Отчет по лабораторной работе №14

Пузырев Владислав Максимович

04.06.2021

Россия, Москва

Цель работы

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.

Задание

Задание

Выполнить следующие пункты:

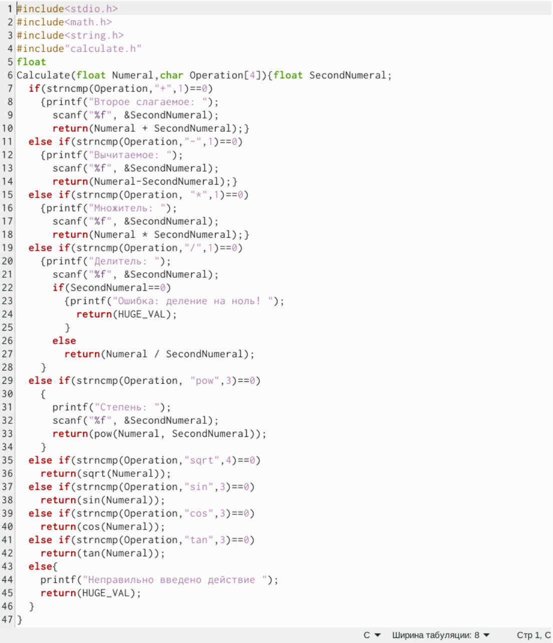
1. В домашнем каталоге создайте подкаталог~/Lab_OS/lab14/lab_prog.
2. Создайте в нём файлы:calculate.h,calculate.c,main.c.
3. Выполните компиляцию программы посредством gcc: > gcc -c calculate.c > gcc -c main.c > gcc calculate.o main.o -o calcul -lm
4. Создайте Makefile со следующим содержанием.
5. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile)
6. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

Выполнение лабораторной работы

Выполнение лабораторной работы

Перед выполнением лабораторной работы я хорошо ознакомился с теоритическим материалом для её выполнения

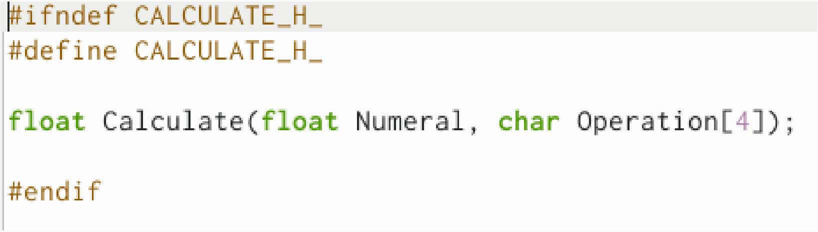
Слайд 1



```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);
        }
      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);
  }
}
```

C ▾ Ширина табуляции: 8 ▾ Стр 1, С

Слайд 2



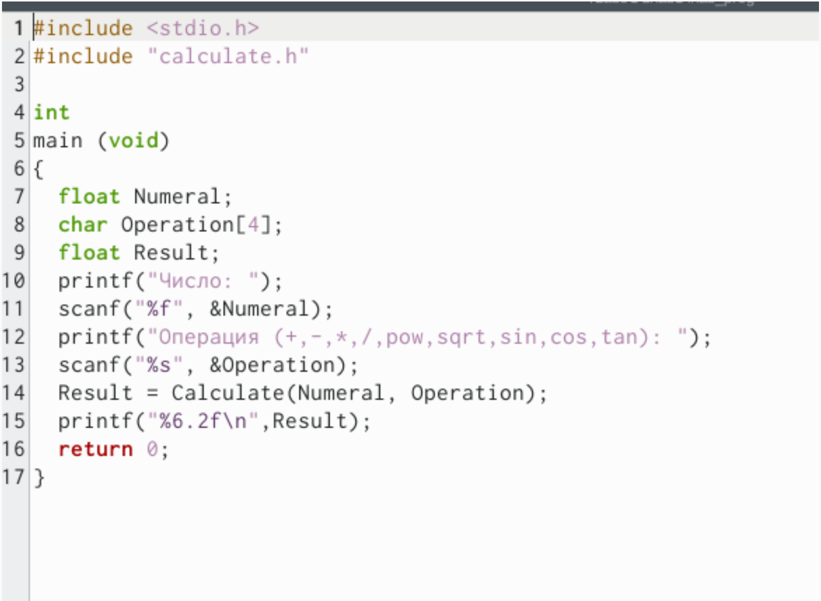
```c
#ifndef CALCULATE_H_
#define CALCULATE_H_

float Calculate(float Numeral, char Operation[4]);

#endif
```

Figure 2: Файл calculate.h

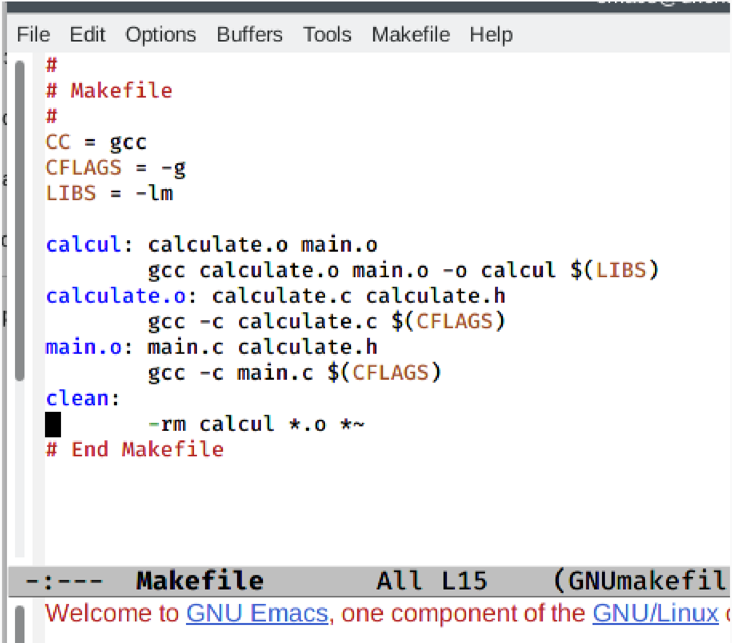
Слайд 3



```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;
}
```

Слайд 4



File Edit Options Buffers Tools Makefile Help

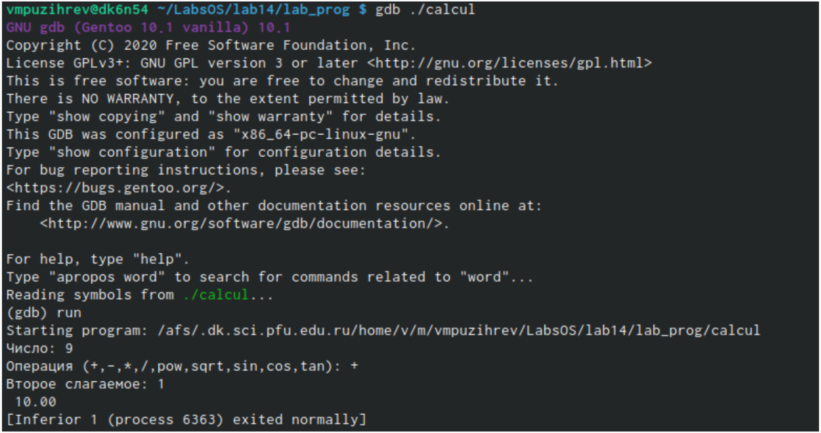
```
#
# Makefile
#
CC = gcc
CFLAGS = -g
LIBS = -lm

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
```

-:--- Makefile All L15 (GNUmakefil

Welcome to GNU Emacs, one component of the GNU/Linux

Слайд 5

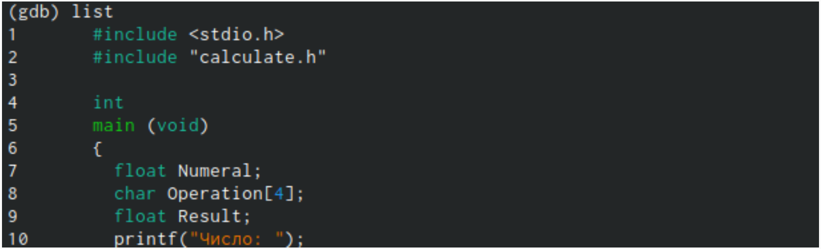


```
vmpuzihrev@dk6n54 ~/LabsOS/lab14/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 10.1 vanilla) 10.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-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"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/v/m/vmpuzihrev/LabsOS/lab14/lab_prog/calcul
Число: 9
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 1
 10.00
[Inferior 1 (process 6363) exited normally]
```

Figure 5: Отладку программы calcul

Слайд 6

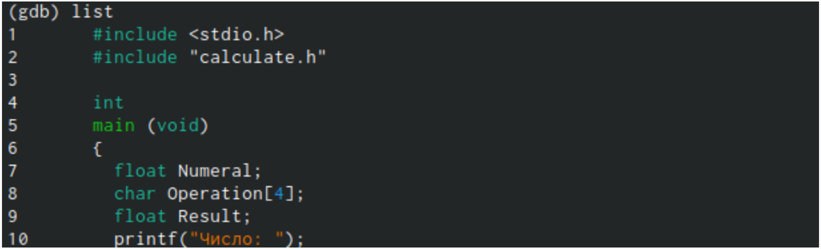


```
(gdb) list
        #include <stdio.h>
        #include "calculate.h"

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

Figure 6: Просмотр кода

Слайд 7

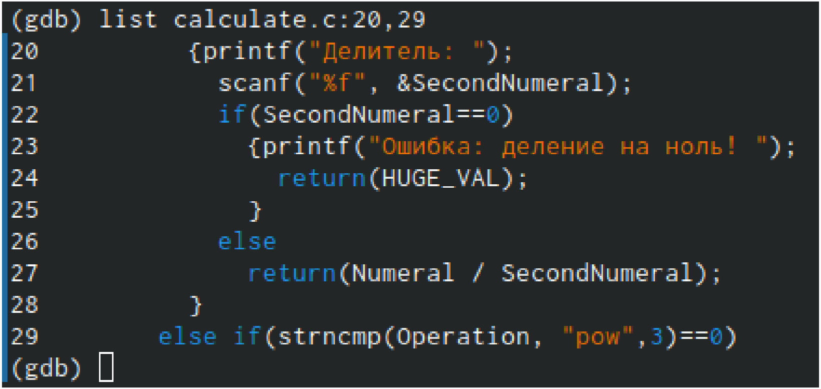


```
(gdb) list
        #include <stdio.h>
        #include "calculate.h"

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

Figure 7: Просмотр кода

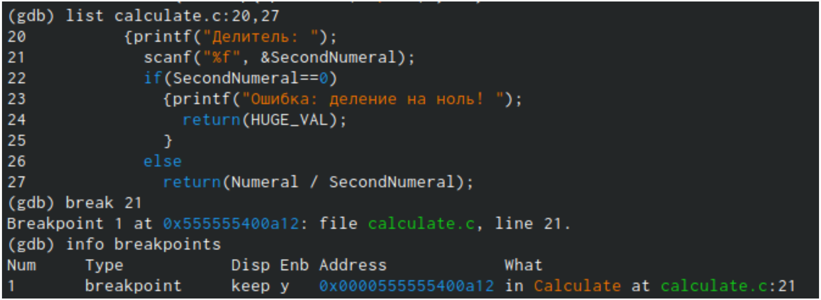
Слайд 8



```
(gdb) list calculate.c:20,29
20            {printf("Делитель: ");
21              scanf("%f", &SecondNumeral);
22              if(SecondNumeral==0)
23                {printf("Ошибка: деление на ноль! ");
24                  return(HUGE_VAL);
25                }
26              else
27                return(Numeral / SecondNumeral);
28            }
29          else if(strncmp(Operation, "pow",3)==0)
(gdb) 
```

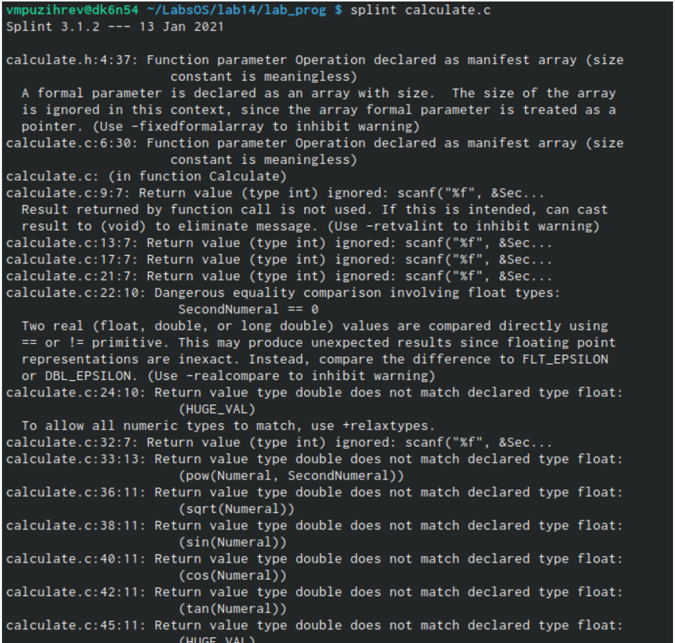
Figure 8: Установка и удаление точки останова

Слайд 9



```
(gdb) list calculate.c:20,27
20	        {printf("Делитель: ");
21	          scanf("%f", &SecondNumeral);
22	          if(SecondNumeral==0)
23	            {printf("Ошибка: деление на ноль! ");
24	              return(HUGE_VAL);
25	            }
26	          else
27	            return(Numeral / SecondNumeral);
(gdb) break 21
Breakpoint 1 at 0x555555400a12: file calculate.c, line 21.
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555400a12 in Calculate at calculate.c:21
```

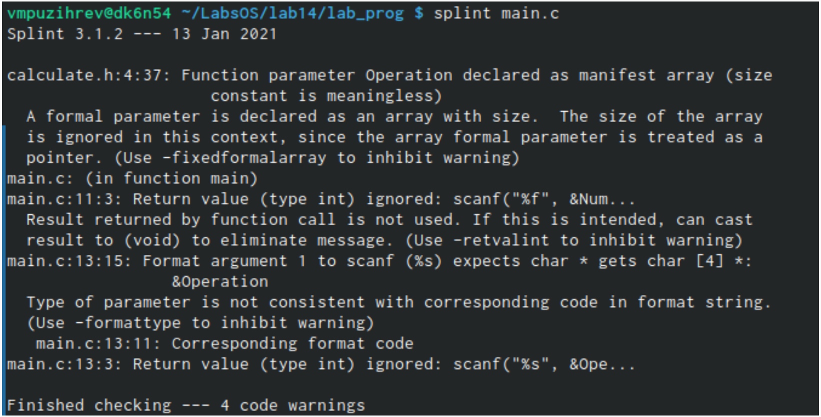
Слайд 10



```
vmpuzihrev@dk6n54 ~/LabsOS/lab14/lab_prog $ splint calculate.c
Splint 3.1.2 --- 13 Jan 2021

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:6:30: Function parameter Operation declared as manifest array (size
                     constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:9:7: 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:13:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:17:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:21:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:22:10: 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:24:10: Return value type double does not match declared type float:
                      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:32:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:33:13: Return value type double does not match declared type float:
                      (pow(Numeral, SecondNumeral))
calculate.c:36:11: Return value type double does not match declared type float:
                      (sqrt(Numeral))
calculate.c:38:11: Return value type double does not match declared type float:
                      (sin(Numeral))
calculate.c:40:11: Return value type double does not match declared type float:
                      (cos(Numeral))
calculate.c:42:11: Return value type double does not match declared type float:
                      (tan(Numeral))
calculate.c:45:11: Return value type double does not match declared type float:
                      (HUGE_VAL)
```

Слайд 11



```
vmpuzihrev@dk6n54 ~/LabsOS/lab14/lab_prog $ splint main.c
Splint 3.1.2 --- 13 Jan 2021

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:11:3: 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:13:15: 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:13:11: Corresponding format code
main.c:13:3: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
```

Figure 10: Код файла main.c