

Лабораторная работа №13

Архитектура компьютеров

Скандарова П. Ю.

29 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Скандарова Полина Юрьевна
- Российский университет дружбы народов
- 1132221815@pfur.ru
- <https://pyskandarova.github.io/ru/>

Вводная часть

Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.

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.

```
[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
```

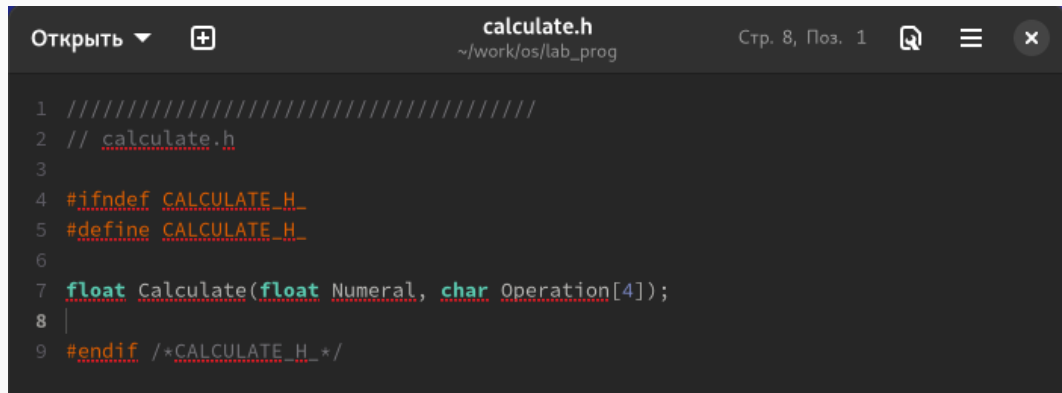
Рис. 1: Перечисленные действия

Выполнение лабораторной работы



The screenshot shows a code editor window titled "calculate.c" with the file path "~/work/05/lab_prog". The code is a C program for a simple calculator. It includes headers for stdio, math, and string, and a custom header "calculate.h". The main function "Calculate" takes a float "Numeral" and a character array "Operation" of size 4. It uses a series of if-else statements to check the operation: addition (+), subtraction (-), multiplication (*), and division (/). For each operation, it prompts the user for a second number, reads it, and performs the calculation. Division includes a check for a zero denominator, returning "Ошибка: деление на ноль!" if it occurs. A "pow" operation is also partially implemented. The code is line-numbered from 1 to 45.

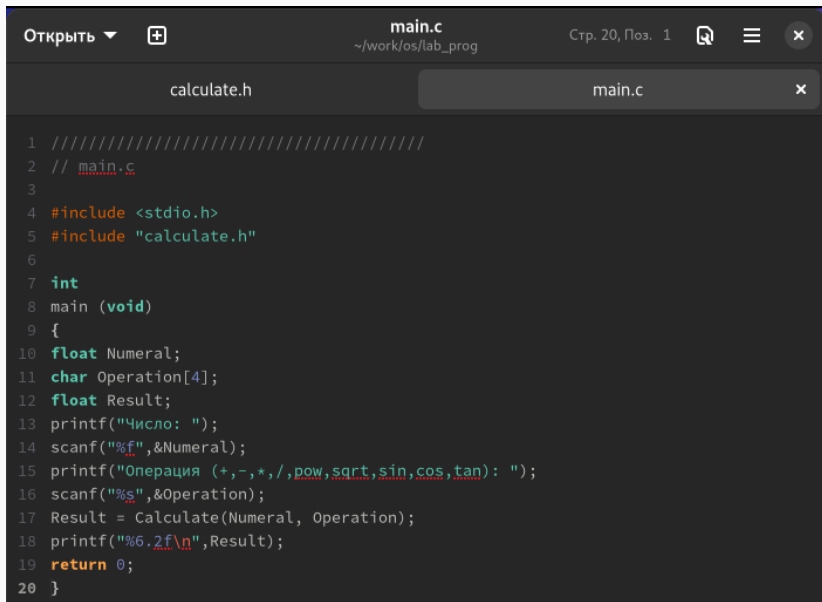
```
1 //////////////////////////////////////////////////
2 // calculate.c
3
4 #include <stdio.h>
5 #include <math.h>
6 #include <string.h>
7 #include "calculate.h"
8
9 float
10 Calculate(float Numeral, char Operation[4])
11 {
12     float SecondNumeral;
13     if(strcmp(Operation, "+") == 0)
14     {
15         printf("Второе слагаемое: ");
16         scanf("%f", &SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strcmp(Operation, "-") == 0)
20     {
21         printf("Вычитаемое: ");
22         scanf("%f", &SecondNumeral);
23         return(Numeral - SecondNumeral);
24     }
25     else if(strcmp(Operation, "*") == 0)
26     {
27         printf("Множитель: ");
28         scanf("%f", &SecondNumeral);
29         return(Numeral * SecondNumeral);
30     }
31     else if(strcmp(Operation, "/") == 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(strcmp(Operation, "pow", 3) == 0)
44     {
45         printf("Второе слагаемое: ");
```

```
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. 3: Интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора

Выполнение лабораторной работы



```
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("%.2f\n",Result);
19 return 0;
20 }
```

3. Выполните компиляцию программы посредством gcc:

```
[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. 5: 3. Выполните компиляцию программы посредством gcc:

4. При необходимости исправьте синтаксические ошибки.
5. Создайте Makefile со следующим содержанием:

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

Рис. 6: Создайте Makefile со следующим содержанием:

Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.

Неспасибо за невнимание
