

Отчёт по лабораторной работе №13

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

Федюшина Ярослава Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	18

Список иллюстраций

3.1	lab_prog	8
3.2	создание файлов	9
3.3	calculate.h	10
3.4	calculate.h	11
3.5	main.c	12
3.6	компиляция программы	13
3.7	13
3.8	сами файлы	14
3.9	код makefile	15
3.10	исправление кода	16
3.11	запуск 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 код 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 точка останова:
 - Запустите программу внутри отладчика и убедитесь, что программа оста-

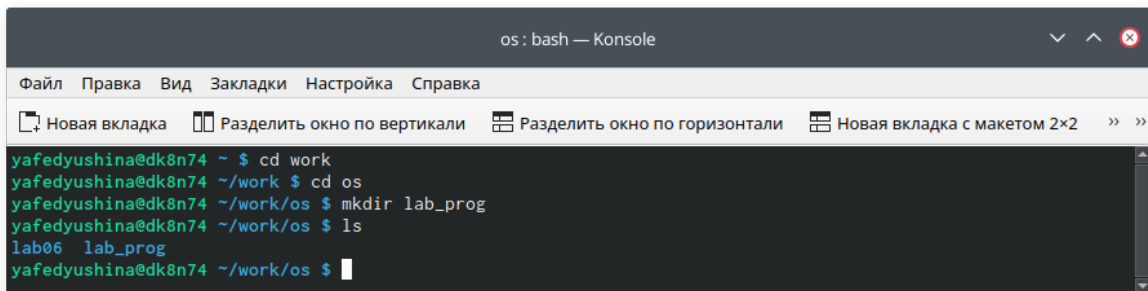
новится в момент прохождения точки останова:

- 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



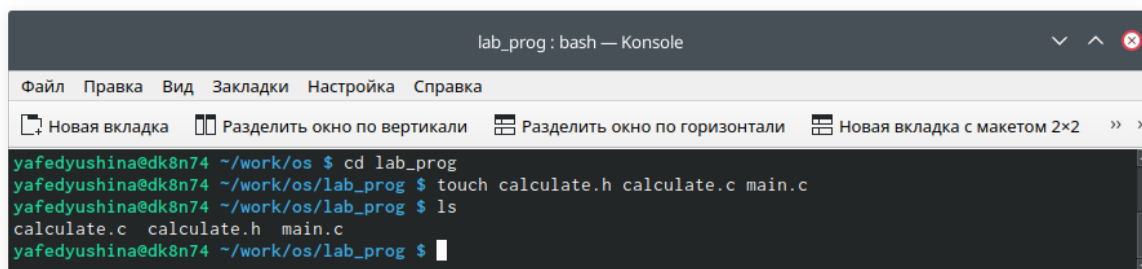
```
os : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[icon] Новая вкладка  [icon] Разделить окно по вертикали  [icon] Разделить окно по горизонтали  [icon] Новая вкладка с макетом 2x2  >> >>
yafedyushina@dk8n74 ~ $ cd work
yafedyushina@dk8n74 ~/work $ cd os
yafedyushina@dk8n74 ~/work/os $ mkdir lab_prog
yafedyushina@dk8n74 ~/work/os $ ls
lab06  lab_prog
yafedyushina@dk8n74 ~/work/os $
```

Рис. 3.1: lab_prog

3.2

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

3.3



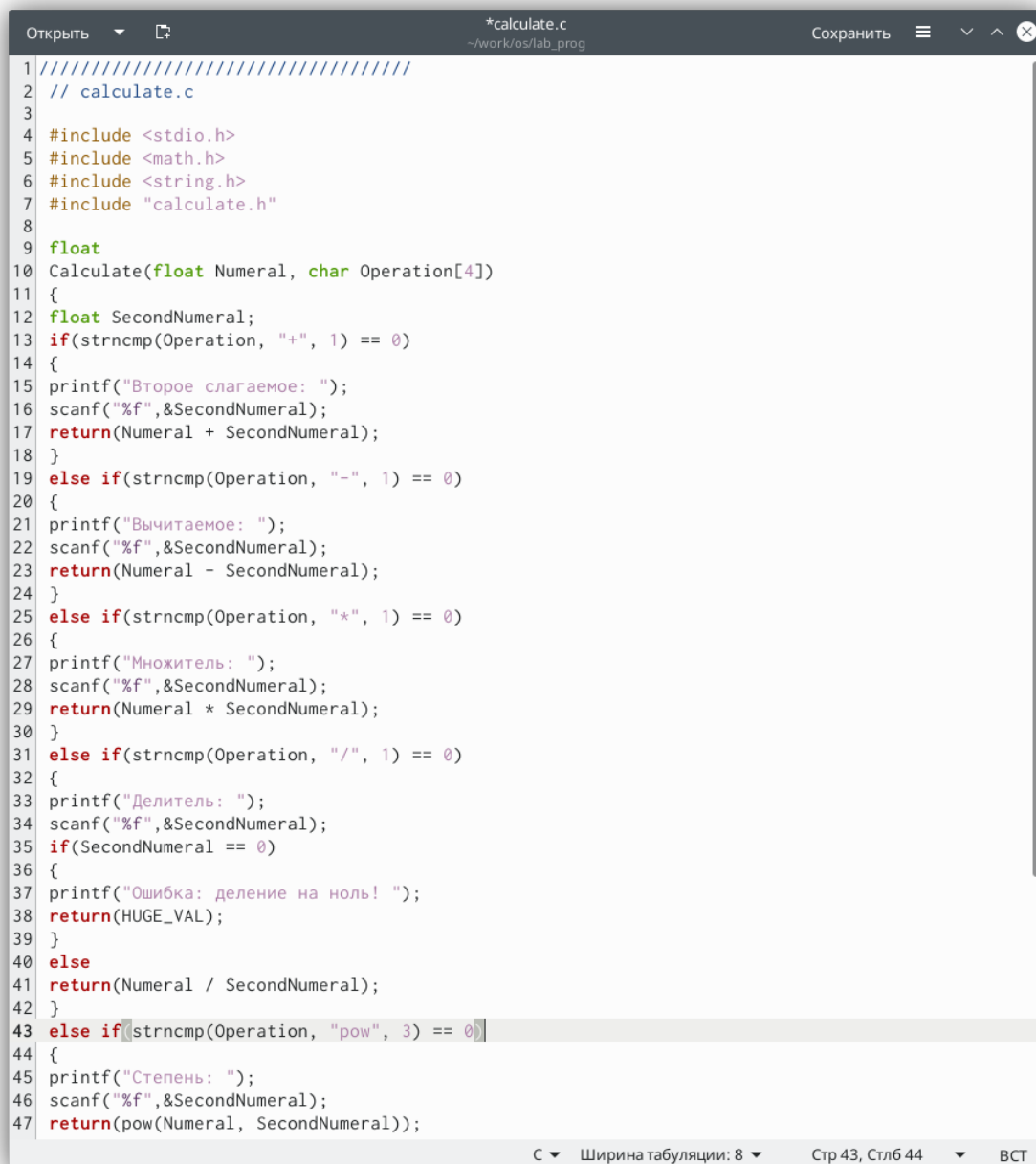
```
lab_prog : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[+] Новая вкладка  [||] Разделить окно по вертикали  [|||] Разделить окно по горизонтали  [|||] Новая вкладка с макетом 2x2  >> >>
yafedyushina@dk8n74 ~/work/os $ cd lab_prog
yafedyushina@dk8n74 ~/work/os/lab_prog $ touch calculate.h calculate.c main.c
yafedyushina@dk8n74 ~/work/os/lab_prog $ ls
calculate.c  calculate.h  main.c
yafedyushina@dk8n74 ~/work/os/lab_prog $
```

Рис. 3.2: создание файлов

3.4

Реализация функций калькулятора в файле calculate.h

3.5



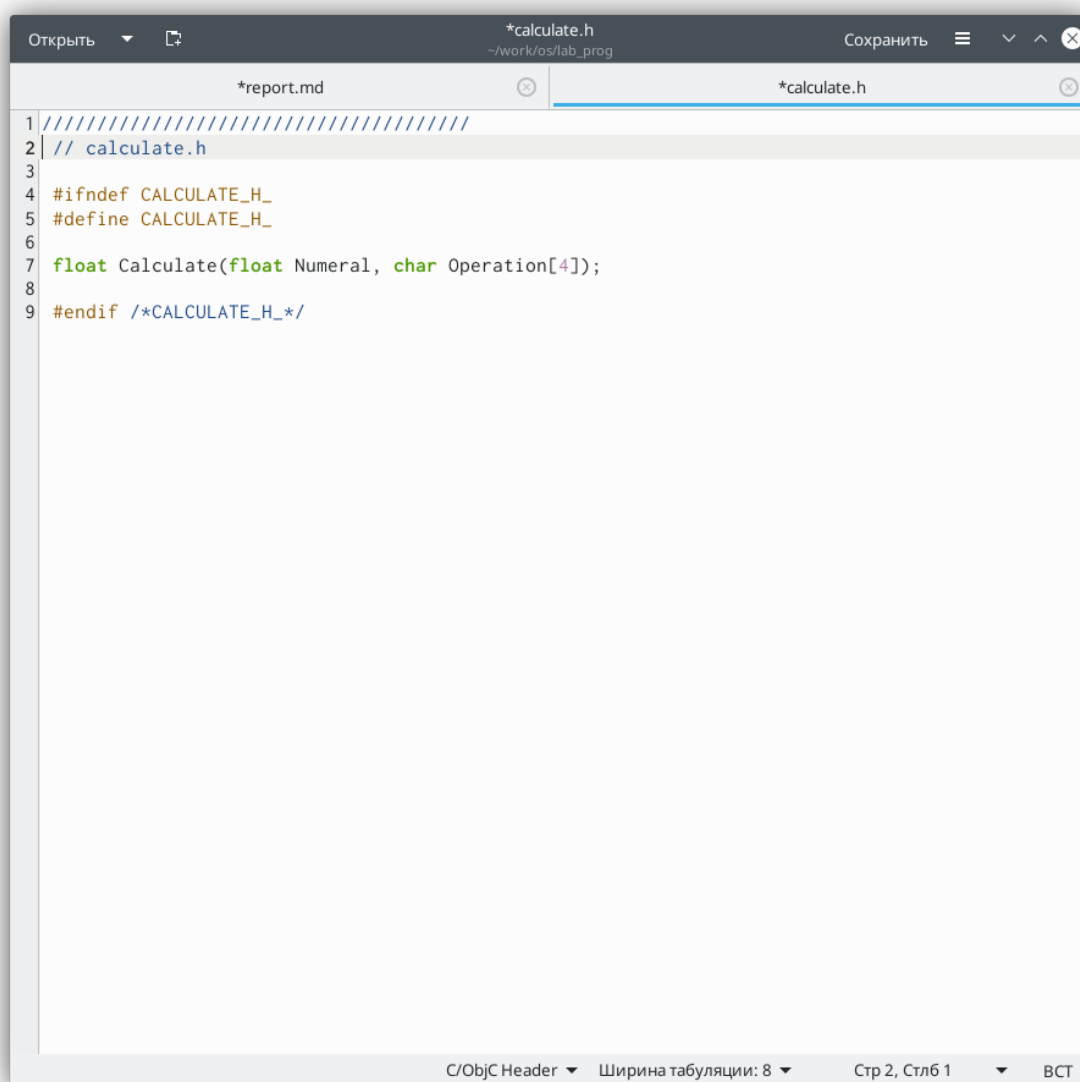
```
1 //////////////////////////////////////////////////
2 // calculate.c
3
4 #include <stdio.h>
5 #include <math.h>
6 #include <string.h>
7 #include "calculate.h"
8
9 float
10 Calculate(float Numeral, char Operation[4])
11 {
12     float SecondNumeral;
13     if(strncmp(Operation, "+", 1) == 0)
14     {
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));
```

Рис. 3.3: calculate.h

3.6

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

3.7



The image shows a code editor window with two tabs: `*report.md` and `*calculate.h`. The `*calculate.h` tab is active, displaying the following C header file content:

```
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_*/
```

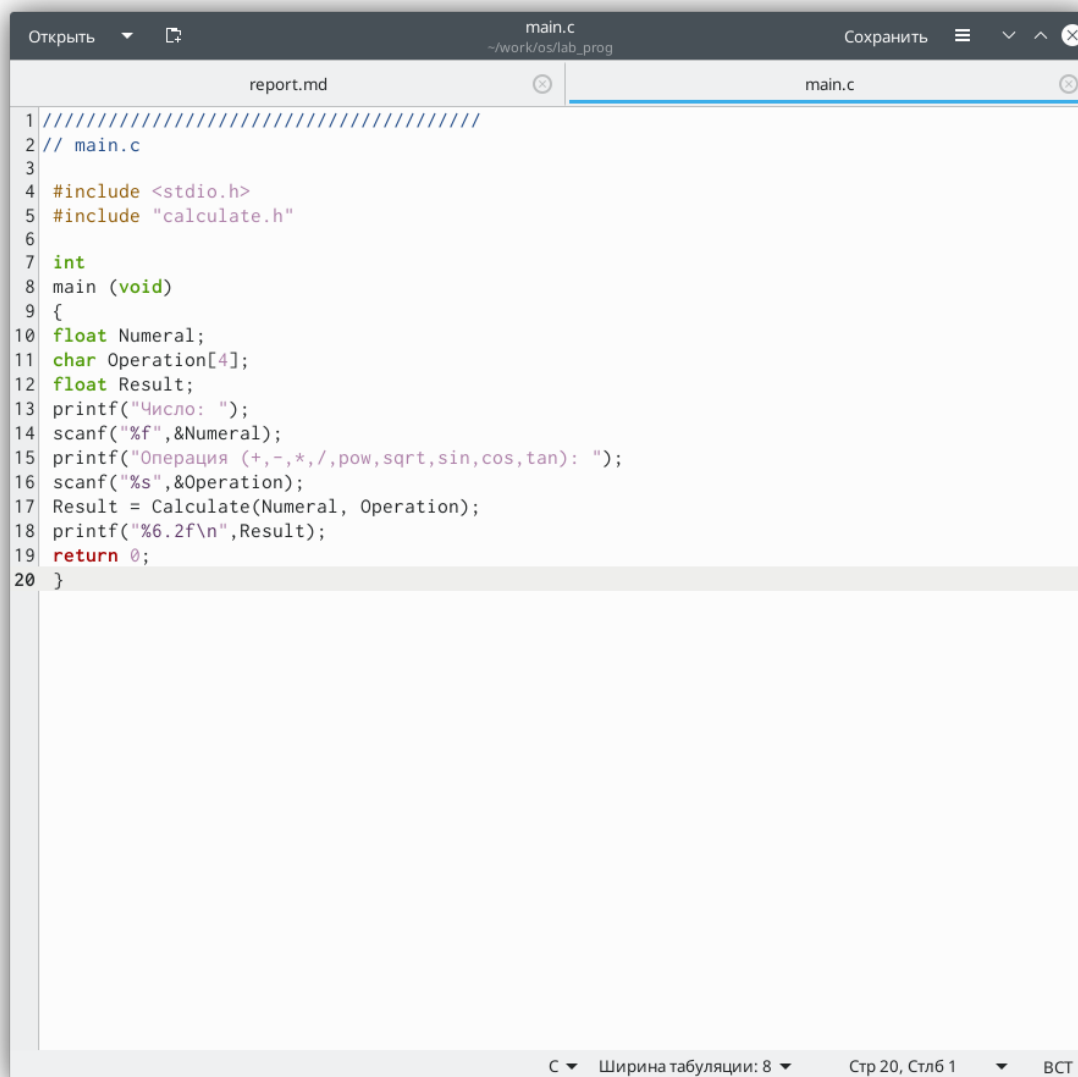
The editor's status bar at the bottom indicates: C/ObjC Header, Ширина табуляции: 8, Стр 2, Стлб 1, ВСТ.

Рис. 3.4: `calculate.h`

3.8

Основной файл main.c, реализующий интерфейс пользователя калькулятору

3.9



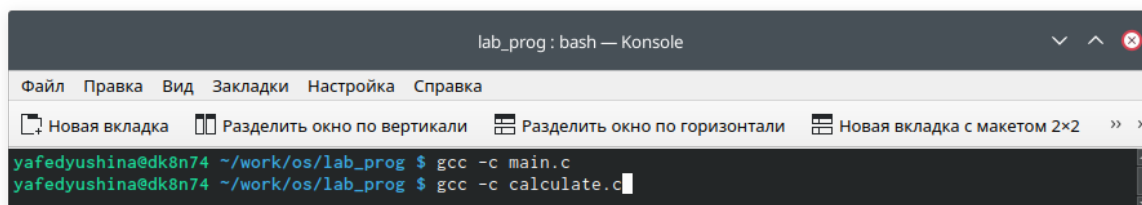
```
1 //////////////////////////////////////////////////
2 // main.c
3
4 #include <stdio.h>
5 #include "calculate.h"
6
7 int
8 main (void)
9 {
10 float Numeral;
11 char Operation[4];
12 float Result;
13 printf("Число: ");
14 scanf("%f",&Numeral);
15 printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");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

Рис. 3.5: main.c

3.10

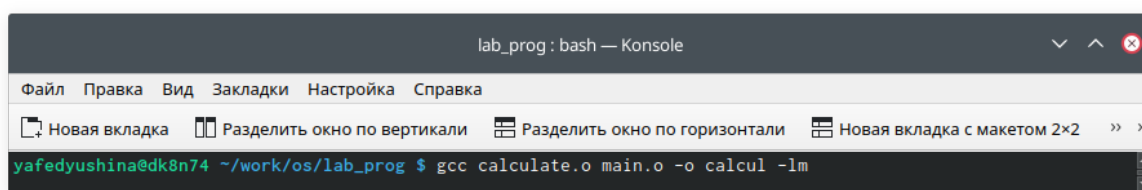
Далее выполняю компиляцию программы посредством gcc

3.11



```
lab_prog : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[+] Новая вкладка  [||] Разделить окно по вертикали  [≡] Разделить окно по горизонтали  [≡] Новая вкладка с макетом 2×2  >> >>
yafedyushina@dk8n74 ~/work/os/lab_prog $ gcc -c main.c
yafedyushina@dk8n74 ~/work/os/lab_prog $ gcc -c calculate.c
```

Рис. 3.6: компиляция программы



```
lab_prog : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[+] Новая вкладка  [||] Разделить окно по вертикали  [≡] Разделить окно по горизонтали  [≡] Новая вкладка с макетом 2×2  >> >>
yafedyushina@dk8n74 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
```

Рис. 3.7: .

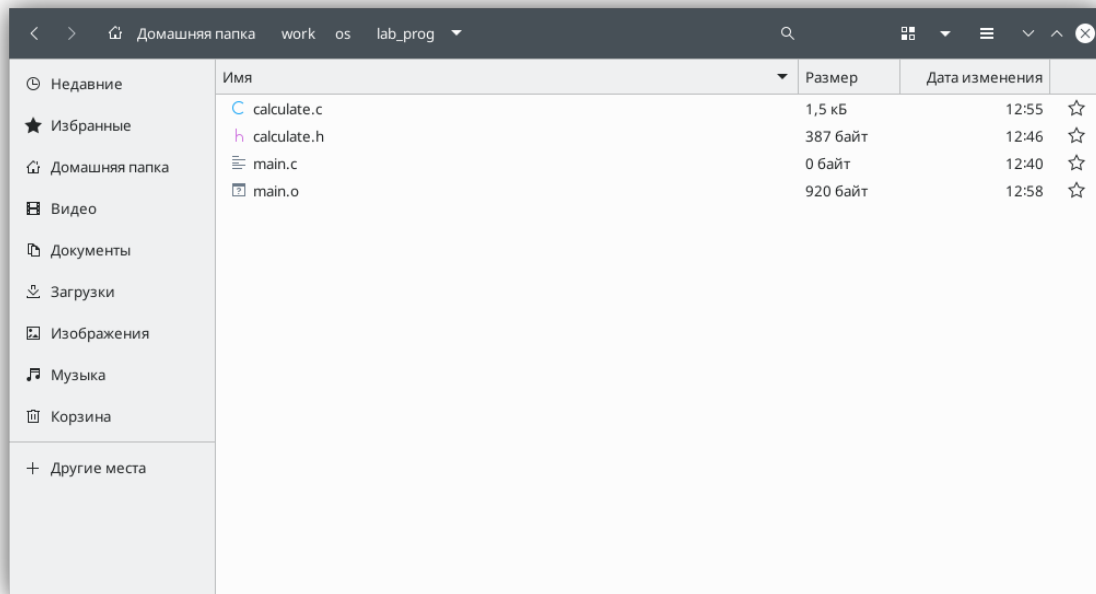
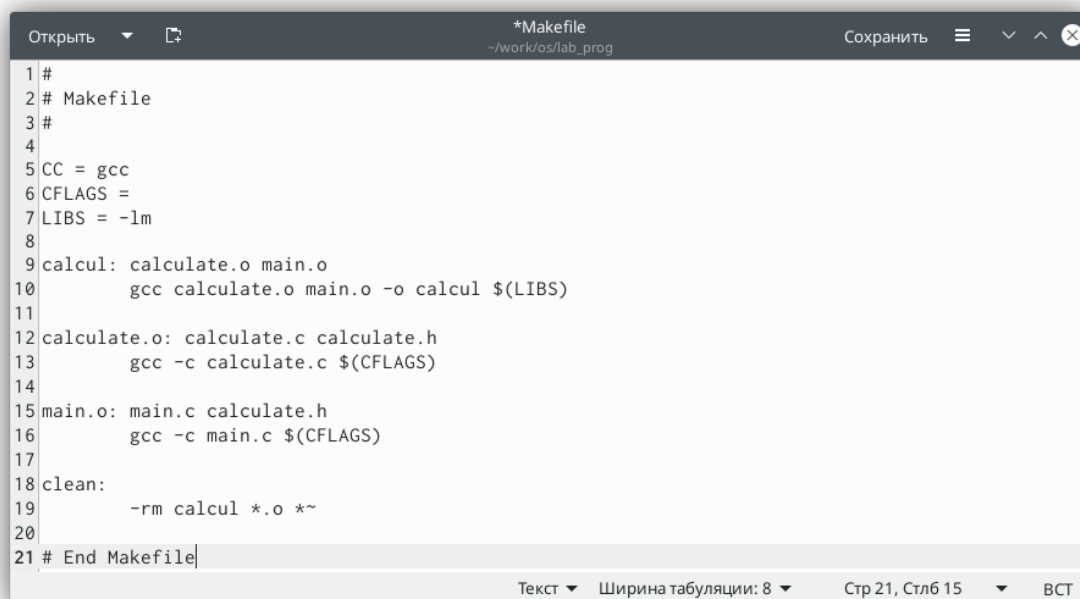


Рис. 3.8: сами файлы

3.12

Создаю Makefile

3.13



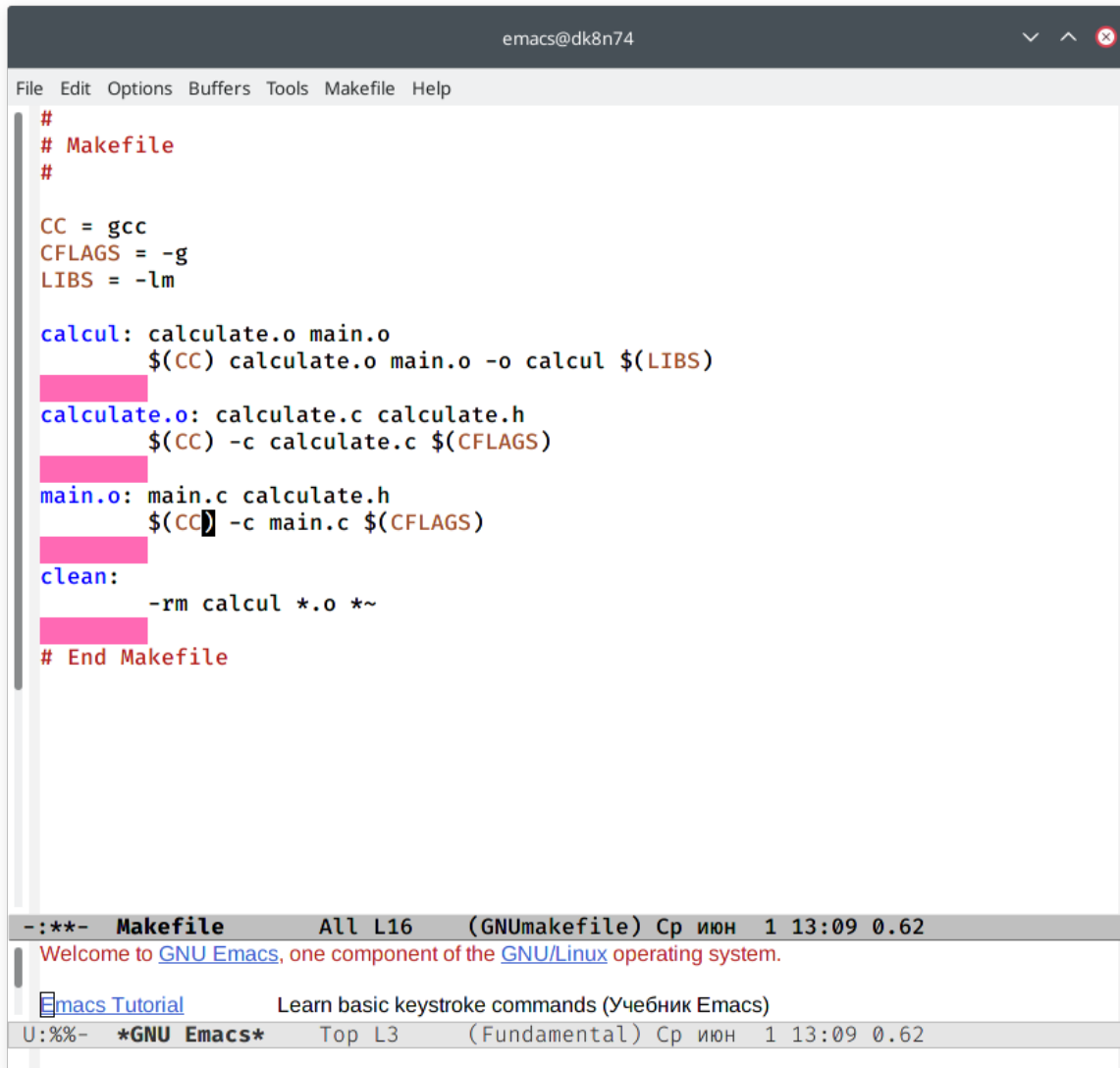
```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10     gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13     gcc -c calculate.c $(CFLAGS)
14
15 main.o: main.c calculate.h
16     gcc -c main.c $(CFLAGS)
17
18 clean:
19 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

Рис. 3.9: код makefile

3.14

Далее исправляю код для Makefile

3.15



The screenshot shows an Emacs window titled 'emacs@dk8n74'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Makefile', and 'Help'. The main text area contains a Makefile with the following content:

```
#
# Makefile
#

CC = gcc
CFLAGS = -g
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
```

At the bottom of the window, there is a status bar with the following text:

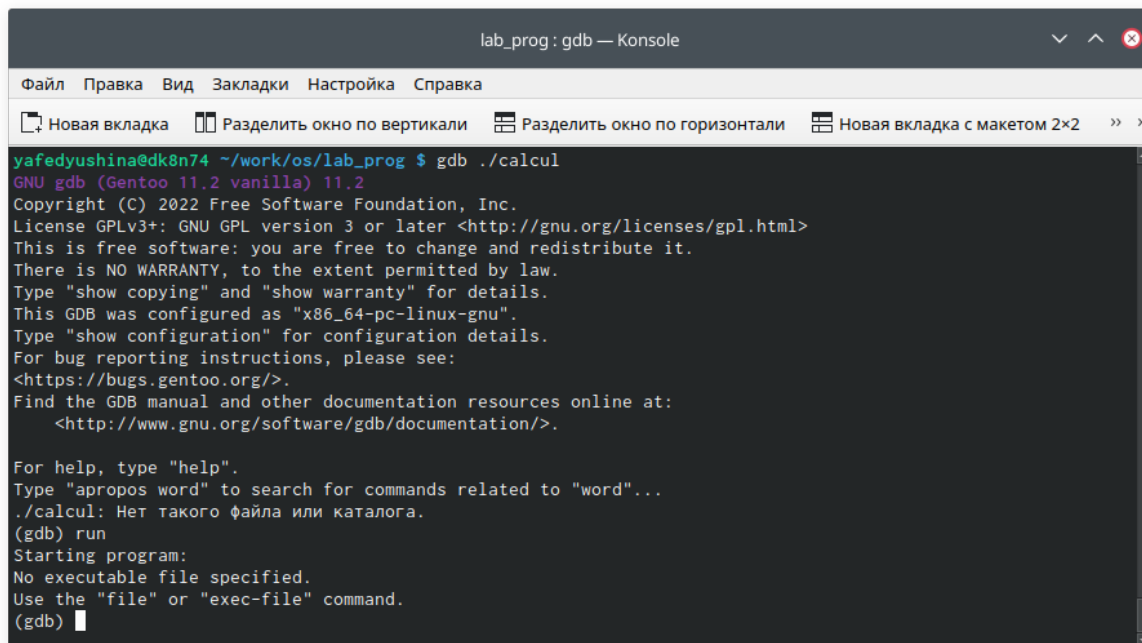
```
-:***- Makefile All L16 (GNUmakefile) Ср июн 1 13:09 0.62
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
Emacs Tutorial Learn basic keystroke commands (Учебник Emacs)
U:%%- *GNU Emacs* Top L3 (Fundamental) Ср июн 1 13:09 0.62
```

Рис. 3.10: 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



The screenshot shows a terminal window titled "lab_prog : gdb — Konsole". The window has a menu bar with "Файл", "Правка", "Вид", "Закладки", "Настройка", and "Справка". Below the menu bar is a toolbar with icons for "Новая вкладка", "Разделить окно по вертикали", "Разделить окно по горизонтали", and "Новая вкладка с макетом 2x2", followed by right arrow icons. The terminal content shows a user prompt "yafedyushina@dk8n74 ~/work/os/lab_prog \$" followed by the command "gdb ./calcul". The GDB startup sequence follows, displaying version information (GNU gdb (Gentoo 11.2 vanilla) 11.2), copyright (C) 2022 Free Software Foundation, Inc., and the GPL license. It then shows the configuration "x86_64-pc-linux-gnu" and provides links for bug reporting and documentation. After the introductory text, the user enters "(gdb) run". The response is "Starting program: No executable file specified. Use the 'file' or 'exec-file' command." The prompt "(gdb) " is shown at the end of the line.

```
lab_prog : gdb — Konsole
Файл Правка Вид Закладки Настройка Справка
[+] Новая вкладка [||] Разделить окно по вертикали [|||] Разделить окно по горизонтали [|||] Новая вкладка с макетом 2x2 » »
yafedyushina@dk8n74 ~/work/os/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 11.2 vanilla) 11.2
Copyright (C) 2022 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.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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) run
Starting program:
No executable file specified.
Use the "file" or "exec-file" command.
(gdb) 
```

Рис. 3.11: 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