

Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

Мурашко В.В.

03.06.2021

Отчёт по лабораторной работе №14

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.

Выполнение лабораторной работы

```
vvmurashko1@dk8n70 ~ $ mkdir work
vvmurashko1@dk8n70 ~ $ cd work
vvmurashko1@dk8n70 ~/work $ mkdir os
vvmurashko1@dk8n70 ~/work $ cd ~/work/os
vvmurashko1@dk8n70 ~/work/os $ mkdir lab_prog
vvmurashko1@dk8n70 ~/work/os $ cd ~/work/os/lab_prog
vvmurashko1@dk8n70 ~/work/os/lab_prog $ █
```

```
vvmurashko1@dk8n70 ~/work/os/lab_prog $ touch calculate.h
vvmurashko1@dk8n70 ~/work/os/lab_prog $ touch calculate.c
vvmurashko1@dk8n70 ~/work/os/lab_prog $ touch main.c
vvmurashko1@dk8n70 ~/work/os/lab_prog $ ls
calculate.c calculate.h main.c
vvmurashko1@dk8n70 ~/work/os/lab_prog $ █
```

Реализация функций калькулятора в файле calculate.c

```
File Edit Options Buffers Tools C Help
#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)
```

Реализация функций калькулятора в файле calculate.c

```
File Edit Options Buffers Tools C Help
}
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);
}
}
```


Интерфейсный файл calculate.h

File Edit Options Buffers Tools C Help

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

Основной файл main.c

File Edit Options Buffers Tools C Help

```
#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("%.2f\n",Result);
    return 0;
}
```

```
vvmurashko1@dk8n70 ~/work/os/lab_prog $ gcc -c calculate.c  
vvmurashko1@dk8n70 ~/work/os/lab_prog $ gcc -c main.c
```

```
vvmurashko1@dk8n70 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
```

*Makefile

```
1 CC = gcc
2 CFLAGS = -g
3 LIBS = -lm
4
5 calcul: calculate.o main.o
6     gcc calculate.o main.o -o calcul $(LIBS)
7
8 calculate.o: calculate.c calculate.h
9     gcc -c calculate.c $(CFLAGS)
10
11 main.o: main.c calculate.h
12     gcc -c main.c $(CFLAGS)
13
14 clean:
15     -@rm calcul *.o *~
16
```

Отладчик GDB и run

```
vvmurashko1@dk8n70 ~/work/os/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...
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/v/v/vvmurashko1/work/os/lab_prog/calcul
Число: 3
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): *
Множитель: 4
12.00
[Inferior 1 (process 19068) exited normally]
(gdb) █
```

```
(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3      int
4      main (void)
5      {
6          float Numeral;
7          char Operation[4];
8          float Result;
9          printf("Число: ");
10         scanf("%f",&Numeral);
(gdb) list 12,15
12         scanf("%s",&Operation);
13         Result = Calculate(Numeral, Operation);
14         printf("%6.2f\n",Result);
15         return 0;
```

Команда list, точка останова и информация

```
20     }
21     else if(strncmp(Operation, "*", 1) == 0)
22     {
23         printf("Множитель: ");
24         scanf("%f",&SecondNumeral);
25         return(Numeral * SecondNumeral);
26     }
27     else if(strncmp(Operation, "/", 1) == 0)
28     {
29         printf("Делитель: ");
(gdb) list calculate.c:20,27
20     }
21     else if(strncmp(Operation, "*", 1) == 0)
22     {
23         printf("Множитель: ");
24         scanf("%f",&SecondNumeral);
25         return(Numeral * SecondNumeral);
26     }
27     else if(strncmp(Operation, "/", 1) == 0)
(gdb) break 21
Breakpoint 1 at 0x991: file calculate.c, line 21.
(gdb) info breakpoints
Num   Type             Disp Enb Address            What
1     breakpoint       keep y   0x0000000000000991 in Calculate at calculate.c:21
(gdb)
```


Run, команда backtrace, Numeral и удаление точки останова

```
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/v/v/vvmurashko1/work/os/lab_prog/calcul
Число: 7
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): pow

Breakpoint 1, Calculate (Numeral=7, Operation=0x7fffffff14 "pow") at calculate.c:21
21     else if(strncmp(Operation, "*", 1) == 0)
(gdb) backtrace
#0 Calculate (Numeral=7, Operation=0x7fffffff14 "pow") at calculate.c:21
#1 0x000055555400c31 in main () at main.c:13
(gdb) print Numeral
$1 = 7
(gdb) display Numeral
1: Numeral = 7
(gdb) info breakpoints
Num   Type             Disp Enb Address            What
1     breakpoint        keep y   0x000055555400991 in Calculate at calculate.c:21
      breakpoint already hit 1 time
(gdb) delete 1
```

```
vyvmurashko1@dk8n70 ~/work/os/lab_prog $ splint calculate.c
```

```
Splint 3.1.2 --- 13 Jan 2021
```

```
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:42:6: Return value (type int) ignored: scanf("%f", &Sec...
```

```
calculate.c:43:12: Return value type double does not match declared type float:
```

```
vvmurashko1@edk&n70 ~/work/os/lab_prog $ splint main.c
Splint 3.1.2 --- 13 Jan 2021

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)
main.c: (in function main)
main.c:10:3: 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:12:14: 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:12:11: Corresponding format code
main.c:12:3: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings _
```

Я 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.