Лабораторная работа №14

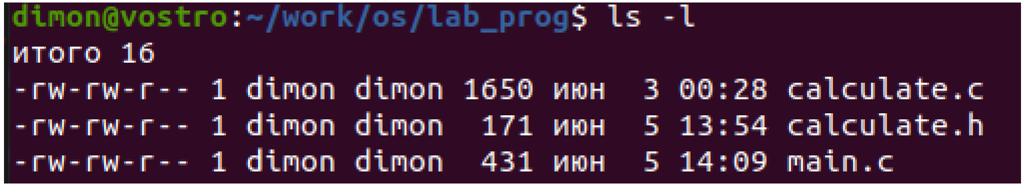
Студент: Евсельев Д.

Группа: НПМбд-01-20

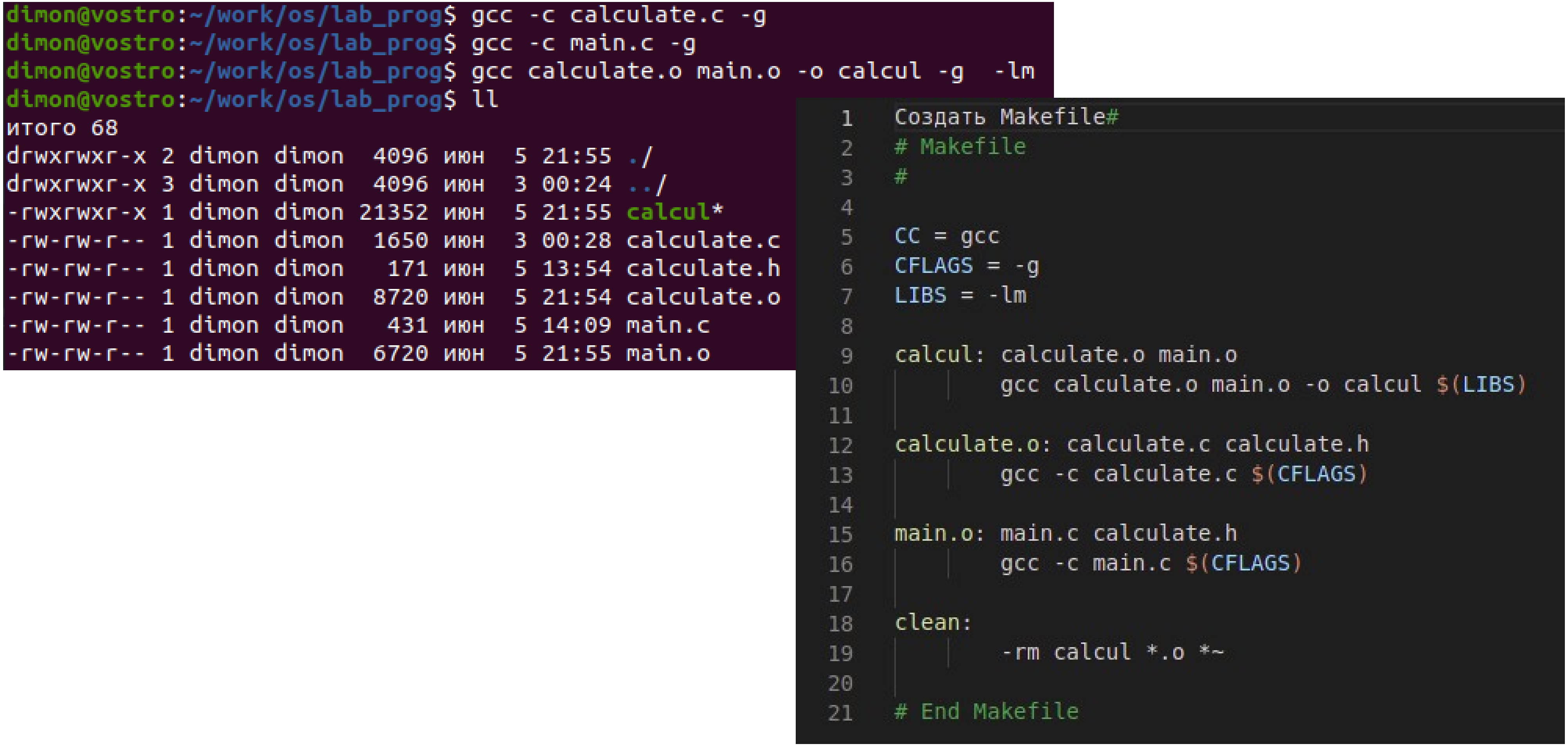
Преподаватель: Курячиий Г.

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.

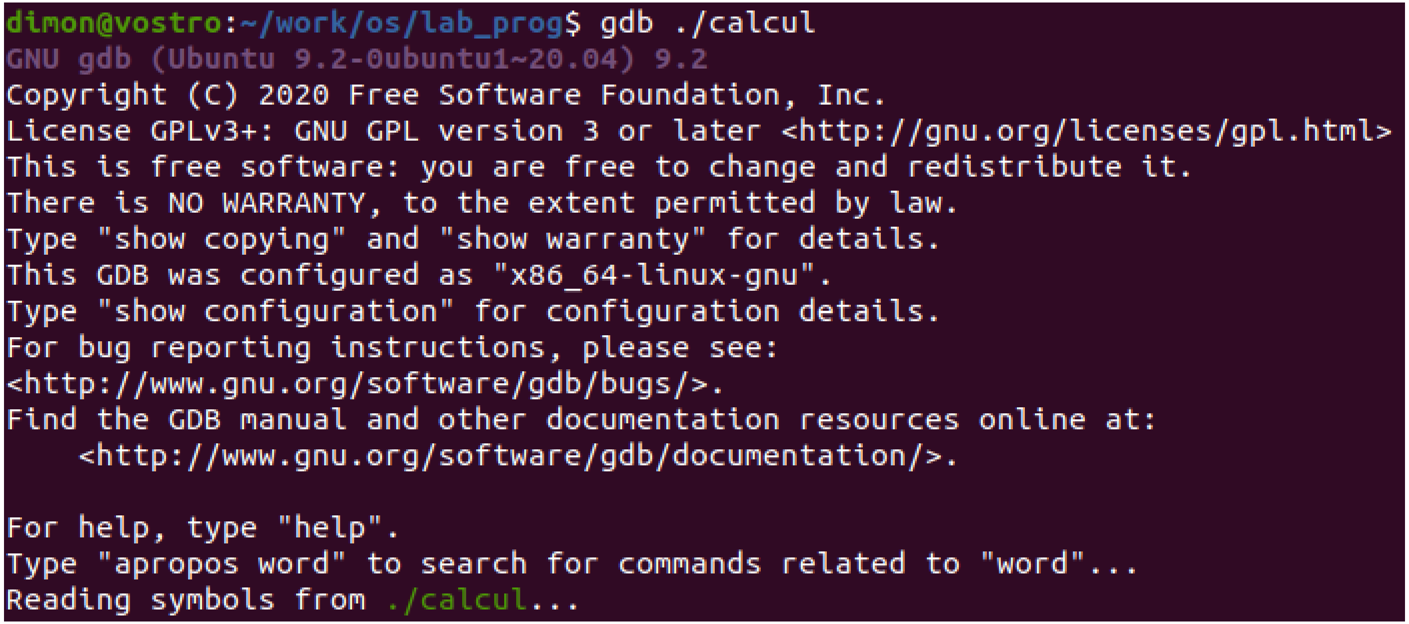


```
dimon@vostro:~/work/os/lab_prog$ ls -l
итого 16
-rw-rw-r-- 1 dimon dimon 1650 июн  3 00:28 calculate.c
-rw-rw-r-- 1 dimon dimon  171 июн  5 13:54 calculate.h
-rw-rw-r-- 1 dimon dimon  431 июн  5 14:09 main.c
```

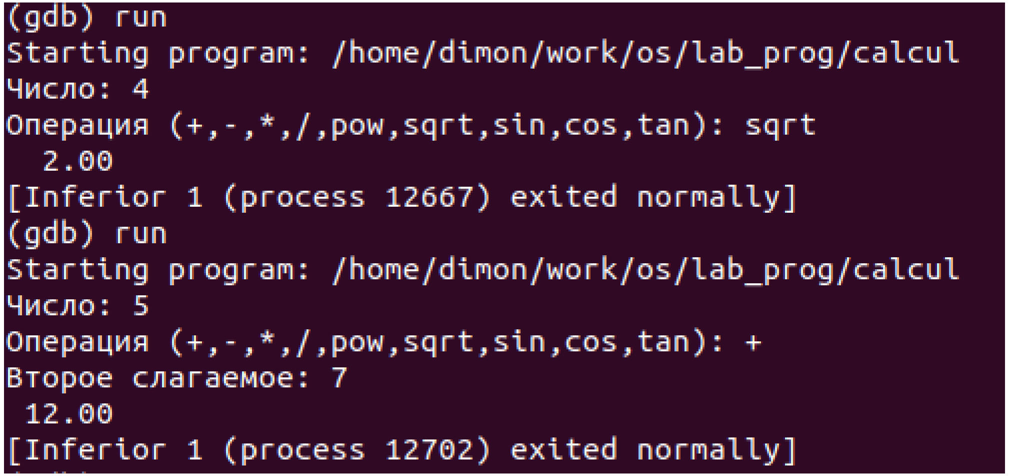


```
dimon@vostro:~/work/os/lab_prog$ gcc -c calculate.c -g
dimon@vostro:~/work/os/lab_prog$ gcc -c main.c -g
dimon@vostro:~/work/os/lab_prog$ gcc calculate.o main.o -o calcul -g  -lm
dimon@vostro:~/work/os/lab_prog$ ll
итого 68
drwxrwxr-x 2 dimon dimon  4096 июн  5 21:55 ./
drwxrwxr-x 3 dimon dimon  4096 июн  3 00:24 ../
-rwxrwxr-x 1 dimon dimon 21352 июн  5 21:55 calcul*
-rw-rw-r-- 1 dimon dimon  1650 июн  3 00:28 calculate.c
-rw-rw-r-- 1 dimon dimon   171 июн  5 13:54 calculate.h
-rw-rw-r-- 1 dimon dimon  8720 июн  5 21:54 calculate.o
-rw-rw-r-- 1 dimon dimon   431 июн  5 14:09 main.c
-rw-rw-r-- 1 dimon dimon  6720 июн  5 21:55 main.o
```

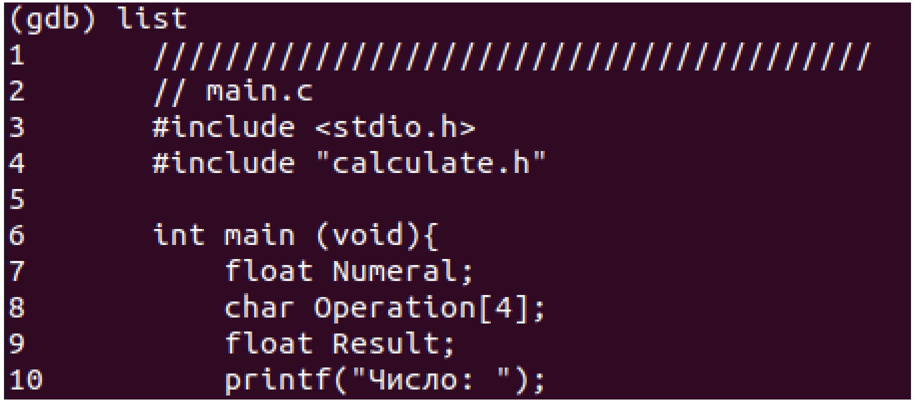
```
Создать Makefile#
# Makefile
#

CC = gcc
CFLAGS = -g
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 $(CFLAGS)

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

clean:
	-rm calcul *.o *~

# End Makefile
```



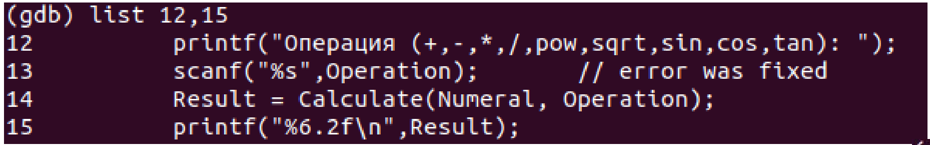
```
dimon@vostro:~/work/os/lab_prog$ gdb ./calcul
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, 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...
```



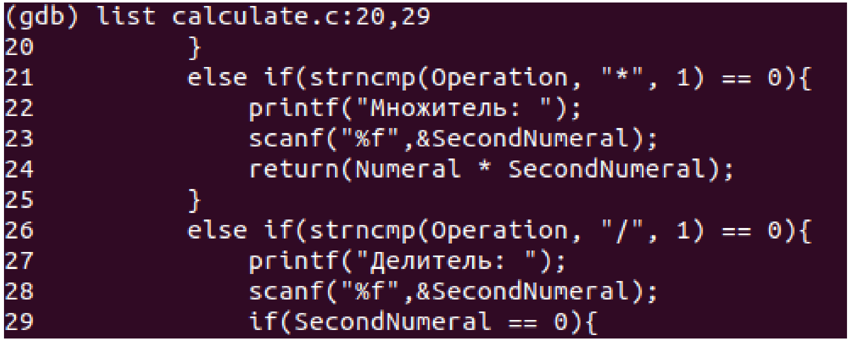
```
(gdb) run
Starting program: /home/dimon/work/os/lab_prog/calcul
Число: 4
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): sqrt
  2.00
[Inferior 1 (process 12667) exited normally]
(gdb) run
Starting program: /home/dimon/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 7
 12.00
[Inferior 1 (process 12702) exited normally]
```



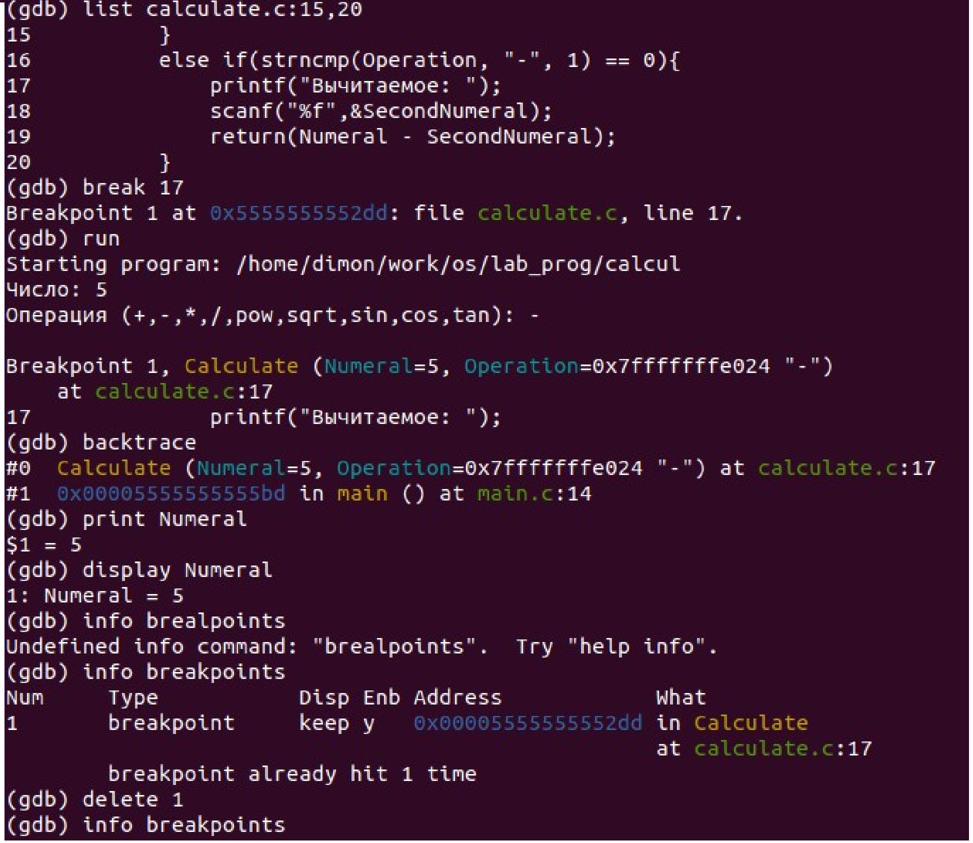
```
(gdb) list
        ////////////////////////////////////////
        // main.c
        #include <stdio.h>
        #include "calculate.h"

        int main (void){
            float Numeral;
            char Operation[4];
            float Result;
            printf("Число: ");
```



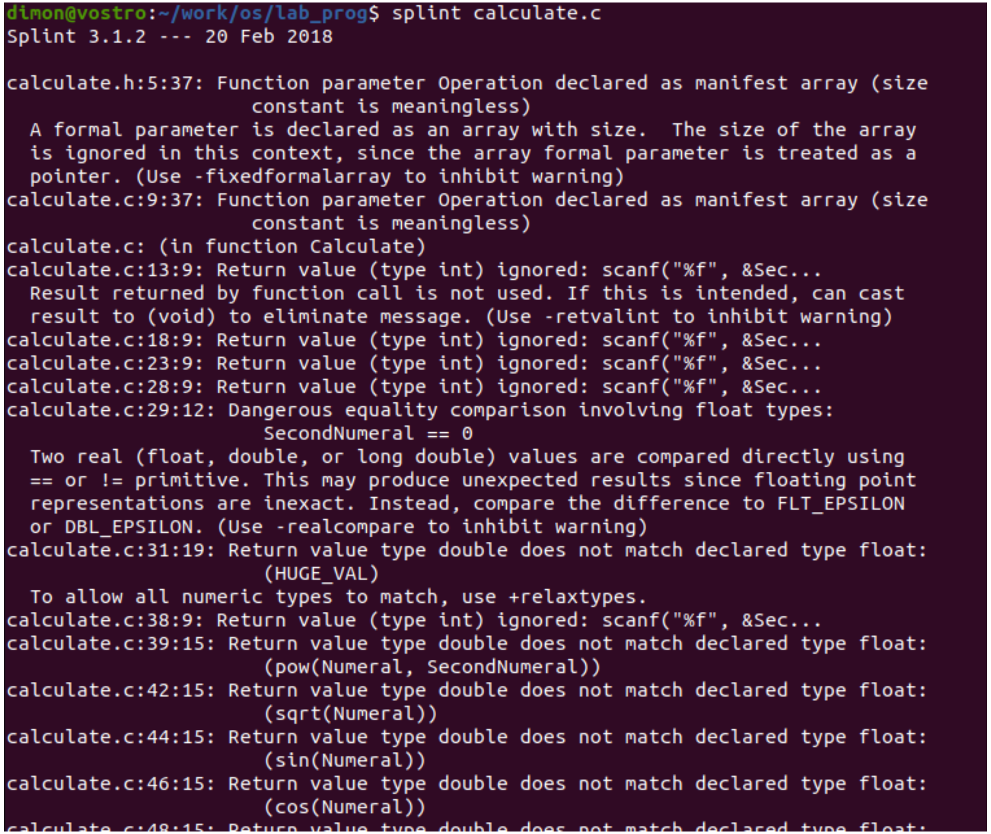
```
(gdb) list 12,15
12          printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
13          scanf("%s",Operation);       // error was fixed
14          Result = Calculate(Numeral, Operation);
15          printf("%6.2f\n",Result);
```



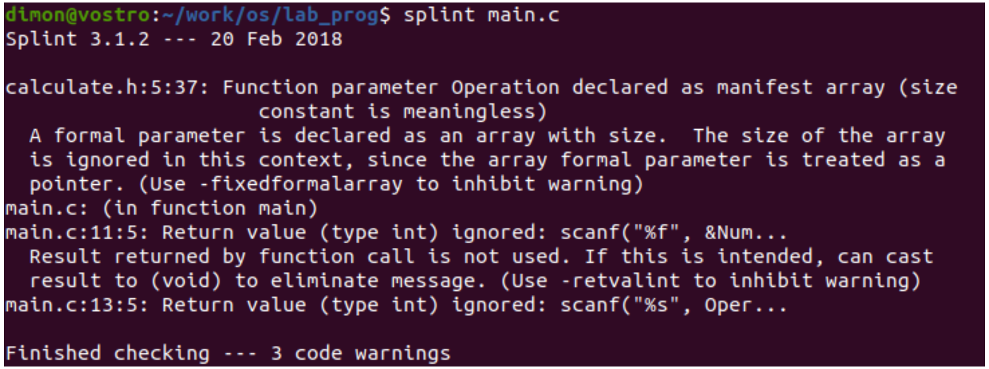
```
(gdb) list calculate.c:20,29
20            }
21            else if(strncmp(Operation, "*", 1) == 0){
22                printf("Множитель: ");
23                scanf("%f",&SecondNumeral);
24                return(Numeral * SecondNumeral);
25            }
26            else if(strncmp(Operation, "/", 1) == 0){
27                printf("Делитель: ");
28                scanf("%f",&SecondNumeral);
29                if(SecondNumeral == 0){
```



```
(gdb) list calculate.c:15,20
15            }
16            else if(strncmp(Operation, "-", 1) == 0){
17                printf("Вычитаемое: ");
18                scanf("%f",&SecondNumeral);
19                return(Numeral - SecondNumeral);
20            }
(gdb) break 17
Breakpoint 1 at 0x5555555552dd: file calculate.c, line 17.
(gdb) run
Starting program: /home/dimon/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffe024 "-")
    at calculate.c:17
17                printf("Вычитаемое: ");
(gdb) backtrace
#0  Calculate (Numeral=5, Operation=0x7fffffffe024 "-") at calculate.c:17
#1  0x00005555555553bd in main () at main.c:14
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) info brealpoints
Undefined info command: "brealpoints".  Try "help info".
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00005555555552dd in Calculate
                                                   at calculate.c:17
        breakpoint already hit 1 time
(gdb) delete 1
(gdb) info breakpoints
```



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

calculate.h:5:37: 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:9:37: Function parameter Operation declared as manifest array (size
                     constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:13:9: 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:18:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:23:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:29:12: 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:31:19: Return value type double does not match declared type float:
                      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:38:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:39:15: Return value type double does not match declared type float:
                      (pow(Numeral, SecondNumeral))
calculate.c:42:15: Return value type double does not match declared type float:
                      (sqrt(Numeral))
calculate.c:44:15: Return value type double does not match declared type float:
                      (sin(Numeral))
calculate.c:46:15: Return value type double does not match declared type float:
                      (cos(Numeral))
calculate.c:48:15: Return value type double does not match declared type float:
```



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

calculate.h:5:37: 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:11:5: 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:13:5: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 3 code warnings
```

Вывод

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