Отчёт по лабораторной работе 13

Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

Осокин Г.И

Содержание

## 0.1 Цель работы

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.

# 1 Выполнение лабораторной работы

## 1.1 Написание программ

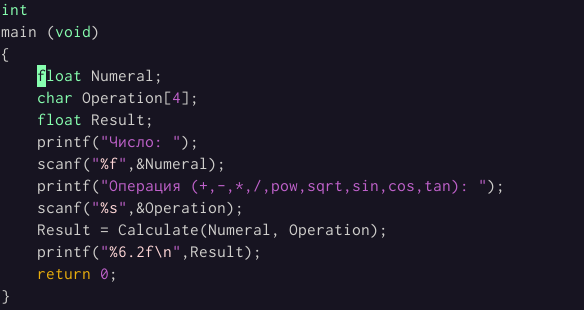
Создали директорию ~/work/os/lab\_prog:

Создание директории

Создание директории

Написали программу main.c:

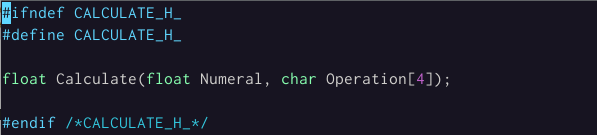
#include <stdio.h>  
#include "calculate.h"  
  
int main (void)  
{  
 float Numeral;  
 char Operation[4];  
 float Result;  
 printf("Число: ");  
 scanf("%f",&Numeral);  
 printf("Операция (+,-,\*,/,pow,sqrt,sin,cos,tan): ");  
 scanf("%s",Operation);  
 Result = Calculate(Numeral, Operation);  
 printf("%6.2f\n",Result);  
 return 0;  
}



main.c

Написали программу calculate.h:

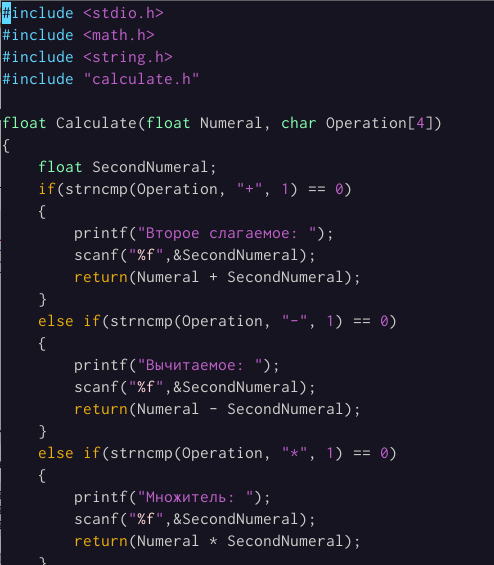
#ifndef CALCULATE\_H\_  
#define CALCULATE\_H\_  
  
float Calculate(float Numeral, char Operation[4]);  
  
#endif /\*CALCULATE\_H\_\*/



calculate.h

Написали программу calculate.c:

#include <stdio.h>  
#include <math.h>  
#include <string.h>  
#include "calculate.h"  
  
float Calculate(float Numeral, char Operation[4])  
{  
 float SecondNumeral;  
 if(strncmp(Operation, "+", 1) == 0)  
 {  
 printf("Второе слагаемое: ");  
 scanf("%f",&SecondNumeral);  
 return(Numeral + SecondNumeral);  
 }  
 else if(strncmp(Operation, "-", 1) == 0)  
 {  
 printf("Вычитаемое: ");  
 scanf("%f",&SecondNumeral);  
 return(Numeral - SecondNumeral);  
 }  
 else if(strncmp(Operation, "\*", 1) == 0)  
 {  
 printf("Множитель: ");  
 scanf("%f",&SecondNumeral);  
 return(Numeral \* SecondNumeral);  
 }  
 else if(strncmp(Operation, "/", 1) == 0)  
 {  
 printf("Делитель: ");  
 scanf("%f",&SecondNumeral);  
 if(SecondNumeral == 0)  
 {  
 printf("Ошибка: деление на ноль! ");  
 return(HUGE\_VAL);  
 }  
 else  
 return(Numeral / SecondNumeral);  
 }  
 else if(strncmp(Operation, "pow", 3) == 0)  
 {  
 printf("Степень: ");  
 scanf("%f",&SecondNumeral);  
 return(pow(Numeral, SecondNumeral));  
 }  
 else if(strncmp(Operation, "sqrt", 4) == 0)  
 return sqrt(Numeral);  
 else if(strncmp(Operation, "sin", 3) == 0)  
 return(sin(Numeral));  
 else if(strncmp(Operation, "cos", 3) == 0)  
 return(cos(Numeral));  
 else if(strncmp(Operation, "tan", 3) == 0)  
 return(tan(Numeral));  
 else  
 {  
 printf("Неправильно введено действие ");  
 return(HUGE\_VAL);  
 }  
}



calculate.c

Скомпилировали программы при помощи команд

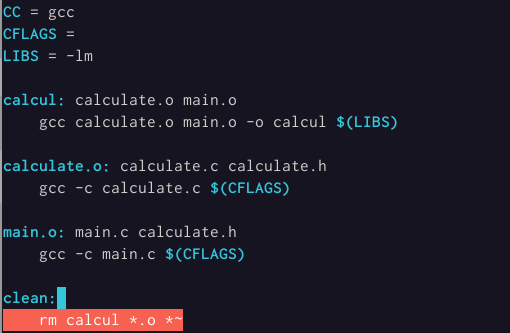
gcc -c calculate.c  
gcc -c main.c  
gcc calculate.o main.o -o calcul -lm

Компиляция программ

Компиляция программ

Для автоматизации сборки написали Makefile:

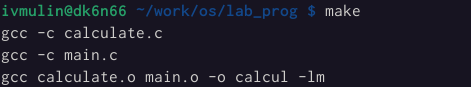
CC = gcc  
CFLAGS =  
LIBS = -lm  
  
calcul: calculate.o main.o  
 gcc calculate.o main.o -o calcul $(LIBS)  
  
calculate.o: calculate.c calculate.h  
 gcc -c calculate.c $(CFLAGS)  
  
main.o: main.c calculate.h  
 gcc -c main.c $(CFLAGS)  
  
clean:  
  
 -rm calcul \*.o \*~



Makefile

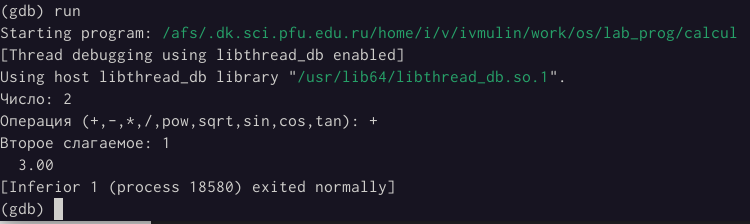
## 1.2 Отладка программы

Прежде чем начать отладку, запустили Makefile:



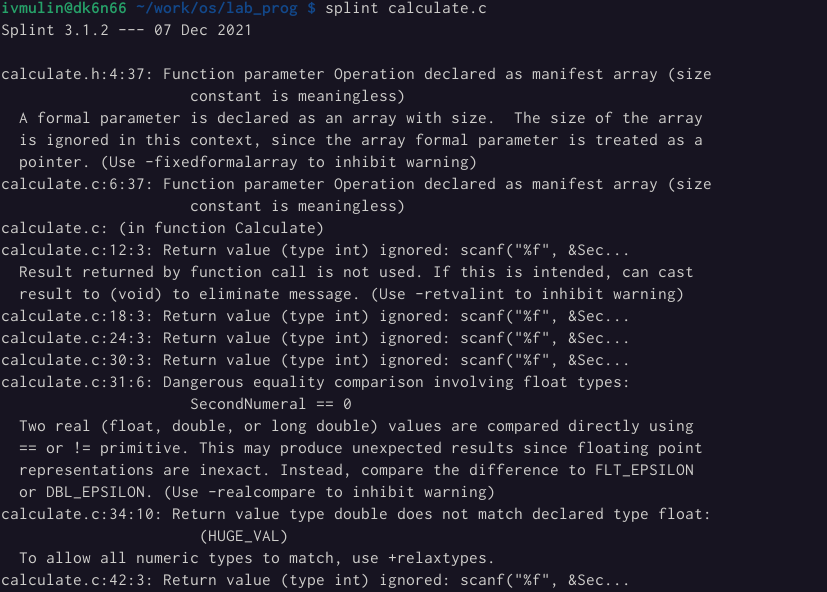
Запуск Makefile

Запустили отладчик gdb и провели отладку кода:

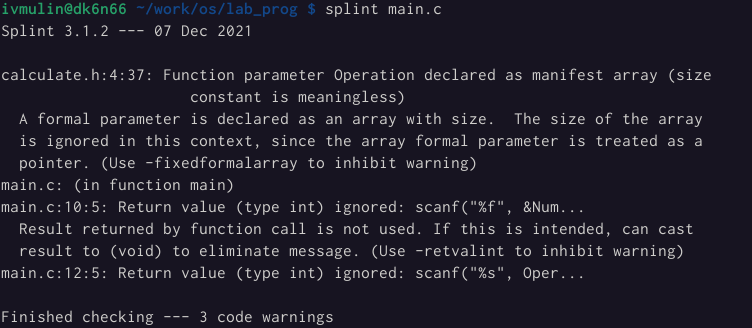


Отладка кода

При помощи команды splint проанализировали код программ calculate.c и main.c:



Отладка кода программы calculate.c



Отладка кода программы main.c

## 1.3 Ответы на контрольные вопросы

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

*Ответ*: при помощи программы man.

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

*Ответ*: 1. Выбор названия 2. Выбор языка программирования 3. Попытка выполнить работу всю зараз 4. Отрицание 5. Гнев 6. Торг 7. Депрессия 8. Принятие

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

*Ответ*: финальная часть названия программы, обычно отделяемая точкой.

1. Каково основное назначение компилятора языка С в UNIX?

*Ответ*: компилятор языка C в UNIX в основном компилирует программы языка C в UNIX, написанные на языке C в UNIX.

1. Для чего предназначена утилита make?

*Ответ*: для того чтобы делать.

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

*Ответ*: всё выполнил, как и просили.

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

*Ответ*: свойство - анализ кода; для анализа необходимо скомпилировать программу.

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

*Ответ*: см. ответ к вопросу 6.

1. Опишите по шагам схему отладки программы, которую вы использовали при выполнении лабораторной работы.

*Ответ*: 1. Вначале я запустил gdb 2. Затем я его закрыл

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

*Ответ*: когда я увидел реакцию компилятора на синтаксические ошибки в программе при его первом запуске,я был возмущён, поражён, обескуражен, ошеломлён, фрустрирован и изумлён. Но использовал совершенно другие выражения.

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

*Ответ*: здравый смысл и нормальный код.

1. Каковы основные задачи, решаемые программой splint?

*Ответ*: анализ кода.

## 1.4 Заключение

В ходе выполнения лабораторной работы были изучены простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями. Цель работы была достигнута.