Лабораторная работа №9

Архитектура компьютера

Иванов Александр Олегович

Содержание

# 1 Цель работы

Приобрести навыки написания программ с использованием подпрограмм. Знакомство с методами откладки при помощи GDB и его основными возможностями

# 2 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.

Подпрограмма — это, как правило, функционально законченный участок кода, который можно многократно вызывать из разных мест программы. В отличие от простых переходов из подпрограмм существует возврат на команду, следующую за вызовом. Если в программе встречается одинаковый участок кода, его можно оформить в виде подпрограммы, а во всех нужных местах поставить её вызов. При этом подпрограмма бу- дет содержаться в коде в одном экземпляре, что позволит уменьшить размер кода всей программы

# 3 Выполнение лабораторной работы

Записываю в lab09-1.asm программу из листинга,при этом добавив вторую подпрограмму для подсчета сложной функции. (рис. [1](#fig:001)).

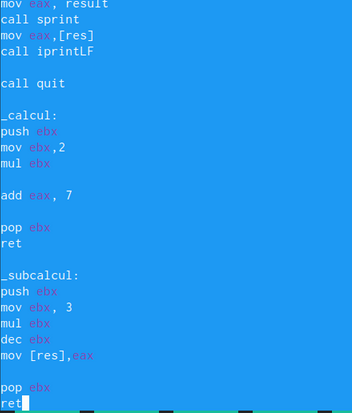


Figure 1: Окно MidnightCommander

Проверка вывода пргораммы (рис. [2](#fig:002)).

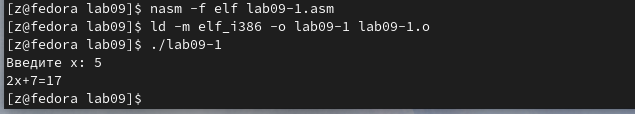


Figure 2: запуск lab09-1

Создаю файл lab9-2.asm и проассемблируем его с другими ключами, чтобы была возможность открыть этот файл через gdb (рис. [3](#fig:003)).

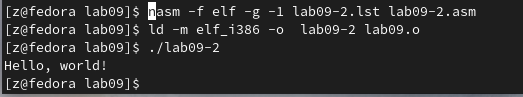


Figure 3: запуск lab09-2

Открываю файл lab9-2 с помощью gdb (рис. [4](#fig:004)).

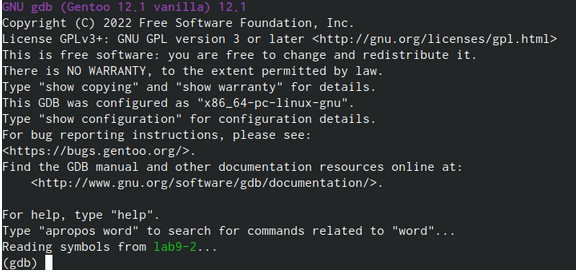


Figure 4: запуск lab09-2

Поставим точку остановки на метке \_start (рис. [5](#fig:005)).

Figure 5: окно gdb

Figure 5: окно gdb

В представлении АТТ в виде 16-ричного числа записаны первые аргументы всех комманд, а в представлении intel так записываются адреса вторых аргумантов. Включим режим псевдографики, с помощью которго отбражается код программы и содержимое регистров(рис. [6](#fig:006)).

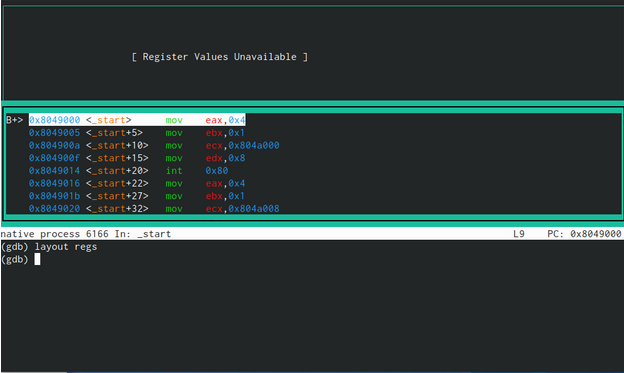


Figure 6: окно gdb

Поставим точку остановки на метке \_start (рис. [7](#fig:007)).

Figure 7: окно gdb

Figure 7: окно gdb

Установим еще одну точку остановк (рис. [8](#fig:008)).

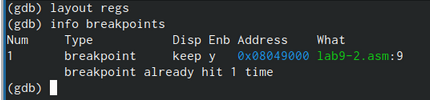


Figure 8: окно gdb

Посмотрим информацию о наших точках останова, используя команду i b (рис. [9](#fig:009)).

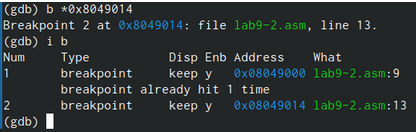


Figure 9: окно gdb

В откладчике можно вывести текущее значение переменных (рис. [10](#fig:010)).

Figure 10: окно gdb

Figure 10: окно gdb

Также можно обращаться по адресам переменных. Здесь был заменен первый символ переменной msg2 на символ отступа (рис. [11](#fig:011)).

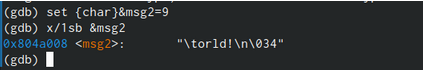


Figure 11: окно gdb

Зададю регистру значения (рис. [12](#fig:012)).



Figure 12: окно gdb

Копирую файл из предыдущей лабораторной работы, переименовываю и создаю исполняемый файл. Создаю точку останова на метке \_start и запускаю программу. (рис. [13](#fig:013)).

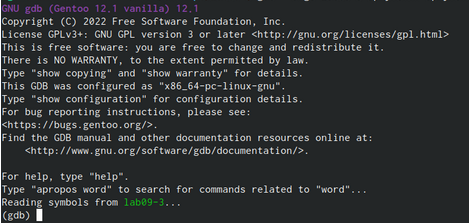


Figure 13: окно gdb

Заметим что, все остальные адреса аргументов располагаются в 4 байтах друг от друга. (рис. [14](#fig:014)).

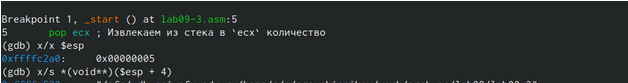


Figure 14: окно gdb

# 4 Самостоятельная работа

Программа из лабороторной 8, но с использованием подпрограмм. (рис. [15](#fig:015)).

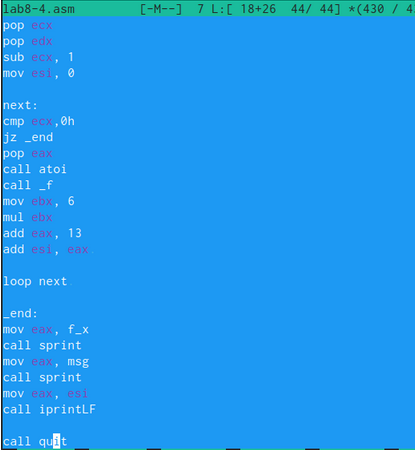


Figure 15: окно Midnightcommander

Проверка вывода программы. (рис. [16](#fig:016)).

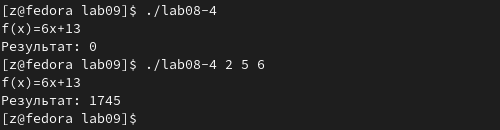


Figure 16: запуск lab08-4

# 5 Выводы

В ходе выполнения лабораторной работы я приобрел навыки написания программ с использованием подпрограмм. Я ознакомился с методами откладки при помощи GDB и его основными возможностями. # Список литературы{.unnumbered}