

# **Отчёт по лабораторной работе №13**

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

Световидова Полина Михайловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Выводы</b>	<b>18</b>

# Список иллюстраций

figno1b_prog . . . . .	8
figno2создание файлов . . . . .	9
figno3calculate.h . . . . .	11
figno4calculate.h . . . . .	12
figno5main.c . . . . .	13
figno6компиляция программы . . . . .	13
figno7. . . . .	14
figno8все файлы . . . . .	14
figno9од makefile . . . . .	15
figno10управление кода . . . . .	16
figno11пуск отладчика . . . . .	17

## **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 Задание

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.
3. Выполните компиляцию программы посредством `gcc`:
4. При необходимости исправьте синтаксические ошибки.
5. Создайте `Makefile` со следующим содержанием:
6. С помощью `gdb` выполните отладку программы `calcul` (перед использованием `gdb` исправьте `Makefile`):
  - Запустите отладчик GDB, загрузив в него программу для отладки
  - Для запуска программы внутри отладчика введите команду `run`:
  - Для постраничного (по 9 строк) просмотра исходного кода используйте команду `list`:
  - Для просмотра строк с 12 по 15 основного файла используйте `list` с параметрами:
  - Для просмотра определённых строк не основного файла используйте `list` с параметрами:
  - Установите точку останова в файле `calculate.c` на строке номер 21:
  - Выведите информацию об имеющихся в проекте точках останова:
  - Запустите программу внутри отладчика и убедитесь, что программа оста-

новится в момент прохождения точки останова:

- Отладчик выдаст следующую информацию:
- Посмотрите, чему равно на этом этапе значение переменной Numeral На экран должно быть выведено число 5.
- Сравните с результатом вывода на экран после использования команды:
- Уберите точки останова:

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

## 3 Выполнение лабораторной работы

В домашнем каталоге создаю подкаталог ~/work/os/lab\_prog.

### 3.1

```
[root@10 ~]# cd work
[root@10 work]# cd os
-bash: cd: os: Нет такого файла или каталога
[root@10 work]#
[root@10 work]# mc

[root@10 work]# mkdir os
[root@10 work]# cd os
[root@10 os]# mkdir lab_prog
[root@10 os]# ls
lab_prog
[root@10 os]#
```

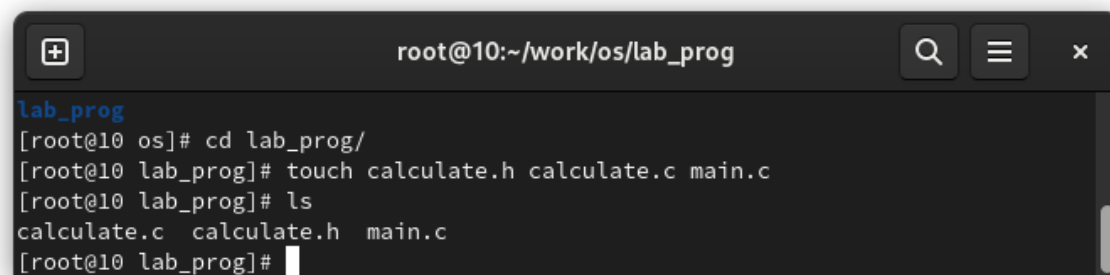
lab\_prog

### 3.2

Создаю в нём файлы calculate.h, calculate.c и main.c



### 3.3

A terminal window with a dark background. The title bar shows 'root@10:~/work/os/lab\_prog' and standard window controls. The terminal content shows a sequence of commands: 'cd lab\_prog/' to change the directory, 'touch calculate.h calculate.c main.c' to create three files, and 'ls' to list them. The output of 'ls' shows 'calculate.c calculate.h main.c'. The prompt is currently at '[root@10 lab\_prog]#'.

```
lab_prog
[root@10 os]# cd lab_prog/
[root@10 lab_prog]# touch calculate.h calculate.c main.c
[root@10 lab_prog]# ls
calculate.c calculate.h main.c
[root@10 lab_prog]#
```

создание файлов

### 3.4

Реализация функций калькулятора в файле calculate.h



## 3.5



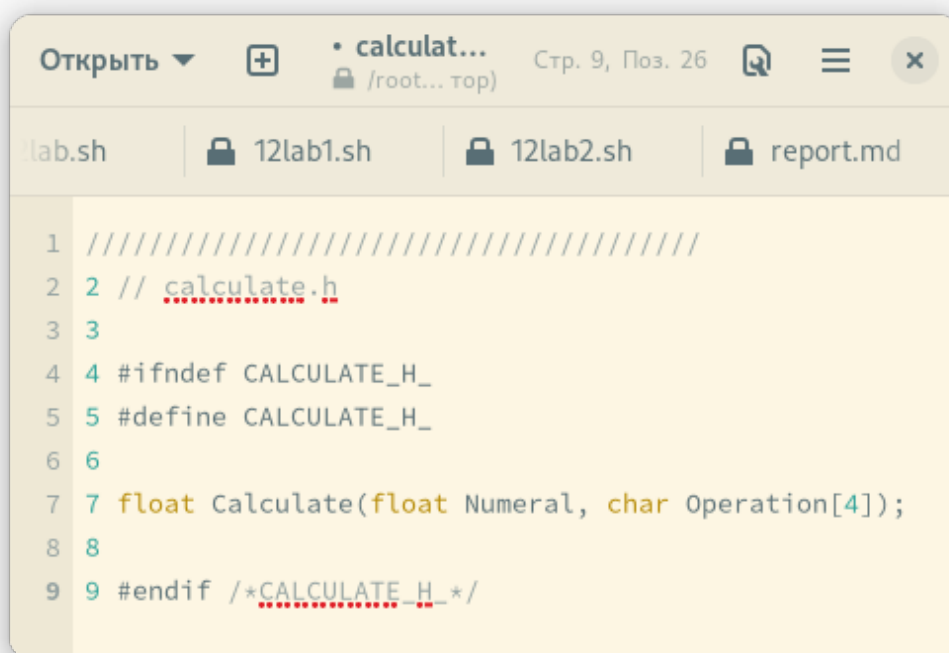
The image shows a code editor window with the title bar "calculate.c" and "Стр. 47, Поз. 37". The editor contains the following C code:

```
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: ");
46         scanf("%f", &SecondNumeral);
47         return(pow(Numeral, SecondNumeral));
```

## 3.6

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

## 3.7

A screenshot of a code editor window. The title bar shows 'Открыть' (Open) with a dropdown arrow, a plus icon, and the file name 'calculat...' with a lock icon and path '/root... top'. It also shows 'Стр. 9, Поз. 26' (Page 9, Position 26) and icons for search, menu, and close. The editor has four tabs: 'lab.sh', '12lab1.sh', '12lab2.sh', and 'report.md'. The active tab is '12lab2.sh'. The code is as follows:

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

`calculate.h`

## 3.8

Основной файл `main.c`, реализующий интерфейс пользователя калькулятору

## 3.9



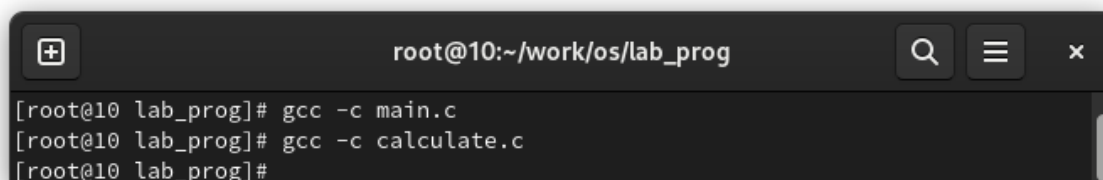
```
1 //////////////////////////////////////////////////
2 // main.c
3
4 #include <stdio.h>
5 #include "calculate.h"
6
7 int
8 main (void)
9 {
10 float Numeral;
11 char Operation[4];
12 float Result;
13 printf("Число: ");
14 scanf("%f",&Numeral);
15 printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");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

main.c

## 3.10

Далее выполняю компиляцию программы посредством gcc

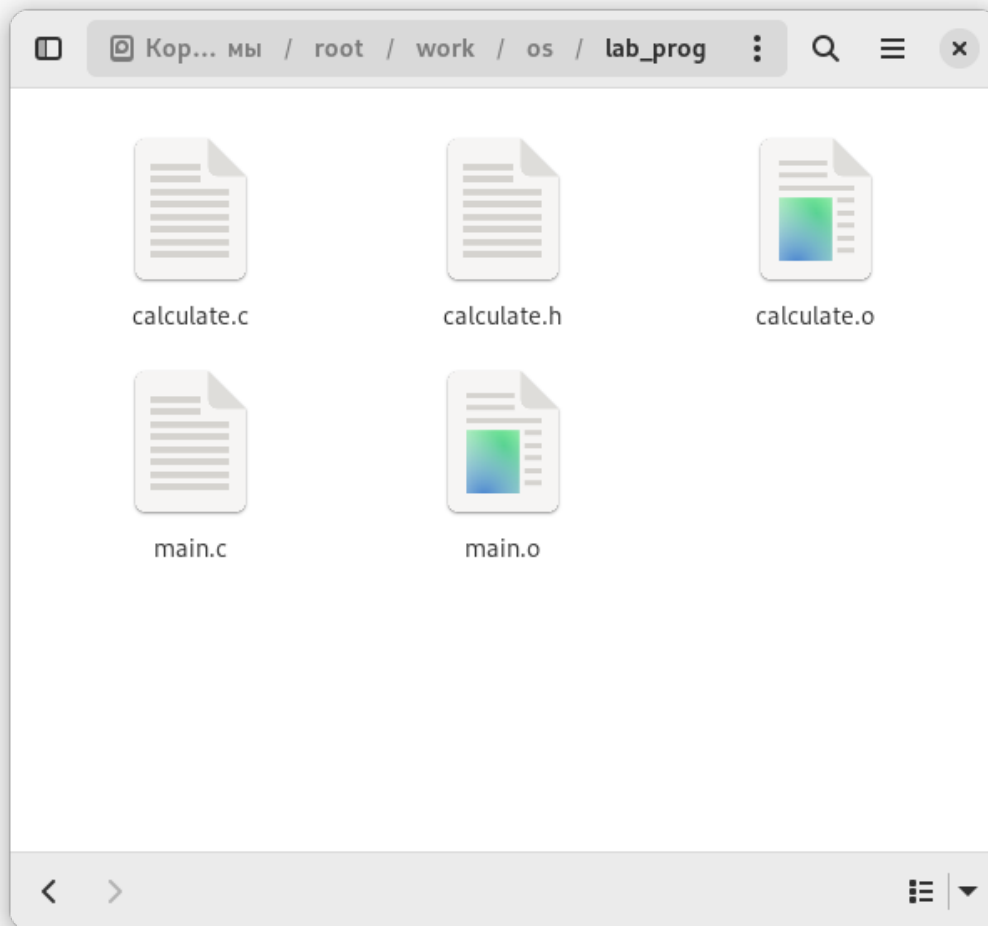
## 3.11



```
root@10:~/work/os/lab_prog
[root@10 lab_prog]# gcc -c main.c
[root@10 lab_prog]# gcc -c calculate.c
[root@10 lab_prog]#
```

компиляция программы

```
[root@10 lab_prog]# gcc calculate.o main.o -o calcul -lm
```



сами файлы

## 3.12

Создаю Makefile

## 3.13

A screenshot of a code editor window titled "makefile" with a path of "/root/work/os/lab\_prog (Администратор)". The editor shows a Makefile with the following content:

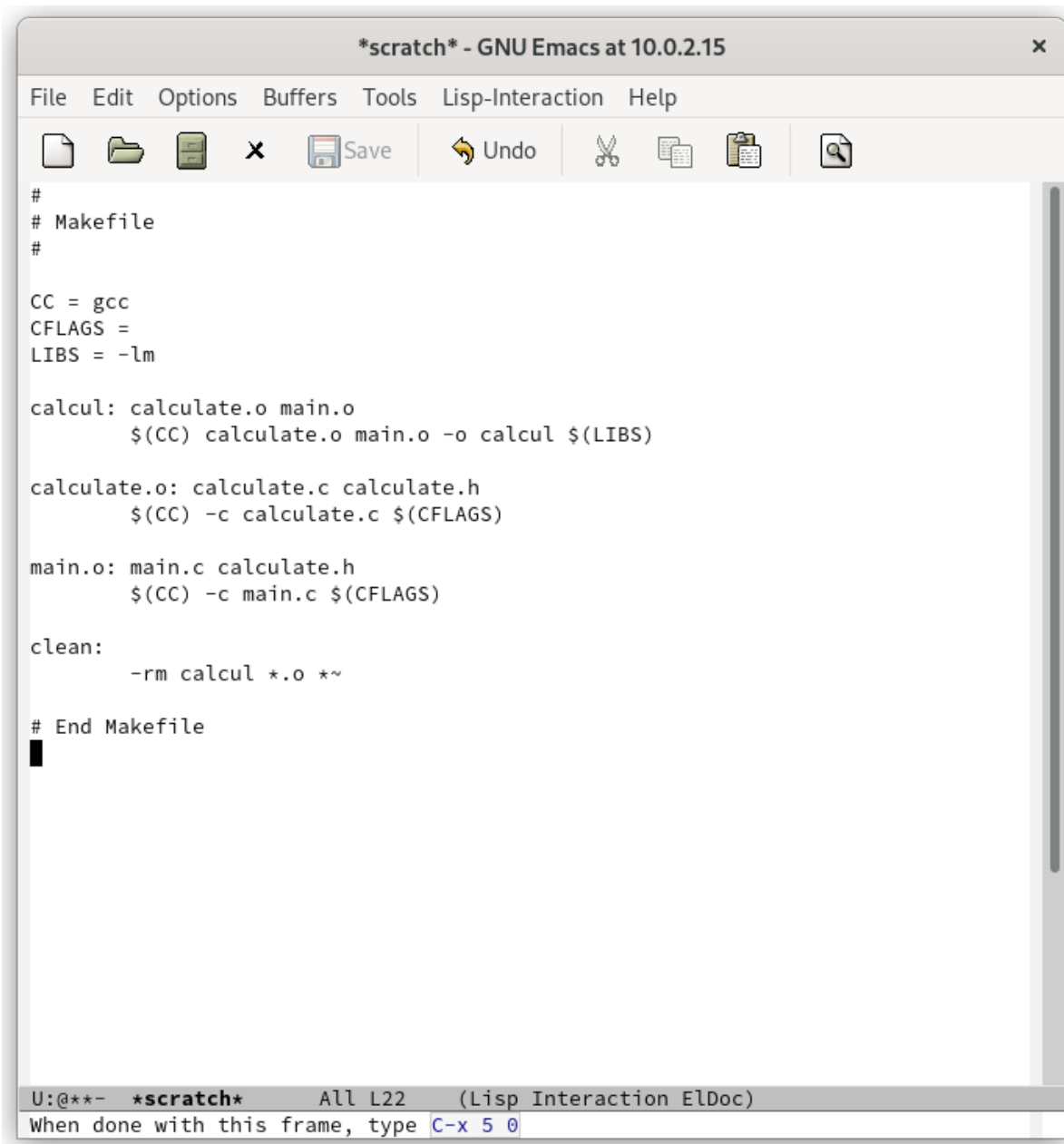
```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10 gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13 gcc -c calculate.c $(CFLAGS)
14
15 main.o: main.c calculate.h
16 gcc -c main.c $(CFLAGS)
17
18 clean:
19 -rm calcul *.o *~
20
```

код makefile

## 3.14

Далее исправляю код для Makefile

## 3.15



The screenshot shows the GNU Emacs editor window titled "\*scratch\* - GNU Emacs at 10.0.2.15". The menu bar includes File, Edit, Options, Buffers, Tools, Lisp-Interaction, and Help. The toolbar contains icons for file operations (new, open, save, close), editing (undo, redo, cut, copy, paste), and search. The main text area contains a Makefile with the following content:

```
#
# Makefile
#

CC = gcc
CFLAGS =
LIBS = -lm

calcul: calculate.o main.o
    $(CC) calculate.o main.o -o calcul $(LIBS)

calculate.o: calculate.c calculate.h
    $(CC) -c calculate.c $(CFLAGS)

main.o: main.c calculate.h
    $(CC) -c main.c $(CFLAGS)

clean:
    -rm calcul *.o *~

# End Makefile
```

The status bar at the bottom shows "U:@\*- \*scratch\* All L22 (Lisp Interaction ElDoc)" and a prompt "When done with this frame, type C-x 5 0".

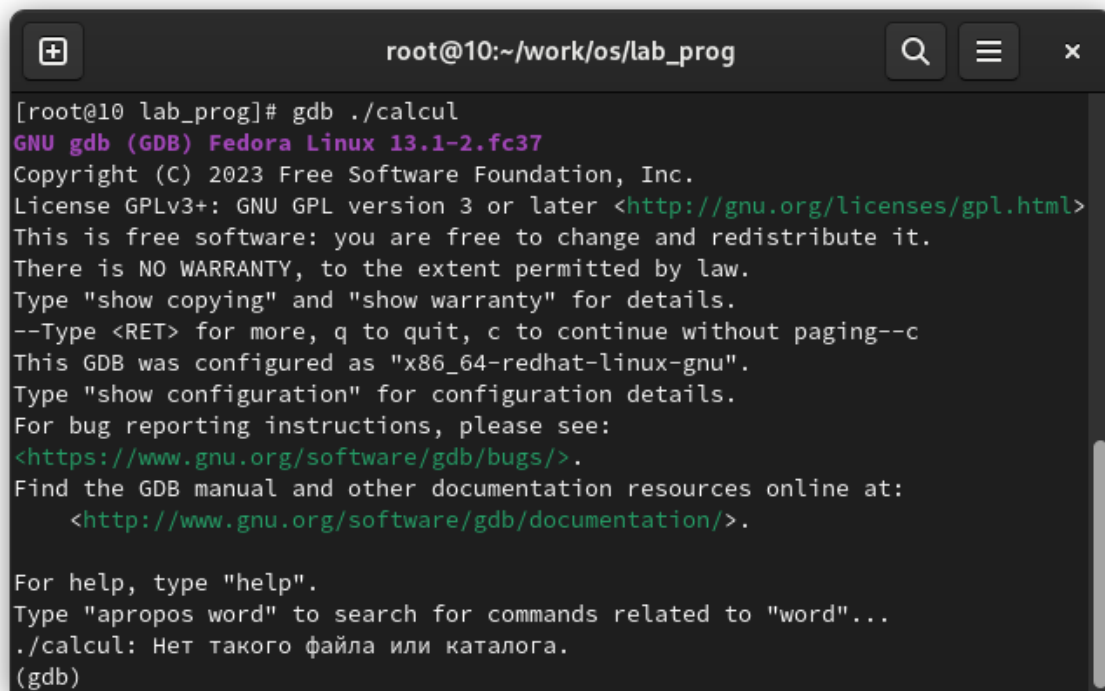
исправление кода



## 3.16

С помощью gdb выполняю отладку программы calcul и запускаю отладчик GDB, загрузив в него программу для отладки. Для запуска ввожу команду run

## 3.17

A screenshot of a terminal window with a dark background. The title bar shows 'root@10:~/work/os/lab\_prog'. The terminal content shows the command 'gdb ./calcul' being executed. The GDB startup banner is displayed, including copyright information for 2023, the GNU GPL license, and various help prompts. The prompt '(gdb)' is visible at the bottom.

```
[root@10 lab_prog]# gdb ./calcul
GNU gdb (GDB) Fedora Linux 13.1-2.fc37
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
--Type <RET> for more, q to quit, c to continue without paging--c
This GDB was configured as "x86_64-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
./calcul: Нет такого файла или каталога.
(gdb)
```

запуск отладчика

## 4 Выводы

В ходе выполнения лабораторной работы №13 я приобрела простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями