Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

Аникин Константин Сергеевич

# Цели работы

Приобрести простейшие навыки разработки, анализа,

тестирования и отладки приложений в ОС типа UNIX/Linux

на примере создания на языке программирования С калькулятора с простейшими функциями.

# Задание

* Ознакомиться с теоретическим материалом.
* Выполнить задание

# Теоретическое введение

Стадии цикла разработки ПО [[1]](https://qalight.ua/ru/baza-znaniy/stadii-tsikla-razrabotki-po/):

Анализ требований, Проектирование, Разработка, Документация, Тестирование, Внедрение

Стандартным средством для компиляции программ в ОС типа UNIX является GCC (GNU Compiler Collection). Это набор компиляторов для разного рода языков программирования (С, C++, Java, Фортран и др.). Работа с GCC производится при помощи одноимённой управляющей программы gcc, которая интерпретирует аргументы командной строки, определяет и осуществляет запуск нужного компилятора для входного файла.

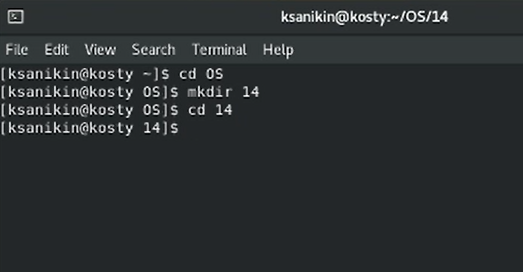
Во время работы над кодом программы программист неизбежно сталкивается с появлением ошибок в ней. Использование отладчика для поиска и устранения ошибок в программе существенно облегчает жизнь программиста. В комплект программ GNU для ОС типа UNIX входит отладчик GDB (GNU Debugger).

Ещё одним средством проверки исходных кодов программ, написанных на языке C, является утилита splint. Эта утилита анализирует программный код, проверяет корректность задания аргументов использованных в программе функций и типов возвращаемых значений, обнаруживает синтаксические и семантические ошибки.

Краткий справочник команд Linux [[2]](https://hpc.icc.ru/documentation/cmnds.pdf)

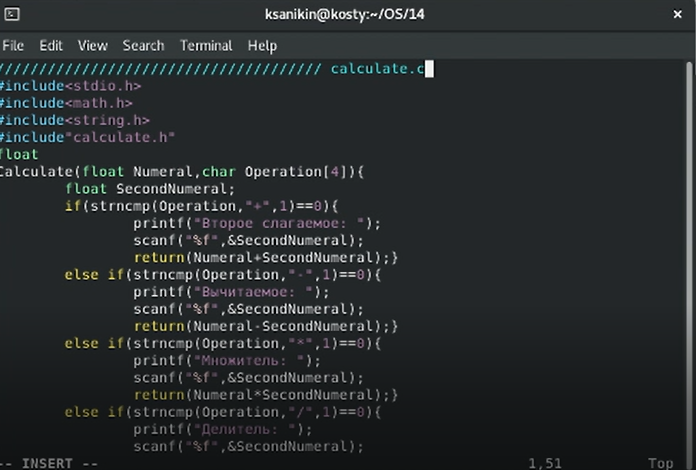
# Выполнение лабораторной работы

1. Создание папки. Я придерживаюсь другой структуры, отличной от указанной в работе (рис. 1)



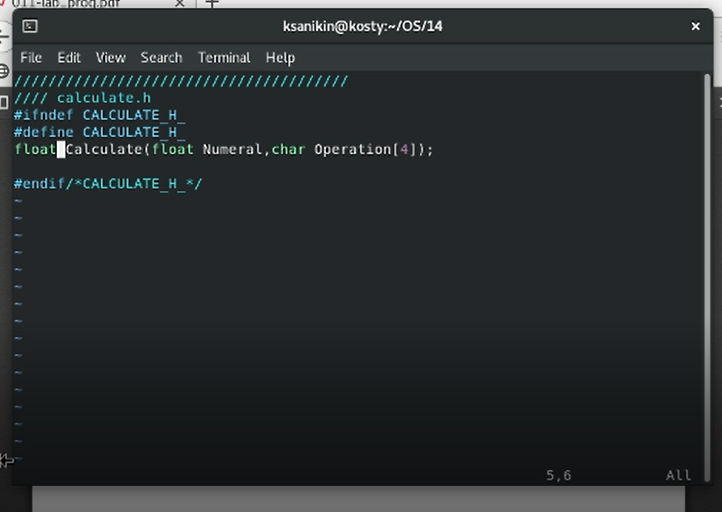
*Рис. 1: Создание папки*

2.1 Текст файла calculate.c (рис. 2)



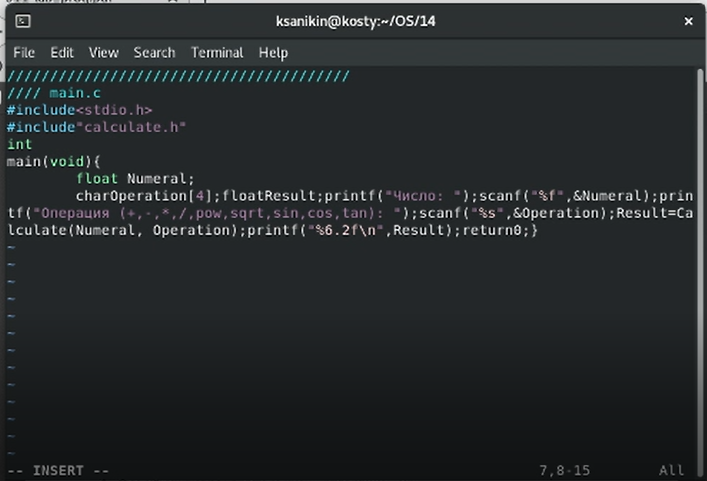
*Рис. 2: Текст файла calculate.c*

2.2 Текст файла calculate.h (рис. 3)



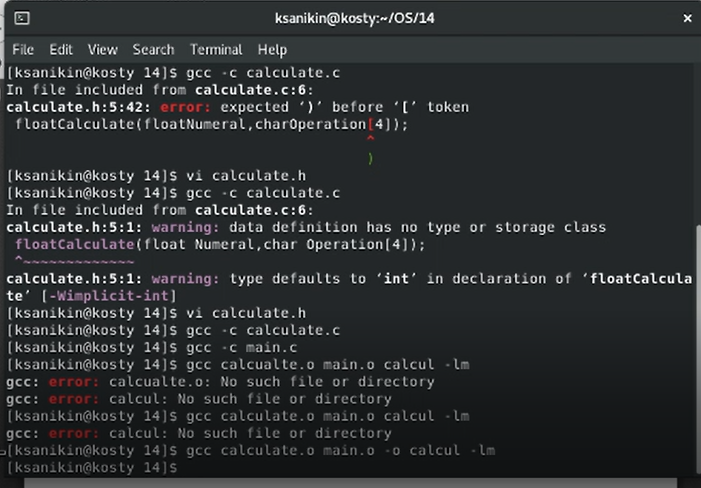
*Рис. 3: Текст файла calculate.h*

2.3 Текст файла main.c (рис. 4)



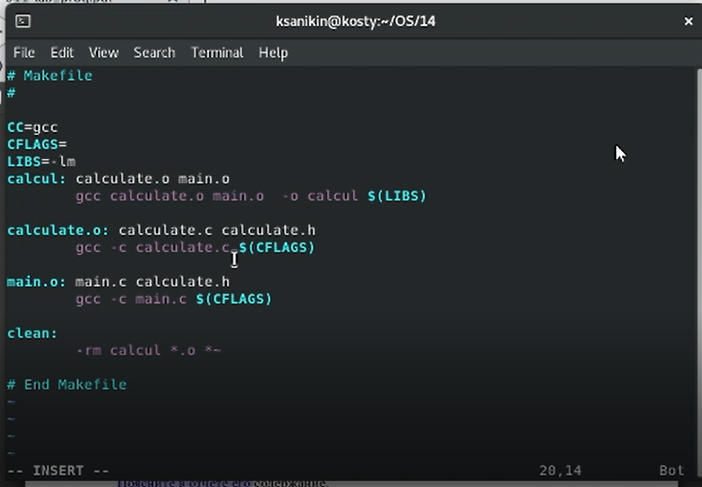
*Рис. 4: Текст файла main.c*

1. Создание исполняемых файлов с помощью gcc (рис. 5)



*Рис. 5: Создание исполняемых файлов с помощью gcc*

1. Готовый Makefile. Анализ его вынесен отдельным пунктом в конце работы (рис. 6)



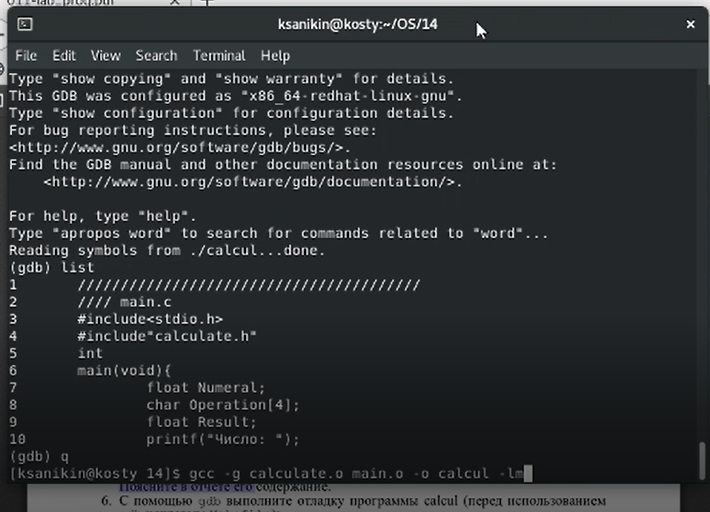
*Рис. 6: Готовый Makefile*

5.5 Далее в работе упоминается исправление makefile для работы с calculом

Однако для нормального дебаггинга в gdb понадобилось вовсе не исправить makefile

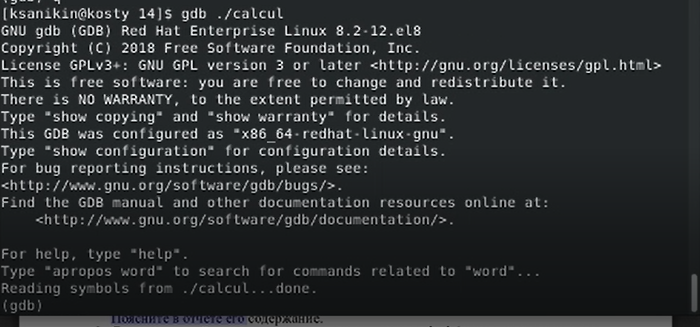
(который в ходе работы вообще не используется)

а добавить атрибут -g ко всем командам пункта 3 (рис. 7)



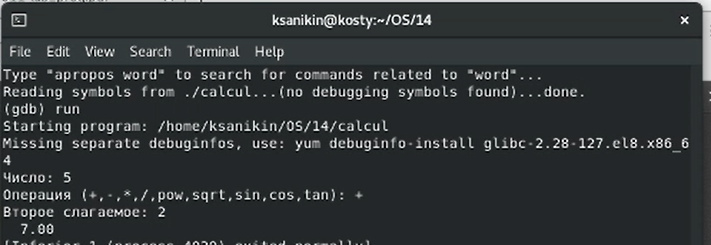
*Рис. 7: Добавление атрибута -g*

6.1 Запуск gdb (рис. 8)



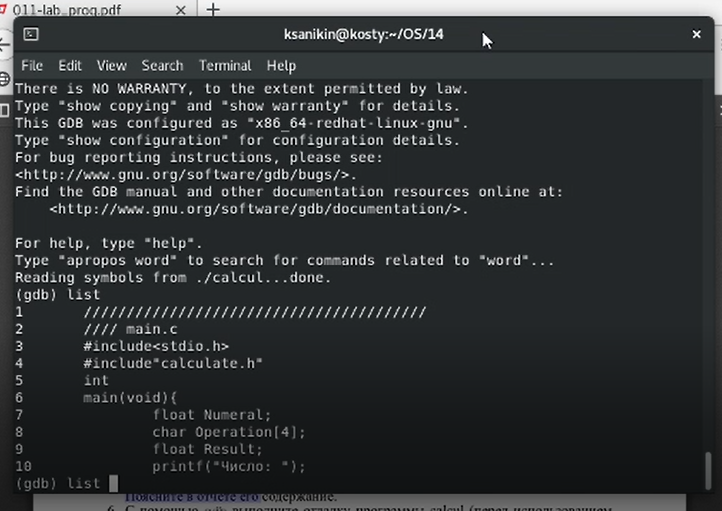
*Рис. 8: gdb./calcul*

6.2 Запуск программы командой run (рис. 9)



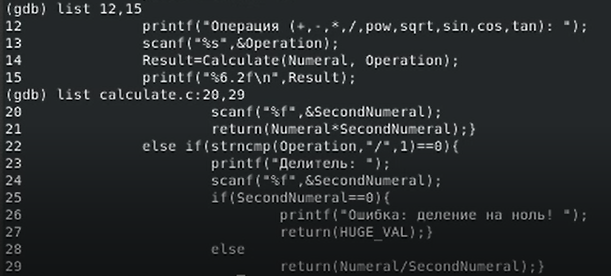
*Рис. 9: Команда run*

6.3 Команда list (рис. 10)



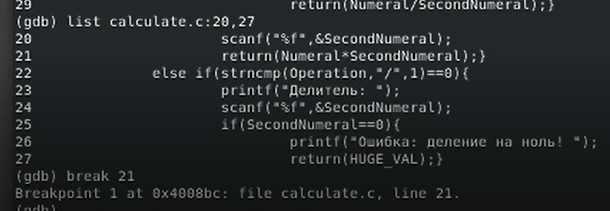
*Рис. 10: Команда list*

6.4-6.5 Еще две команды list в других расцветках (рис. 11)



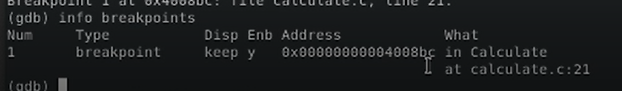
*Рис. 11: Команды list*

6.6-6.7 Еще одна команда list, установка брейк-поинта (рис. 12)



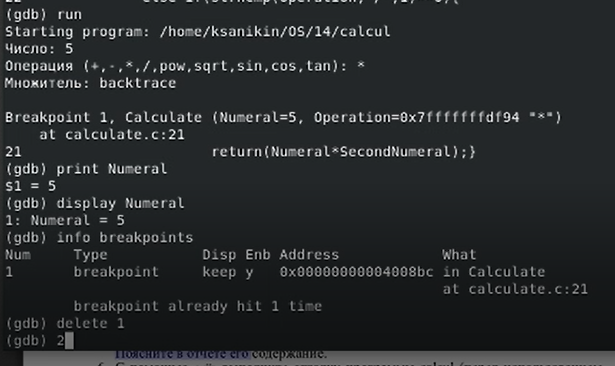
*Рис. 12: Установка брейк-поинта*

6.8 Информация о брейк-поинтах (рис. 13)



*Рис. 13: info breakpoints*

6.9 backtrace, сравнение print и display, удаление точки останова (рис. 14)



*Рис. 14: Оставшиеся пункты работы*

1. Для этого пункта необходим splint, который я не понял, как установить

# Анализ Makefile

Первые три не-комментария - объявление переменных.

Последняя - аргументы для создания calcul

Вторая - пустая, используется при создании исполняемых файлов. Видимо, её можно как-то заполнить извне

Первая нигде не используется. Её смысл мне неясен.

Далее идут четыре метода(?) создания файла.

Первые три идентичны командам создания файлов, вводимых в работе.

Причём первый в ходе работы вызывает второй и третий, как я понял.

Последний же убирает все файлы, по сути - метод удаления.

# Выводы

Работа выполнена полностью, если не считать проблем со splintом

## Библиографический список

[1]: [Стадии цикла разработки ПО](https://qalight.ua/ru/baza-znaniy/stadii-tsikla-razrabotki-po/)

[2]: [Краткий справочник команд Linux](https://hpc.icc.ru/documentation/cmnds.pdf)