Презентация Лабораторной работы №14

Лабораторная работа 14

Новосельцев.Д.С. НФИбд-02-20

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.

Ход работы:

```
danila@dsnovoseljcev-VirtualBox:~/work/os/lab_p
rog$ touch calculate.h calculate.c main.c
danila@dsnovoseljcev-VirtualBox:~/work/os/lab_p
rog$
```

```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("Second term: ");
      scanf("%f", &SecondNumeral);
      return(Numeral + SecondNumeral);
    }
  else if(strncmp(Operation, "-", 1) == 0)
    {
      printf("Subtrahend: ");
      scanf("%f", &SecondNumeral);
      return(Numeral - SecondNumeral);
    }
  else if(strncmp(Operation, "*", 1) == 0)
```

U:--- **calculate.c**   Top L21   (C/*l Abbrev)

Beginning of buffer

```
    {
      printf("Factor: ");
      scanf("%f", &SecondNumeral);
      return(Numeral * SecondNumeral);
    }
  else if(strncmp(Operation, "/", 1) == 0)
    {
      printf("Divisor: ");
      scanf("%f", &SecondNumeral);
      if(SecondNumeral == 0)
        {
          printf("Error: division by zero!");
          return(HUGE_VAL);
        }
      else
        return(Numeral / SecondNumeral);
    }
  else if(strncmp(Operation, "pow", 3) == 0)
    {
      printf("Degree: ");
U:---  calculate.c     34% L31     (C/*l Abbrev)
```

```
      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("Incorrectly entered action ");
      return(HIGE_VAL);
    }
}
```

U:--- **calculate.c**   Bot L41   (C/*l Abbrev)

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

```
U:---  calculate.h     All L4      (C/*l Abbrev)
```

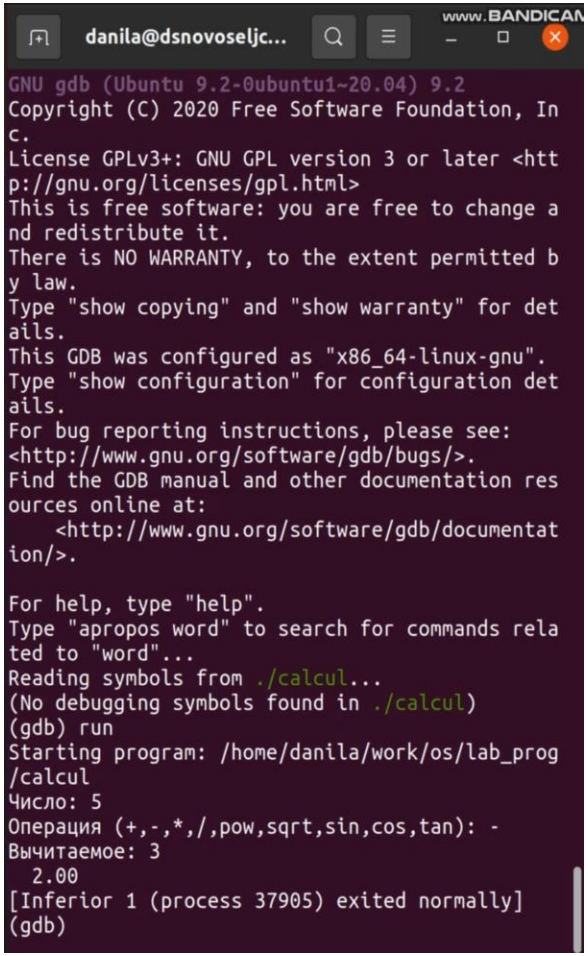
```c
#include <stdio.h>
#include <calculate.h>
int
main (void)
{
  float Numeral;
  char Operation[4];
  float Result;
  printf("Numeral: ");
  scanf("%f", &Numeral);
  printf("Operation (+,-,*,/,pow,sqrt,sin,cos,tan): ");
  scanf("%s", &Operation);
  Result = Calculate(Numeral, Operation);
  printf("%6.2f\n",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

U:--- **main.c** All L17 (C/*l Abbrev)

```
danila@dsnovoseljcev-VirtualBox:~/work/os/lab_p
rog$ make
gcc -c calculate.c
gcc -c main.c
gcc calculate.o main.o -o calcul -lm
```

```
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)


danila@dsnovoseljc...
www.BANDICAM
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/danila/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 3
2.00
[Inferior 1 (process 37905) exited normally]
(gdb)

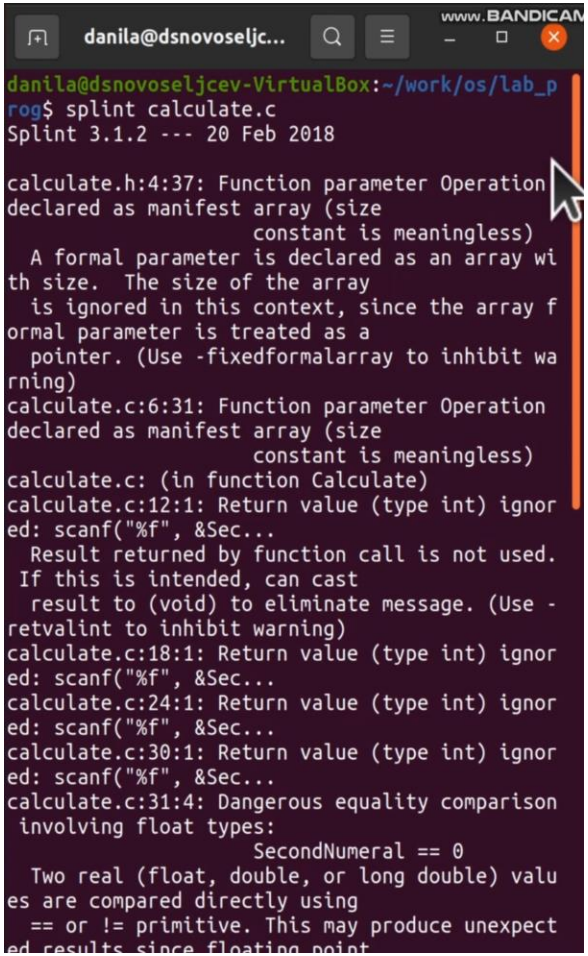
```
(gdb) list
1       #include <stdio.h>
2       #include <math.h>
3       #include <string.h>
4       #include "calculate.h"
5       float
6       Calculate(float Numeral, char Operation[4])
7       {
8         float SecondNumeral;
9         if(strncmp(Operation, "+", 1) == 0)
10          {
(gdb) list 12,15
12            scanf("%f", &SecondNumeral);
13            return(Numeral + SecondNumeral);
14          }
15        else if(strncmp(Operation, "-", 1) == 0)
(gdb)
```

```
(gdb) list
1       #include <stdio.h>
2       #include <math.h>
3       #include <string.h>
4       #include "calculate.h"
5       float
6       Calculate(float Numeral, char Operation[4])
7       {
8         float SecondNumeral;
9         if(strncmp(Operation, "+", 1) == 0)
10          {
(gdb) list 12,15
12            scanf("%f", &SecondNumeral);
13            return(Numeral + SecondNumeral);
14          }
15        else if(strncmp(Operation, "-", 1) == 0)
(gdb)
```

```
(gdb) list calculate.c:20,29
20          }
21        else if(strncmp(Operation, "*", 1) == 0)
22          {
23            printf("Factor: ");
24            scanf("%f", &SecondNumeral);
25            return(Numeral * SecondNumeral);
26          }
27        else if(strncmp(Operation, "/", 1) == 0)
28          {
29            printf("Divisor: ");
(gdb) list calculate.c:20,27
20          }
21        else if(strncmp(Operation, "*", 1) == 0)
22          {
23            printf("Factor: ");
24            scanf("%f", &SecondNumeral);
25            return(Numeral * SecondNumeral);
26          }
27        else if(strncmp(Operation, "/", 1) == 0)
(gdb) break 21
Breakpoint 1 at 0x1319: file calculate.c, line 21.
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000001319 in Calculate at calculate.c:21
(gdb)
```

```
(gdb) run
Starting program: /home/pdarzhankina/work/os/lab_prog/calcul
Numeral: 7
Operation (+,-,*,/,pow,sqrt,sin,cos,tan): pow

Breakpoint 1, Calculate (Numeral=7, Operation=0x7fffffffde14 "pow") at calculate.c:21
21          else if(strncmp(Operation, "*", 1) == 0)
(gdb) backtrace
#0  Calculate (Numeral=7, Operation=0x7fffffffde14 "pow") at calculate.c:21
#1  0x00005555555555bd in main ()
(gdb) print Numeral
$1 = 7
(gdb) display Numeral
1: Numeral = 7
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555555319 in Calculate at calculate.c:21
        breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```


danila@dsnovoseljc...
www.BANDICAM
danila@dsnovoseljcev-VirtualBox:~/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:6:31: Function parameter Operation declared as manifest array (size
constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:12:1: 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:18:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:24:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:30:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:4: 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

```
danila@dsnovoseljcev-VirtualBox:~/work/os/lab_p
rog$ 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 wi
th size.  The size of the array
  is ignored in this context, since the array f
ormal parameter is treated as a
  pointer. (Use -fixedformalarray to inhibit wa
rning)
main.c: (in function main)
main.c:10:1: Return value (type int) ignored: s
canf("%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:12:1: Return value (type int) ignored: s
canf("%s", Oper...

Finished checking --- 3 code warnings
danila@dsnovoseljcev-VirtualBox:~/work/os/lab_p
rog$
```

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