

# Лабораторная работа №14

---

Калистратова Ксения Евгеньевна

2 июня, 2021, Москва, Россия

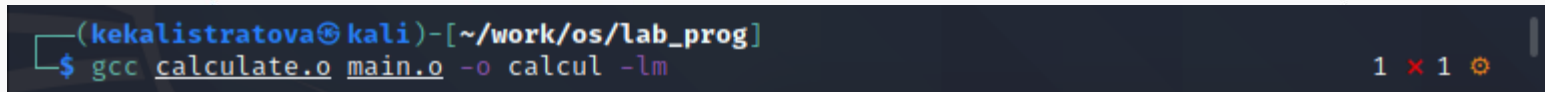
<sup>1</sup>RUDN University, Moscow, Russian Federation

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. Познакомиться со стандартным средством для компиляции программ в ОС типа UNIX - GCC (GNU Compiler Collection).
2. Познакомиться с отладчиком GDB (GNU Debugger).
3. Познакомиться с утилитой splint.
4. В ходе работы проанализировать и выполнить данные программы.
5. Выполнить отчет.

# Выполнение лабораторной работы

В ходе работы мы создали каталог, а в нем 3 файла: calculate.h, calculate.c, main.c. Это примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. Мы выполнили компиляцию программы посредством gcc. (рис. 1)

A terminal window with a dark background. The prompt is `(kekalistratova@kali) - [~/work/os/lab_prog]`. The command entered is `$ gcc calculate.o main.o -o calcul -lm`. On the right side of the terminal, there is a status bar showing `1 x 1` and a gear icon.

```
(kekalistratova@kali) - [~/work/os/lab_prog]
$ gcc calculate.o main.o -o calcul -lm
```

Figure 1: Компиляция программы

# Выполнение лабораторной работы

Мы создали Makefile. С помощью gdb выполнили отладку программы calcul. (рис. 2)

```
(gdb) list calculate.c:20,27
20      {
21          printf("Вычитаемое: ");
22          scanf("%f",&SecondNumeral);
23          return(Numeral-SecondNumeral);
24      }
25      else if(strncmp(Operation,"*",1) == 0)
26      {
27          printf("Множитель: ");
(gdb) break 21
Breakpoint 1 at 0x555555400955: file calculate.c, line 21.
(gdb) █
```

Figure 2: Установка точки останова

# Выполнение лабораторной работы

С помощью утилиты splint проанализировали коды файлов calculate.c и main.c. (рис. 3)

```
Splint 3.1.2 --- 13 Jan 2021

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)
main.c: (in function main)
main.c:14:3: Return value (type int) ignored: scanf("%f", &Num...
    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)
main.c:16:14: Unallocated storage Operation passed as out parameter to scanf:
                    Operation
    An rvalue is used that may not be initialized to a value on some execution
    path. (Use -usedef to inhibit warning)
main.c:16:3: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 4 code warnings
```

Figure 3: splint main.c

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 C калькулятора с простейшими функциями.