# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»

**Факультет физико-математических и естественных наук Кафедра информационных технологий**

# ОТЧЕТ

по лабораторной работе 13

**ТЕМА** «Средства, применяемые при разработке программного обеспечения в ОС типа

UNIX/Linux» по дисциплине «Операционные системы»

# Выполнил/лa:

**Студент/ка группы** НПИбд-02-21

**Студенческий билет No** 1032205421

**Студент/кa:** Стелина Петрити

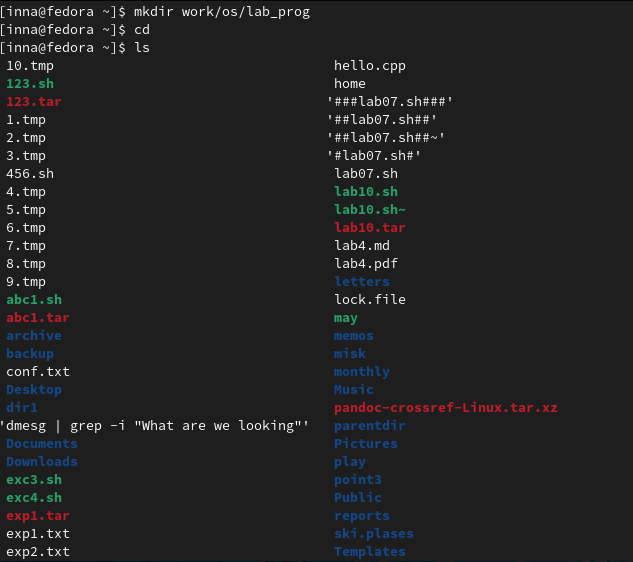
## Цель работы

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.

## Последовательность выполнения работы

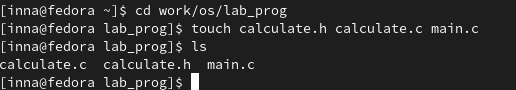
1. *В домашнем каталоге создайте подкаталог ~/work/os/lab\_prog*



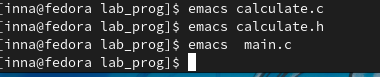
* 1. *подкаталог ~/work/os/lab\_prog*

1. *Создайте в нём файлы: calculate.h, calculate.c, main.c. Это будет примитивнейший*

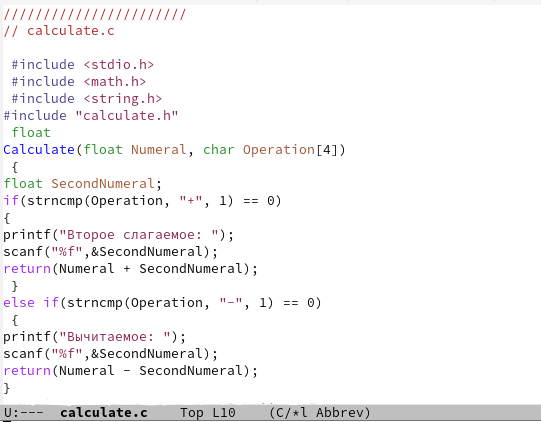
*калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. При запуске он будет*

*запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.*

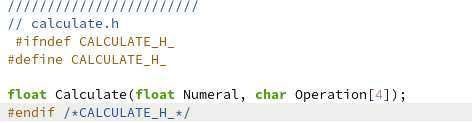
* 1. *Создание файлы- calculate.h, calculate.c, main.c.*



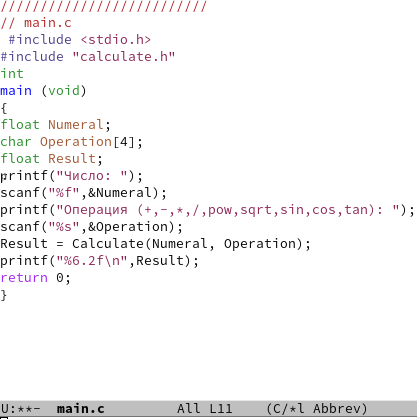
* 1. *emacs: calculate.h, calculate.c, main.c.*



* 1. *функция калькулятора calculate.h*

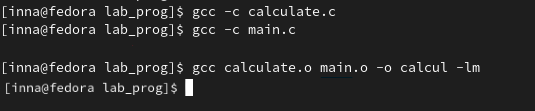


* 1. *Интерфейсный файл calculate.h*



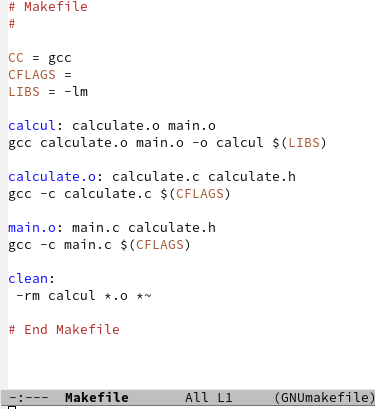
* 1. *файл main*

1. *Выполните компиляцию программы посредством gcc:*



* 1. *Выполнение компиляцию программы*

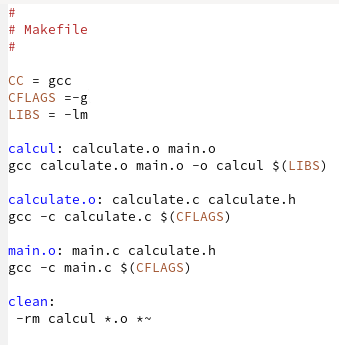
1. *Создайте Makefile со следующим содержанием: В этом файле мы создаем переменные CC, CFLAGS, LIBS. Инициализируем и создаем блоки.*



*4.1.Makefile*

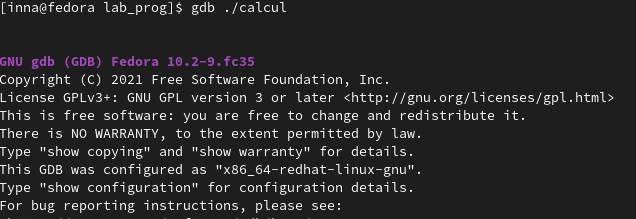
1. *С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте*

*Makefile):*



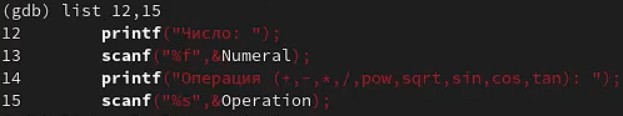
* 1. *отладка программы calcul*

*Запустим отладчик GDB, загрузив в него программу для отладки (рис. [-@fig:008]).*



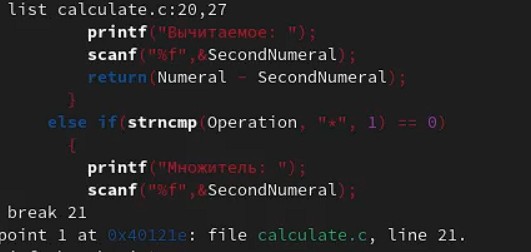
* 1. *Запустим отладчик GDB*

*Для просмотра строк с 12 по 15 основного файла используйте list с параметрами*



* 1. *строк с 12 по 15*

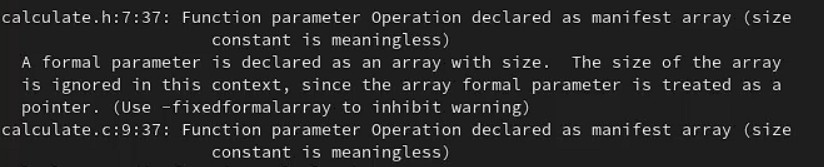
Установите точку останова в файле calculate.c на строке номер 21



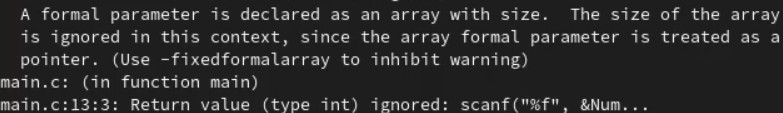
* 1. *просмотра строк номер 21*

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

*и main.c : splint calculate.c*



* 1. *splint calculate.c*



* 1. *splint main.c*

## Выводы

Во время этой лабораторной работы я работала с самыми простыми навыками. Эти простейшие навыки включают в себя навыки разработки, анализа, тестирования и отладки приложений. Я использовала в качестве примера создание калькулятора с простейшими функциями.

## Контрольные вопросы

### Как получить информацию о возможностях программ gcc, make, gdb и др.? c командой

man.

### Назовите и дайте краткую характеристику основным этапам разработки приложений в UNIX.

тестирование, отладка, компиляция исходного текста, создание исходного кода

### Что такое суффикс в контексте языка программирования? Приведите примеры использования.

Суффикс .с; с программой на Си

1. **Каково основное назначение компилятора языка С в UNIX?** Основное назначение компилятора языка С в UNIX -суффиксы и префиксы **5.Для чего предназначена утилита make?**

утилита Make- автоматизирующая процесс преобразования файлов из одной формы в другую

### Приведите пример структуры Makefile. Дайте характеристику основным элементам этого файла.

make- Правила преобразования задаются в скрипте с именем Makefile, который должен находиться в корне рабочей директории проекта. Сам скрипт состоит из набора правил, которые в свою очередь описываются:

* 1. целями ( данное правило делает);
  2. реквизитами ( для выполнения правила и получения целей);
  3. командами (выполняющими данные преобразования). пример:

# Индентация осуществляется исключительно при помощи символов табуляции,

# каждой команде должен предшествовать отступ

<цели>:

<команда #1> ...

<команда

### Назовите основное свойство, присущее всем программам отладки. Что необходимо сделать, чтобы его можно было использовать?

Для отладки нужно запустить приложение с отладчиком, подключенным к процессу приложения. Для этого чаще всего используется клавиша F5 . Однако сейчас у вас, возможно, не задано ни одной точки останова для проверки кода приложения, поэтому мы сначала зададим их, а затем начнем отладку. Точки останова — это один из самых простых и важных компонентов надежной отладки. Точка останова указывает, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

### Назовите и дайте основную характеристику основным командам отладчика gdb.

* next – пошаговое выполнение программы
* print – выводит значение какого-либо выражения (выражение пере- даётся в качестве параметра)
* run – запускает программу на выполнение
* set – устанавливает новое значение переменной
* step – пошаговое выполнение программы
* backtrace – выводит весь путь к текущей точке останова
* break – устанавливает точку останова
* clear – удаляет все точки останова на текущем уровне стека
* continue – продолжает выполнение программы от текущей точки до конца;
* delete – удаляет точку останова
* display – добавляет выражение в список выражений
* finish – выполняет программу до выхода из текущей функции
* info breakpoints – выводит список всех имеющихся точек останова
* info watchpoints – выводит список всех имеющихся контрольных выражений
* list – выводит исходный код
* watch – устанавливает контрольное выражение

### Опишите по шагам схему отладки программы, которую Вы использовали при выполнении лабораторной работы.

запустилa отладчик, запустила программу, выполнения, введя команды, отобразила данные.

### Прокомментируйте реакцию компилятора на синтаксические ошибки в программе при его первом запуске.

Не было синтаксических ошибок.

### Назовите основные средства, повышающие понимание исходного кода программы.

cscope - это инструмент программирования, который работает с текстовым интерфейсом, который позволяет программистам.

Lint, или линтер, - это инструмент статического анализа кода, используемый для выявления ошибок программирования,стилистических ошибок, ошибок и подозрительных конструкций.

### Каковы основные задачи, решаемые программой splint?

Splint - это инструмент для статической проверки программ на языке Си,то инструмент статического анализа кода, используемый для выявления ошибок

программирования,стилистических ошибок, ошибок и подозрительных конструкций.