Отчёт по лабораторной работе №9

Архитектура компьютера

Алехин Давид Андреевич

Содержание

# 1 Цель работы

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

# 2 Задание

# 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

Создаю lab09/lab9-1.asm. (рис. 1).

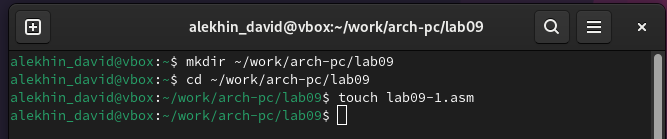


Рис. 1: lab09/lab9-1.asm

Вставляю туда Листинг 9.1, компаную и запускаю. (рис. 2).

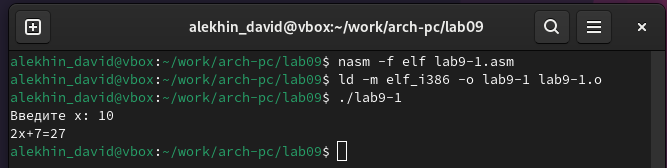


Рис. 2: Листинг 9.1

Изменяю текст программы, чтобы она решала выражение f(g(x)) и запускаю. (рис. 3), (рис. 4).

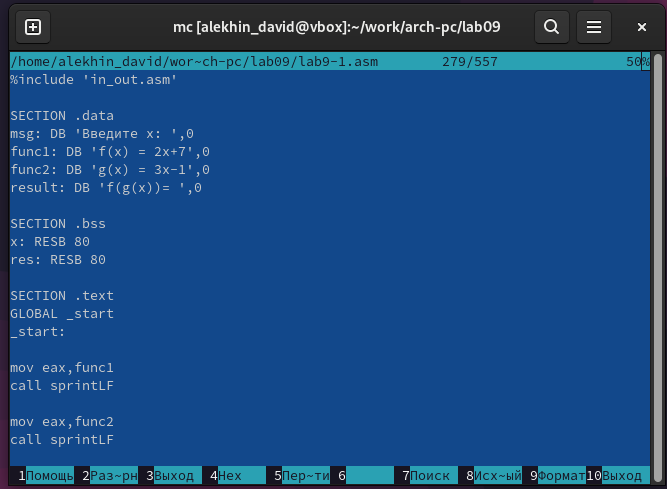


Рис. 3: Изменённый 9.1

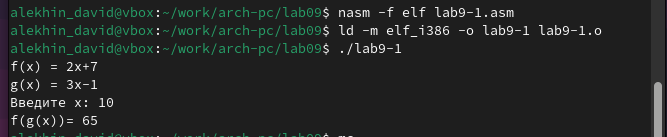


Рис. 4: ./9.1 изменённый

Создаю файл вписываю туда текст Листинг 9.2, компаную и запускаю через отладчик. (рис. 5).

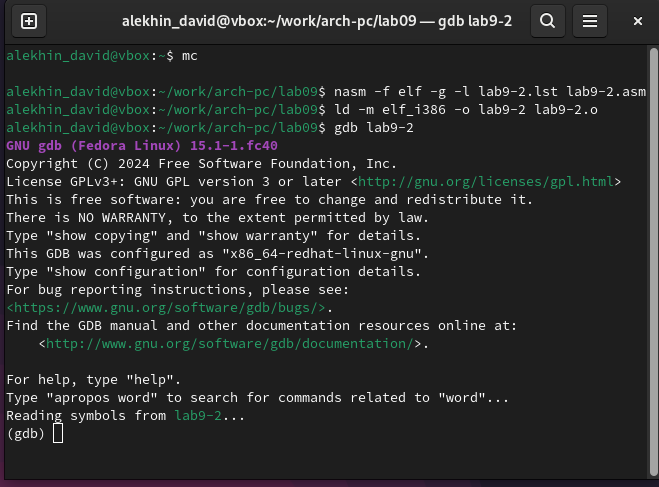


Рис. 5: Отладка 9.2

Запускаю 9.2 через отладчик. (рис. 6).

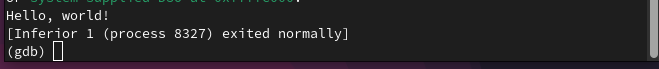


Рис. 6: ./9.2 через отладчик

Ставлю брекпоинт на метку \_start и запустил программу. (рис. 7).

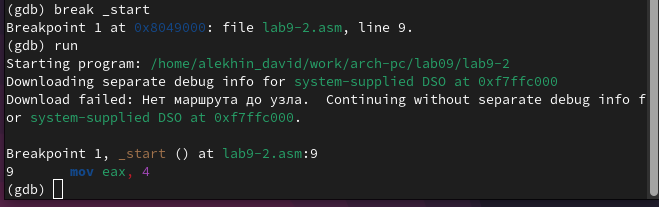


Рис. 7: Брекпоинт на метку \_start

Смотрю дисассимплированный код программы начиная с метки. (рис. 8).

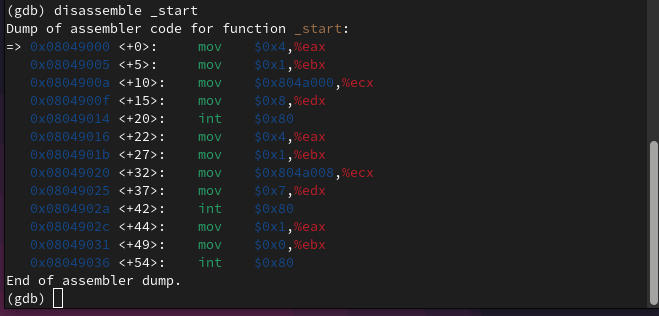


Рис. 8: Дисассимплированный код

С помощью команды я переключился на intel’овское отображение синтаксиса. Отличие заключается в командах, в диссамилированном отображении в командах используют % и $, а в Intel отображение эти символы не используются. (рис. 9).

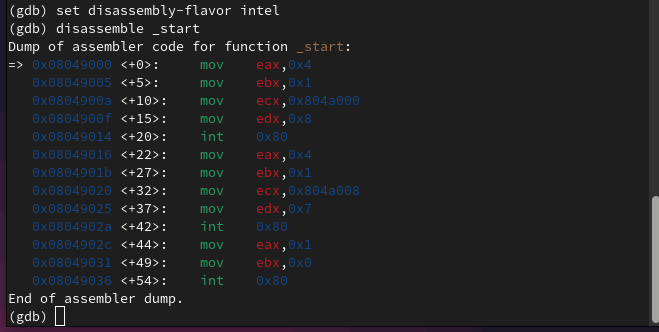


Рис. 9: Intel’овское отображение

Включаю режим псевдографики. (рис. 10).

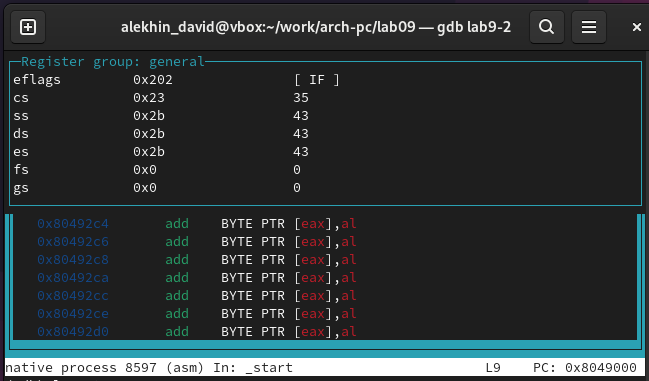


Рис. 10: Псевдографика

Смотрю наличие брейкпоинтов. (рис. 11).

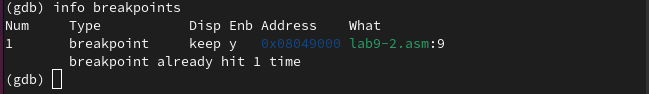


Рис. 11: Смотрю наличие брейкпоинтов

Добавляю метку на предпоследнюю строку. (рис. 12).

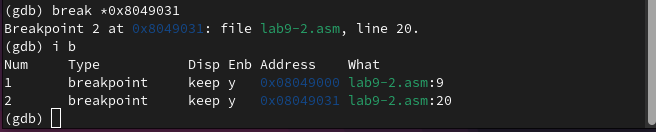


Рис. 12: Добавляю метку

С помощью команды si просматриваю регистры. (рис. 13).

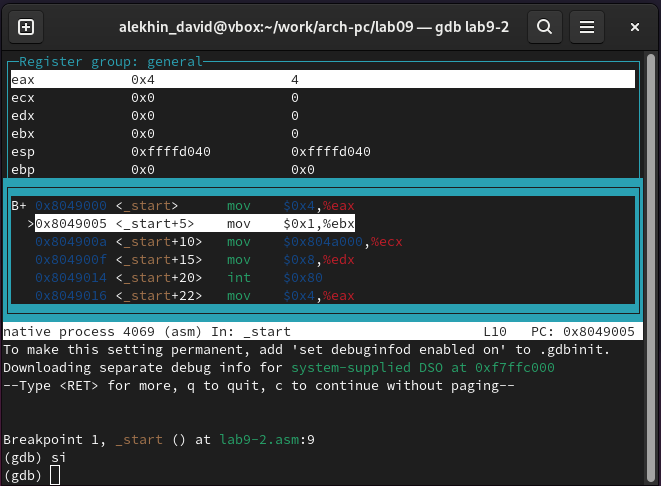


Рис. 13: Просмотр регистров

С помощью команды смотрю значение переменной msg1. (рис. 14).

Рис. 14: Просмотр значения переменной

Рис. 14: Просмотр значения переменной

Просматриваю значение второй переменной msg2. (рис. 15).

Рис. 15: Значение переменной msg2

Рис. 15: Значение переменной msg2

С помощью команды set меняю значение переменной msg1. (рис. 16).

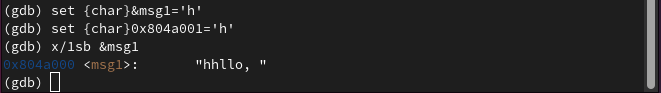


Рис. 16: Изменение значения переменной

Я изменяю переменную msg2. (рис. 17).

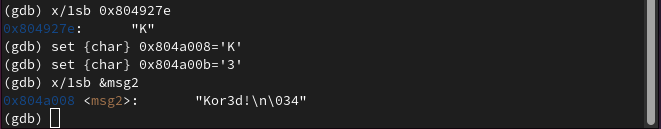


Рис. 17: Изменение msg2

Вывожу значение регистров ecx и eax. (рис. 18).

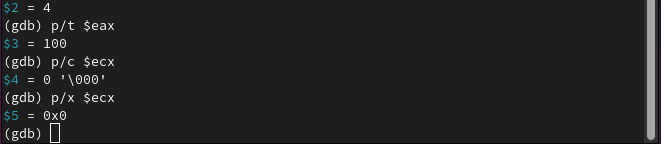


Рис. 18: Значение регистров ecx и eax

Меняю значение регистра ebx. Команда выводит два разных значения так как в первый раз мы вносим значение 2, а во второй раз регистр равен двум. (рис. 19).

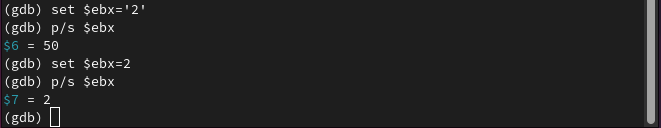


Рис. 19: Меняю значение регистров ebx

Копирую файл lab8-2.asm и запускаю в отладчике. (рис. 20).

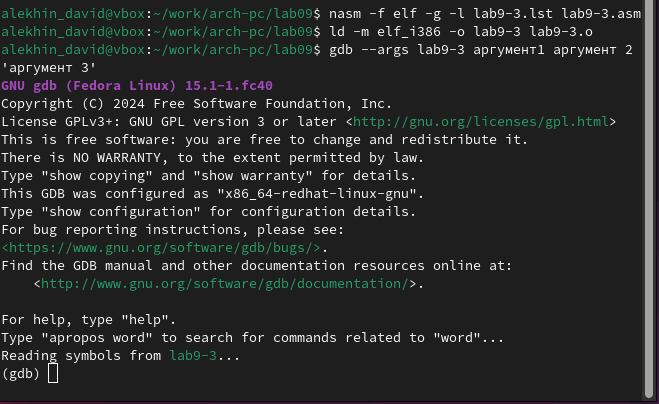


Рис. 20: Запуск файла в отладчике

Ставлю метку на \_start и запускаю файл. (рис. 21).

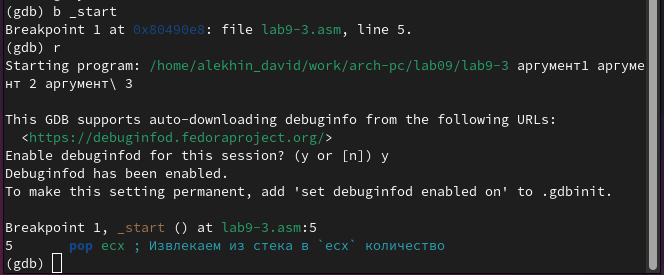


Рис. 21: Запуск файла lab10-3 через метку

Проверяю адрес вершины стека. (рис. 22).

Рис. 22: Адрес вершины стека

Рис. 22: Адрес вершины стека

Я смотрю все позиции стека. Элементы расположены с интервалом в 4 единицы, так как стек может хранить до 4 байт, и для того чтобы данные сохранялись нормально и без помех, компьютер использует новый стек для новой информации.(рис. 23).

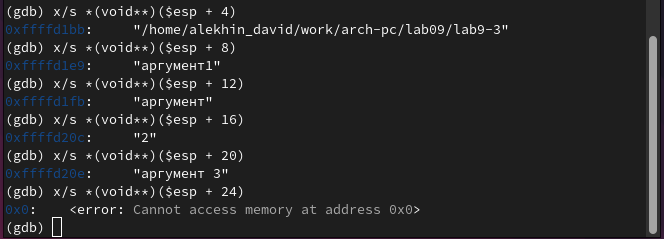


Рис. 23: Все позиции стека

Перехожу к самостоятельной работе. Преобразую программу из лабораторной работы №8 и реализую вычисления как подпрограмму. Запускаю. (рис. 24), (рис. 25).

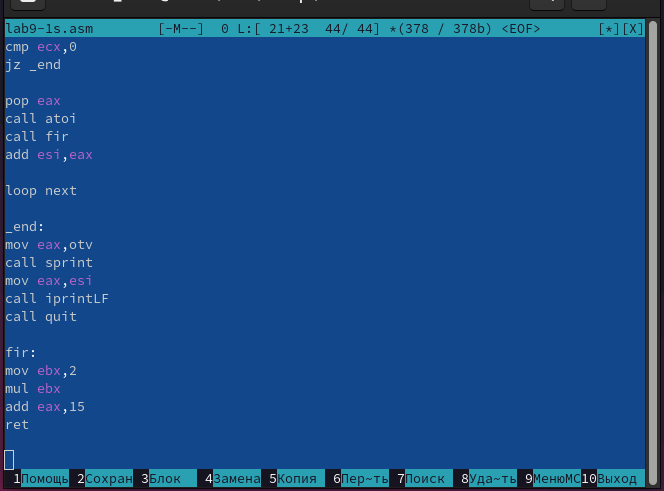


Рис. 24: Текст программы

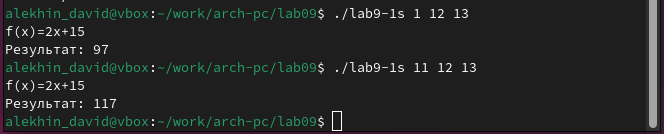


Рис. 25: Запуск программы

Копирую программу и запускаю, плучается арифметическая ошибка.(рис. 26).

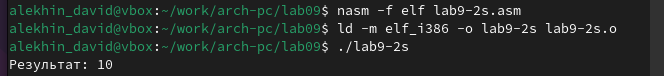


Рис. 26: Запуск программы

Запускаю программу в отладчике. (рис. 27).

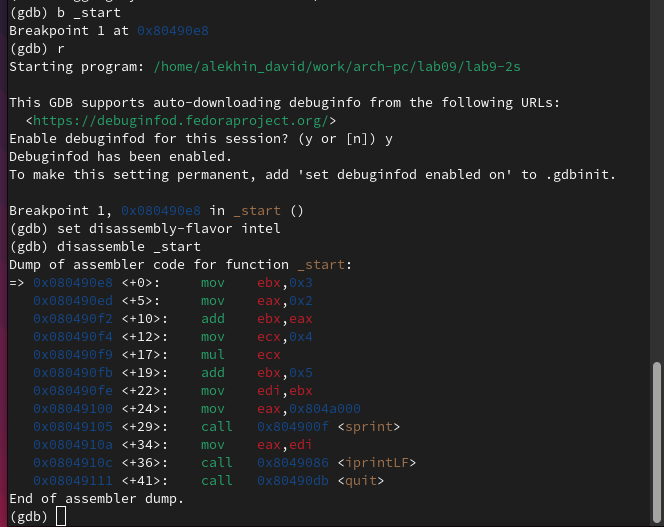


Рис. 27: Запуск программы в отладчике

Анализирую регистры, некоторые из них стоят не на своих местах. (рис. 28).

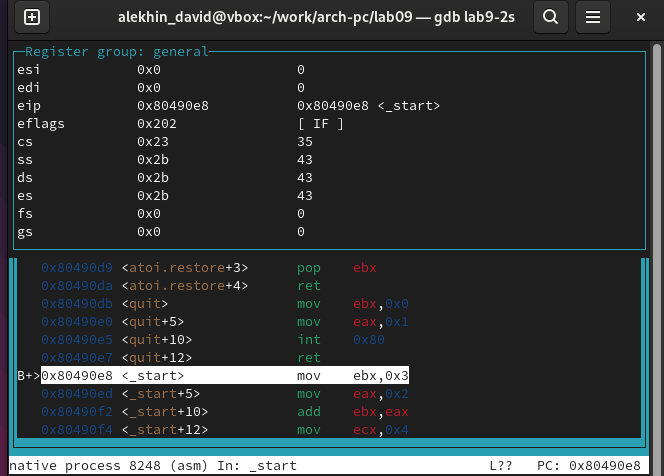


Рис. 28: Анализ регистров

Исправляю ошибки и запускаю пргораммы, получаем верный результат. (рис. 29).

Рис. 29: Запуск

Рис. 29: Запуск

# 5 Выводы

Я приобрел навыки написания программ использованием подпрограмм. Обучился отладке программ.

# Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/Learning- bash-Shell-Programming-Nutshell/dp/0596009658.
6. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm\_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
17. — 1120 с. — (Классика Computer Science).