

Лабораторная работа №14

Егорова Александра¹

3 июня, 2021, Москва, Россия

¹RUDN University, Moscow, Russian Federation

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) В домашнем каталоге создаю подкаталог `~/work/os/lab_prog`.
Создала в каталоге файлы. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять \sin , \cos , \tan . (рис. -fig. 1) (рис. -fig. 2)

```
aegorova@aegorova:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[aegorova@aegorova ~]$ mkdir -p ~/work/os/lab_prog
```

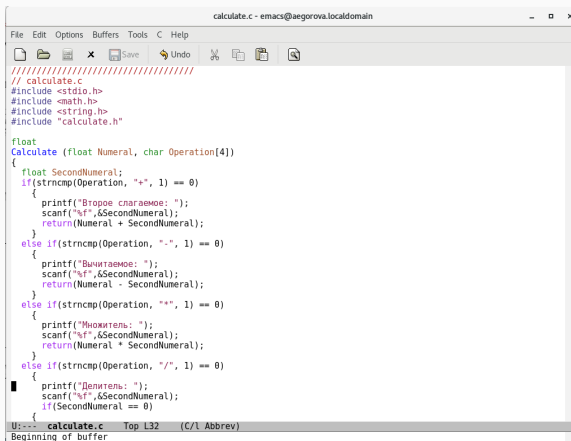
Figure 1: Создание подкаталога

```
[aegorova@aegorova ~]$ cd ~/work/os/lab_prog  
[aegorova@aegorova lab_prog]$ touch calculate.h  calculate.c  main.c  
[aegorova@aegorova lab_prog]$ ls  
calculate.c  calculate.h  main.c
```

Figure 2: Создание файлов

Выполнение лабораторной работы

Реализация функций калькулятора в файле calculate.c. Интерфейсный файл calculate.h , описывающий формат вызова функции-калькулятора. Основной файл main.c , реализующий интерфейс пользователя к калькулятору. (рис. -fig. 3)

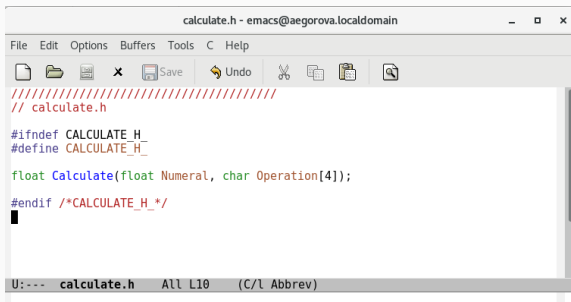


```
calculate.c - emacs@aegorova.localdomain
File Edit Options Buffers Tools C Help
[Icons] Save Undo [Icons]
////////////////////////////////////
// calculate.c
#include <stdio.h>
#include <math.h>
#include <string.h>
#include "calculate.h"

float
Calculate (float Numeral, char Operation[4])
{
    float SecondNumeral;
    if (strcmp(Operation, "+") == 0)
    {
        printf("Второе слагаемое: ");
        scanf("%f", &SecondNumeral);
        return(Numeral + SecondNumeral);
    }
    else if (strcmp(Operation, "-") == 0)
    {
        printf("Вычитаемое: ");
        scanf("%f", &SecondNumeral);
        return(Numeral - SecondNumeral);
    }
    else if (strcmp(Operation, "*") == 0)
    {
        printf("Множитель: ");
        scanf("%f", &SecondNumeral);
        return(Numeral * SecondNumeral);
    }
    else if (strcmp(Operation, "/") == 0)
    {
        printf("Делитель: ");
        scanf("%f", &SecondNumeral);
        if (SecondNumeral == 0)
        {
            // Error handling for division by zero
        }
    }
}
```

U:-- calculate.c Top L32 (C/L Abbrev)
Beginning of buffer

calculate.h (рис. -fig. 4)

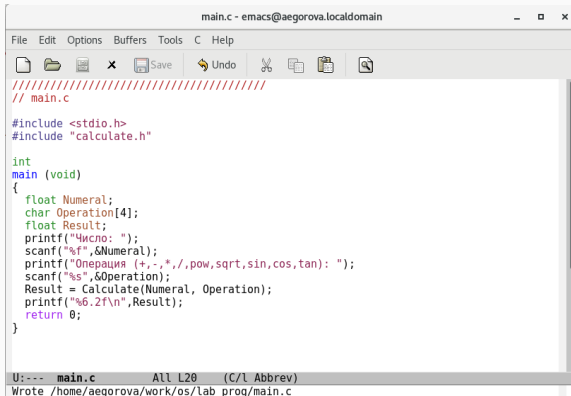


```
calculate.h - emacs@aegorova.localdomain
File Edit Options Buffers Tools C Help
////////////////////
// calculate.h
#ifndef CALCULATE_H
#define CALCULATE_H
float Calculate(float Numeral, char Operation[4]);
#endif /*CALCULATE_H*/
█
U: --- calculate.h All L10 (C/l Abbrev)
```

Figure 4: Файл calculate.h

Выполнение лабораторной работы

main.c. (рис. -fig. 5)



```
main.c - emacs@aegorova.localdomain
File Edit Options Buffers Tools C Help

////////////////////////////////////
// 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;
}

U:--- main.c All L20 (C/l Abbrev)
Wrote /home/aegorova/work/os/lab_prog/main.c
```

Figure 5: Файл main.c

- 3) Выполнила компиляцию программы посредством gcc : gcc -c calculate.c; gcc -c main.c; gcc calculate.o main.o -o calcul -lm. (рис. -fig. 6)

```
[aegorova@aegorova lab_prog]$ gcc -c calculate.c
[1] Done                               emacs
[2]- Done                               emacs
[3]+ Done                               emacs
[aegorova@aegorova lab_prog]$ gcc -c main.c
[aegorova@aegorova lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

Figure 6: Компиляция программы

С помощью утилиты splint попробовала проанализировать коды файлов calculate.c и main.c. (рис. -fig. 7)

```
[aegorova@aegorova lab_prog]$ splint calculate.c
Splint 3.1.2 --- 11 Oct 2015

calculate.h:7:37: Function parameter Operation declared as manifest array (size
        constant is meaningless)
    A formal parameter is declared as an array with size. The size of the array
    is ignored in this context, since the array formal parameter is treated as a
    pointer. (Use -fixedformalarray to inhibit warning)
calculate.c:9:31: Function parameter Operation declared as manifest array (size
        constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:15:7: Return value (type int) ignored: scanf("%f", &Sec...
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalint to inhibit warning)
calculate.c:21:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:27:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:33:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:10: Dangerous equality comparison involving float types:
        SecondNumeral == 0
    Two real (float, double, or long double) values are compared directly using
    == or != primitive. This may produce unexpected results since floating point
    representations are inexact. Instead, compare the difference to FLT_EPSILON
    or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:37:10: Return value type double does not match declared type float:
        (HUGE_VAL)
```

Figure 7: splint

В ходе выполнения 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.