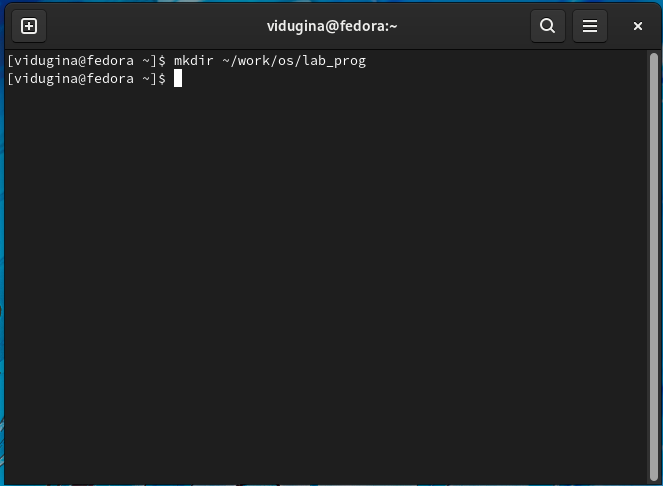
## **Отчёт по лабораторной работе №13**

#### **Дугина Виктория Игоревна**

**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.

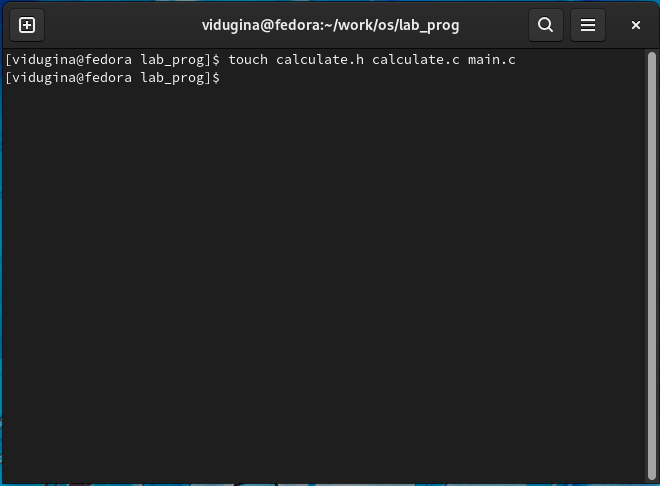
### **Выполнение работы**

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

 *Рис 1.1*

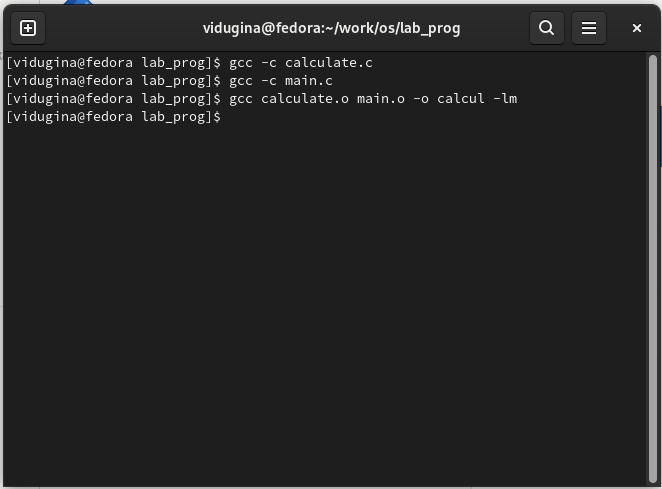
#### 2. Создайте в нём файлы: calculate.h, calculate.c, main.c. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. При запуске он будет запрашивать первое число, операцию, второе число.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

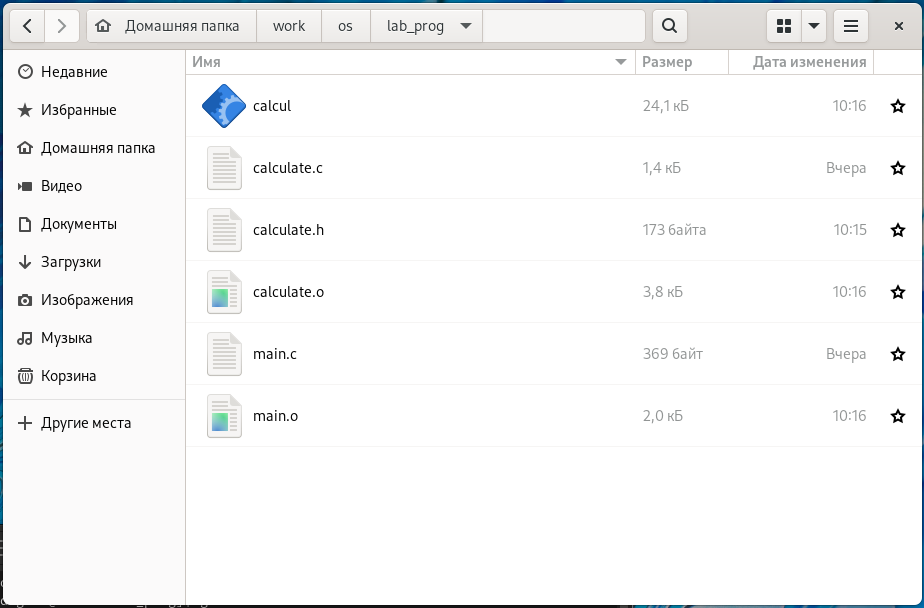
Используя команду touch, создаём файлы, после записывая в них нужный код.

 *Рис 2.1*

#### 3. Выполните компиляцию программы посредством gcc.

Компилируем программу.

 *Рис 3.1*

 *Рис 3.2*

#### 4. При необходимости исправьте синтаксические ошибки.

Синтаксических ошибок нет.

#### 5. Создайте Makefile со следующим содержанием...Поясните в отчёте его содержание.

 *Рис 5.1*

В данном Makefile существует 3 переменные и 4 цели:

Переменные:

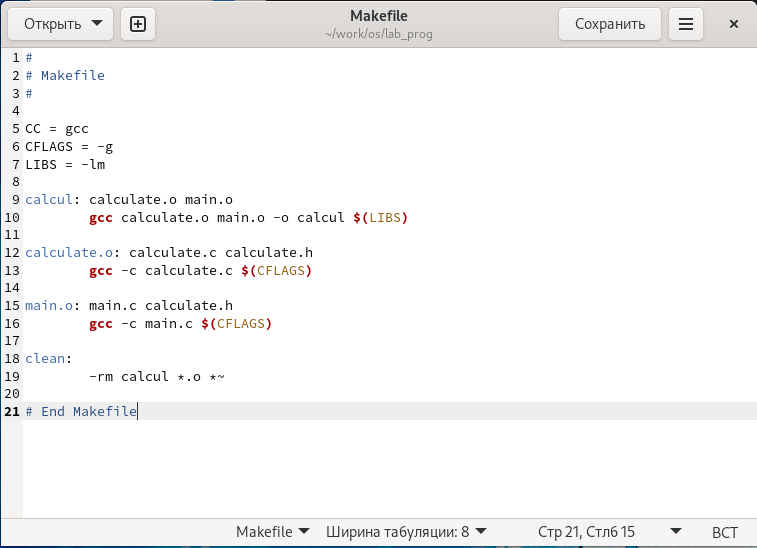
* СС = gcc: далее не использется
* CFLAGS = : не сожержит ничего
* LIBS = -lm: флаг gcc, находит и приязывает библиотеку libm, содержащую <math.h>

Цели:

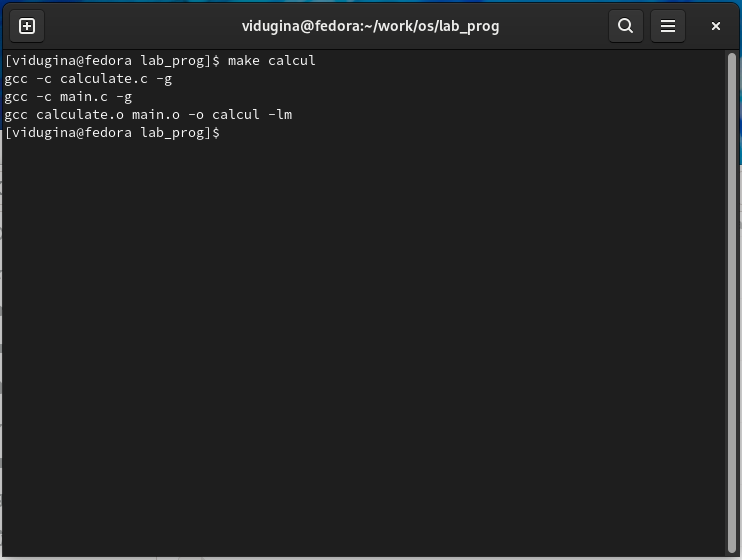
* calcul: Зависит от целей calculate.o и main.o. То есть будет выполнена, только при успешном выполнении этих команд (а значит необходимые файлы будут созданны). Последнее действие для компиляции программы.
* calculate.o: Зависит от файлов calculate.c и calculate.h. Компилирует файл calculate.с.
* main.o: Зависит от файлов main.c и calculate.h. Компилирует файл main.с.
* clean: Удаляет файлы calcul, те которые заканчиваются ".o".

#### 6. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile):

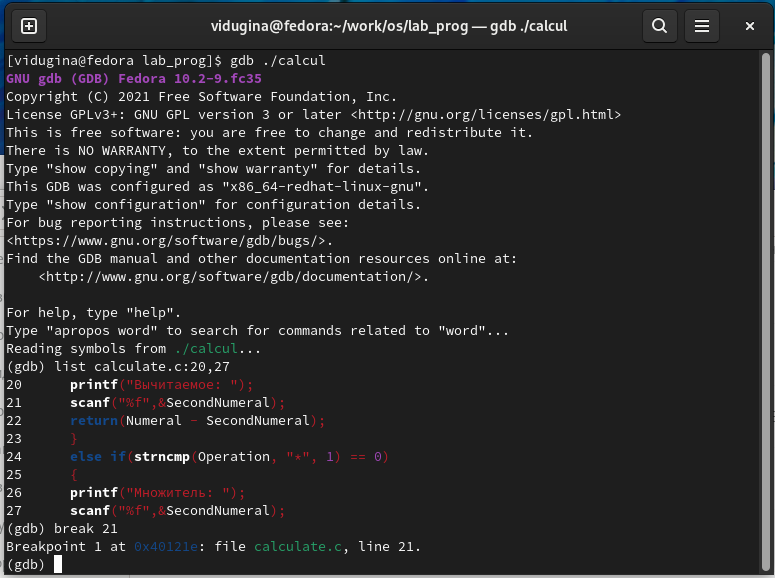
Добавляем в переменную CFLAGS -g. Данная переменая стоит в строках 13 и 16 и тем самым при компиляции добавится возможность работать в gdb.

 *Рис 6.1 - Изменённый Makefile*

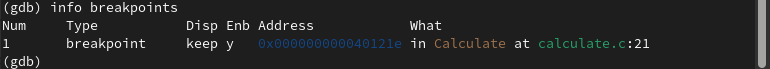
Компилируем программу, используя Makefile.

 *Рис 6.2*

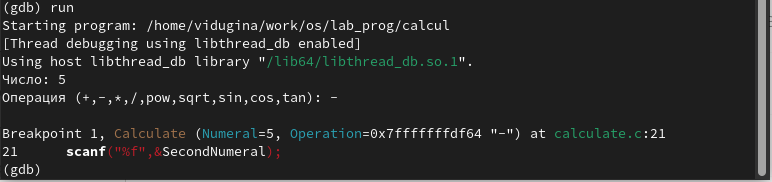
Запусeкаем отладчик GDB, загрузив в него calcul. Просматриваем строки 20-27 в файле calculate.c (list calculate.c:20,27) и добавлем точку останова на строке 21.

 *Рис 6.3*

Выводим информацию о точках останова.

 *Рис 6.4*

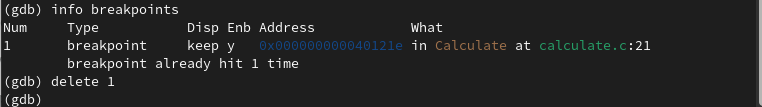
Запускаем программу и убеждаемся, что она останавливается в нужном месте.

 *Рис 6.5*

Смотрим значение переменной, используя сначала print, а потом display. Значения они выводят одинаковое, но оформляя по разному.

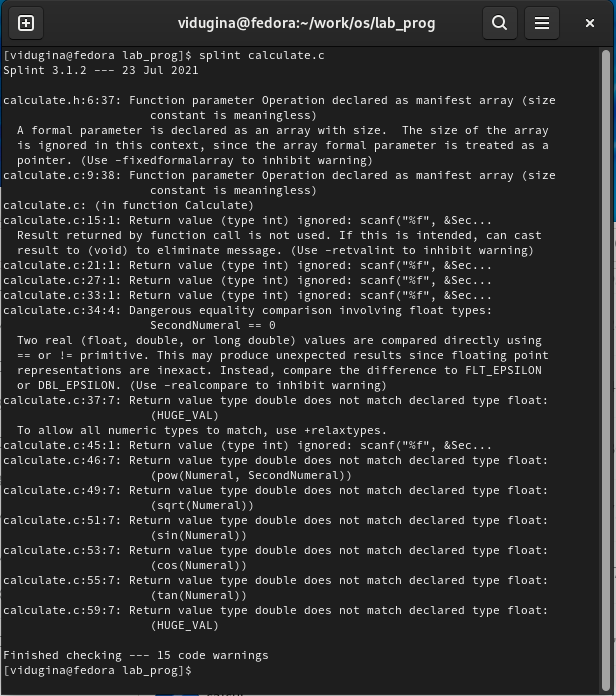
 *Рис 6.6*

Удаляем точку останова.

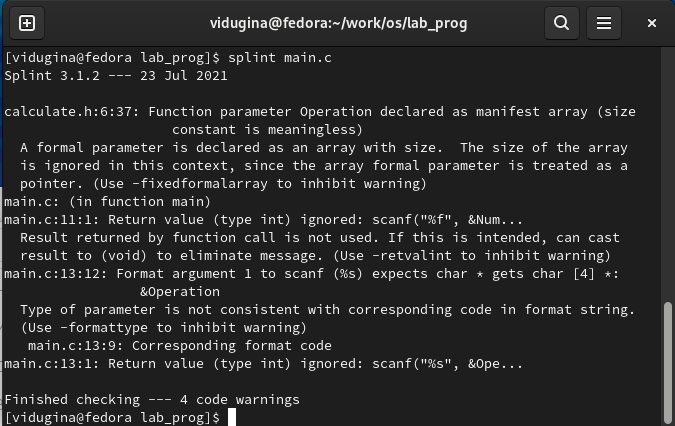
 *Рис 6.7*

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

Вводим splint calculate.c.

 *Рис 7.1*

Вводим splint main.c.

 *Рис 7.2*

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

### **Контрольные вопросы**

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

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

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

#### 4. Каково основное назначение компилятора языка С в UNIX?

#### 5. Для чего предназначена утилита make?

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

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

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

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

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

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

#### 12. Каковы основные задачи, решаемые программой splint?