

Лабораторная работа No 13. Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

Leysan R. Abdullina

NEC-2022, 31 May

RUDN University, Moscow, Russian Federation

Лабораторная работа No 13.
Средства, применяемые при
разработке программного
обеспечения в ОС типа UNIX/Linux

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

Процесс разработки программного обеспечения обычно разделяется на следующие этапы: - планирование, включающее сбор и анализ требований к функционалу и другим характеристикам разрабатываемого приложения;

- проектирование, включающее в себя разработку базовых алгоритмов и спецификаций, определение языка программирования;
- непосредственная разработка приложения:
- кодирование — по сути создание исходного текста программы (возможно в нескольких вариантах);
- анализ разработанного кода;
- сборка, компиляция и разработка исполняемого модуля;
- тестирование и отладка, сохранение произведённых изменений;
- документирование.

1. В домашнем каталоге создадим подкаталог ~/work/os/lab_prog - через команду mkdir (скриншот 1)

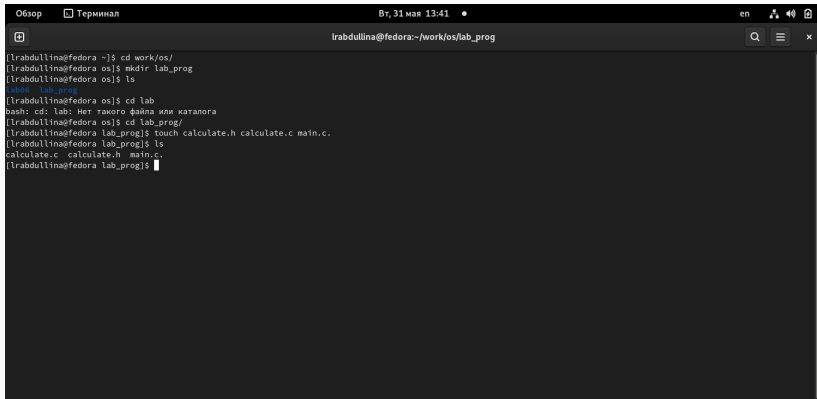


The screenshot shows a terminal window titled "Терминал" (Terminal) with the user "lrabdullina@fedora" and the current directory "~/work/os". The terminal output shows the following commands and results:

```
[lrabdullina@fedora ~]$ cd work/os/  
[lrabdullina@fedora os]$ mkdir lab_prog  
[lrabdullina@fedora os]$ ls  
labos  lab_prog  
[lrabdullina@fedora os]$
```

Figure 1: Созданный каталог

2. Создадим в нём файлы: calculate.h,calculate.c,main.c - через команду touch (скриншот 2)

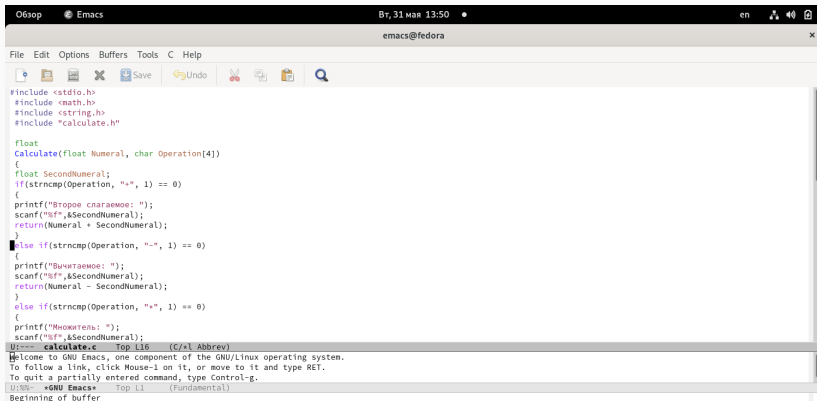
A screenshot of a terminal window titled "Терминал" (Terminal) with a dark background. The window shows a series of commands and their outputs. The user starts by navigating to the directory ~/work/os/ and creating a subdirectory named lab_prog. Then, they navigate into lab_prog and use the touch command to create three files: calculate.h, calculate.c, and main.c. The terminal output shows the successful creation of these files.

```
lrabdullina@fedora: ~/work/os/lab_prog
[~]$ cd work/os/
[~]$ mkdir lab_prog
[~]$ ls
lab_prog
[~]$ cd lab_prog
bash: cd: lab: Нет такого файла или каталога
[~]$ cd lab_prog/
[~]$ touch calculate.h calculate.c main.c
[~]$ ls
calculate.c calculate.h main.c
[~]$
```

Figure 2: Созданные файлы

Выполнение работы

Это примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять \sin , \cos , \tan . При запуске он запрашивает первое число, операцию, второе число. После этого программа выведет результат и останавливается. (скриншоты 3,4,5)



The screenshot shows the Emacs editor interface. The title bar at the top indicates the window is titled "Обзор" (Overview) and "Emacs", with the system clock showing "Вт, 31 мая 13:50" (Tue, 31 May 13:50) and the user "en". The Emacs status bar at the top right shows "emacs@fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The toolbar contains icons for file operations and search. The main text area displays a C program named "calculate.c". The code includes standard headers and defines a "Calculate" function that takes a number and an operation string as input. It uses a switch statement to perform addition, subtraction, multiplication, and division. The program also includes a main function that prompts the user for input and calls the "Calculate" function. The bottom status bar shows the current buffer is "calculate.c" and the line number is 116.

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include "calculate.h"

float
Calculate(float Numeral, char Operation[4])
{
    float SecondNumeral;
    if(strncmp(Operation, "+", 1) == 0)
    {
        printf("Второе число: ");
        scanf("%f", &SecondNumeral);
        return(Numeral + SecondNumeral);
    }
    else if(strncmp(Operation, "-", 1) == 0)
    {
        printf("Вывчитаемое: ");
        scanf("%f", &SecondNumeral);
        return(Numeral - SecondNumeral);
    }
    else if(strncmp(Operation, "*", 1) == 0)
    {
        printf("Умножитель: ");
        scanf("%f", &SecondNumeral);
    }
}
```

U:~ calculate.c Top 116 (C/*L Abbrev)

Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

U:~ GNU Emacs Top 1 (Fundamental)

Beginning of buffer

Выполнение работы

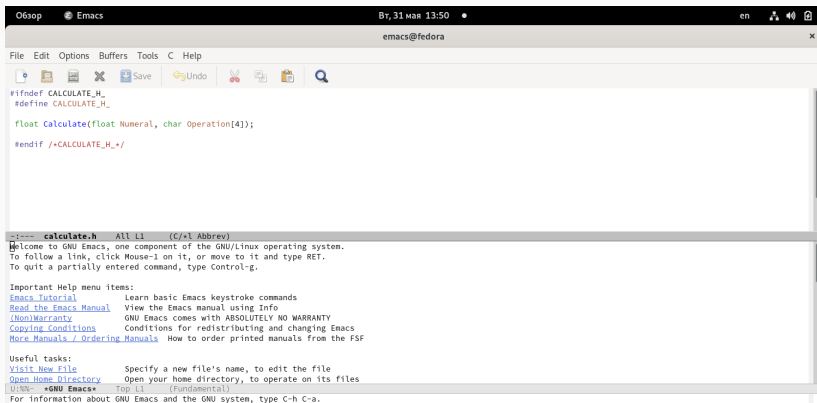
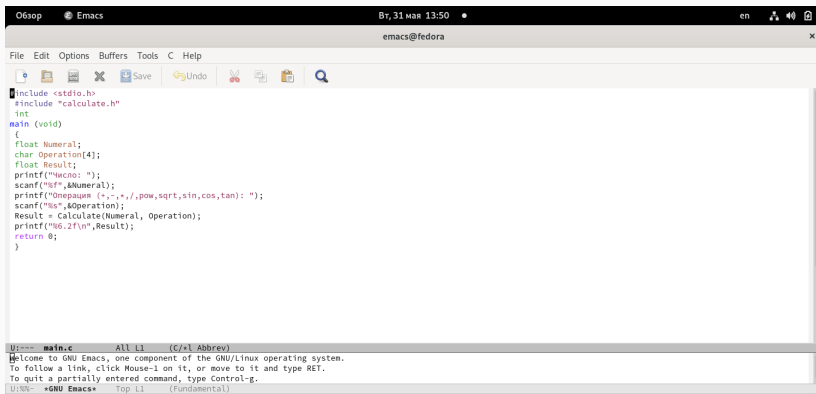


Figure 4: Интерфейсный файл calculate.h, описывающий формат вызова функции калькулятора

Выполнение работы



```
#include <stdio.h>
#include "calculate.h"
int
main (void)
{
  float Numeral;
  char Operation[4];
  float Result;
  printf("Число: ");
  scanf("%f",&Numeral);
  printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
  scanf("%s",&Operation);
  Result = Calculate(Numeral, Operation);
  printf("%6.2f\n",Result);
  return 0;
}
```

U:~ - main.c All L1 (C/++ Abbrev)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.
U:~% - *GNU Emacs* Top L1 (Fundamental)

Figure 5: Основной файл main.c, реализующий интерфейс пользователя к калькулятору

3. Выполним компиляцию программы посредством gcc, используя команды (скриншот 6):
 - `gcc -c calculate.c`
 - `gcc -c main.c`
 - `gcc calculate.o main.o -o calcul -lm`

Выполнение работы

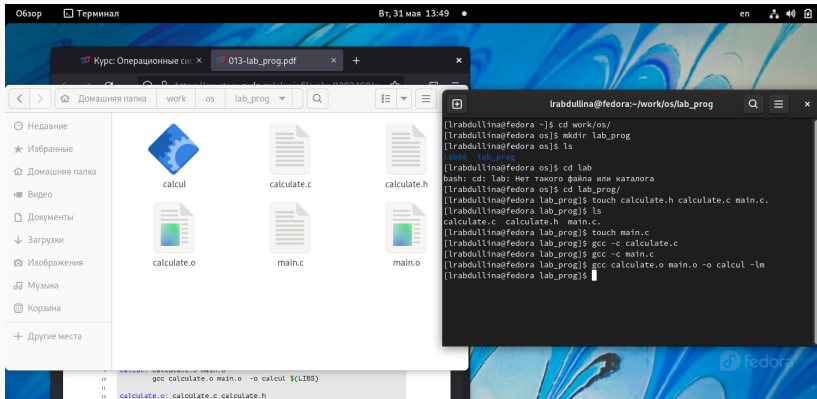
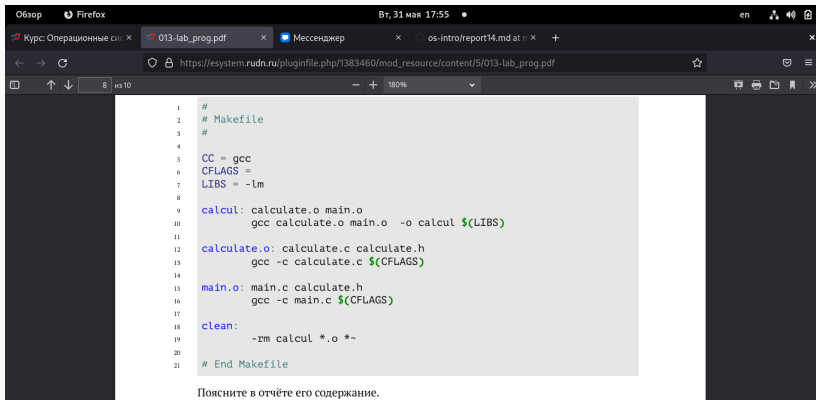


Figure 6: Созданные файлы

4. Необходимости исправлять синтаксические ошибки не было.

5. Создадим Makefile со следующим содержанием (скриншот 7):

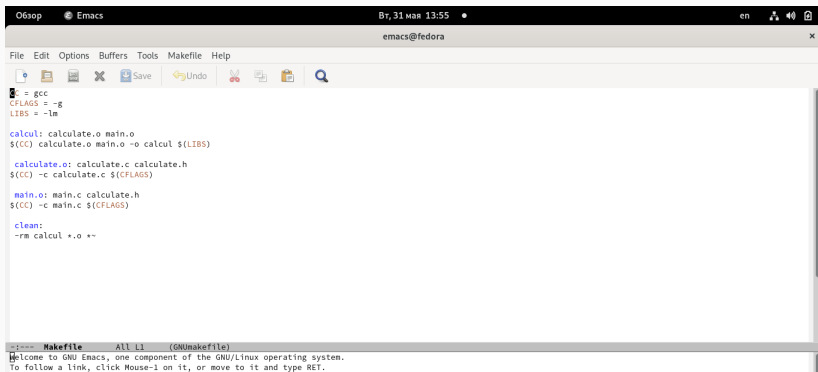
A screenshot of a Firefox browser window. The address bar shows the URL: https://esystem.rudn.ru/pluginfile.php/1383460/mod_resource/content/5/013-lab_prog.pdf. The page content is a code editor displaying a Makefile. The code is as follows:

```
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
```

Figure 7: Созданный файл

Данный файл необходим для автоматической компиляции файлов `calculate.c` (цель `calculate.o`), `main.c` (цель `main.o`), а также их объединения в один исполняемый файл `calcul` (цель `calcul`). Цель `clean` нужна для автоматического удаления файлов. Переменная `CC` отвечает за утилиту для компиляции. Переменная `CFLAGS` отвечает за опции в данной утилите. Переменная `LBS` отвечает за опции для объединения объектных файлов в один исполняемый файл.

6. С помощью gdb выполним отладку программы calcul, но перед использованием gdb исправим Makefile. В переменную CFLAGS добавим опцию -g, необходимую для компиляции объектных файлов и их использования в программе отладчика GDB. Сделаем так, что утилита компиляции выбирается с помощью переменной CC. (скриншот 8)



```
Обзор Emacs Бт, 31 мая 13:55 en emacs@fedora x
File Edit Options Buffers Tools Makefile Help
[Icons: Open, Save, Undo, Redo, Find, etc.]

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)

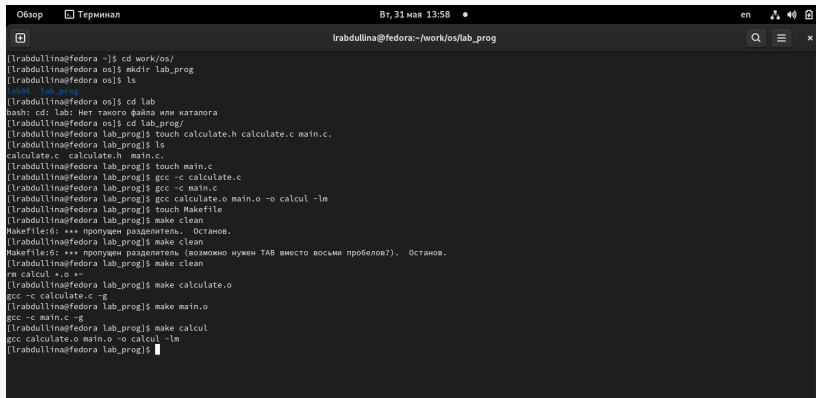
main.o: main.c calculate.h
$(CC) -c main.c $(CFLAGS)

clean:
-rm calcul *.o *~

-:==== Makefile All L1 (GNUmakefile)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
```

Выполнение работы

После этого удалим исполняемые и объектные файлы из каталога с помощью команды «make clean». Выполним компиляцию файлов, используя команды «make calculate.o», «make main.o», «male calcul» (скриншот 9)



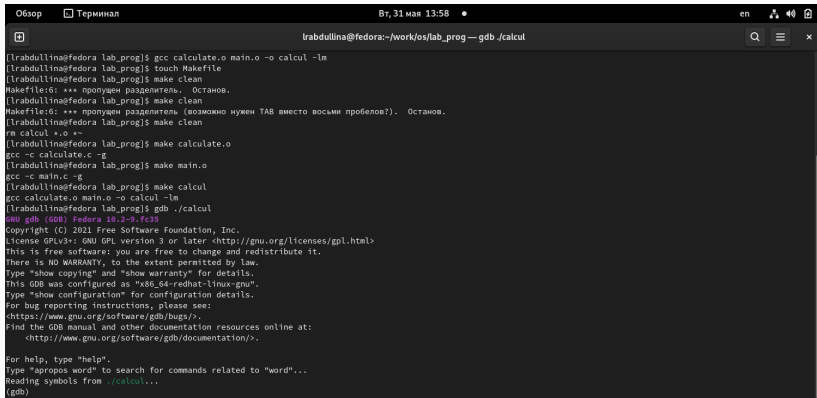
```
Обзор Терминал Вт, 31 мая 13:58 en
lrabdu1lina@fedora:~/work/os/lab_prog

[lrabdu1lina@fedora ~]$ cd work/os/
[lrabdu1lina@fedora os]$ mkdir lab_prog
[lrabdu1lina@fedora os]$ ls
labos  lab_prog
[lrabdu1lina@fedora os]$ cd lab
bash: cd: lab: Нет такого файла или каталога
[lrabdu1lina@fedora os]$ cd lab_prog/
[lrabdu1lina@fedora lab_prog]$ touch calculate.h calculate.c main.c.
[lrabdu1lina@fedora lab_prog]$ ls
calculate.c calculate.h main.c.
[lrabdu1lina@fedora lab_prog]$ touch main.c
[lrabdu1lina@fedora lab_prog]$ gcc -c calculate.c
[lrabdu1lina@fedora lab_prog]$ gcc -c main.c
[lrabdu1lina@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[lrabdu1lina@fedora lab_prog]$ touch Makefile
[lrabdu1lina@fedora lab_prog]$ make clean
Makefile:6: *** пропущен разделитель. Останов.
[lrabdu1lina@fedora lab_prog]$ make clean
Makefile:6: *** пропущен разделитель (возможно нужен TAB вместо восьми пробелов?). Останов.
[lrabdu1lina@fedora lab_prog]$ make clean
rm calcul *.o +-
[lrabdu1lina@fedora lab_prog]$ make calculate.o
gcc -c calculate.c -g
[lrabdu1lina@fedora lab_prog]$ make main.o
gcc -c main.c -g
[lrabdu1lina@fedora lab_prog]$ make calcul
gcc calculate.o main.o -o calcul -lm
[lrabdu1lina@fedora lab_prog]$
```

Figure 9: Компиляция файлов

Выполнение работы

- Запустим отладчик GDB, загрузив в него программу для отладки “gdb ./calcul” (скриншот 10)



```
Обзор Терминал Вт, 31 мая 13:58 • en
lrabdullina@fedora: ~/work/os/lab_prog — gdb ./calcul

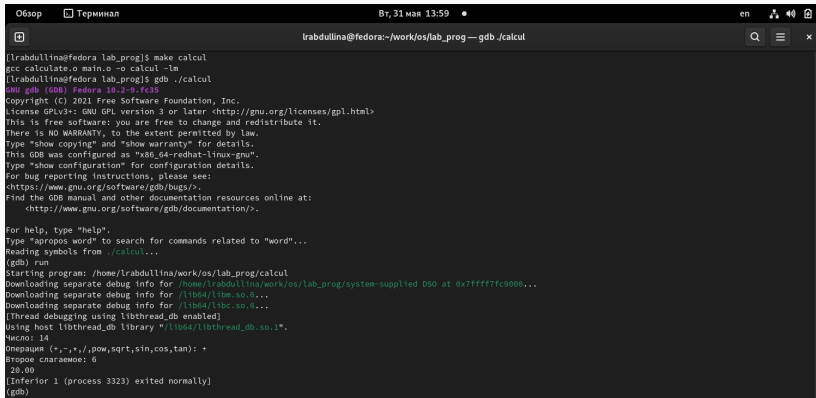
[lrabdullina@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[lrabdullina@fedora lab_prog]$ touch Makefile
[lrabdullina@fedora lab_prog]$ make clean
Makefile:6: *** пропущен разделитель. Останов.
[lrabdullina@fedora lab_prog]$ make clean
Makefile:6: *** пропущен разделитель (возможно нужен TAB вместо восьми пробелов?). Останов.
[lrabdullina@fedora lab_prog]$ make clean
rm calcul *.o *~
[lrabdullina@fedora lab_prog]$ make calculate.o
gcc -c calculate.c -g
[lrabdullina@fedora lab_prog]$ make main.o
gcc -c main.c -g
[lrabdullina@fedora lab_prog]$ make calcul
gcc calculate.o main.o -o calcul -lm
[lrabdullina@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 10.2-9.fc30
Copyright (C) 2021 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-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"...
Reading symbols from ./calcul...
(gdb)
```

Figure 10: Запуск отладчика

Выполнение работы

- Для запуска программы внутри отладчика введем команду “run” и произведем арифметическую операцию (скриншот 11)



```
Обзор Терминал Вт, 31 мая 13:59 • en
lrabduullina@fedora:~/work/os/lab_prog — gdb ./calcul

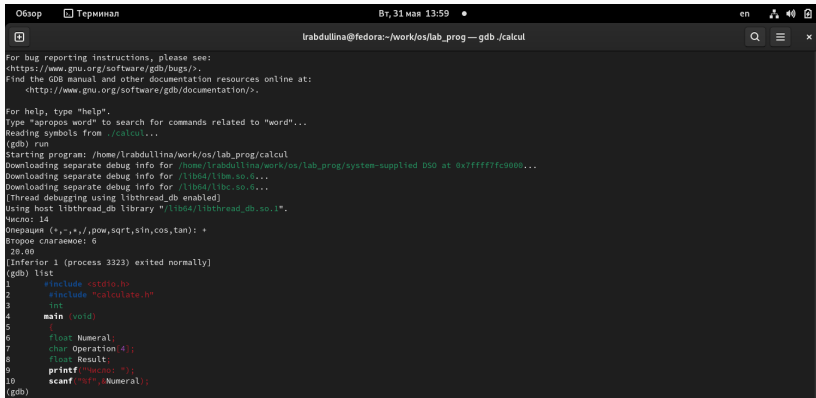
[lrabduullina@fedora lab_prog]$ make calcul
gcc calculate.o main.o -o calcul -lm
[lrabduullina@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 10.2-9.fc35
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: 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-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"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /home/lrabduullina/work/os/lab_prog/calcul
Downloading separate debug info for /home/lrabduullina/work/os/lab_prog/system-supplied DSO at 0x7ffff7fc9000...
Downloading separate debug info for /lib64/libm.so.6...
Downloading separate debug info for /lib64/libc.so.6...
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 14
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 6
20.00
[Inferior 1 (process 3323) exited normally]
(gdb)
```

Figure 11: Запуск программы внутри отладчика

Выполнение работы

- Для постраничного (по 9 строк) просмотра исходного код используем команду “list”(скриншот 12)



```
Обзор Терминал  Вт, 31 мая 13:59  en  🔊  🏠  🔍  ☰  ✕

lrabdullina@fedora:~/work/os/lab_prog — gdb ./calcul

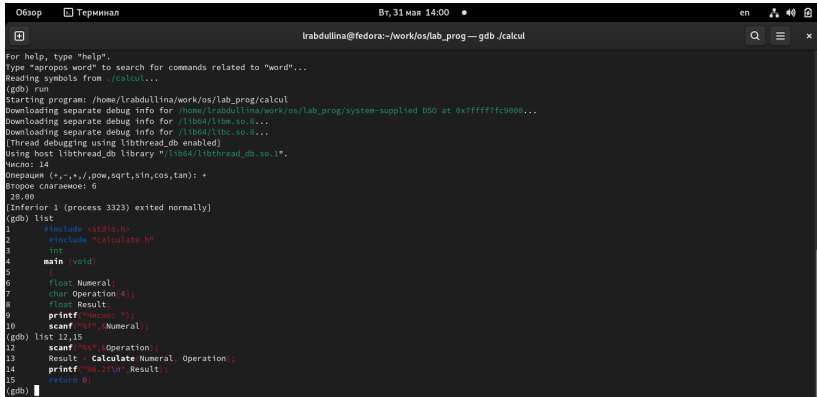
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /home/lrabdullina/work/os/lab_prog/calcul
Downloading separate debug info for /home/lrabdullina/work/os/lab_prog/system-supplied 050 at 0x7ffff7fc9000...
Downloading separate debug info for /lib64/libm.so.6...
Downloading separate debug info for /lib64/libc.so.6...
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Wcno: 14
Операция (+, -, *, /, pow, sqrt, sin, cos, tan): +
Стопое слагаемое: 6
28.00
[Inferior 1 (process 3323) exited normally]
(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3      int
4      main (void)
5      {
6          float Numeral;
7          char Operation[4];
8          float Result;
9          printf("Wcno: ");
10         scanf("%f",&Numeral);
(gdb)
```

Figure 12: Просмотр кода (постранично)

Выполнение работы

- Для просмотра строк с 12 по 15 основного файла используем “list 12,15” (скриншот 13)



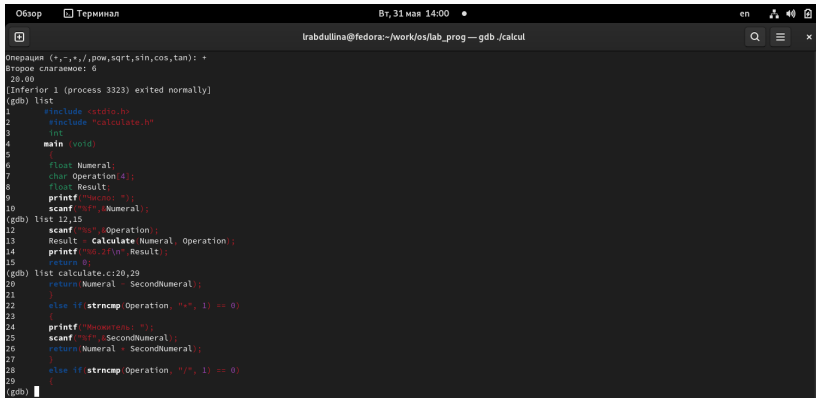
```
Обзор Терминал Вт, 31 мая 14:00 en
lrabdullina@fedora:~/work/os/lab_prog — gdb ./calcul

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /home/lrabdullina/work/os/lab_prog/calcul
Downloading separate debug info for /home/lrabdullina/work/os/lab_prog/system-supplied DSO at 0x7ffff7fc9008...
Downloading separate debug info for /lib64/libm.so.6...
Downloading separate debug info for /lib64/libc.so.6...
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 14
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 6
20.90
[Inferior 1 (process 3323) exited normally]
(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3      int
4      main (void)
5      {
6          float Numeral;
7          char Operation[4];
8          float Result;
9          printf("Число: ");
10         scanf("%f",&Numeral);
(gdb) list 12,15
12         scanf("%s",&Operation);
13         Result = Calculate(Numeral, Operation);
14         printf("%g.2f\n",Result);
15         return 0;
(gdb) |
```

Figure 13: Просмотр строк 12-15

Выполнение работы

- Для просмотра определённых строк не основного файла используем “list calculate.c:20,29” (скриншот 14)



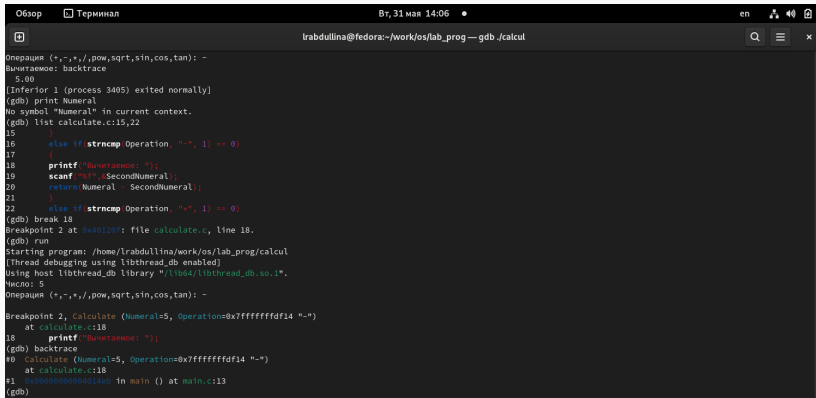
```
Обзор Терминал Вт, 31 мая 14:00 • en
lrabduullina@fedora: ~/work/os/lab_prog — gdb ./calcul

Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 6
20.00
[Inferior 1 (process 3323) exited normally]
(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3      int
4      main (void)
5      {
6          float Numeral;
7          char Operation(4);
8          float Result;
9          printf("Введите: ");
10         scanf("%f",&Numeral);
(gdb) list 12,15
12         scanf("%s",&Operation);
13         Result = calculate(Numeral, Operation);
14         printf("%0.2f\n",Result);
15         return 0;
(gdb) list calculate.c:20,29
20     return(Numeral * SecondNumeral);
21     }
22     else if(strcmp( Operation, "+" ) == 0)
23     {
24         printf("Умножение: ");
25         scanf("%f",&SecondNumeral);
26         return(Numeral * SecondNumeral);
27     }
28     else if(strcmp( Operation, "/" ) == 0)
29     {
(gdb) |
```

Figure 14: Просмотр строк не основного файла

Выполнение работы

- Установим точку останова в файле calculate.c на строке номер 18 с командами “list calculate.c:20,27”, “break 18” (скриншот 15)



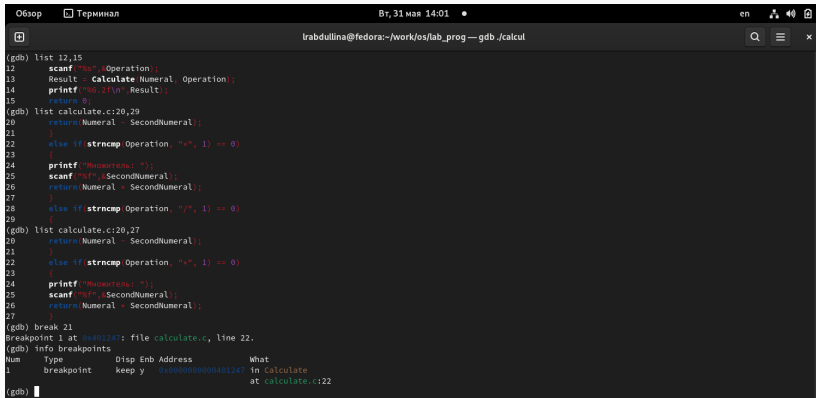
```
Обзор Терминал Вт, 31 мая 14:06 • en
lrabdu...@fedora: ~/work/os/lab_prog — gdb ./calcul

Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: backtrack
5,00
[Inferior 1 (process 3405) exited normally]
(gdb) print Numeral
No symbol "Numeral" in current context.
(gdb) list calculate.c:15,22
15     }
16     else if(strncmp Operation, "--", 1) == 0)
17     {
18         printf("Вычитаемое: ");
19         scanf("%f",&SecondNumeral);
20         return (Numeral - SecondNumeral);
21     }
22     else if(strncmp Operation, "+", 1) == 0)
(gdb) break 18
Breakpoint 2 at 0x40120f: file calculate.c, line 18.
(gdb) run
Starting program: /home/lrabdu.../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): -
Breakpoint 2, Calculate (Numeral=5, Operation=0x7fffffffdf14 "--")
at calculate.c:18
18     printf("Вычитаемое: ");
(gdb) backtrack
#0 Calculate (Numeral=5, Operation=0x7fffffffdf14 "--")
at calculate.c:18
#1 0x00000000004014eb in main () at main.c:13
(gdb)
```

Figure 15: Установка точки останова

Выполнение работы

- Выведем информацию об имеющихся в проекте точках останова “info breakpoints” (скриншот 16)

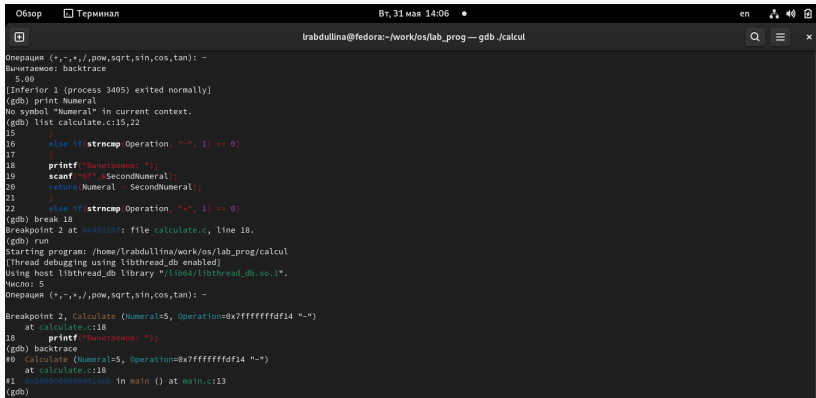


```
(gdb) list 12,15
12      scanf("%s",&Operation);
13      Result = Calculate_Numeral_Operation);
14      printf("%g.2f\n",Result);
15      return 0;
(gdb) list calculate.c:20,29
20      return (Numer1 - SecondNumer1);
21  }
22      else if(strcmp Operation, "+", 1) == 0)
23  {
24      printf("Множительство: ");
25      scanf("%f",&SecondNumer1);
26      return (Numer1 + SecondNumer1);
27  }
28      else if(strcmp Operation, "/", 1) == 0)
29  {
(gdb) list calculate.c:20,27
20      return (Numer1 - SecondNumer1);
21  }
22      else if(strcmp Operation, "+", 1) == 0)
23  {
24      printf("Множительство: ");
25      scanf("%f",&SecondNumer1);
26      return (Numer1 + SecondNumer1);
27  }
(gdb) break 21
Breakpoint 1 at 0x401247: file calculate.c, line 22.
(gdb) info breakpoints
Num   Type             Disp Enb Address            What
1     breakpoint       keep y   0x00000000401247   in Calculate
                                at calculate.c:22
(gdb)
```

Figure 16: Информация о точках останова

Выполнение работы

- Запустите программу внутри отладчика и убедимся, что программа остановится в момент прохождения точки останова. (скриншот 17)



```
Обзор Терминал Bt, 31 мая 14:06 • en
lrabdullina@fedora: ~/work/os/lab_prog — gdb ./calcul

Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: backtrack
5,00
[Inferior 1 (process 3405) exited normally]
(gdb) print Numeral
No symbol "Numeral" in current context.
(gdb) list calculate.c:15,22
15      }
16      else if(strcmp Operation, "-") == 0)
17      {
18          printf("Вычитаемое: ");
19          scanf("%f",&SecondNumeral);
20          return (Numeral - SecondNumeral);
21      }
22      else if(strcmp Operation, "+") == 0)
(gdb) break 18
Breakpoint 2 at 0x40120f: file calculate.c, line 18.
(gdb) run
Starting program: /home/lrabdullina/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): -

Breakpoint 2, Calculate (Numeral=5, Operation=0x7fffffffdf14 "-")
at calculate.c:18
18      printf("Вычитаемое: ");
(gdb) backtrack
#0 Calculate (Numeral=5, Operation=0x7fffffffdf14 "-")
at calculate.c:18
#1 0x00000000004014eb in main () at main.c:13
(gdb)
```

Figure 17: Остановка у точки

- Отладчик выдал следующую информацию: #0 Calculate (Numeral=5, Operation=0x7fffffff280 "-") at calculate.c:21 #1 0x000000000400b2b in main () at main.c:17

а команда “backtrace” показала весь стек вызываемых функций от начала программы до текущего места.

Выполнение работы

- Посмотрим, чему равно на этом этапе значение переменной Numeral, введя “print Numeral”. На экране выведено число 5. (скриншот 18)

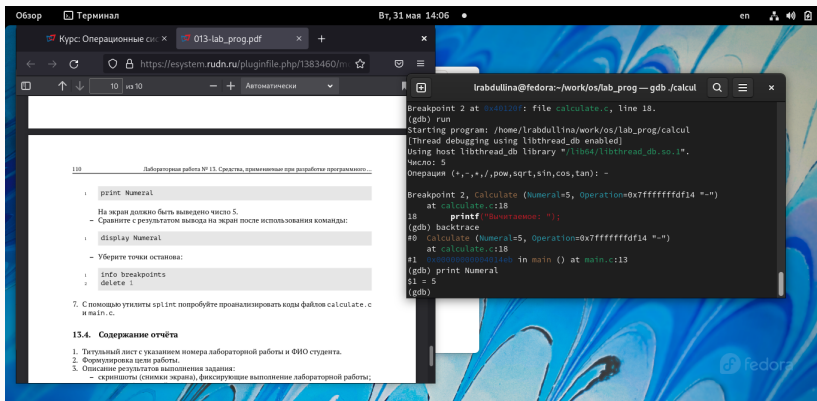


Figure 18: Значение Numeral

Выполнение работы

- Сравним с результатом вывода на экран после использования команды “display Numeral”. Значения совпадают (скриншот 19)

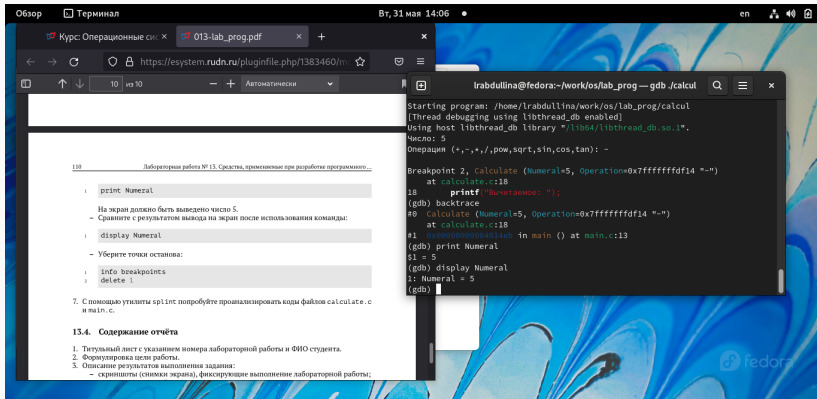


Figure 19: Значение Numeral

- Уберем точки останова “info breakpoints”, “delete 1” (скриншот 20)

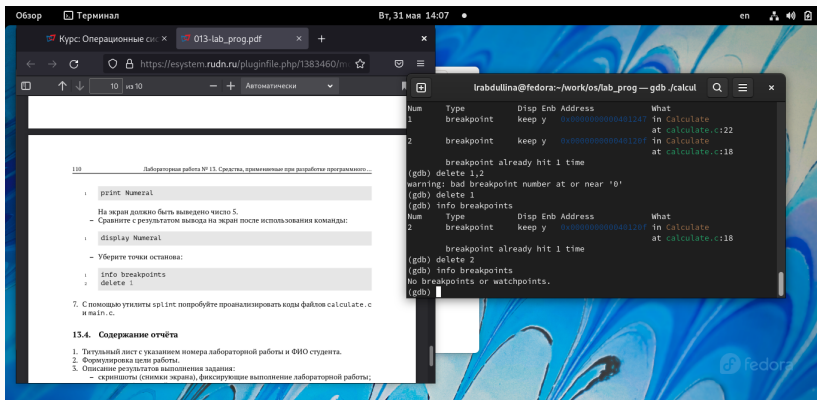
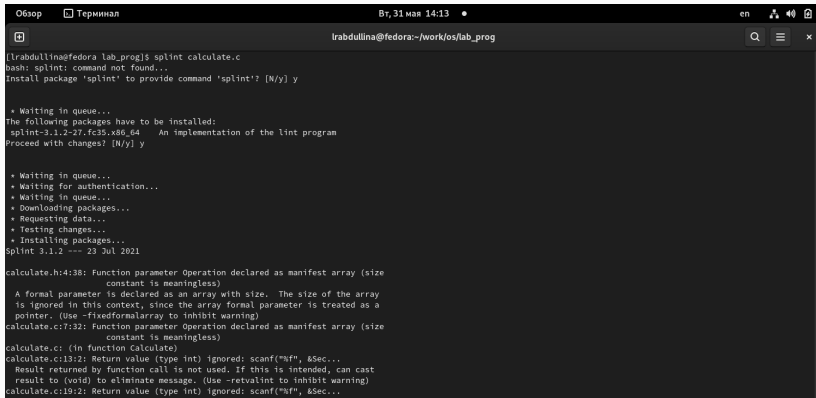


Figure 20: Убираем точки останова

7. С помощью утилиты splint попробуем проанализировать коды файлов calculate.c и main.c. (скриншот 21,22, 23)



```
Обзор Терминал  Вт, 31 мая 14:13  en  [иконки]
lrabduullina@fedora:~/work/os/lab_prog

[lrabduullina@fedora lab_prog]$ splint calculate.c
bash: splint: command not found...
Install package 'splint' to provide command 'splint'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
splint-3.1.2-27.fc35.x86_64  An implementation of the lint program
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
Splint 3.1.2 --- 23 Jul 2021

calculate.h:4:38: 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:7:32: Function parameter Operation declared as manifest array (size
        constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:13:2: 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:19:2: Return value (type int) ignored: scanf("%f", &Sec...
```

Figure 21: Загрузка утилиты

Выполнение работы

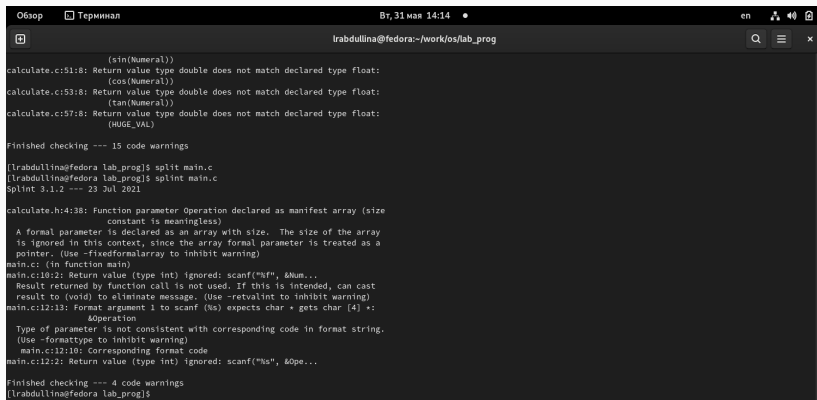
```
Обзор Терминал Брт, 31 мая 14:13 en
```

```
lrabduлина@fedora:~/work/os/lab_prog
```

```
calculate.c:13:2: 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:19:2: Return value (type int) ignored: scanf("%f", &Sec...  
calculate.c:25:2: Return value (type int) ignored: scanf("%f", &Sec...  
calculate.c:31:1: Return value (type int) ignored: scanf("%f", &Sec...  
calculate.c:32:5: 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  
representations are inexact. Instead, compare the difference to FLT_EPSILON  
or DBL_EPSILON. (Use -realcompare to inhibit warning)  
calculate.c:35:8: Return value type double does not match declared type float:  
    (HUGE_VAL)  
To allow all numeric types to match, use +relaxtypes.  
calculate.c:43:2: Return value (type int) ignored: scanf("%f", &Sec...  
calculate.c:44:8: Return value type double does not match declared type float:  
    (pow(Numeral, SecondNumeral))  
calculate.c:47:8: Return value type double does not match declared type float:  
    (sqrt(Numeral))  
calculate.c:49:8: Return value type double does not match declared type float:  
    (sin(Numeral))  
calculate.c:51:8: Return value type double does not match declared type float:  
    (cos(Numeral))  
calculate.c:53:8: Return value type double does not match declared type float:  
    (tan(Numeral))  
calculate.c:57:8: Return value type double does not match declared type float:  
    (HUGE_VAL)  
  
Finished checking --- 15 code warnings  
  
[lrabduлина@fedora lab_prog]$
```

Figure 22: Информация о calculate.c

Выполнение работы



```
Обзор Терминал Вт, 31 мая 14:14 en
lrabduлина@fedora:~/work/os/lab_prog

(sin(Numeral))
calculate.c:51:8: Return value type double does not match declared type float:
(cos(Numeral))
calculate.c:53:8: Return value type double does not match declared type float:
(tan(Numeral))
calculate.c:57:8: Return value type double does not match declared type float:
(HUGE_VAL)

Finished checking --- 15 code warnings

[lrabduлина@fedora lab_prog]$ split main.c
[lrabduлина@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2021

calculate.h:4:38: 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:10:2: 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:12:13: 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:12:10: Corresponding format code
main.c:12:2: Return value (type int) ignored: scanf("%s", &Ope...

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

Figure 23: Информация о main.c

С помощью утилиты `splint` выяснилось, что в файлах `calculate.c` и `main.c` присутствует функция чтения `scanf`, возвращающая целое число (тип `int`), но эти числа не используются и нигде не сохраняются. Утилита вывела предупреждение о том, что в файле `calculate.c` происходит сравнение вещественного числа с нулем. Также возвращаемые значения (тип `double`) в функциях `pow`, `sqrt`, `sin`, `cos` и `tan` записываются в переменную типа `float`, что свидетельствует о потере данных.

В ходе лабораторной работы мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.