

Лабораторная работа №13

Архитектура компьютеров

Скандарова Полина Юрьевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

Список иллюстраций

2.1	Перечисленные действия	6
2.2	Реализация функций калькулятора в файле calculate.h	7
2.3	Интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора	7
2.4	Основной файл main.c, реализующий интерфейс пользователя к калькулятору	8
2.5	3. Выполните компиляцию программы посредством gcc:	8
2.6	Создайте Makefile со следующим содержанием:	8

Список таблиц

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 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. fig. 2.1) (рис. fig. 2.2) (рис. fig. 2.3) (рис. fig. 2.4).

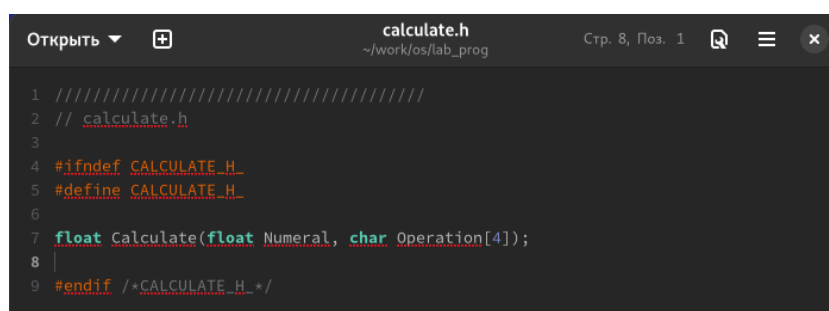
```
[pyskandarova@fedora ~]$ mkdir ~/work/os/lab_prog  
[pyskandarova@fedora ~]$ cd ~/work/os/lab_prog  
[pyskandarova@fedora lab_prog]$ touch calculate.h calculate.c main.c
```

Рис. 2.1: Перечисленные действия



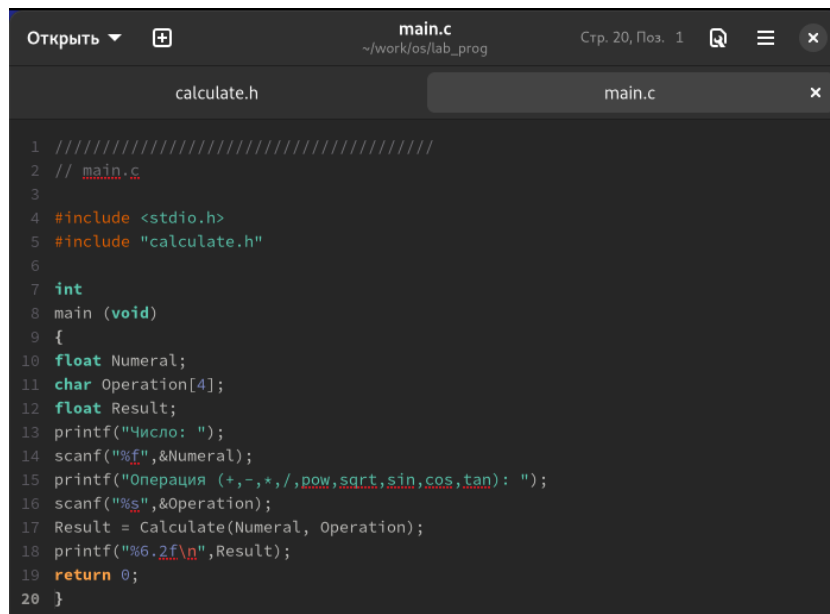
```
1 //////////////////////////////////////////////////
2 // calculate.c
3
4 #include <stdio.h>
5 #include <math.h>
6 #include <string.h>
7 #include "calculate.h"
8
9 float
10 Calculate(float Numeral, char Operation[4])
11 {
12     float SecondNumeral;
13     if(strncmp(Operation, "+", 1) == 0)
14     {
15         printf("Второе слагаемое: ");
16         scanf("%f", &SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strncmp(Operation, "-", 1) == 0)
20     {
21         printf("Вычитаемое: ");
22         scanf("%f", &SecondNumeral);
23         return(Numeral - SecondNumeral);
24     }
25     else if(strncmp(Operation, "*", 1) == 0)
26     {
27         printf("Множитель: ");
28         scanf("%f", &SecondNumeral);
29         return(Numeral * SecondNumeral);
30     }
31     else if(strncmp(Operation, "/", 1) == 0)
32     {
33         printf("Делитель: ");
34         scanf("%f", &SecondNumeral);
35         if(SecondNumeral == 0)
36         {
37             printf("Ошибка: деление на ноль! ");
38             return(HUGE_VAL);
39         }
40         else
41             return(Numeral / SecondNumeral);
42     }
43     else if(strncmp(Operation, "pow", 3) == 0)
44     {
45         printf("Степень: ");
```

Рис. 2.2: Реализация функций калькулятора в файле calculate.h



```
1 //////////////////////////////////////////////////
2 // calculate.h
3
4 #ifndef CALCULATE_H_
5 #define CALCULATE_H_
6
7 float Calculate(float Numeral, char Operation[4]);
8
9 #endif /*CALCULATE_H_*/
```

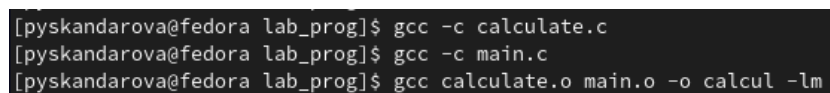
Рис. 2.3: Интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора



```
1 //////////////////////////////////////////////////
2 // main.c
3
4 #include <stdio.h>
5 #include "calculate.h"
6
7 int
8 main (void)
9 {
10     float Numeral;
11     char Operation[4];
12     float Result;
13     printf("Число: ");
14     scanf("%f",&Numeral);
15     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
16     scanf("%s",&Operation);
17     Result = Calculate(Numeral, Operation);
18     printf("%6.2f\n",Result);
19     return 0;
20 }
```

Рис. 2.4: Основной файл main.c, реализующий интерфейс пользователя к калькулятору

3. Выполните компиляцию программы посредством gcc: (рис. fig. 2.5).

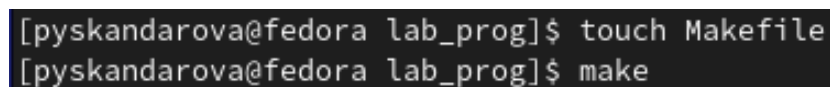


```
[pyskandarova@fedora lab_prog]$ gcc -c calculate.c
[pyskandarova@fedora lab_prog]$ gcc -c main.c
[pyskandarova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

Рис. 2.5: 3. Выполните компиляцию программы посредством gcc:

4. При необходимости исправьте синтаксические ошибки.

5. Создайте Makefile со следующим содержанием: (рис. fig. 2.6).



```
[pyskandarova@fedora lab_prog]$ touch Makefile
[pyskandarova@fedora lab_prog]$ make
```

Рис. 2.6: Создайте Makefile со следующим содержанием:

3 Выводы

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.