Лабораторная работа No 13.

Средства, применяемые приразработке программного обеспечения в ОС типа UNIX/Linux

Паласиос Фелипe

Содержание

# 1 Цель работы

Приобрести простейшие навыки разработки, анализа, тестирования и отладки при- ложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

# 2 Задание

1. В домашнем каталоге создайте подкаталог ~/work/os/lab\_prog.
2. Создайте в нём файлы: calculate.h, calculate.c, main.c. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится. Реализация функций калькулятора в файле calculate.h:

Интерфейсный файл calculate.h, описывающий формат вызова функции- калькулятора:

Основной файл main.c, реализующий интерфейс пользователя к калькулятору:

1. Выполните компиляцию программы посредством gcc:
2. При необходимости исправьте синтаксические ошибки.
3. Создайте Makefile со следующим содержанием:
4. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile): – Запустите отладчик GDB, загрузив в него программу для отладки – Для запуска программы внутри отладчика введите команду run: – Для постраничного (по 9 строк) просмотра исходного код используйте команду list: – Для просмотра строк с 12 по 15 основного файла используйте list с параметрами:

– Для просмотра определённых строк не основного файла используйте list с па- раметрами:

– Установите точку останова в файле calculate.c на строке номер 21:

– Выведите информацию об имеющихся в проекте точка останова:

– Запустите программу внутри отладчика и убедитесь, что программа остановится в момент прохождения точки останова:

– Отладчик выдаст следующую информацию: а команда backtrace покажет весь стек вызываемых функций от начала програм- мы до текущего места.

– Посмотрите, чему равно на этом этапе значение переменной Numeral, введя: На экран должно быть выведено число 5.

– Сравните с результатом вывода на экран после использования команды:

– Уберите точки останова

1. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c

# 3 Выполнение лабораторной работы

1. В домашнем каталоге создайте подкаталог ~/work/os/lab\_prog. (рис. [1](#fig:001))

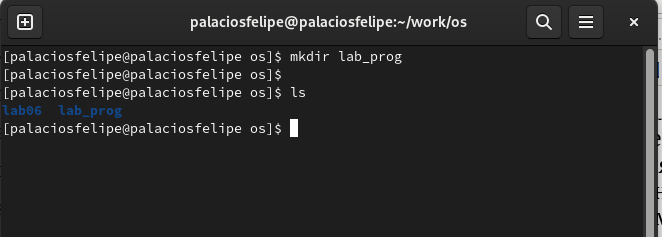


Figure 1: lab\_prog

1. Создайте в нём файлы: calculate.h, calculate.c, main.c. (рис. [2](#fig:002))

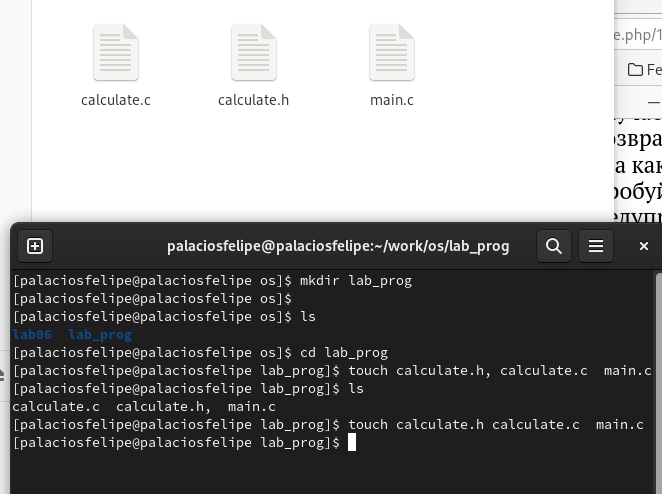


Figure 2: Файлы

Реализация функций калькулятора в файле calculate.c(рис. [3](#fig:003))

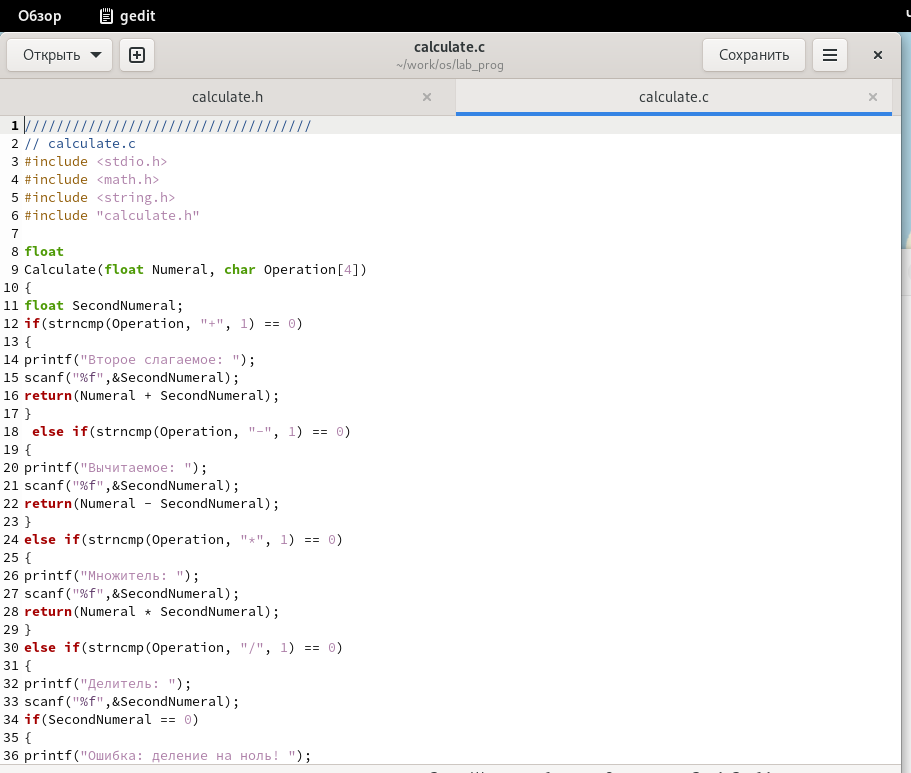


Figure 3: calculate.c

Интерфейсный файл calculate.h, описывающий формат вызова функции- калькулятора:рис. [4](#fig:004))

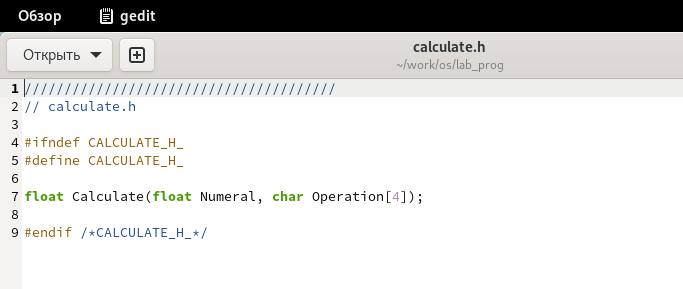


Figure 4: calculate.h

Основной файл main.c, реализующий интерфейс пользователя к калькулятору:рис. [5](#fig:005))

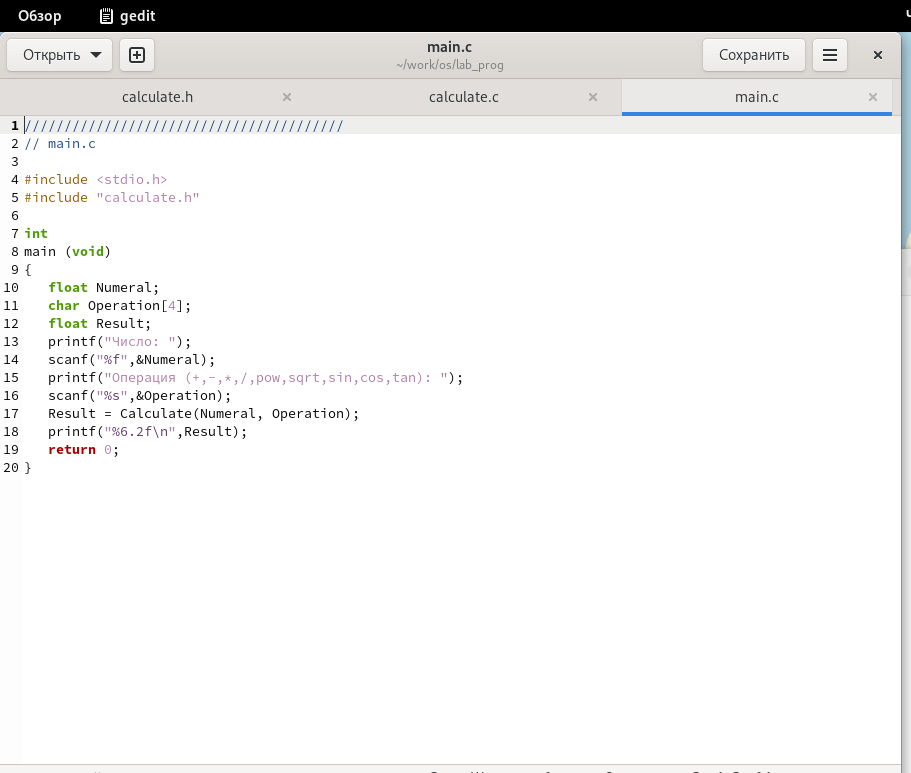


Figure 5: main

1. Выполните компиляцию программы посредством gcc: (рис. [6](#fig:006))

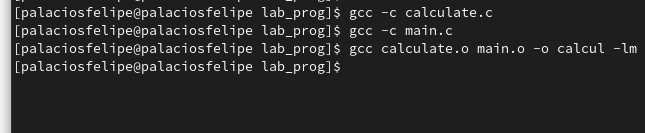


Figure 6: Компиляция

1. При необходимости исправьте синтаксические ошибки.
2. Создайте Makefile со следующим содержанием: (рис. [7](#fig:007))

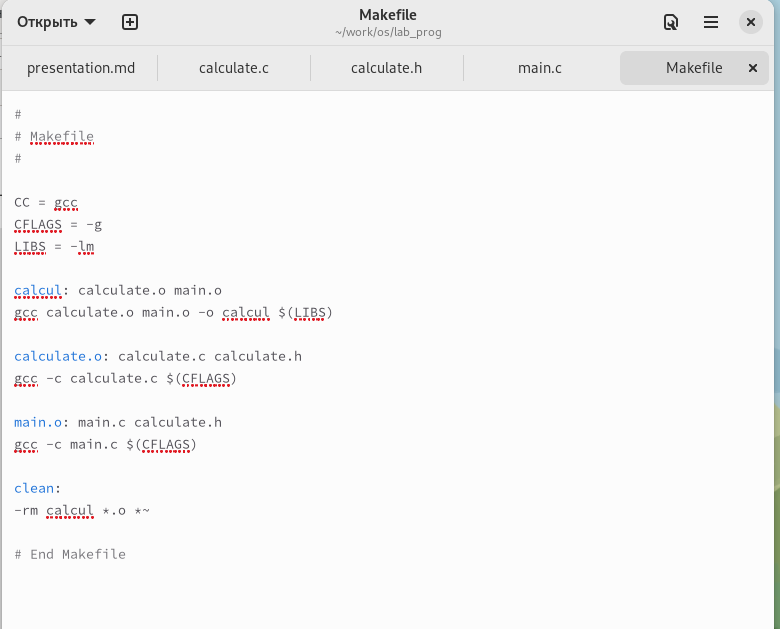


Figure 7: Makefile

1. С помощью gdb выполните отладку программы calcul (перед использованием gdbисправьте Makefile)(рис. [8](#fig:008), [9](#fig:009), [10](#fig:010), [11](#fig:011), [12](#fig:012), [13](#fig:013), [14](#fig:014), [15](#fig:015), [16](#fig:016), [17](#fig:017), [18](#fig:018))

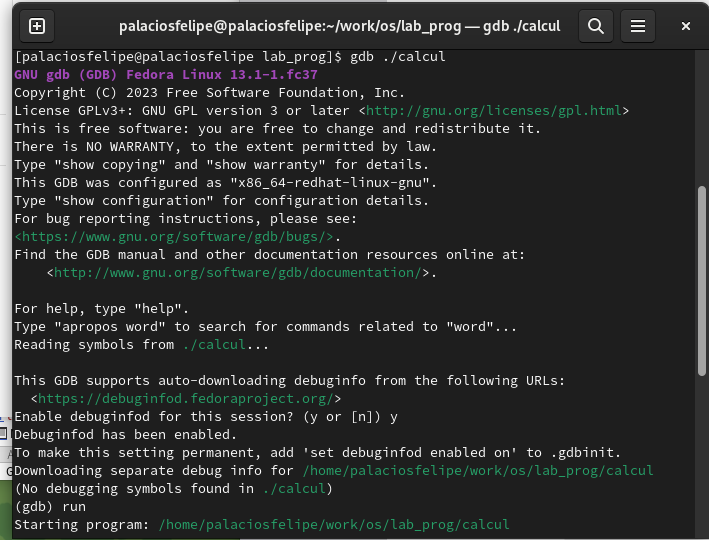


Figure 8: Сalcul

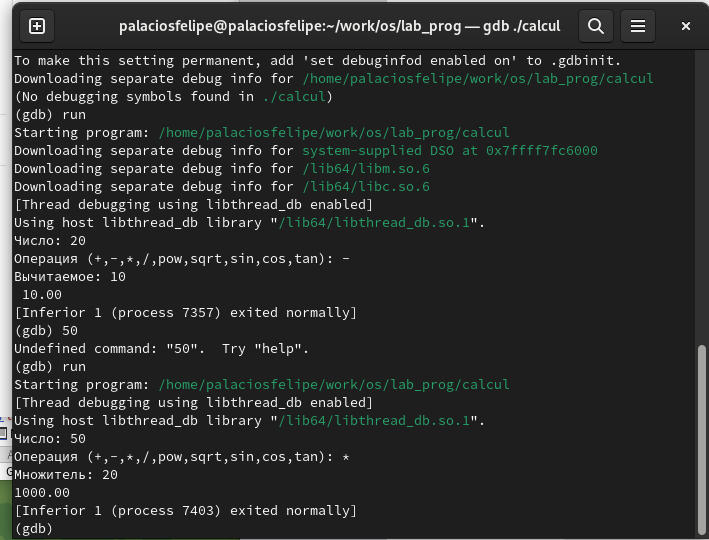


Figure 9: Run

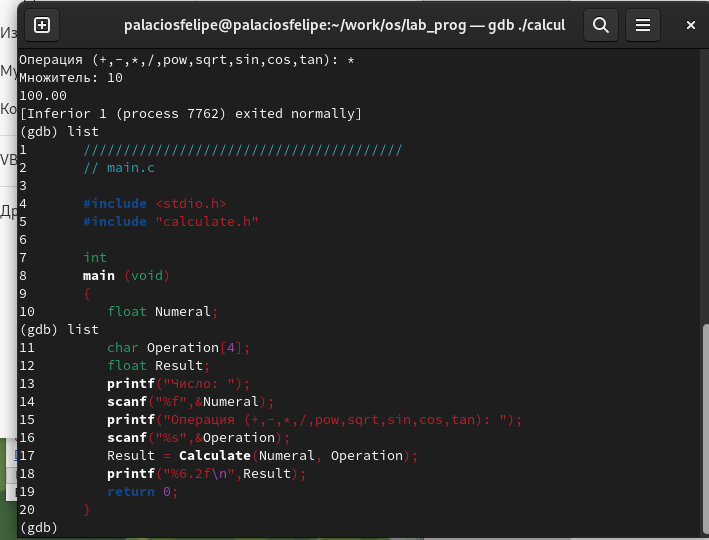


Figure 10: list

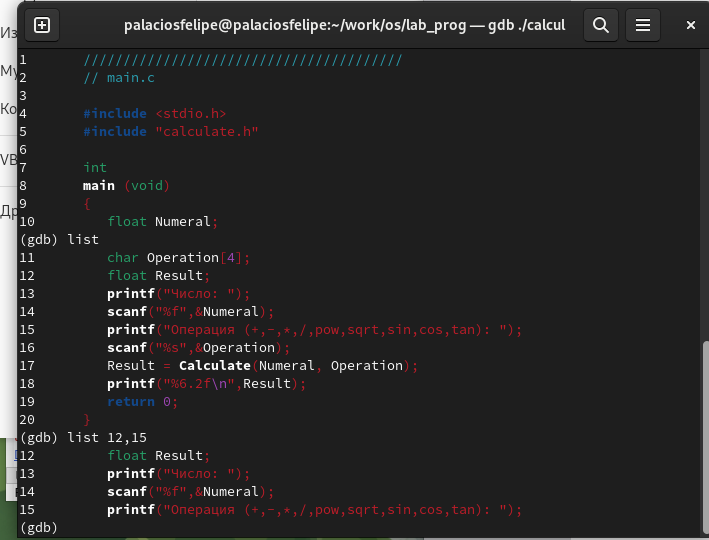


Figure 11: list строк 12-15

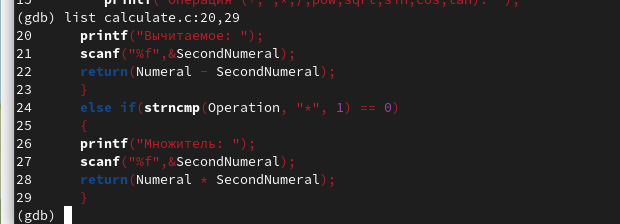


Figure 12: list 20,29

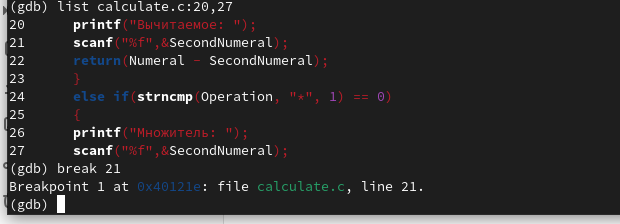


Figure 13: list 20,27, break 21

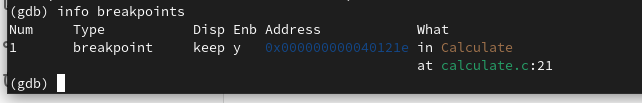


Figure 14: info breakpoints

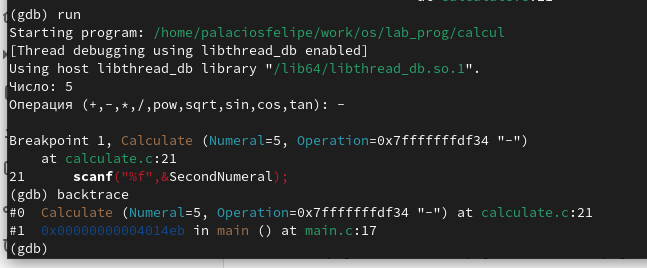


Figure 15: backtrace

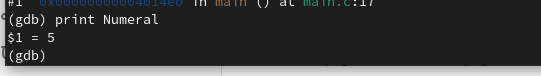


Figure 16: print Numeral

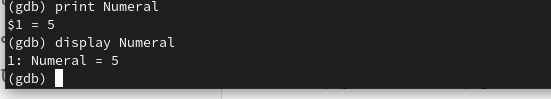


Figure 17: display Numera

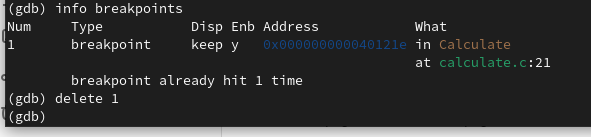


Figure 18: info breakpoints delete 1

1. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c (рис. [19](#fig:019), [20](#fig:020))

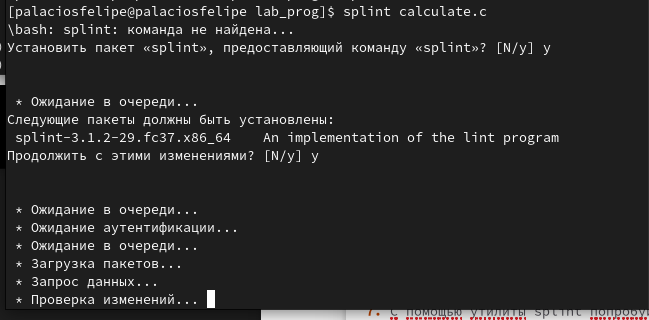


Figure 19: splint calculate.c

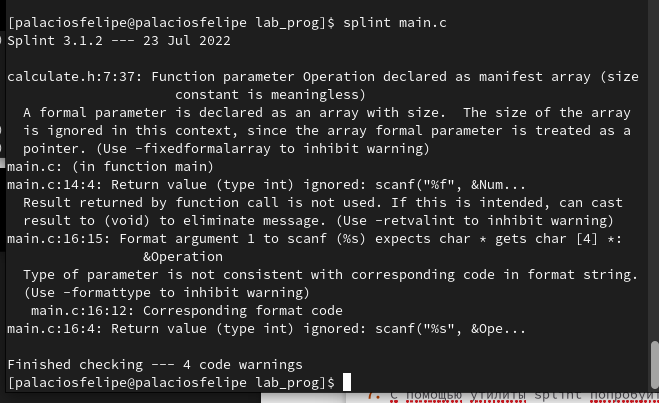


Figure 20: splint main.c

# 4 Выводы

Приобрел простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.