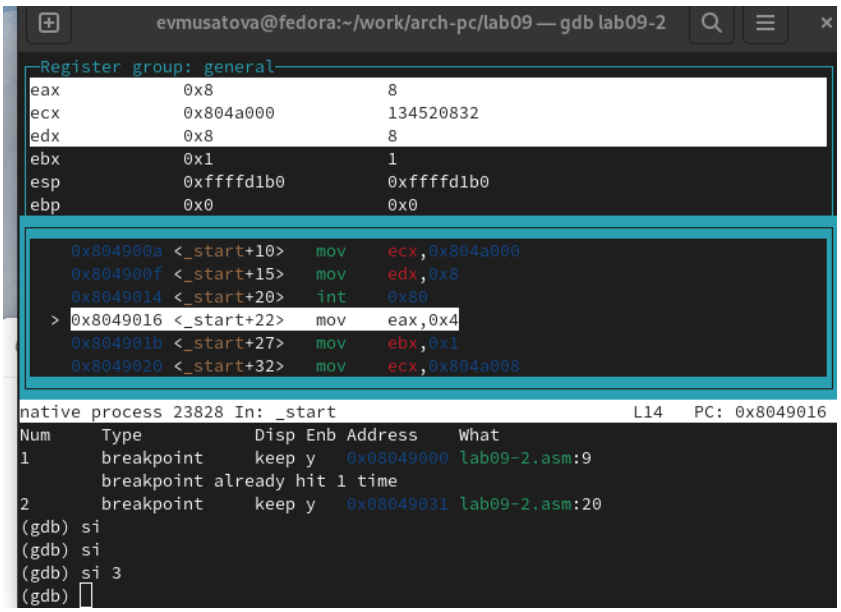

Рис. 2.13: Инструкции

Смотрю значение переменной по имени (рис. 2.14).

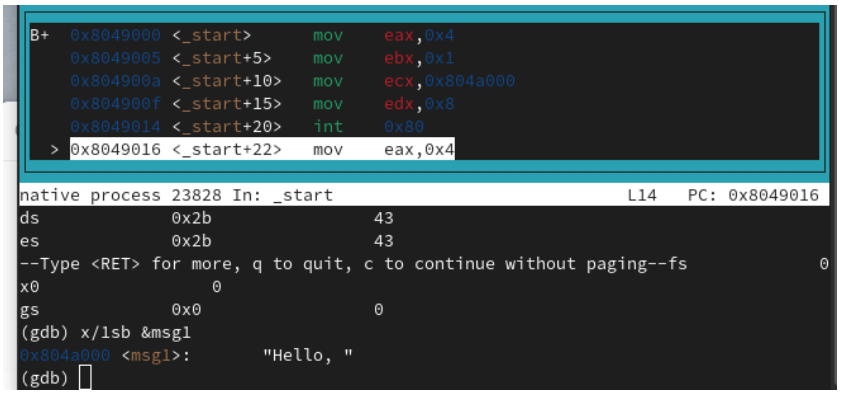

Рис. 2.14: Просмотр

Теперь смотрю значение переменной по адресу (рис. 2.15).

Рис. 2.15: Другой способ посмотреть значение переменной

Изменяю первый символ переменной msg1 (рис. 2.16).

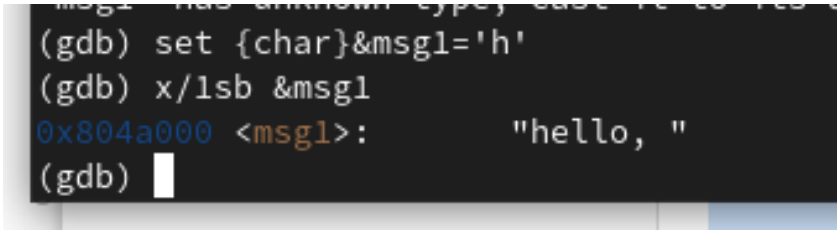

Рис. 2.16: Изменение символа

Теперь заменяю символ во второй переменной (рис. 2.17).

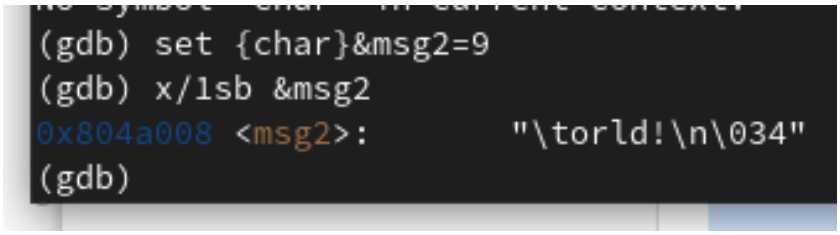

Рис. 2.17: Замена символа в другой прерменной

С помощью команды set изменяю значение регистра ebx (рис. 2.18). Однако при попытке задать строчное значение, происходит ошибка.

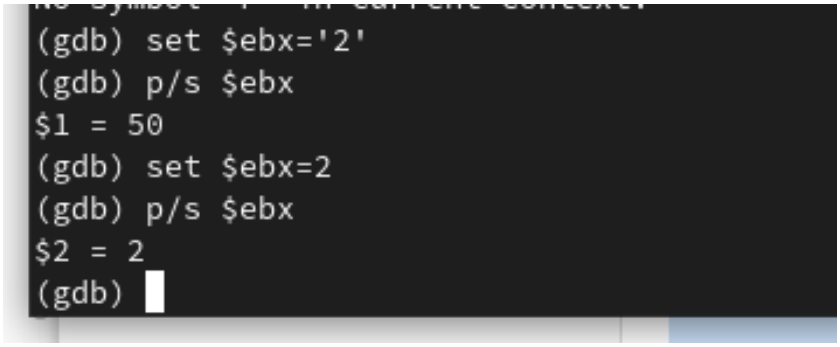

Рис. 2.18: Изменение значения регистра

Завершаю выполнение программы (рис. 2.19).

Рис. 2.19: Завершение

Копирую файл lab8-2.asm, созданный при выполнении лабораторной работы №8, создаю исполняемый файл и загружаю его в отладчик, указав аргументы (рис. 2.20).

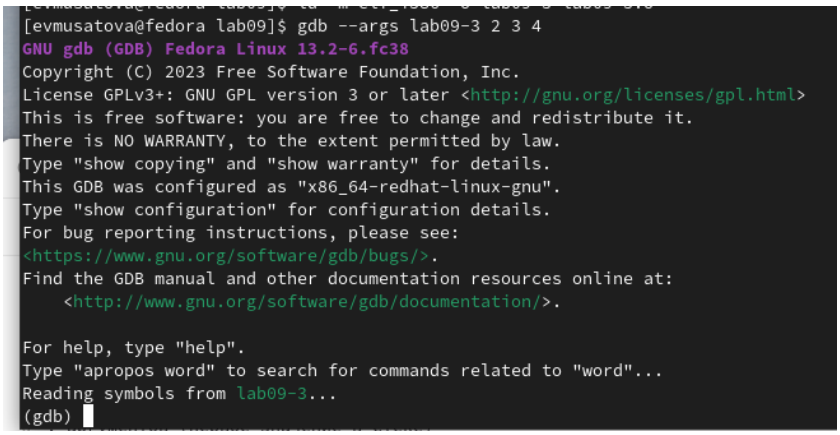

Рис. 2.20: Загрузка в отладчик

Для начала установим точку останова перед первой инструкцией в программе и запустим её (рис. 2.21).

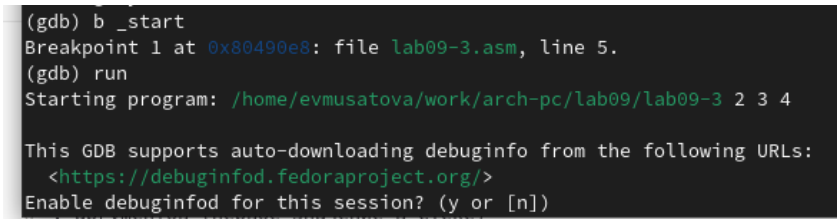

Рис. 2.21: Точка основа и запуск

Смотрим количество аргументов (рис. 2.22).

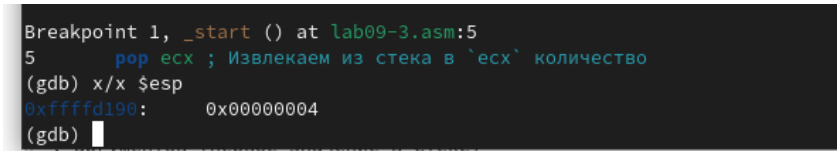

Рис. 2.22: Проверка

Смотрю остальные позиции стека (рис. 2.23). Их адреса распологаются в 4 байтах друг от друга(именно столько заниемает элемент стека.

Рис. 2.23: Просмотр

3 Выводы

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