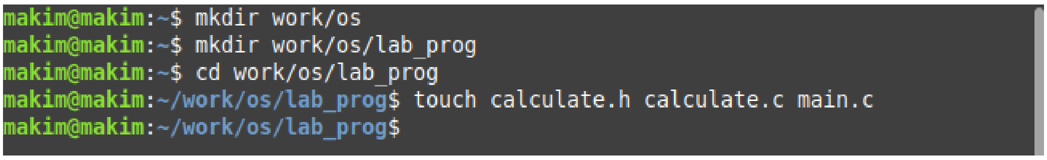
Презентация по лабораторной работе №13

Ким Михаил Алексеевич

1. Цель работы

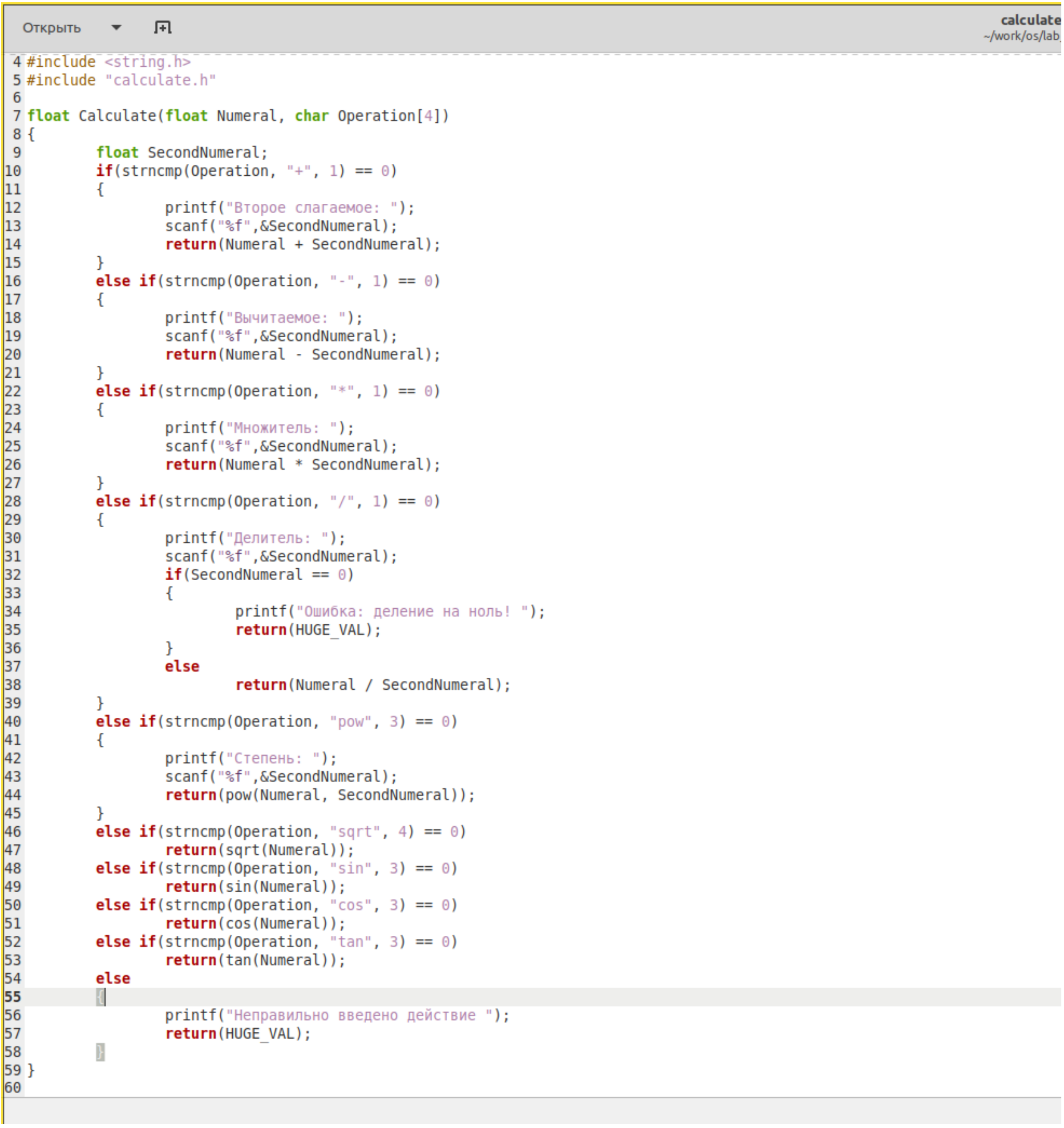
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

2. Выполнение лабораторной работы



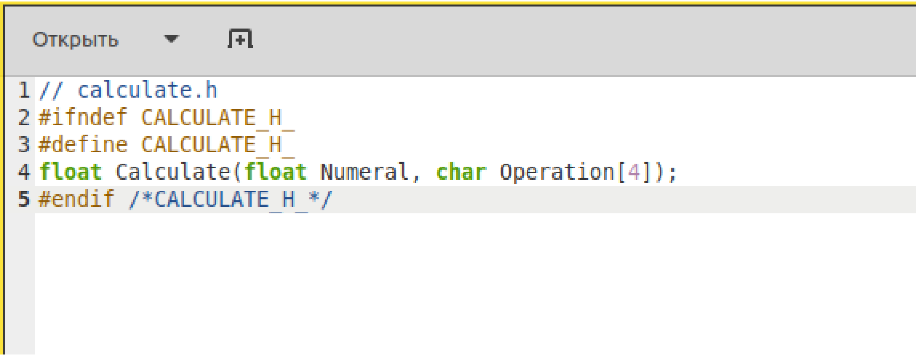
```
makim@makim:~$ mkdir work/os
makim@makim:~$ mkdir work/os/lab_prog
makim@makim:~$ cd work/os/lab_prog
makim@makim:~/work/os/lab_prog$ touch calculate.h calculate.c main.c
makim@makim:~/work/os/lab_prog$
```

Рис. 2.1: Подготовка рабочей среды



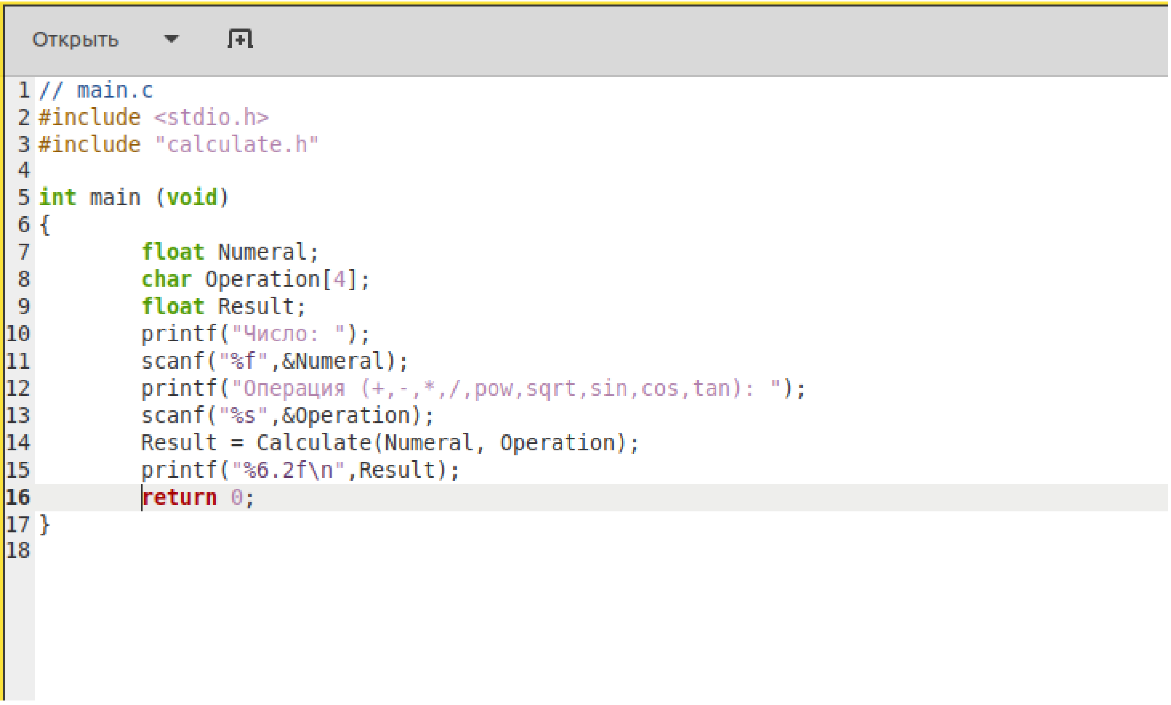
```c
#include <string.h>
#include "calculate.h"

float Calculate(float Numeral, char Operation[4])
{
        float SecondNumeral;
        if(strncmp(Operation, "+", 1) == 0)
        {
                printf("Второе слагаемое: ");
                scanf("%f",&SecondNumeral);
                return(Numeral + SecondNumeral);
        }
        else if(strncmp(Operation, "-", 1) == 0)
        {
                printf("Вычитаемое: ");
                scanf("%f",&SecondNumeral);
                return(Numeral - SecondNumeral);
        }
        else if(strncmp(Operation, "*", 1) == 0)
        {
                printf("Множитель: ");
                scanf("%f",&SecondNumeral);
                return(Numeral * SecondNumeral);
        }
        else if(strncmp(Operation, "/", 1) == 0)
        {
                printf("Делитель: ");
                scanf("%f",&SecondNumeral);
                if(SecondNumeral == 0)
                {
                        printf("Ошибка: деление на ноль! ");
                        return(HUGE_VAL);
                }
                else
                        return(Numeral / SecondNumeral);
        }
        else if(strncmp(Operation, "pow", 3) == 0)
        {
                printf("Степень: ");
                scanf("%f",&SecondNumeral);
                return(pow(Numeral, SecondNumeral));
        }
        else if(strncmp(Operation, "sqrt", 4) == 0)
                return(sqrt(Numeral));
        else if(strncmp(Operation, "sin", 3) == 0)
                return(sin(Numeral));
        else if(strncmp(Operation, "cos", 3) == 0)
                return(cos(Numeral));
        else if(strncmp(Operation, "tan", 3) == 0)
                return(tan(Numeral));
        else
        {
                printf("Неправильно введено действие ");
                return(HUGE_VAL);
        }
}
```

Рис. 2.2: Исходный код calculate.c



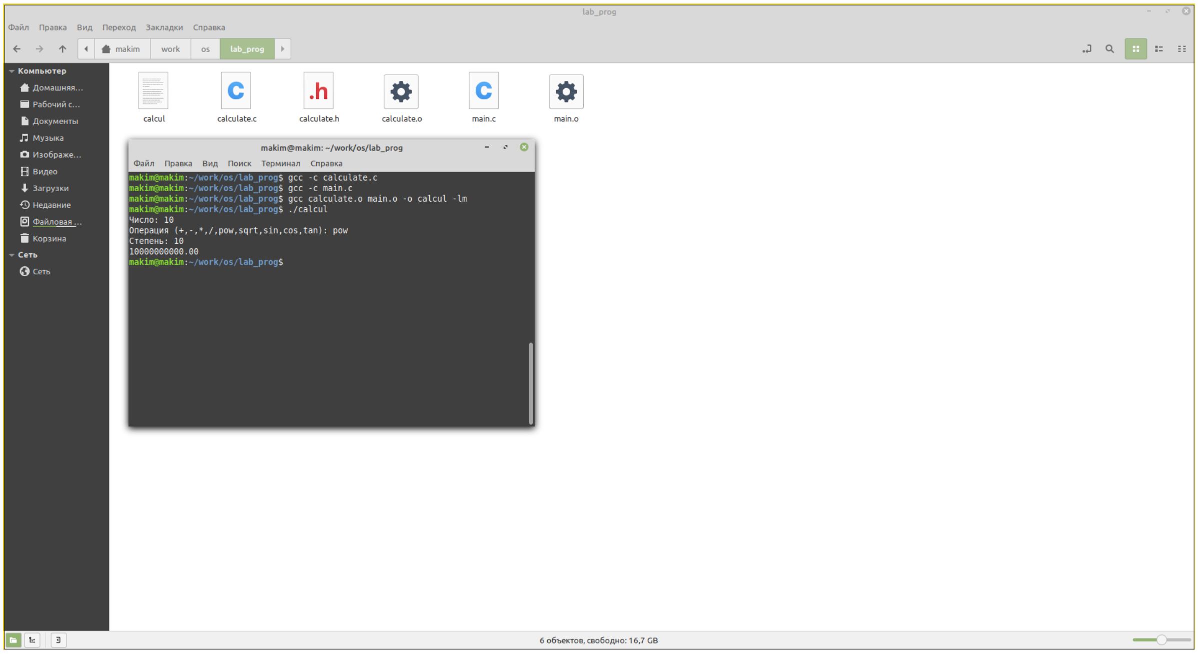
```
// calculate.h
#ifndef CALCULATE_H_
#define CALCULATE_H_
float Calculate(float Numeral, char Operation[4]);
#endif /*CALCULATE_H_*/
```

Рис. 2.3: Исходный код calculate.h



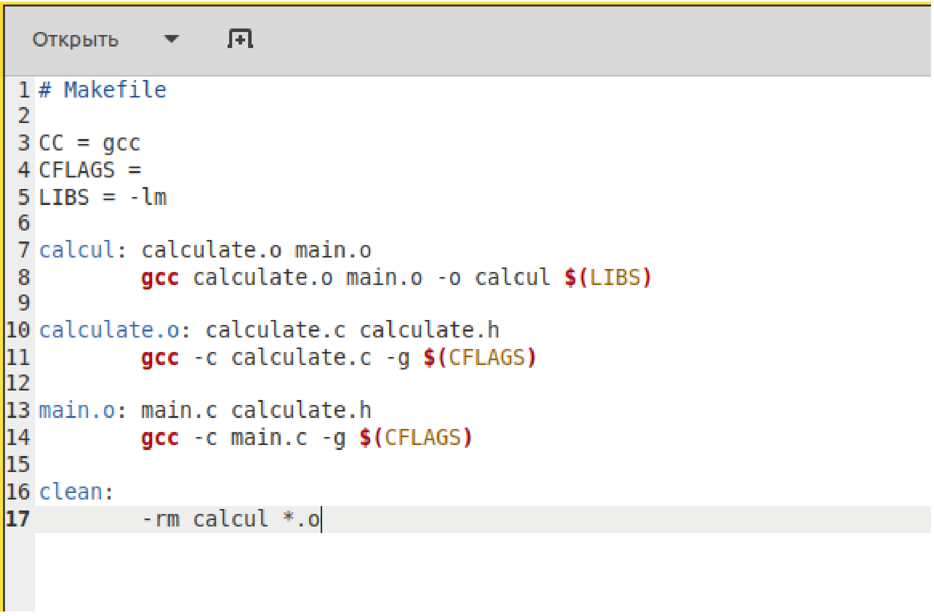
```
// main.c
#include <stdio.h>
#include "calculate.h"

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;
}

```

Рис. 2.4: Исходный код main.c



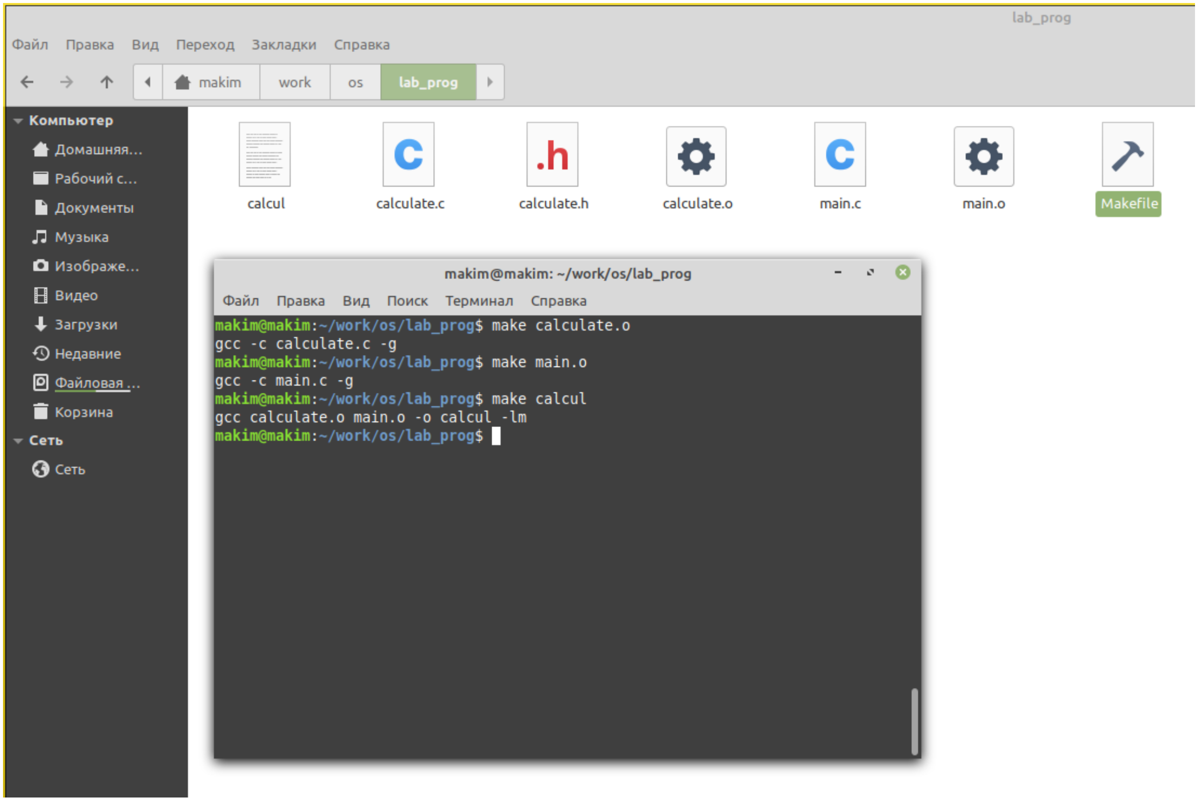
```
makim@makim:~/work/os/lab_prog$ gcc -c calculate.c
makim@makim:~/work/os/lab_prog$ gcc -c main.c
makim@makim:~/work/os/lab_prog$ gcc calculate.o main.o -o calcul -lm
makim@makim:~/work/os/lab_prog$ ./calcul
Число: 10
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): pow
Степень: 10
10000000000.00
makim@makim:~/work/os/lab_prog$
```

Рис. 2.5: Компиляция



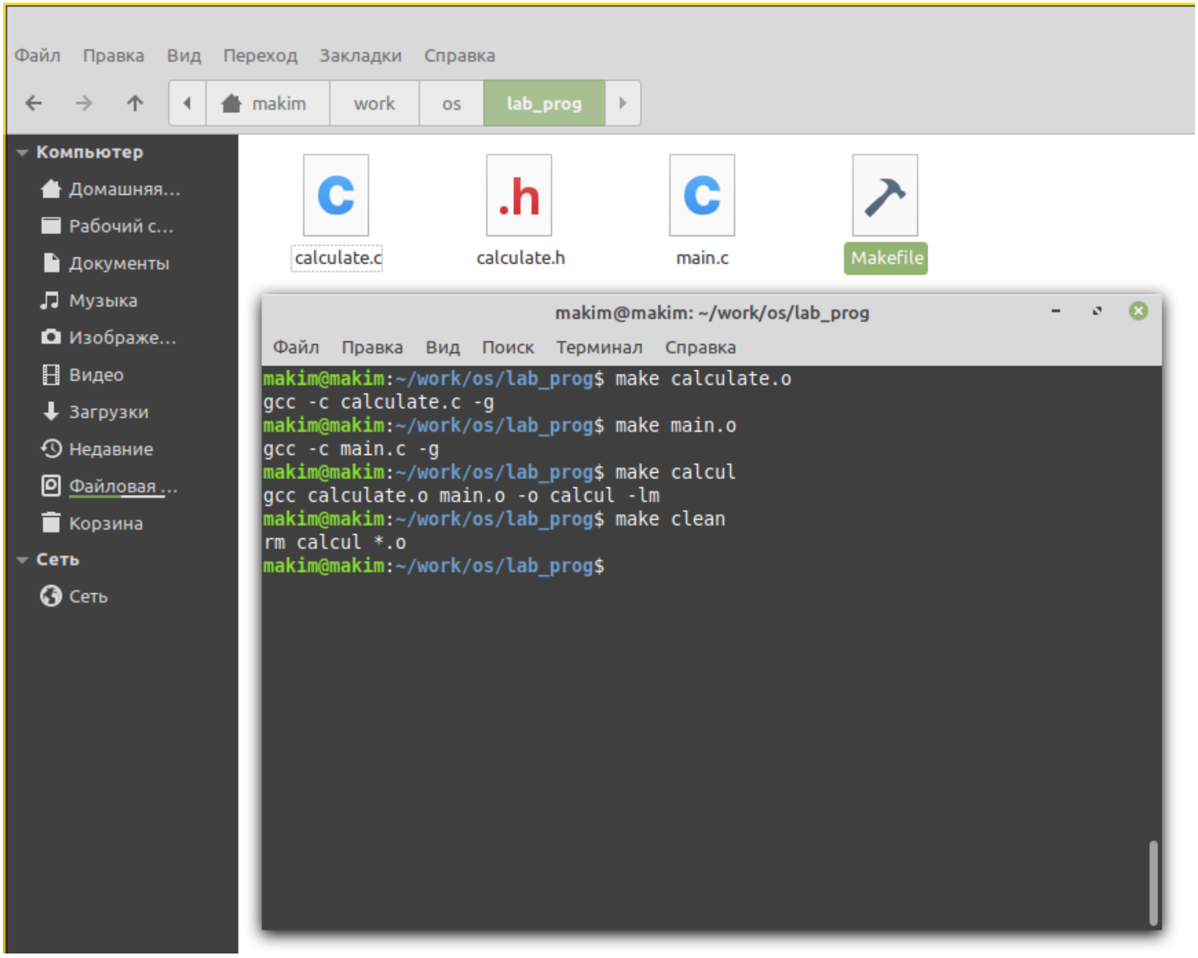
```
# Makefile

CC = gcc
CFLAGS =
LIBS = -lm

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

calculate.o: calculate.c calculate.h
	gcc -c calculate.c -g $(CFLAGS)

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

clean:
	-rm calcul *.o
```

Рис. 2.6: Makefile



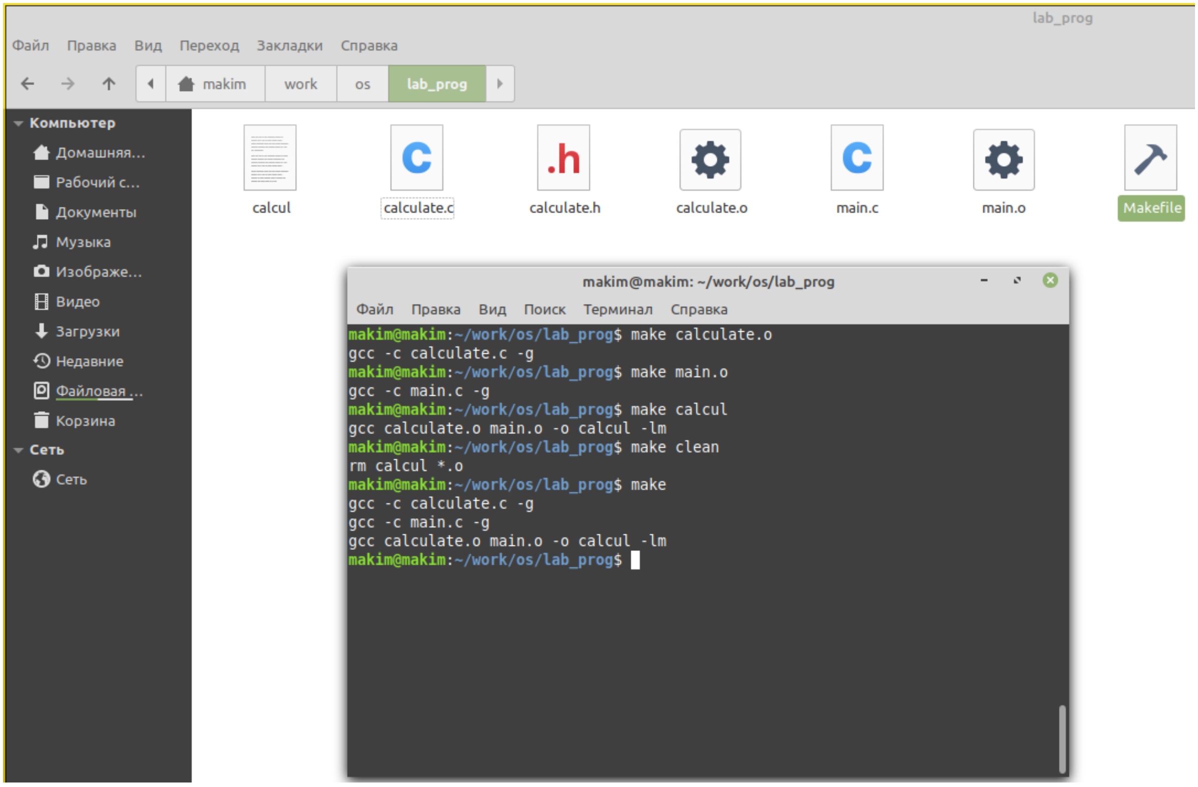
```
makim@makim:~/work/os/lab_prog$ make calculate.o
gcc -c calculate.c -g
makim@makim:~/work/os/lab_prog$ make main.o
gcc -c main.c -g
makim@makim:~/work/os/lab_prog$ make calcul
gcc calculate.o main.o -o calcul -lm
makim@makim:~/work/os/lab_prog$
```

Рис. 2.7: Использование Makefile



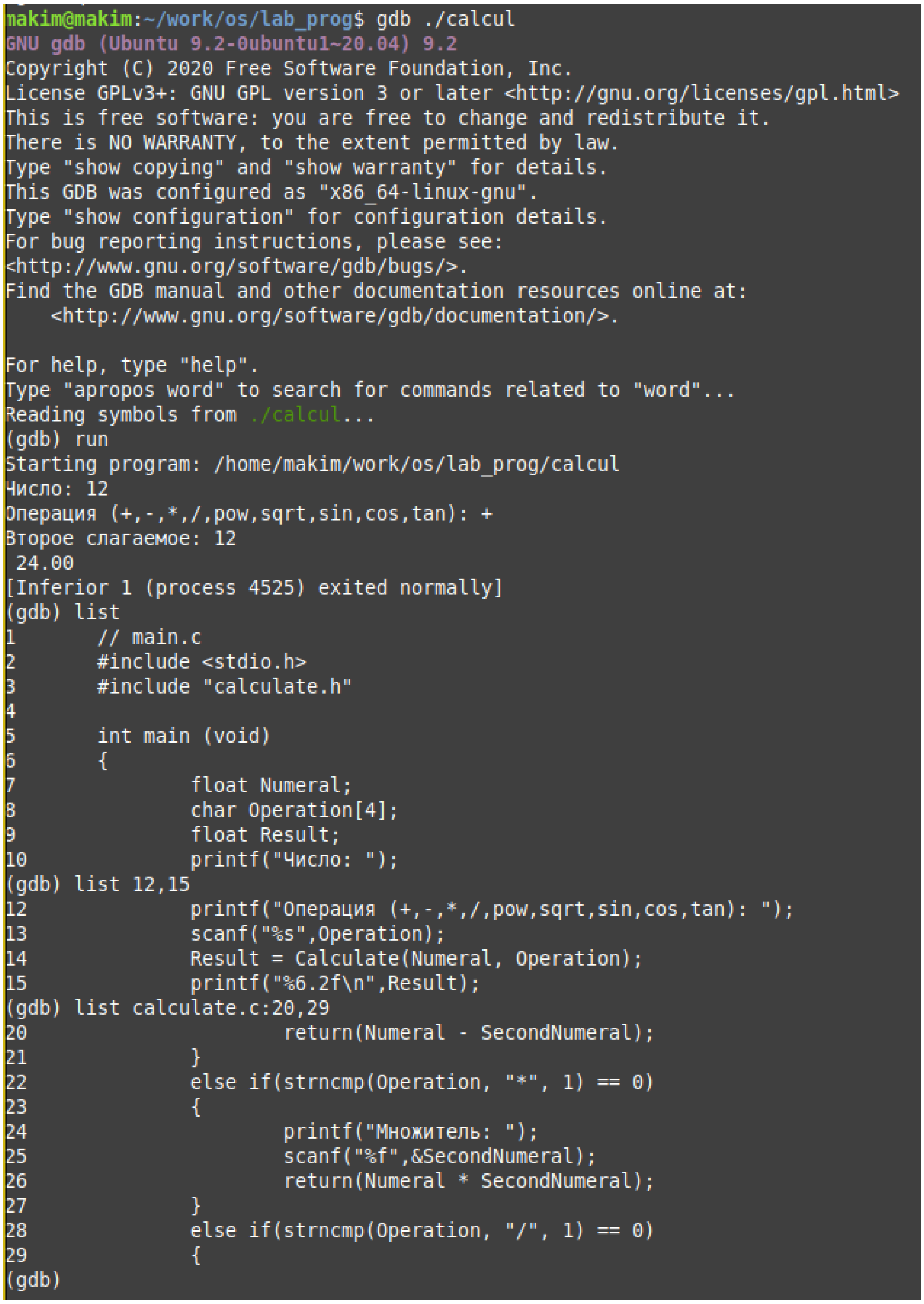
```
makim@makim:~/work/os/lab_prog$ make calculate.o
gcc -c calculate.c -g
makim@makim:~/work/os/lab_prog$ make main.o
gcc -c main.c -g
makim@makim:~/work/os/lab_prog$ make calcul
gcc calculate.o main.o -o calcul -lm
makim@makim:~/work/os/lab_prog$ make clean
rm calcul *.o
makim@makim:~/work/os/lab_prog$
```

Рис. 2.8: Использование Makefile



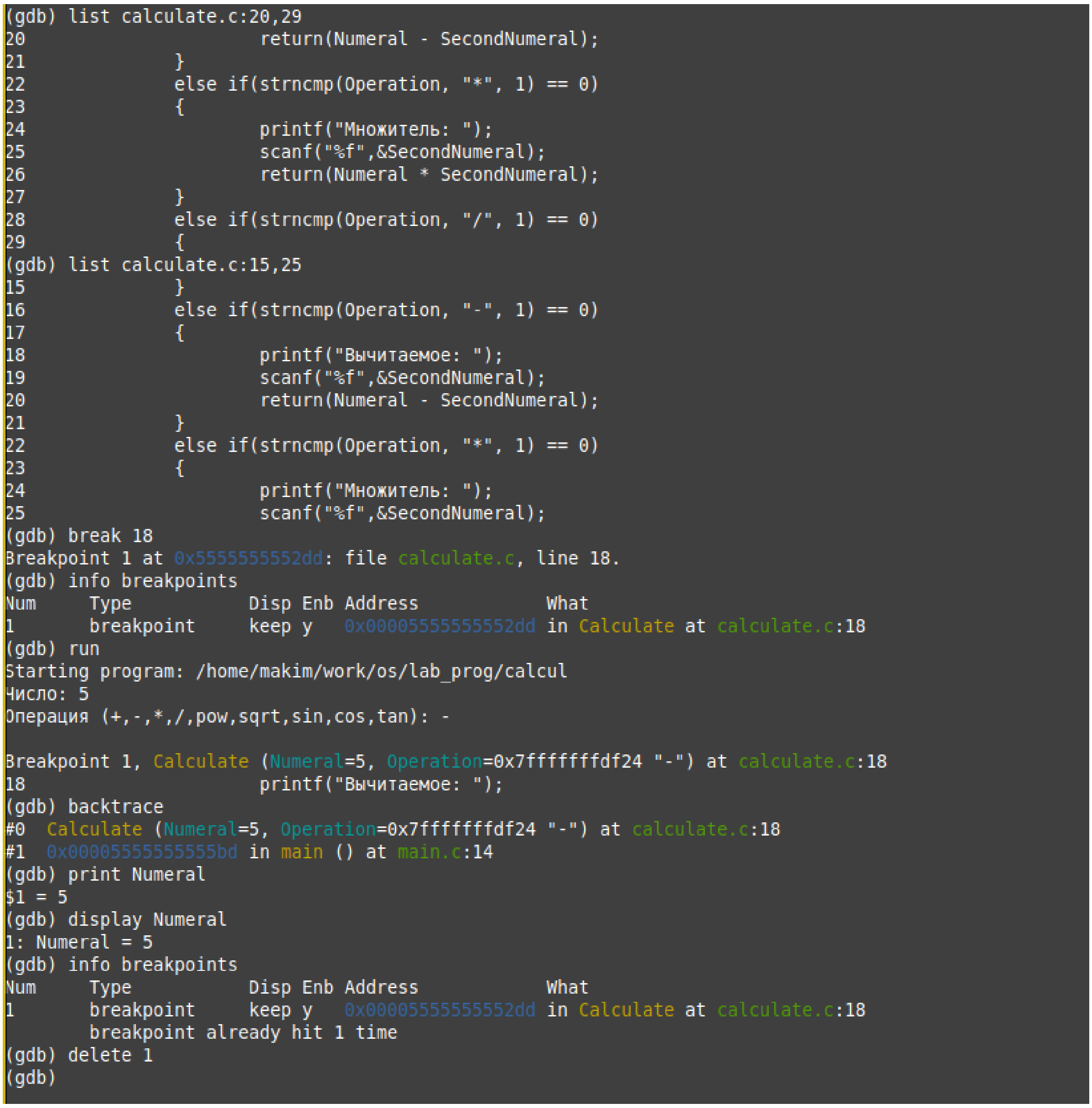
```
makim@makim:~/work/os/lab_prog$ make calculate.o
gcc -c calculate.c -g
makim@makim:~/work/os/lab_prog$ make main.o
gcc -c main.c -g
makim@makim:~/work/os/lab_prog$ make calcul
gcc calculate.o main.o -o calcul -lm
makim@makim:~/work/os/lab_prog$ make clean
rm calcul *.o
makim@makim:~/work/os/lab_prog$ make
gcc -c calculate.c -g
gcc -c main.c -g
gcc calculate.o main.o -o calcul -lm
makim@makim:~/work/os/lab_prog$ 
```

Рис. 2.9: Использование Makefile



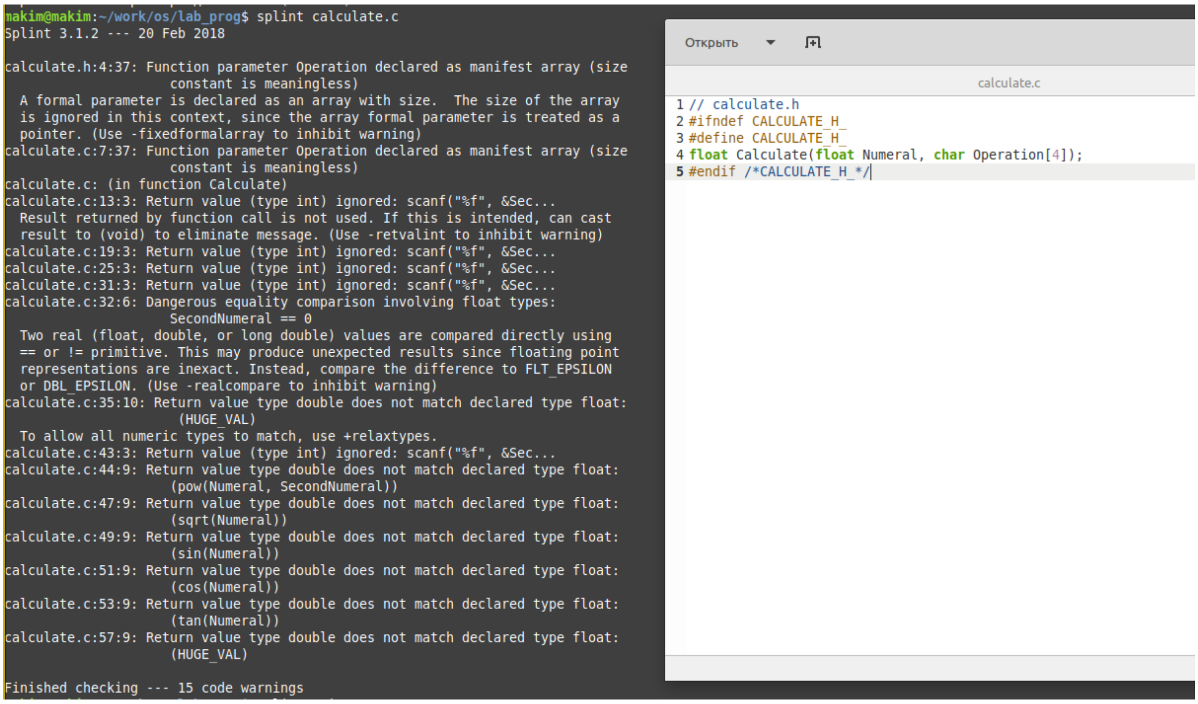
```
makim@makim:~/work/os/lab_prog$ gdb ./calcul
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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/makim/work/os/lab_prog/calcul
Число: 12
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 12
 24.00
[Inferior 1 (process 4525) exited normally]
(gdb) list
1       // main.c
2       #include <stdio.h>
3       #include "calculate.h"
4
5       int main (void)
6       {
7               float Numeral;
8               char Operation[4];
9               float Result;
10              printf("Число: ");
(gdb) list 12,15
12              printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
13              scanf("%s",Operation);
14              Result = Calculate(Numeral, Operation);
15              printf("%6.2f\n",Result);
(gdb) list calculate.c:20,29
20                      return(Numeral - SecondNumeral);
21              }
22              else if(strncmp(Operation, "*", 1) == 0)
23              {
24                      printf("Множитель: ");
25                      scanf("%f",&SecondNumeral);
26                      return(Numeral * SecondNumeral);
27              }
28              else if(strncmp(Operation, "/", 1) == 0)
29              {
(gdb)
```

Рис. 2.10: Отладчик GDB



```
(gdb) list calculate.c:20,29
20                      return(Numeral - SecondNumeral);
21              }
22              else if(strncmp(Operation, "*", 1) == 0)
23              {
24                      printf("Множитель: ");
25                      scanf("%f",&SecondNumeral);
26                      return(Numeral * SecondNumeral);
27              }
28              else if(strncmp(Operation, "/", 1) == 0)
29              {
(gdb) list calculate.c:15,25
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)
23              {
24                      printf("Множитель: ");
25                      scanf("%f",&SecondNumeral);
(gdb) break 18
Breakpoint 1 at 0x5555555552dd: file calculate.c, line 18.
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00005555555552dd in Calculate at calculate.c:18
(gdb) run
Starting program: /home/makim/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf24 "-") at calculate.c:18
18                      printf("Вычитаемое: ");
(gdb) backtrace
#0  Calculate (Numeral=5, Operation=0x7fffffffdf24 "-") at calculate.c:18
#1  0x00005555555555bd in main () at main.c:14
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00005555555552dd in Calculate at calculate.c:18
        breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

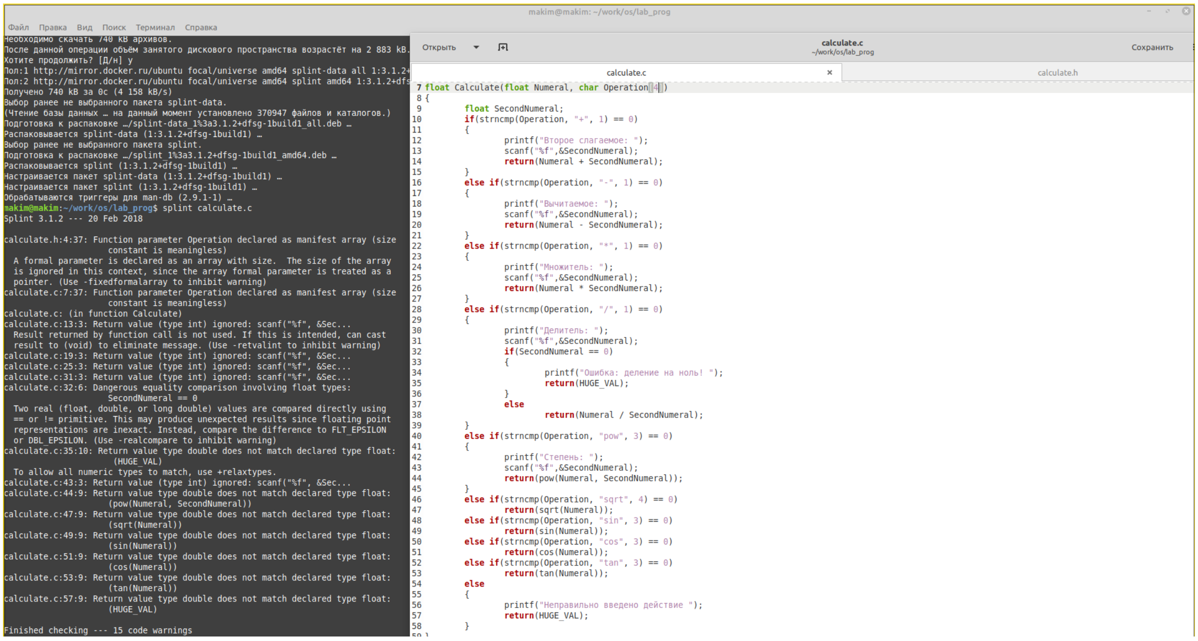
Рис. 2.11: Отладчик GDB



```
makim@makim:~/work/os/lab_prog$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2018

calculate.h:4:37: 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:7:37: Function parameter Operation declared as manifest array (size
                     constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:13:3: 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:19:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:25:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:6: 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:35:10: Return value type double does not match declared type float:
                      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:43:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:9: Return value type double does not match declared type float:
                     (pow(Numeral, SecondNumeral))
calculate.c:47:9: Return value type double does not match declared type float:
                     (sqrt(Numeral))
calculate.c:49:9: Return value type double does not match declared type float:
                     (sin(Numeral))
calculate.c:51:9: Return value type double does not match declared type float:
                     (cos(Numeral))
calculate.c:53:9: Return value type double does not match declared type float:
                     (tan(Numeral))
calculate.c:57:9: Return value type double does not match declared type float:
                     (HUGE_VAL)

Finished checking --- 15 code warnings
```

```c
// calculate.h
#ifndef CALCULATE_H_
#define CALCULATE_H_
float Calculate(float Numeral, char Operation[4]);
#endif /*CALCULATE_H_*/
```

Рис. 2.12: Предупреждения splint (calculate.c)

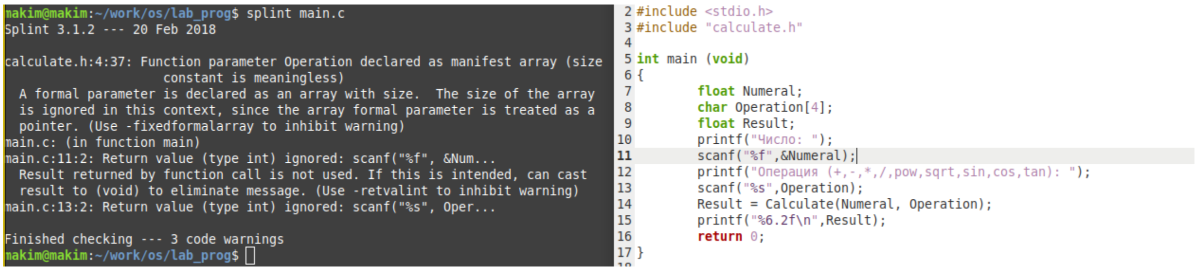


makim@makim: ~/work/os/lab_prog
Файл Правка Вид Поиск Терминал Справка
Открыть
calculate.c
~/work/os/lab_prog
Сохранить
calculate.c
calculate.h

```
Необходимо скачать 740 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 2 883 kB.
Хотите продолжить? [Д/н] y
Пол:1 http://mirror.docker.ru/ubuntu focal/universe amd64 splint-data all 1:3.1.2+
Пол:2 http://mirror.docker.ru/ubuntu focal/universe amd64 splint amd64 1:3.1.2+dfs
Получено 740 kB за 0c (4 158 kB/s)
Выбор ранее не выбранного пакета splint-data.
(Чтение базы данных … на данный момент установлено 370947 файлов и каталогов.)
Подготовка к распаковке …/splint-data_1%3a3.1.2+dfsg-1build1_all.deb …
Распаковывается splint-data (1:3.1.2+dfsg-1build1) …
Выбор ранее не выбранного пакета splint.
Подготовка к распаковке …/splint_1%3a3.1.2+dfsg-1build1_amd64.deb …
Распаковывается splint (1:3.1.2+dfsg-1build1) …
Настраивается пакет splint-data (1:3.1.2+dfsg-1build1) …
Настраивается пакет splint (1:3.1.2+dfsg-1build1) …
Обрабатываются триггеры для man-db (2.9.1-1) …
makim@makim:~/work/os/lab_prog$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2018

calculate.h:4:37: 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:7:37: Function parameter Operation declared as manifest array (size
                     constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:13:3: 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:19:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:25:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:6: 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:35:10: Return value type double does not match declared type float:
                      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:43:3: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:9: Return value type double does not match declared type float:
                     (pow(Numeral, SecondNumeral))
calculate.c:47:9: Return value type double does not match declared type float:
                     (sqrt(Numeral))
calculate.c:49:9: Return value type double does not match declared type float:
                     (sin(Numeral))
calculate.c:51:9: Return value type double does not match declared type float:
                     (cos(Numeral))
calculate.c:53:9: Return value type double does not match declared type float:
                     (tan(Numeral))
calculate.c:57:9: Return value type double does not match declared type float:
                     (HUGE_VAL)

Finished checking --- 15 code warnings
```

```c
float Calculate(float Numeral, char Operation[4])
{
        float SecondNumeral;
        if(strncmp(Operation, "+", 1) == 0)
        {
                printf("Второе слагаемое: ");
                scanf("%f",&SecondNumeral);
                return(Numeral + SecondNumeral);
        }
        else if(strncmp(Operation, "-", 1) == 0)
        {
                printf("Вычитаемое: ");
                scanf("%f",&SecondNumeral);
                return(Numeral - SecondNumeral);
        }
        else if(strncmp(Operation, "*", 1) == 0)
        {
                printf("Множитель: ");
                scanf("%f",&SecondNumeral);
                return(Numeral * SecondNumeral);
        }
        else if(strncmp(Operation, "/", 1) == 0)
        {
                printf("Делитель: ");
                scanf("%f",&SecondNumeral);
                if(SecondNumeral == 0)
                {
                        printf("Ошибка: деление на ноль! ");
                        return(HUGE_VAL);
                }
                else
                        return(Numeral / SecondNumeral);
        }
        else if(strncmp(Operation, "pow", 3) == 0)
        {
                printf("Степень: ");
                scanf("%f",&SecondNumeral);
                return(pow(Numeral, SecondNumeral));
        }
        else if(strncmp(Operation, "sqrt", 4) == 0)
                return(sqrt(Numeral));
        else if(strncmp(Operation, "sin", 3) == 0)
                return(sin(Numeral));
        else if(strncmp(Operation, "cos", 3) == 0)
                return(cos(Numeral));
        else if(strncmp(Operation, "tan", 3) == 0)
                return(tan(Numeral));
        else
        {
                printf("Неправильно введено действие ");
                return(HUGE_VAL);
        }
}
```

Рис. 2.13: Предупреждения splint (main.c)



```
makim@makim:~/work/os/lab_prog$ splint main.c
Splint 3.1.2 --- 20 Feb 2018

calculate.h:4:37: 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:11:2: 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:13:2: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 3 code warnings
makim@makim:~/work/os/lab_prog$ 
```

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

Рис. 2.14: Предупреждения splint

3. Итог

Мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux. Закрепили знания, полученные в прошлых работах. Создали на языке программирования C калькулятор с простейшими функциями и разобрали на нем основные навыки отладки.