# РОCСИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

## Факультет физико-математических и естественных наук

## Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

## ПО ЛАБОРАТОРНОЙ РАБОТЕ № 14

дисциплина: Операционные системы

Студент: Юрченко Артём Алексеевич Группа: НФИбд-02-20

МОСКВА

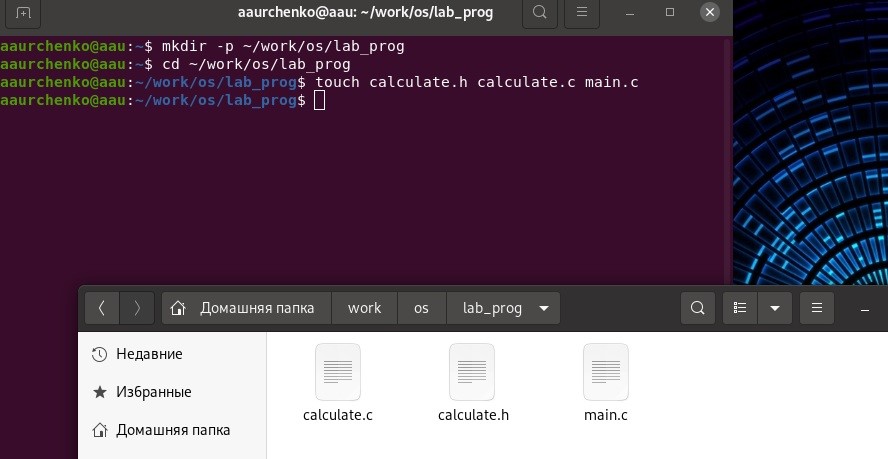
2021 г.

# Цель работы

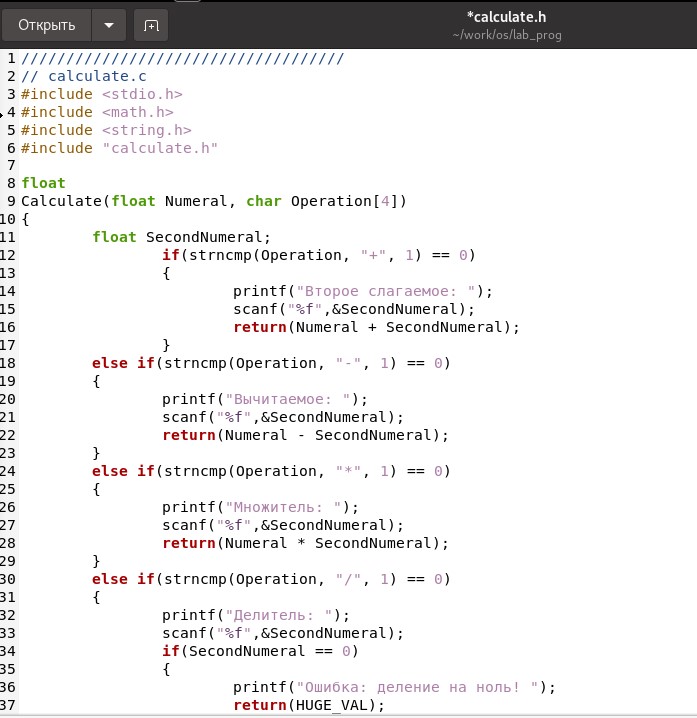
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

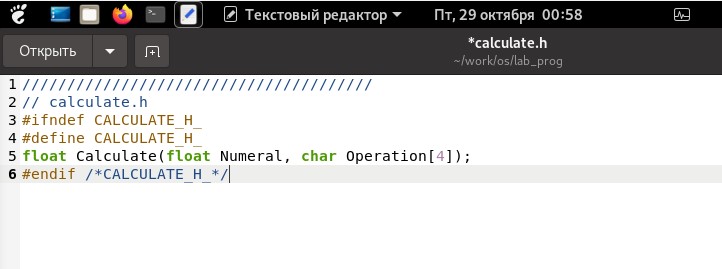
Выполнение лабораторной работы.

1. В домашнем каталоге создаём подкаталог ~/work/os/lab\_prog.

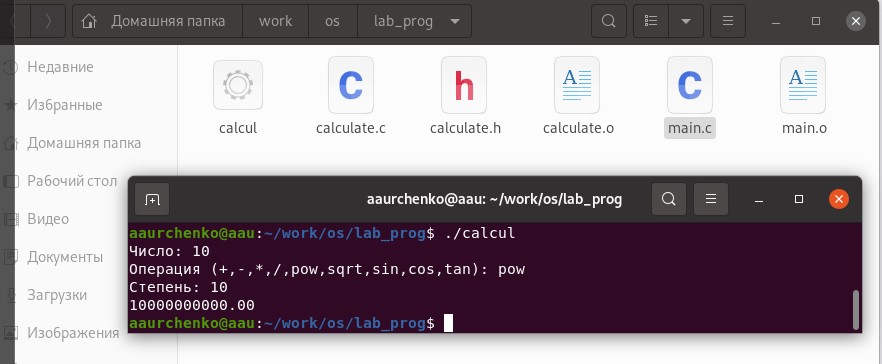


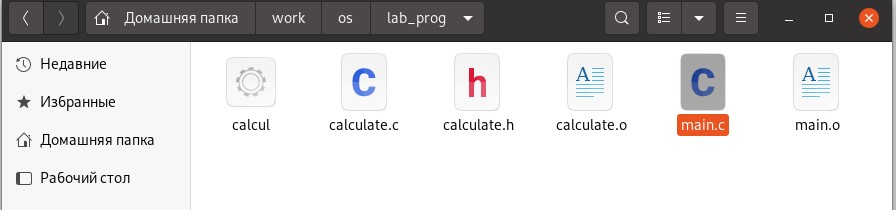
1. Создайте в нём файлы: calculate.h, calculate.c, main.c. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.





1. Реализуем функций калькулятора в файле calculate.c.





1. Реализуем заголовочный файл calculate.h, описывающий формат вызова функции калькулятора.
2. Пишем основной файл main.c, реализующий интерфейс пользователя к калькулятору.
3. Выполняем компиляцию программы посредством gcc и проверяем работу калькулятора.
4. Создаём Makefile. Для создания изпользуем образец из работы. Попутно исправляем ошибки в образце, которые препятсвуют оптимальной работе отладчика.



1. Используем Makefile.

9

.

Запускаем отладчик GDB, загрузив в него программу для отладки.

gdb ./calcul

10

.

Для запуска программы внутри отладчика вводим команду run.

run

11

.

Для постраничного (по 9 строк) просмотра исходного код используем команду list.

list

12. Для просмотра строк с 12 по 15 основного файла используем list с параметрами.

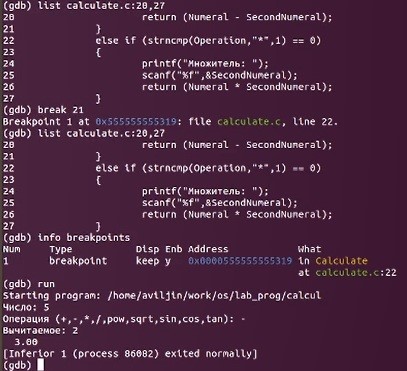
list 12,15

13

.

Для просмотра определённых строк не основного файла используем list с параметрами.

list calculate.c:20,29



14

.

Устанавливаем точку остановки в файле calculate.c на строке номер 22, чтобы программа остановилась, после ввода операции вычетания.

list calculate.c:15,25

break 18

15

.

Выводим информацию об имеющихся в проекте точка останова.

info breakpoints

16

.

Запускаем программу внутри отладчика и убеждаемся, что программа остановилась в момент прохождения точки остановки.

run

5

-

17

.

Используем команду backtrace, которая показывает весь стек вызываемых функций от начала программы до текущего места.

backtrace

18

.

Просматриваем, чему равно на этом этапе значение переменной Numeral.

print Numeral

19

.

Сравниваем с результатом вывода на экран после использования команды display.

display Numeral

20

.

Убераем точки остановки.

info breakpoints

delete 2

21. С помощью утилиты splint пробуем проанализировать коды файлов calculate.c и main.c. Найдены незначительные ошибки, невлияющие на работоспособность кода. Например, утилита предлагает посылать не весь массив Operation в функцию Calculate, а только указатель на него. Это не является ошибкой, но оптимизурет код. Также splint предупреждает, что опасно однозначно сравнивать числа с плавующей точкой и целочисленные числа, опять же хорошоее замечание, но на работу сильно не влияет. В общем, splint не нашел серьёзных ошибок, что, в целом, логично, поскольку код работает.

splint calculate.c

splint main.c

# Выводы

Мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux. Закрепили знания, полученные в прошлых работах. Создали на языке программирования С калькулятор с простейшими функциями и разобрали на нем основные навыки отладки.

# Термины

GCC (GNU Compiler Collection) - этото набор компиляторов для разного рода языков программирования (С, C++, Java, Фортран и др.).

GDB (GNU Debugger) - отладчик для поиска и устранения ошибок в программе. Входит в комплект программ GNU для ОС типа UNIX.

Утилита make позволяет автоматизировать процесс преобразования файлов программы из одной формы в другую, отслеживает взаимосвязи между файлами.

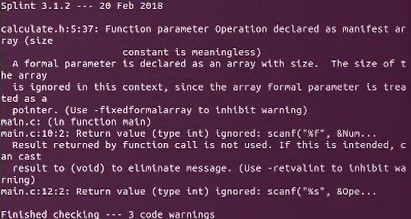
Утилита splint анализирует программный код, проверяет корректность задания аргументов использованных в программе функций и типов возвращаемых значений, обнаруживает синтаксические и семантические ошибки.

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.

Последовательность команд может быть помещена в текстовый файл. Такой файл называется командным.

Каталог, он же директория, (от англисйкого Directory) – это объект в ФС (файловой системе), необходимый для того, чтобы упросить работу с файлами.

Домашний каталог - каталог, предназначенный для хранения собственных данных пользователя Linux. Как правило, является текущим непосредственно после регистрации пользователя в системе.

Команда - записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе.

# Выводы

Мы изучили основы программирования в оболочке ОС UNIX. Закрепили знания, полученные в прошлых работах. Научиись писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Термины

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.

Последовательность команд может быть помещена в текстовый файл. Такой файл называется командным.

Флаги — это опции командной строки, обычно помеченные знаком минус; Например, для команды ls флагом может являться -F.

Каталог, он же директория, (от англисйкого Directory) – это объект в ФС (файловой системе), необходимый для того, чтобы упросить работу с файлами.

Домашний каталог - каталог, предназначенный для хранения собственных данных пользователя Linux. Как правило, является текущим непосредственно после регистрации пользователя в системе.

Команда - записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе.