

Лабораторная работа №12. Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

Цыганков А. Р.

13 апреля 2024

Информация

Докладчик

Вводная часть

Актуальность

- Навыки разработки и отладки программ необходимы любому человеку, связанному с IT и техническими науками.

Объект и предмет исследования

- Командный процессор ОС UNIX

Цели и задачи

- 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.

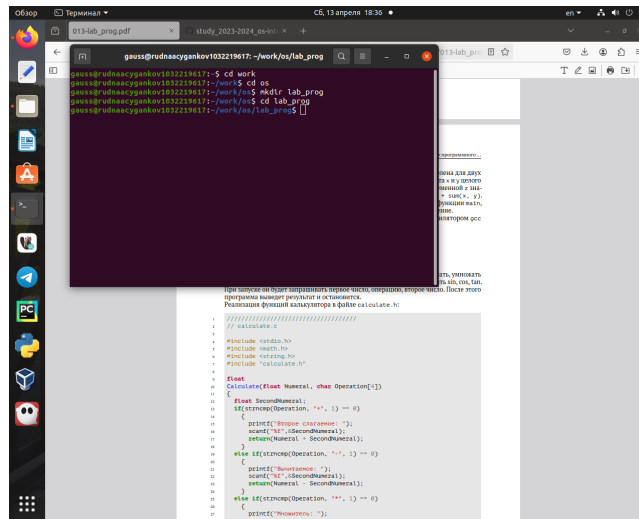


Рис. 1: Создание каталога

Выполнение заданий.

В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`. (рис. @fig:001)

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:002, @fig:003, @fig:004).

Выполните компиляцию программы посредством `gcc`. (рис. @fig:005)

При необходимости исправьте синтаксические ошибки (рис. @fig:006)

Создайте `Makefile` (рис. @fig:007)

С помощью `gdb` выполните отладку программы `calcul` (перед использованием `gdb` исправьте `Makefile`) (рис. @fig:008, @fig:009, @fig:010)

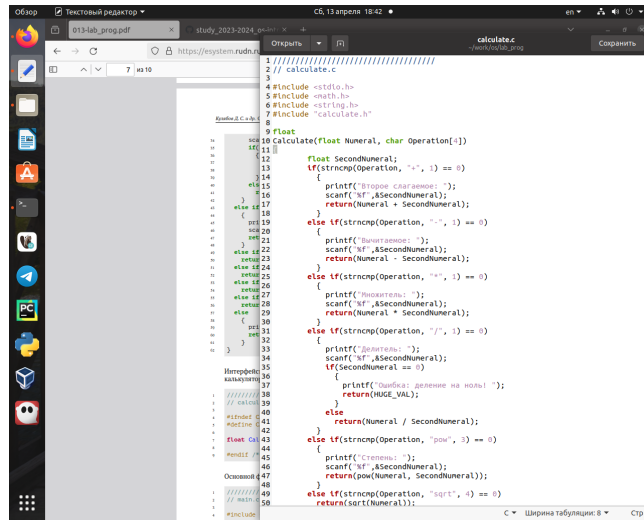
С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c` (рис. @fig:011, @fig:012)

Результаты

2

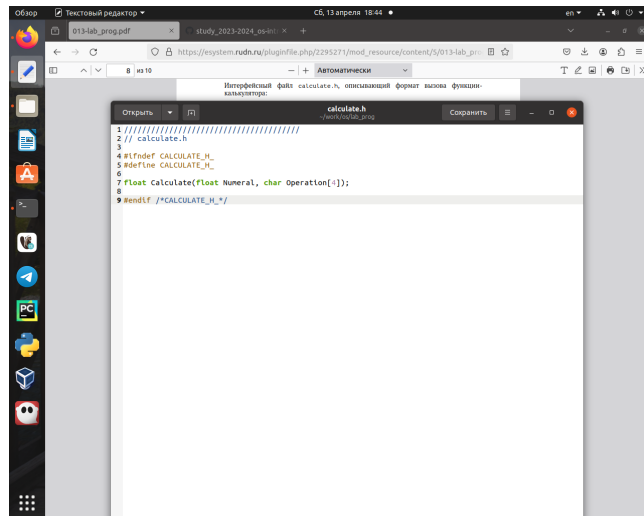
Вывод:

В ходе выполнения лабораторной работы я приобрел простейшие навыки разработки и отладки приложений в ОС типа UNIX/Linux на примере



```
1 // calculate.c
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[])
11 {
12     float SecondNumeral;
13     if(strlen(Operation) == 0)
14     {
15         printf("Стоп! Ошибка!\n");
16         scanf("%f", &SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strlen(Operation) == 1)
20     {
21         printf("Ошибка!\n");
22         scanf("%f", &SecondNumeral);
23         return(Numeral - SecondNumeral);
24     }
25     else if(strlen(Operation) == 2)
26     {
27         printf("Ошибка!\n");
28         scanf("%f", &SecondNumeral);
29         return(Numeral * SecondNumeral);
30     }
31     else if(strlen(Operation) == 3)
32     {
33         printf("Ошибка!\n");
34         scanf("%f", &SecondNumeral);
35         if(SecondNumeral == 0)
36         {
37             printf("Числа, деление на ноль!\n");
38             return(HUGE_VAL);
39         }
40         else
41             return(Numeral / SecondNumeral);
42     }
43     else if(strlen(Operation) == 4)
44     {
45         printf("Ошибка!\n");
46         scanf("%f", &SecondNumeral);
47         return(pow(Numeral, SecondNumeral));
48     }
49     else if(strlen(Operation) == 5)
50         return(sqrt(Numeral));
51 }
```

Рис. 2: calculate.c



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

Рис. 3: calculate.h

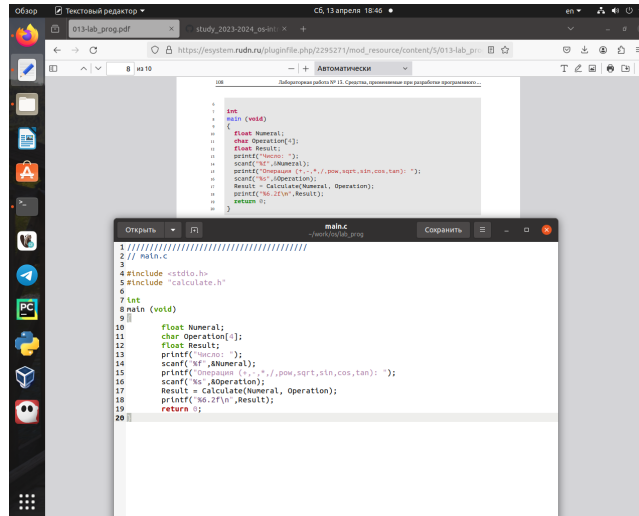


Рис. 4: main.c

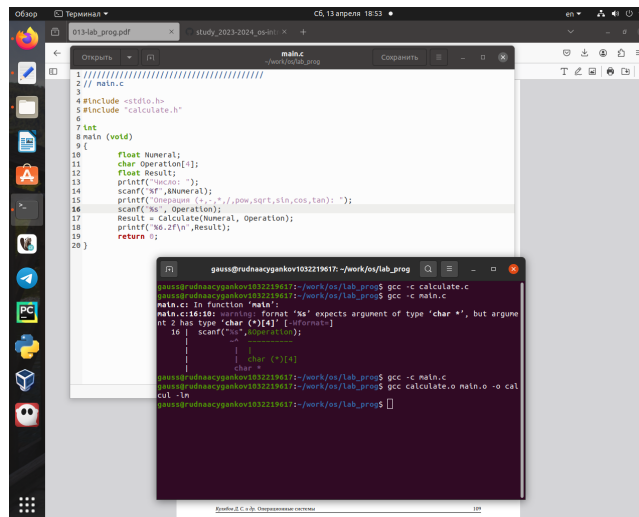


Рис. 5: Попытка компиляции

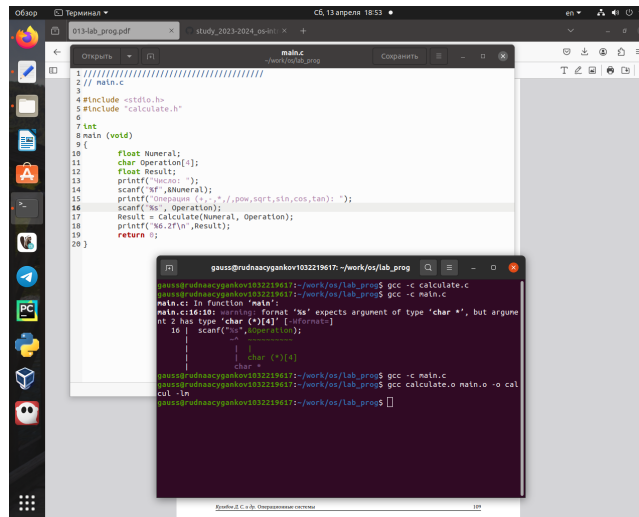


Рис. 6: main.c убрал & перед Operation

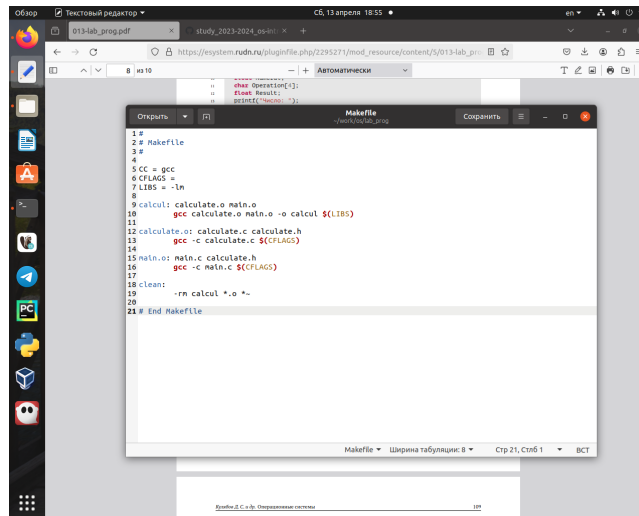


Рис. 7: Компиляция

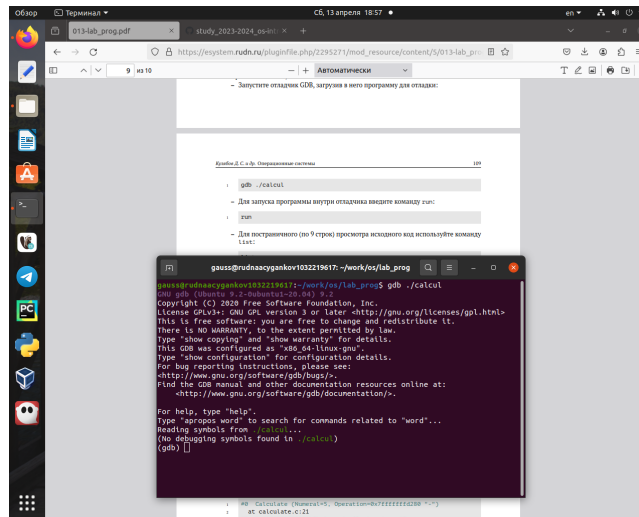


Рис. 8: Запуск программы

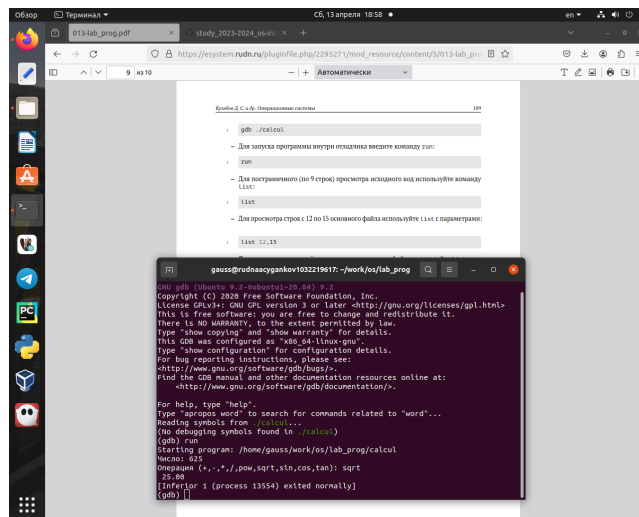
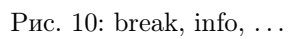


Рис. 9: Работа программы



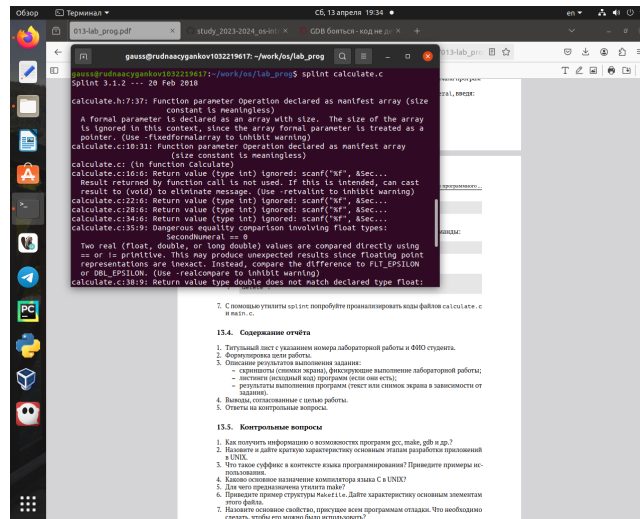


Рис. 12: splint main.c

создания на языке C калькулятора с простейшими функциями.