РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №13

дисциплина: Операционные системы

Студент: Хайдари Ахмад Насир Группа: НБИбд-01-20

МОСКВА

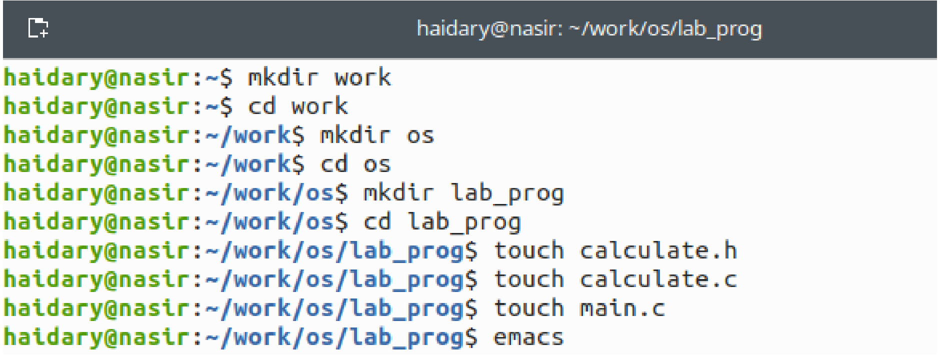
2021 г

1. Цель работы:

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

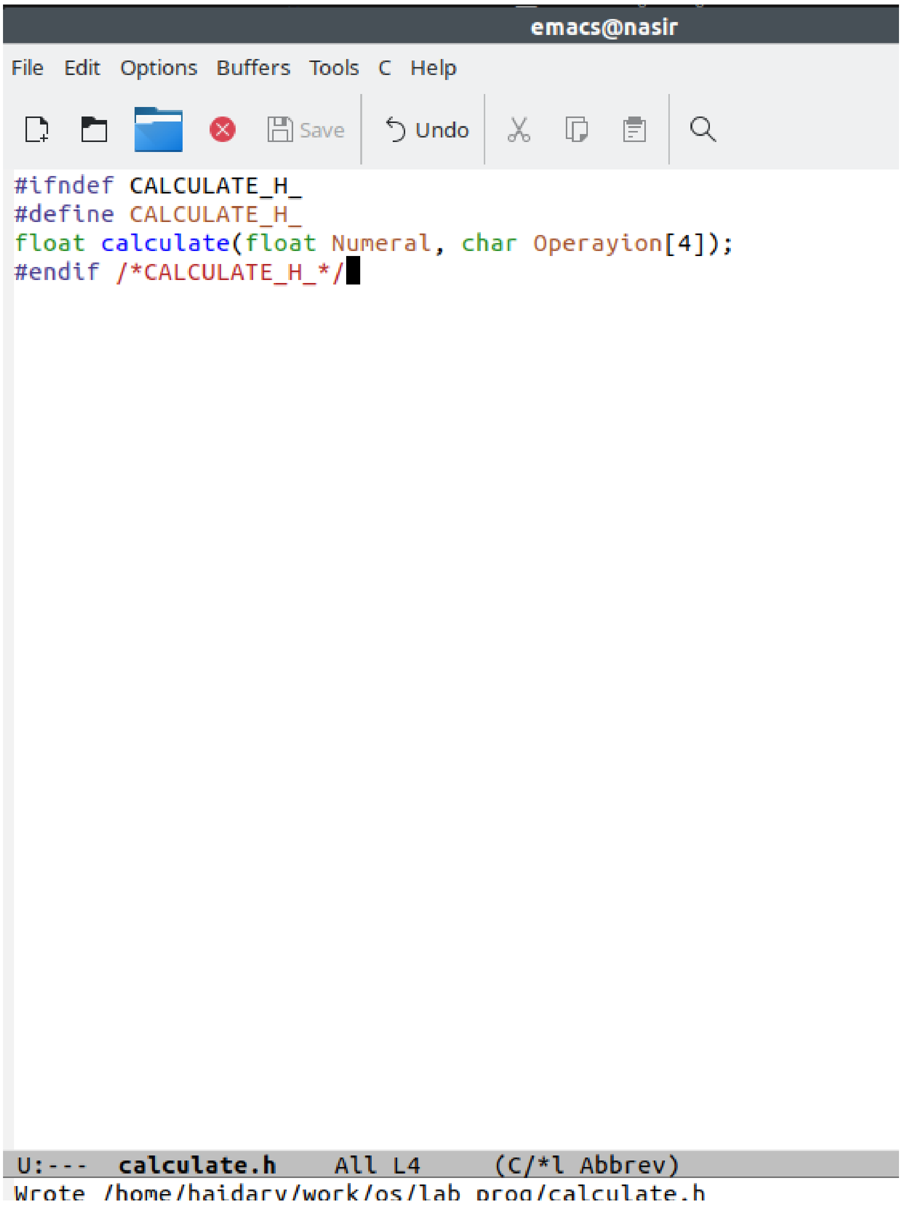
2. Описание процесса выполнения задания.

- В домашнем каталоге создан подкаталог ~/work/os/lab_prog. - Созданы файлы: calculate.h, calculate.c, main.c. (см. рис. 1-4).



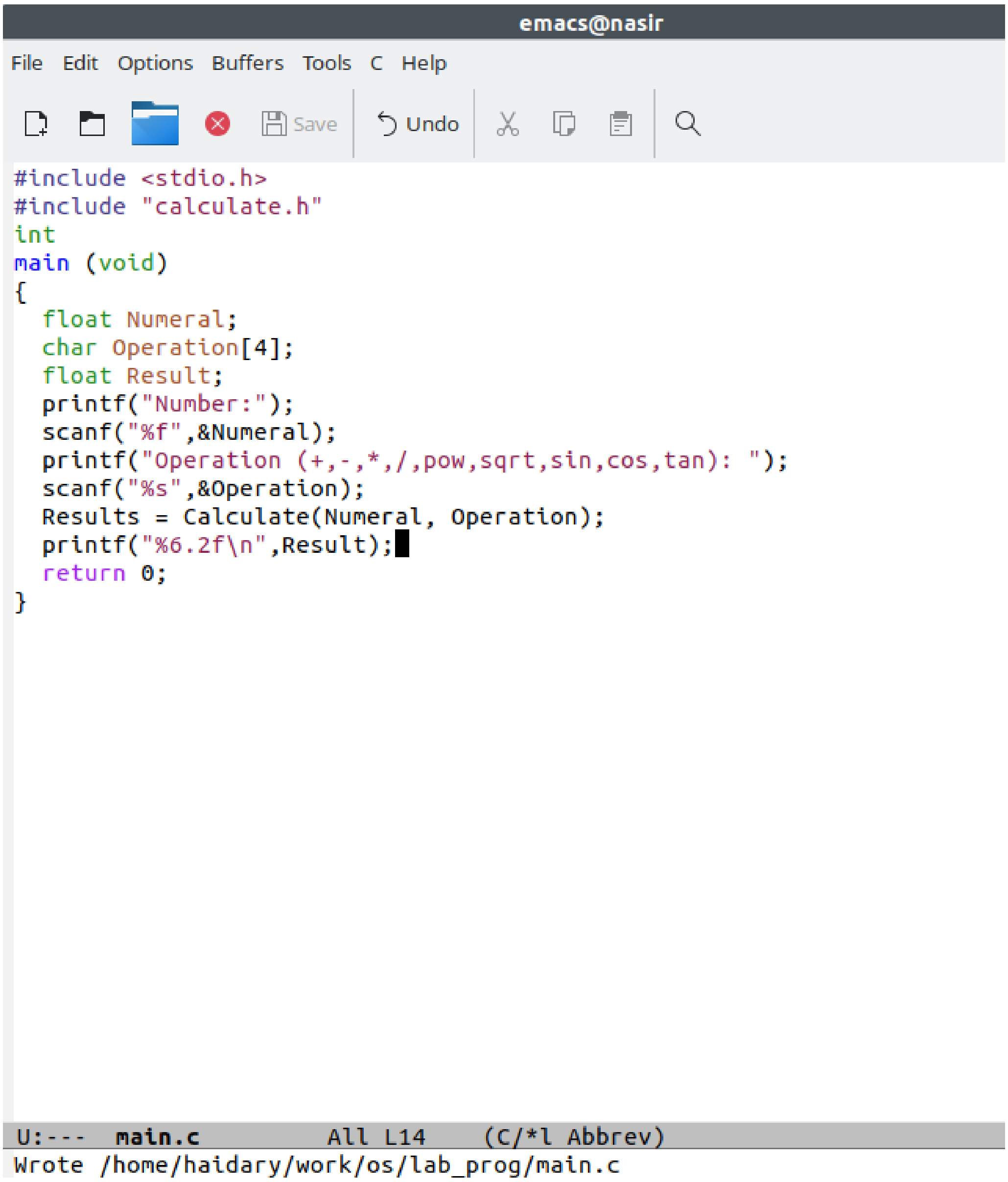
haidary@nasir: ~/work/os/lab_prog

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



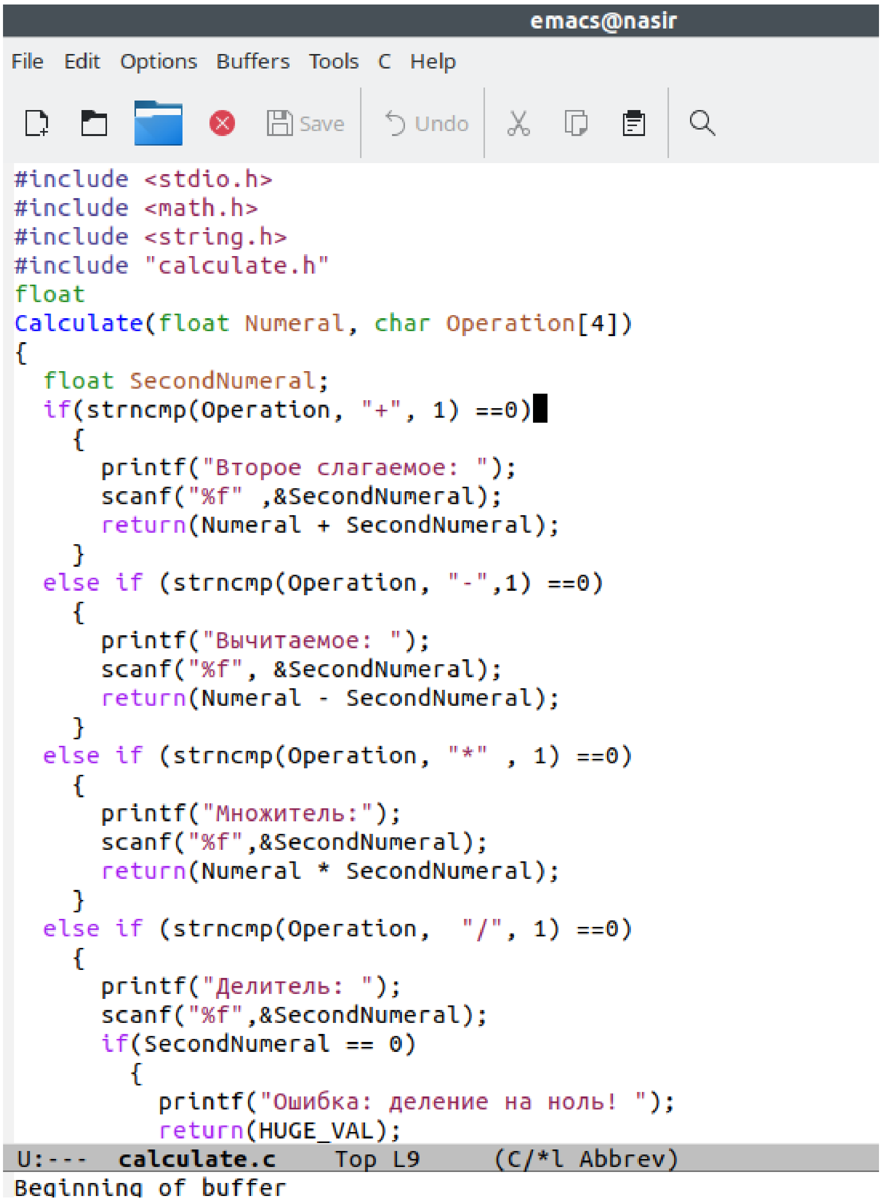
```
#ifndef CALCULATE_H_
#define CALCULATE_H_
float calculate(float Numeral, char Operayion[4]);
#endif /*CALCULATE_H_*/
```

```
U:---  calculate.h    All L4     (C/*l Abbrev)
Wrote /home/haidary/work/os/lab_prog/calculate.h
```



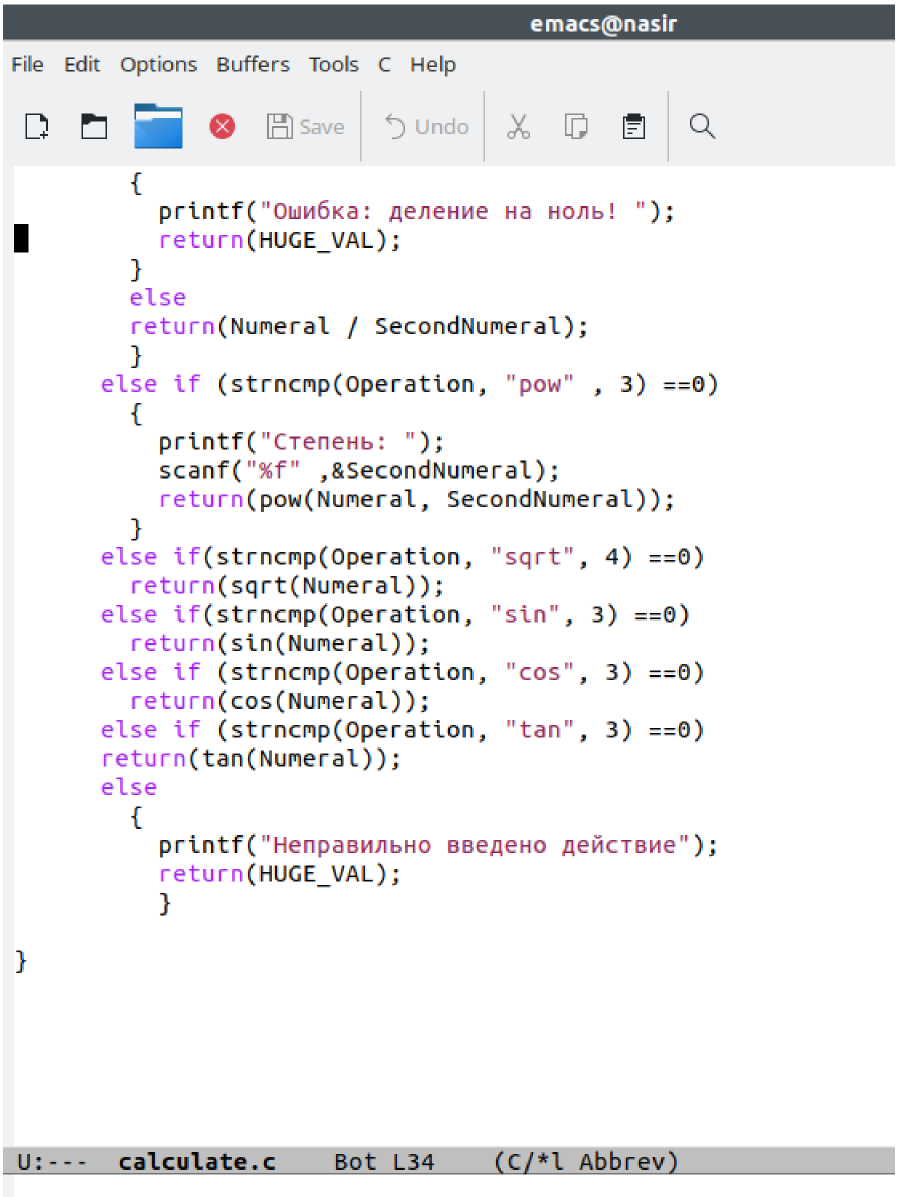
```
#include <stdio.h>
#include "calculate.h"
int
main (void)
{
  float Numeral;
  char Operation[4];
  float Result;
  printf("Number:");
  scanf("%f",&Numeral);
  printf("Operation (+,-,*,/,pow,sqrt,sin,cos,tan): ");
  scanf("%s",&Operation);
  Results = Calculate(Numeral, Operation);
  printf("%6.2f\n",Result);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

```
U:---  main.c        All L14    (C/*l Abbrev)
Wrote /home/haidary/work/os/lab_prog/main.c
```



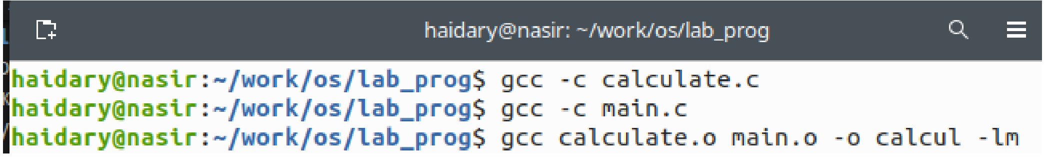
```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include "calculate.h"
float
Calculate(float Numeral, char Operation[4])
{
  float SecondNumeral;
  if(strncmp(Operation, "+", 1) ==0)
    {
      printf("Второе слагаемое: ");
      scanf("%f" ,&SecondNumeral);
      return(Numeral + SecondNumeral);
    }
  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);
```

```
U:---  calculate.c     Top L9      (C/*l Abbrev)
Beginning of buffer
```



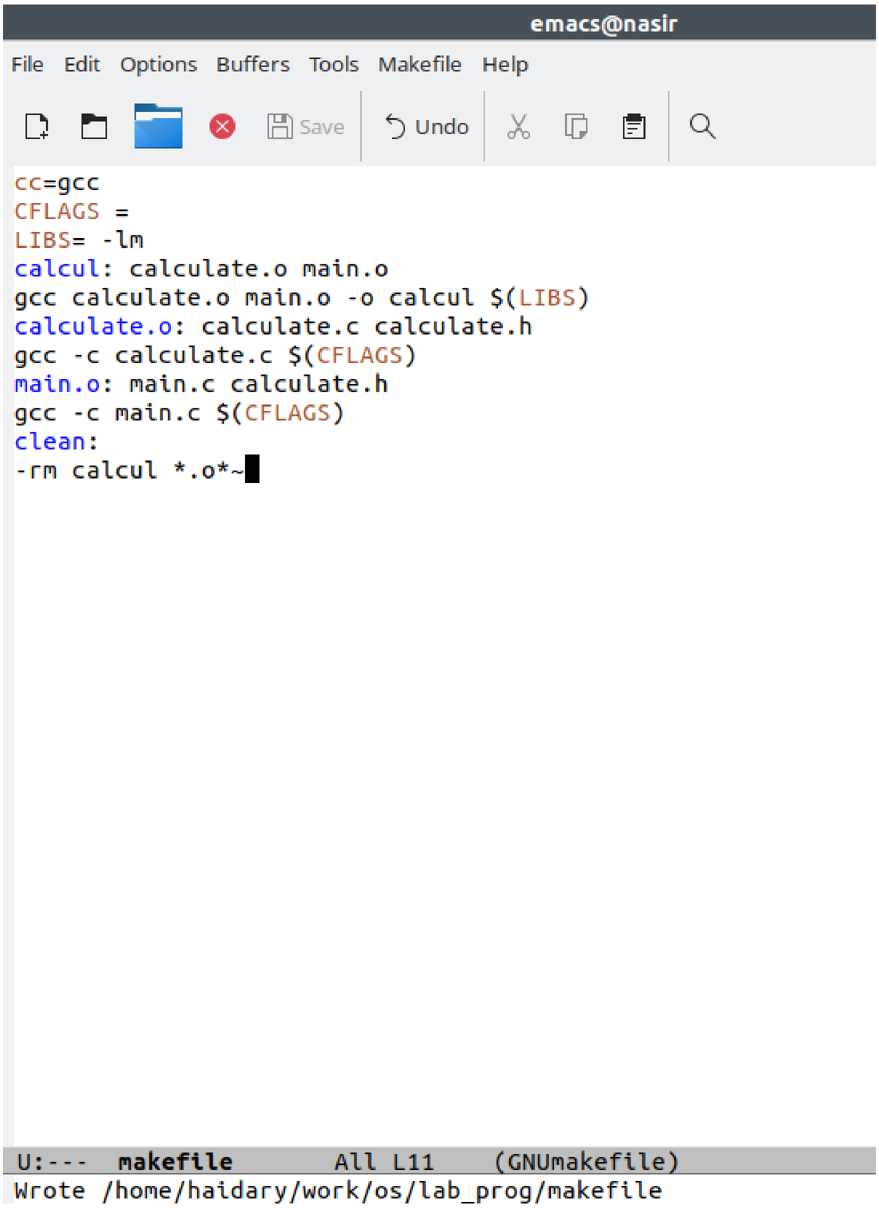
```
        {
          printf("Ошибка: деление на ноль! ");
          return(HUGE_VAL);
        }
        else
        return(Numeral / SecondNumeral);
        }
      else if (strncmp(Operation, "pow" , 3) ==0)
        {
          printf("Степень: ");
          scanf("%f" ,&SecondNumeral);
          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);
          }

}
```

1. Выполнена компиляция программы прсредством gcc. (см. рис.



```
haidary@nasir:~/work/os/lab_prog$ gcc -c calculate.c
haidary@nasir:~/work/os/lab_prog$ gcc -c main.c
haidary@nasir:~/work/os/lab_prog$ gcc calculate.o main.o -o calcul -lm
```

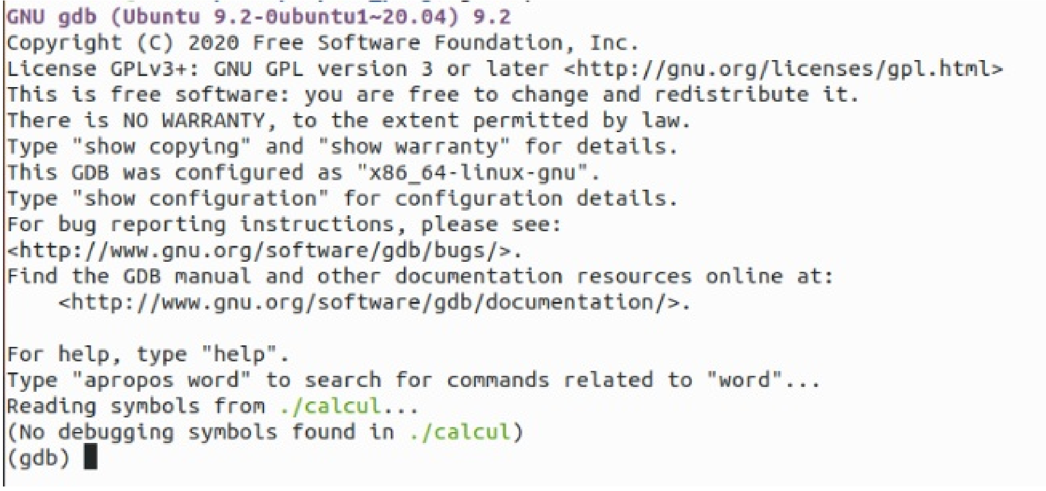
4. При необходимости исправлены синтаксические ошибки.
5. Создан Makefile. (см. рис. 6).



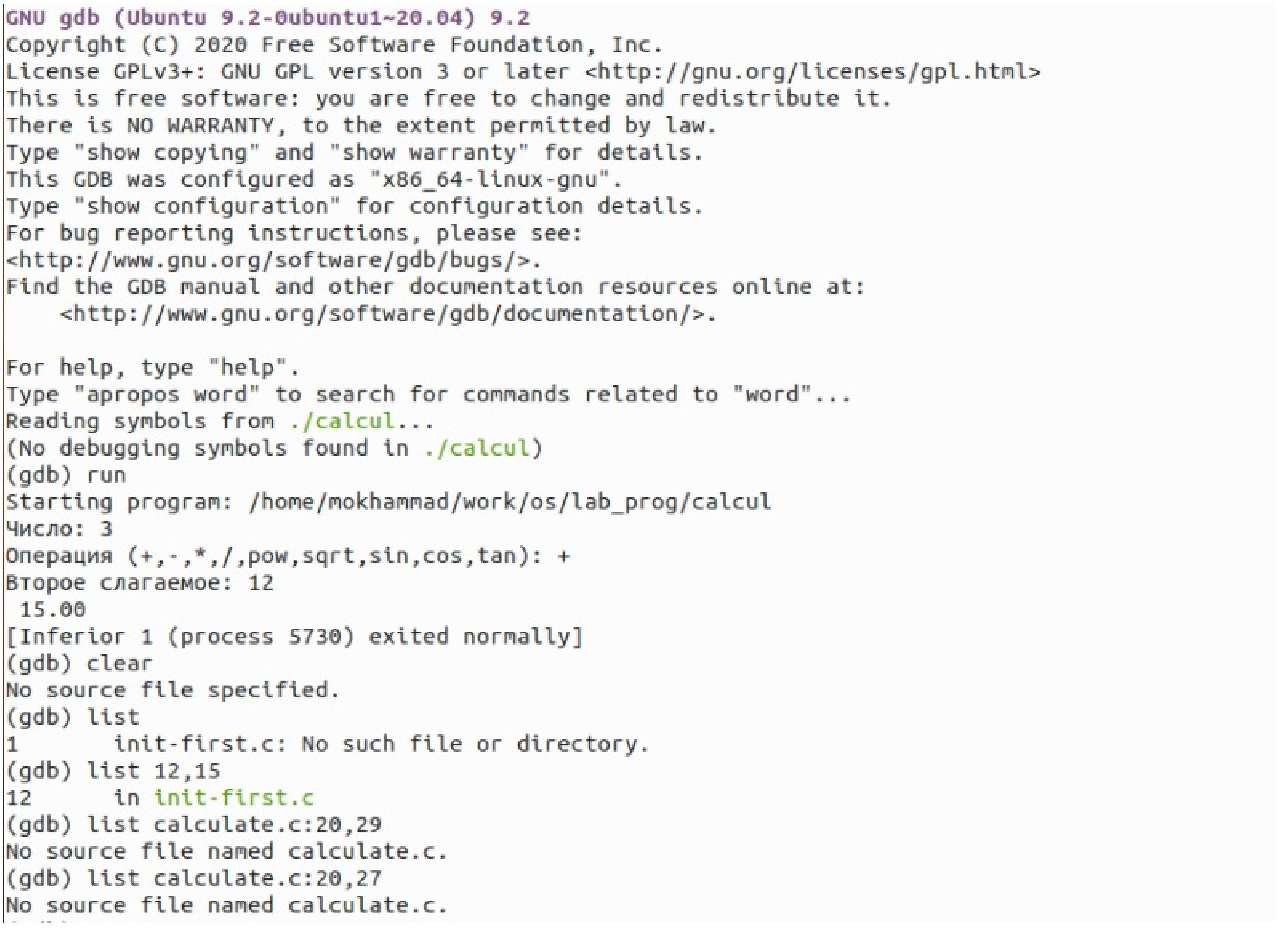
```
cc=gcc
CFLAGS =
LIBS= -lm
calcul: calculate.o main.o
gcc calculate.o main.o -o calcul $(LIBS)
calculate.o: calculate.c calculate.h
gcc -c calculate.c $(CFLAGS)
main.o: main.c calculate.h
gcc -c main.c $(CFLAGS)
clean:
-rm calcul *.o*~
```

```
U:---  makefile        All L11     (GNUmakefile)
Wrote /home/haidary/work/os/lab_prog/makefile
```

6. 6 С помощью gdb выполнена отладка программы calcul. (см. рис. 7, 8).

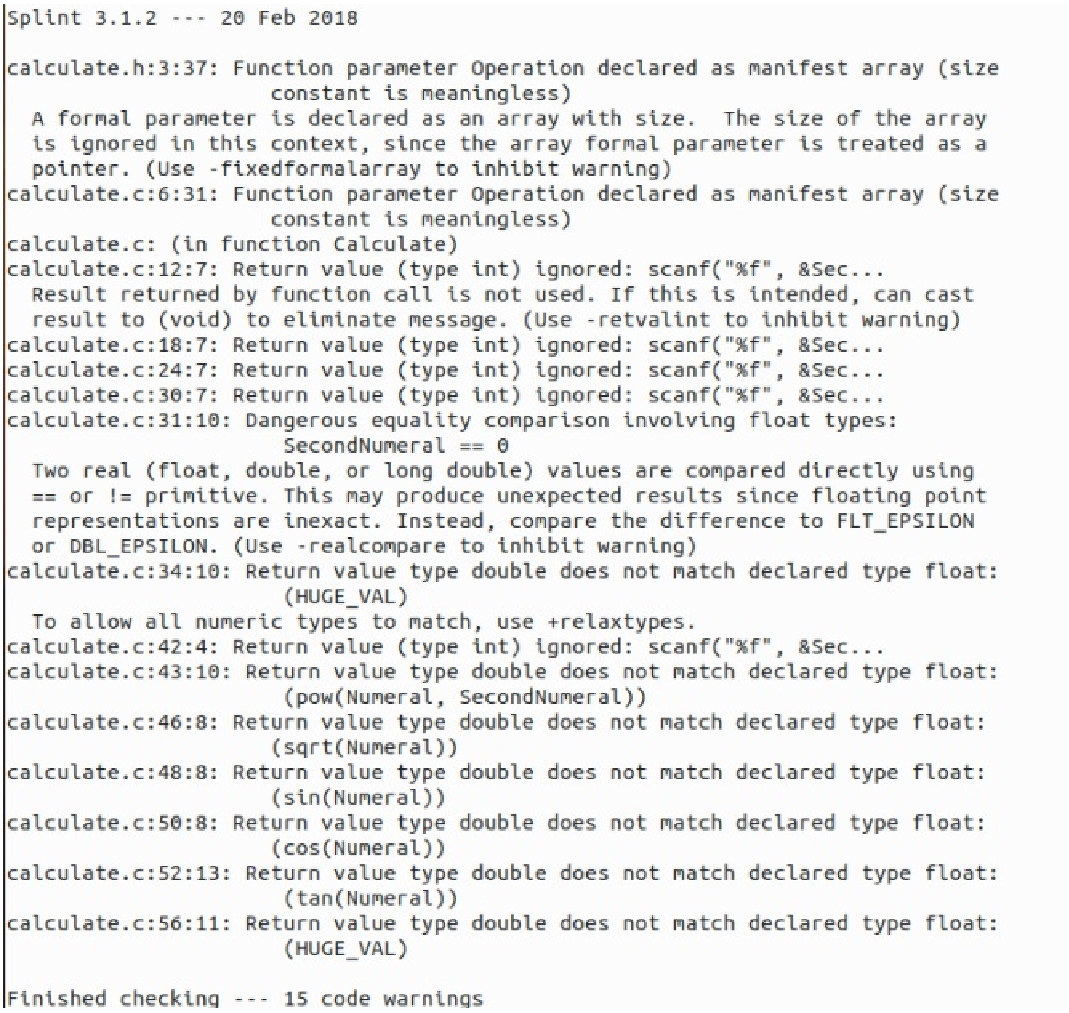


```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show 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...
(No debugging symbols found in ./calcul)
(gdb) 
```



```
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...
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/mokhammad/work/os/lab_prog/calcul
Число: 3
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 12
 15.00
[Inferior 1 (process 5730) exited normally]
(gdb) clear
No source file specified.
(gdb) list
1       init-first.c: No such file or directory.
(gdb) list 12,15
12      in init-first.c
(gdb) list calculate.c:20,29
No source file named calculate.c.
(gdb) list calculate.c:20,27
No source file named calculate.c.
```

7. 6. С помощью утилиты splint проанализированы коды файлов calculate.c и main.c. (см. рис. 9).



```
Splint 3.1.2 --- 20 Feb 2018

calculate.h:3:37: 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:6:31: Function parameter Operation declared as manifest array (size
                     constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:12:7: 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:18:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:24:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:30:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:10: 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:34:10: Return value type double does not match declared type float:
                      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:42:4: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:43:10: Return value type double does not match declared type float:
                      (pow(Numeral, SecondNumeral))
calculate.c:46:8: Return value type double does not match declared type float:
                     (sqrt(Numeral))
calculate.c:48:8: Return value type double does not match declared type float:
                     (sin(Numeral))
calculate.c:50:8: Return value type double does not match declared type float:
                     (cos(Numeral))
calculate.c:52:13: Return value type double does not match declared type float:
                      (tan(Numeral))
calculate.c:56:11: Return value type double does not match declared type float:
                      (HUGE_VAL)

Finished checking --- 15 code warnings
```

Вывод:

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