ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №13

*дисциплина: Операционные системы*

Лихтенштейн Алина Алексеевна

Содержание

# 1 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.

# 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.
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.

# 3 Теоретическое введение

Процесс разработки программного обеспечения обычно разделяется на следующие этапы:

* планирование, включающее сбор и анализ требований к функционалу и другим характеристикам разрабатываемого приложения;
* проектирование, включающее в себя разработку базовых алгоритмов и спецификаций,
* определение языка программирования;
* непосредственная разработка приложения;
* кодирование — по сути создание исходного текста программы (возможно в нескольких вариантах);
* анализ разработанного кода;
* сборка, компиляция и разработка исполняемого модуля;
* тестирование и отладка, сохранение произведённых изменений;
* документирование.

# 4 Выполнение лабораторной работы

# 5 Выполнение лабораторной работы

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 (рис. [1](#fig:001))



Figure 1: Создание файлов

1. Выполните компиляцию программы посредством gcc.
2. При необходимости исправьте синтаксические ошибки (рис. [2](#fig:002))

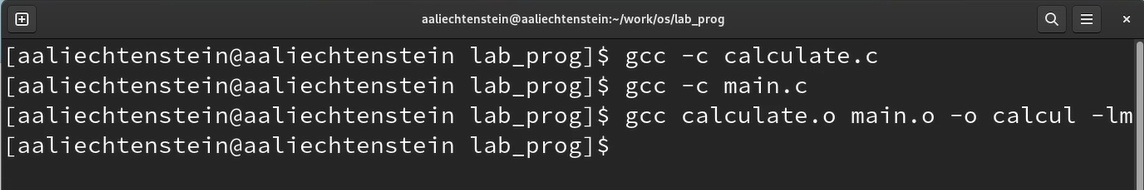


Figure 2: Компиляция

1. Создайте Makefile (рис. [3](#fig:003))

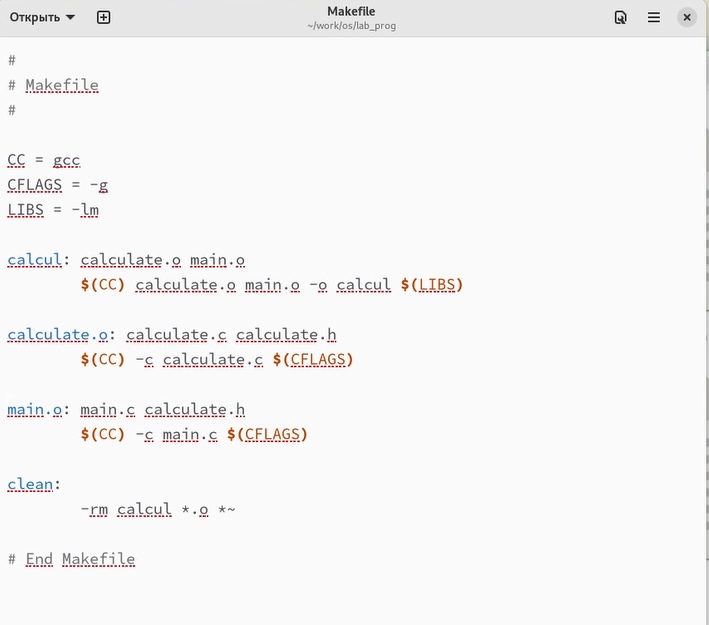


Figure 3: Makefile

Объяснение файла:

#Объявление переменных  
CC = gcc #компилятор  
CFLAGS = -g #опция, которая отладочную информацию положит в результирующий бинарн  
LIBS = -lm  
#Создаем файл calcul из файлов calculate.o main.o  
calcul: calculate.o main.o #ниже обращаемся к содержимому переменной  
 $(CC) calculate.o main.o -o calcul $(LIBS) #добавляем опцию  
#Здесь отражена строчка: gcc calculate.o main.o -o calcul -lm  
#Создаем файл calculate.o  
calculate.o: calculate.c calculate.h #  
 $(CC) -c calculate.c $(CFLAGS) #gcc -c calculate.c -g  
#Создаем файл main.o  
main.o: main.c calculate.h #gcc -c main.c -g  
 $(CC) -c main.c $(CFLAGS) #  
#  
clean: #при вызове make clean будем удалять все файлы с разрешением .о  
 -rm calcul \*.o \*~ #

1. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile)(рис. [4](#fig:004))

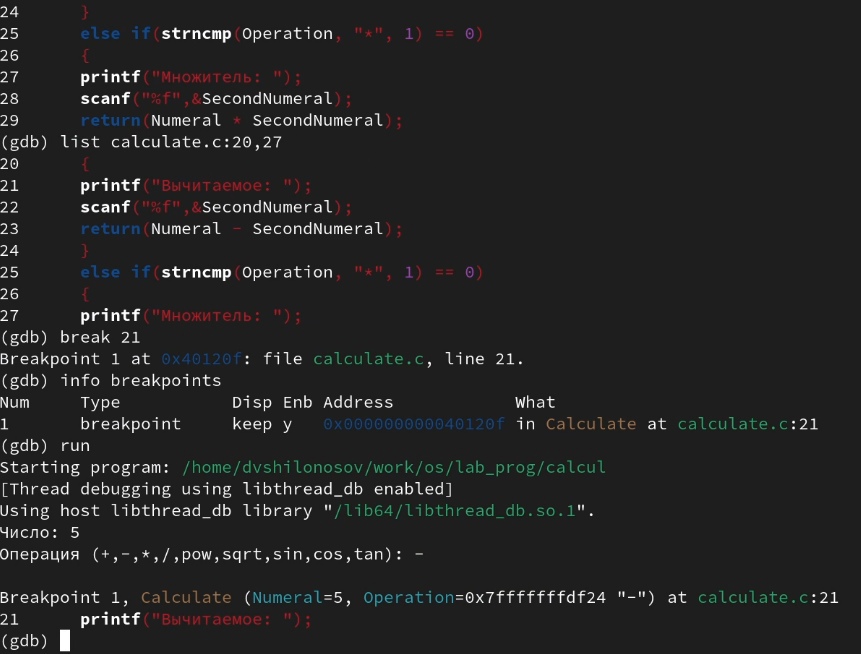


Figure 4: отладкa программы calcul

1. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c (рис. [5](#fig:005), [6](#fig:006))

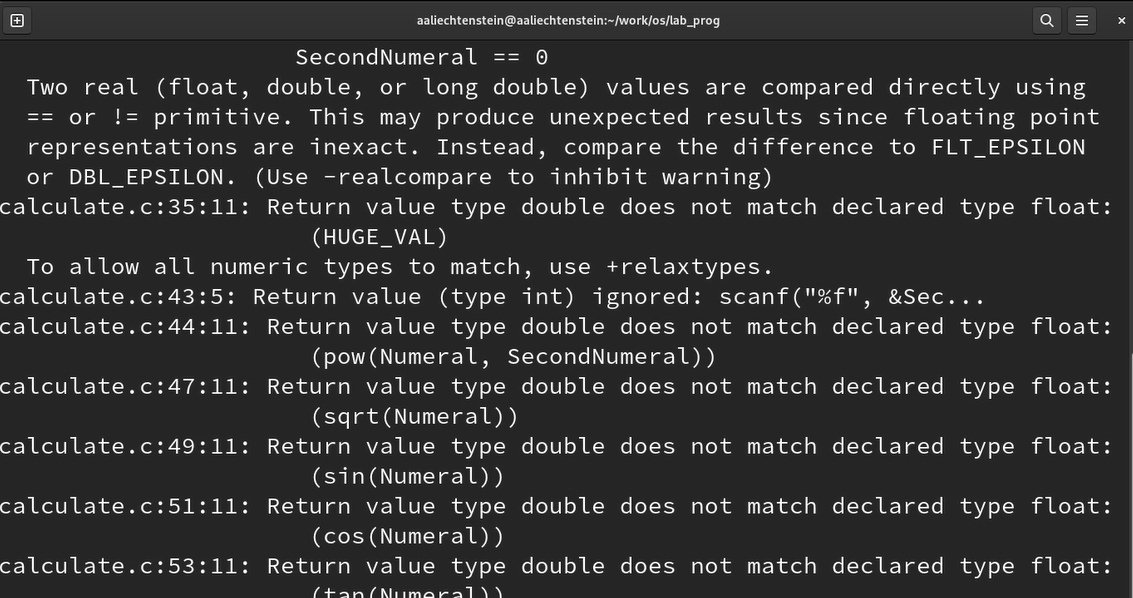


Figure 5: splint calculate.c

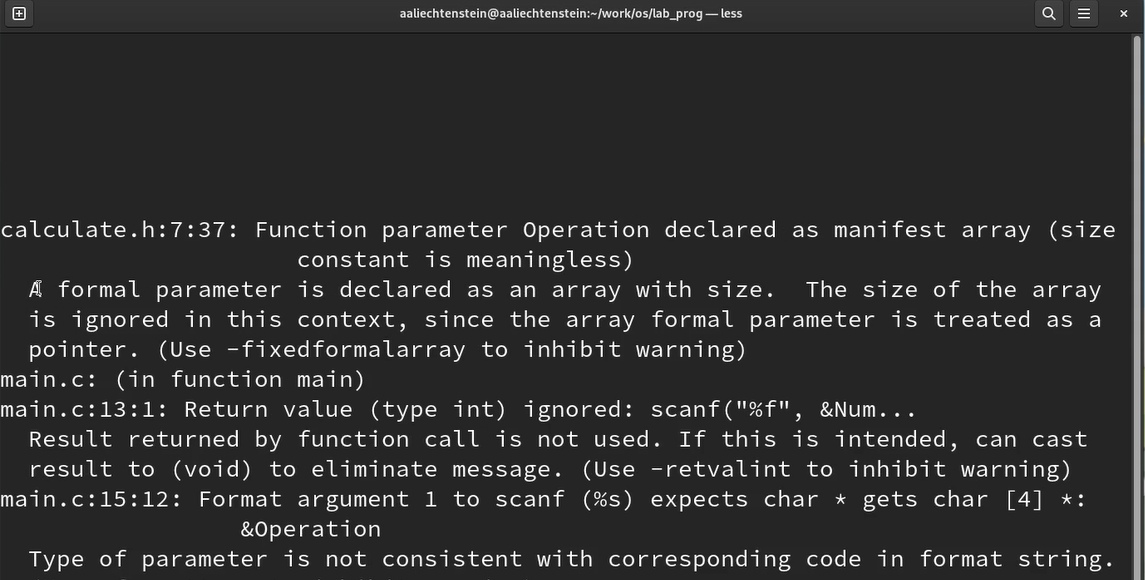


Figure 6: splint main.c

# 6 Выводы

При выполнении данной лабораторной работы 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.