**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Фундаментальная Информатика и Информационные технологии**

**ОТЧЕТ**

**по лабораторной работе № 13**

*дисциплина:* *операционные системы*

Узор-Ежикеме Чинечелум А.

НФИбд-03-21

1032215263

**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.

2. Создайте в нём файлы: calculate.h, calculate.c, main.c.

3. Выполните компиляцию программы посредством gcc:

4. При необходимости исправьте синтаксические ошибки.

5. Создайте Makefile со следующим содержанием:

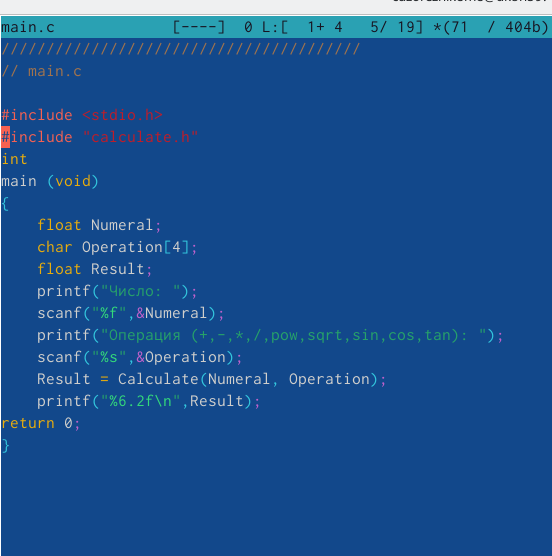
6. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile):

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

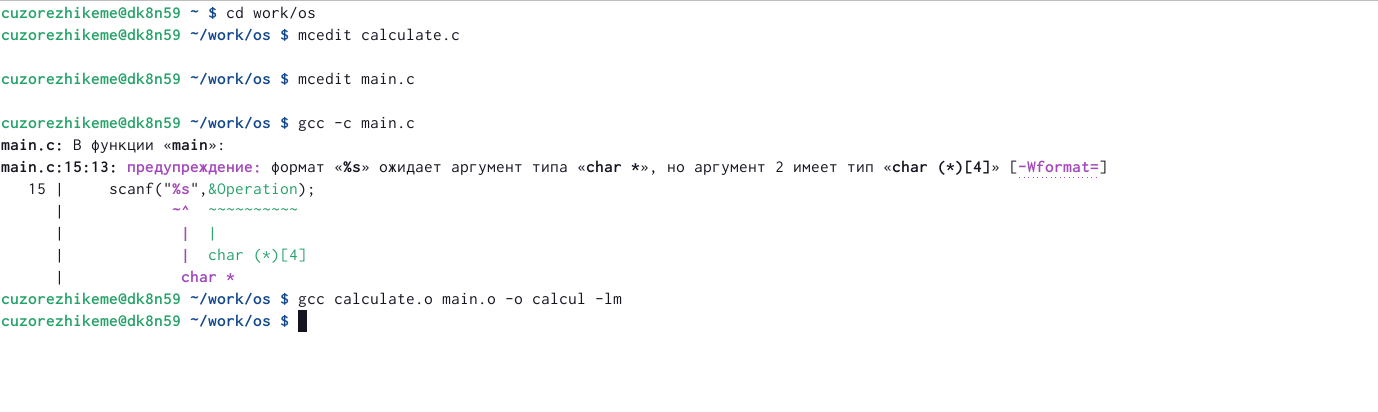
* Создал файлы: calculate.h, calculate.c, main.c. Я написал эскизы, которые будут представлять собой примитивный калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, извлекать квадратный корень, вычислять sin, cos, tan. При запуске он запросит первый номер, операцию, второй номер. После этого программа выведет результат и остановится. Файл интерфейса calculate.h, описывающий формат вызова функции калькулятора:



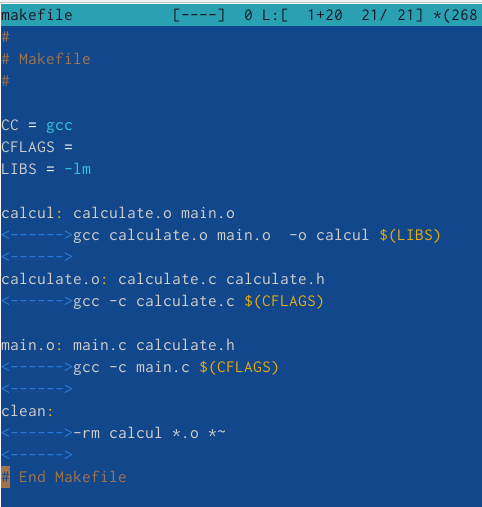
* Основной файл main.c, реализующий интерфейс пользователя к калькулятору:



* Скомпилировал программу с использованием gcc:



* Создал Makefile:



* Использовал gdb для отладки программы calcul.



**Контрольные вопросы**

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

- с помощью команды help.

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

- Суффикс - это условное обозначение добавления одного или нескольких символов к имени файла

3. Каково основное назначение компилятора языка С в UNIX?

- Компилятор берет рецепт (код) для новой программы (написанной на языке высокого уровня) и преобразует этот код в новый язык (машинный язык), который может быть понят самим компьютером.

4. Для чего предназначена утилита make?

- Инструмент, который автоматически определяет, какие исходные файлы программы необходимо перекомпилировать и/или связать.

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

- b main - Устанавливает точку останова в начале программы

b - Устанавливает точку останова в текущей строке

b N - Ставит точку останова на номер строки

b +N - Устанавливает точку останова на N строк ниже текущей строки.

bfn - Ставит точку останова в начале функции "fn"

d N - Удалить номер точки останова N

info break - список точек останова

r - запускайте программу до тех пор, пока не будет достигнута точка останова или ошибка

c - Продолжает выполнение программы до следующей точки останова или ошибки

f - Выполнять до тех пор, пока текущая функция не будет завершена

s - запускает следующую строку программы

s N - Запускает следующие N строк программы

n - Лайков, но он не переходит в функции

u N - Выполняйте до тех пор, пока не получите N строк перед текущей строкой

p var - Выводит текущее значение переменной "var"

bt - Выводит трассировку стека

u - Поднимается на уровень выше в стеке

d - Опускается на уровень ниже в стеке

q - Завершает работу gdb

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

- В принципе, если есть ошибка компиляции, то вы не можете ее запустить. Синтаксическая ошибка означает, что компилятор / интерпретатор говорит: "Я не знаю, что вы имеете в виду под этим", поэтому он не может генерировать какой-либо код или выполнять какие-либо команды, пока вы не исправите это.

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

- Самые основные инструменты - это редактор исходного кода и компилятор или интерпретатор, которые используются повсеместно и постоянно.

8. Каковы основные задачи, решаемые программой splint?

- Split особенно хорош при проверке типов назначений переменных и функций, эффективности, неиспользуемых переменных и идентификаторов функций, недоступного кода и возможных утечек памяти.