

Лабораторная работа No 13

**Средства, применяемые при разработке программного обеспечения в
ОС типа UNIX/Linux**

Акопян Сатеник Манвеловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	рисунок 1	6
2.2	рисунок 2	6
2.3	рисунок 3	6
2.4	рисунок 4	7
2.5	рисунок 5	7
2.6	рисунок 6	8
2.7	рисунок 8	8
2.8	рисунок 7	9
2.9	рисунок 9	9
2.10	рисунок 11	10
2.11	рисунок 12	10
2.12	рисунок 13	11

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.

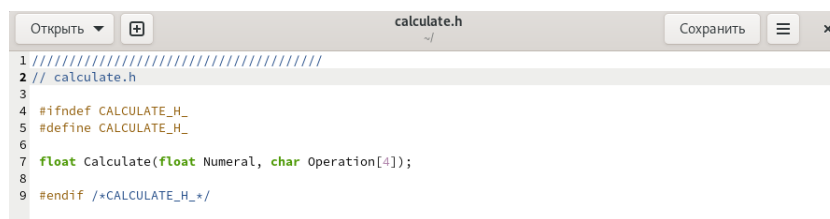
```
[smakopyan@fedora ~]$ cd work
[smakopyan@fedora work]$ mkdir os
[smakopyan@fedora work]$ cd os
[smakopyan@fedora os]$ mkdir lab_prog
[smakopyan@fedora os]$ cd lab_prog
[smakopyan@fedora lab_prog]$
```

Рис. 2.1: рисунок 1

2. Создайте в нём файлы: calculate.h, calculate.c, main.c.

```
[smakopyan@fedora lab_prog]$ touch calculate.h calculate.c main.c
[smakopyan@fedora lab_prog]$ ls
calculate.c calculate.h main.c
[smakopyan@fedora lab_prog]$
```

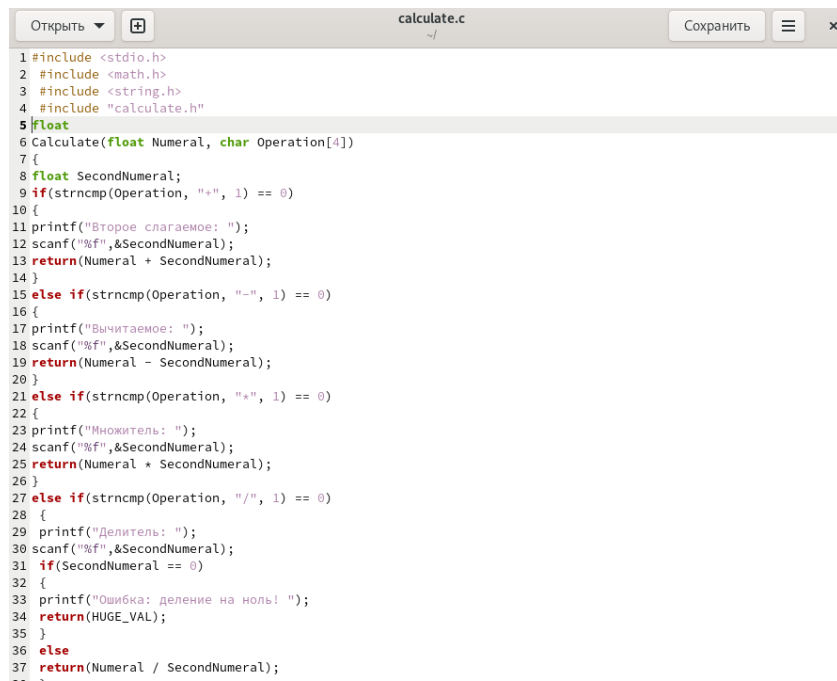
Рис. 2.2: рисунок 2



The screenshot shows a text editor window titled 'calculate.h' with a menu bar containing 'Открыть', '+', 'Сохранить', and a close button. The editor displays the following C header file code:

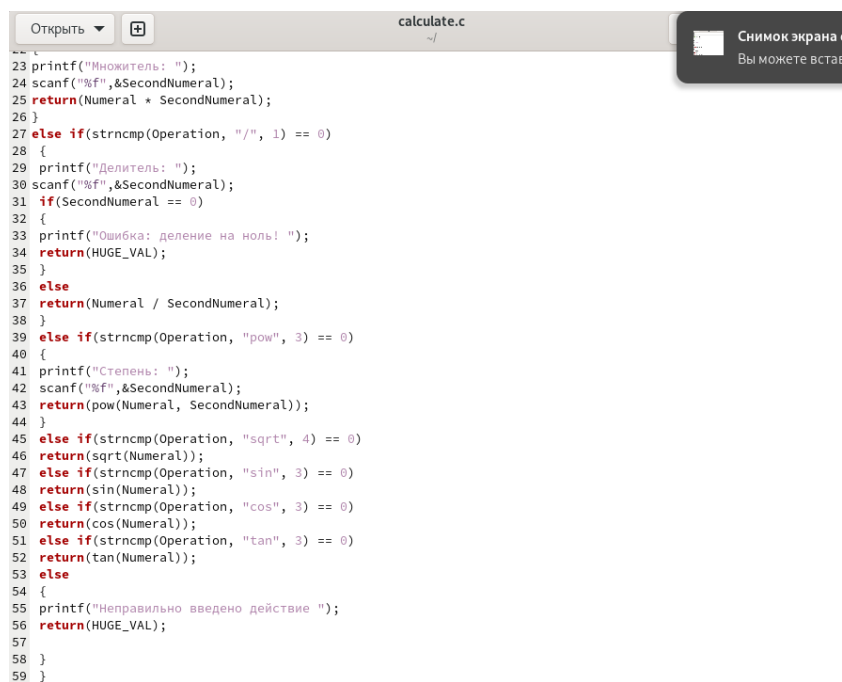
```
1 //////////////////////////////////////////////////
2 // calculate.h
3
4 #ifndef CALCULATE_H_
5 #define CALCULATE_H_
6
7 float Calculate(float Numeral, char Operation[4]);
8
9 #endif /*CALCULATE_H_*/
```

Рис. 2.3: рисунок 3



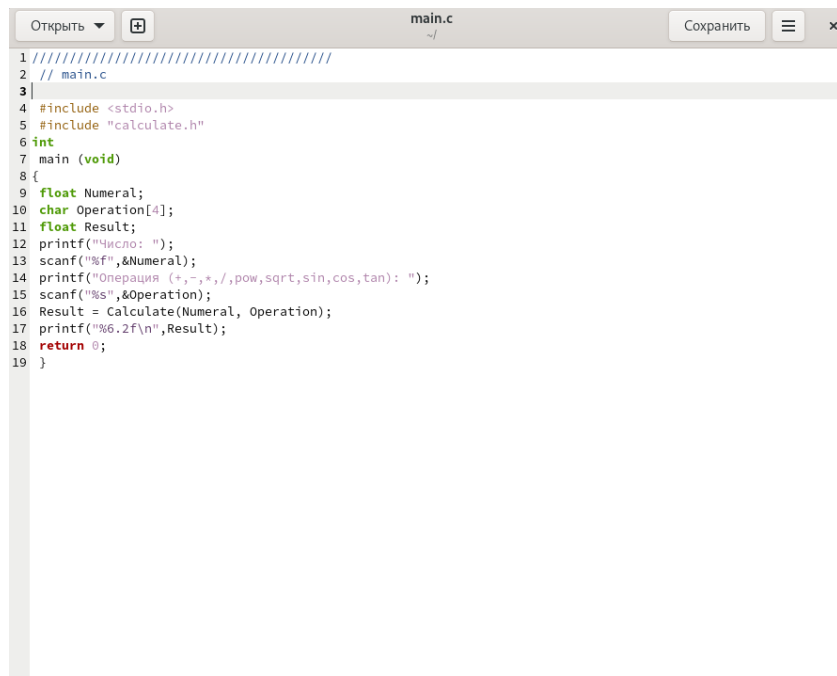
```
1 #include <stdio.h>
2 #include <math.h>
3 #include <string.h>
4 #include "calculate.h"
5 float
6 Calculate(float Numeral, char Operation[4])
7 {
8     float SecondNumeral;
9     if(strcmp(Operation, "+") == 0)
10     {
11         printf("Второе слагаемое: ");
12         scanf("%f",&SecondNumeral);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

Рис. 2.4: рисунок 4



```
23 printf("Множитель: ");
24 scanf("%f",&SecondNumeral);
25 return(Numeral * SecondNumeral);
26 }
27 else if(strcmp(Operation, "/", 1) == 0)
28 {
29     printf("Делитель: ");
30     scanf("%f",&SecondNumeral);
31     if(SecondNumeral == 0)
32     {
33         printf("Ошибка: деление на ноль! ");
34         return(HUGE_VAL);
35     }
36     else
37         return(Numeral / SecondNumeral);
38 }
39 else if(strcmp(Operation, "pow", 3) == 0)
40 {
41     printf("Степень: ");
42     scanf("%f",&SecondNumeral);
43     return(pow(Numeral, SecondNumeral));
44 }
45 else if(strcmp(Operation, "sqrt", 4) == 0)
46     return(sqrt(Numeral));
47 else if(strcmp(Operation, "sin", 3) == 0)
48     return(sin(Numeral));
49 else if(strcmp(Operation, "cos", 3) == 0)
50     return(cos(Numeral));
51 else if(strcmp(Operation, "tan", 3) == 0)
52     return(tan(Numeral));
53 else
54 {
55     printf("Неправильно введено действие ");
56     return(HUGE_VAL);
57 }
58 }
59 }
```

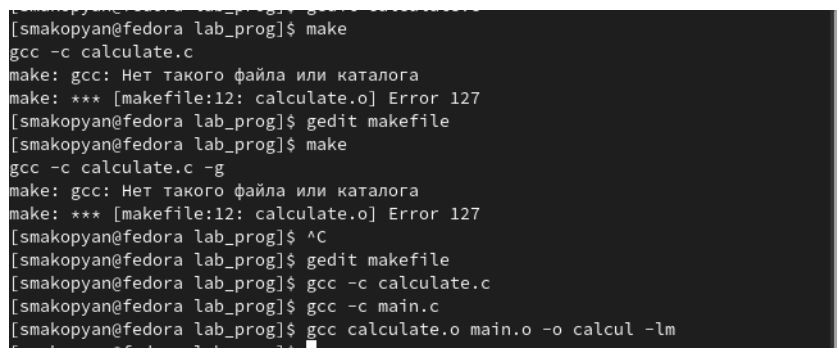
Рис. 2.5: рисунок 5



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

Рис. 2.6: рисунок 6

3. Выполните компиляцию программы посредством gcc



```
[smakopyan@fedora lab_prog]$ make
gcc -c calculate.c
make: gcc: Нет такого файла или каталога
make: *** [makefile:12: calculate.o] Error 127
[smakopyan@fedora lab_prog]$ gedit makefile
[smakopyan@fedora lab_prog]$ make
gcc -c calculate.c -g
make: gcc: Нет такого файла или каталога
make: *** [makefile:12: calculate.o] Error 127
[smakopyan@fedora lab_prog]$ ^C
[smakopyan@fedora lab_prog]$ gedit makefile
[smakopyan@fedora lab_prog]$ gcc -c calculate.c
[smakopyan@fedora lab_prog]$ gcc -c main.c
[smakopyan@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[smakopyan@fedora lab_prog]$
```

Рис. 2.7: рисунок 8

4. Создайте Makefile со следующим содержанием:

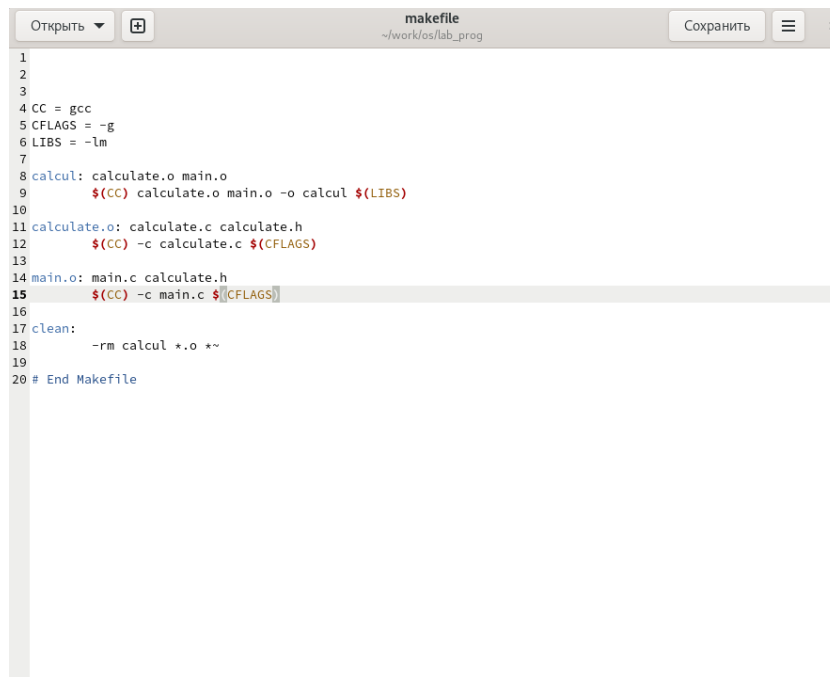


Рис. 2.8: рисунок 7

5. Запустите отладчик GDB, загрузив в него программу для отладки:

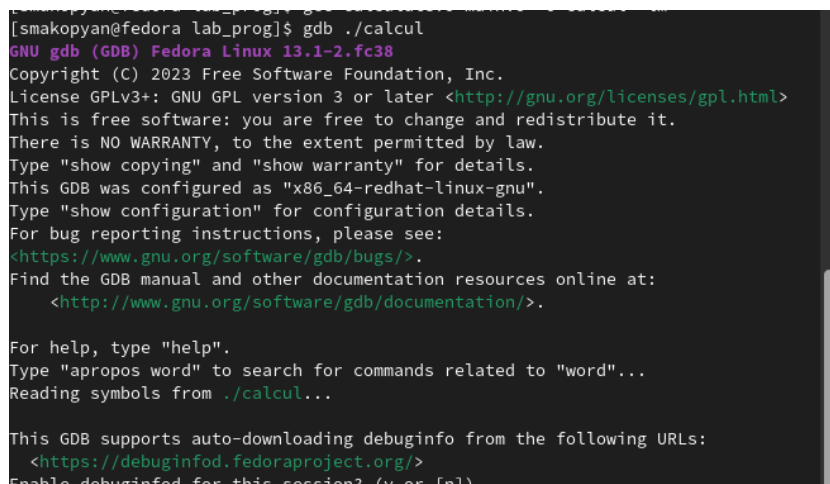


Рис. 2.9: рисунок 9

```

text-user-interface -- TUI is the GDB text based interface.
tracepoints -- Tracing of program execution without stopping the program.
user-defined -- User-defined commands.

Type "help" followed by a class name for a list of commands in that class.
Type "help all" for the list of all commands.
Type "help" followed by command name for full documentation.
Type "apropos word" to search for commands related to "word".
Type "apropos -v word" for full documentation of commands related to "word".
Command name abbreviations are allowed if unambiguous.
(gdb) run
Starting program: /home/smakopyan/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 34
39.00
[Inferior 1 (process 6446) exited normally]
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.37-1.fc38.x86_64
(gdb)

```

Рис. 2.10: рисунок 11

6. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

```

[smakopyan@fedora lab_prog]$ splint calculate.c
Splint 3.1.2 --- 21 Jan 2023

calculate.h:7:38: 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:6:31: Function parameter Operation declared as manifest array (size
constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:12:1: 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:18:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:24:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:30:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:5: 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

```

Рис. 2.11: рисунок 12

```
Finished checking --- 15 code warnings
[smakopyan@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 21 Jan 2023

calculate.h:7:38: 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:13:2: 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:15:13: Format argument 1 to scanf (%s) expects char * gets char [4] *:
        &Operation
    Type of parameter is not consistent with corresponding code in format string.
    (Use -formattype to inhibit warning)
main.c:15:10: Corresponding format code
main.c:15:2: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
[smakopyan@fedora lab_prog]$
```

Рис. 2.12: рисунок 13

3 Выводы

В результате 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.

Список литературы