

# Отчёт по лабораторной работе №13

дисциплина “Операционные системы”

---

Мишонков М. А.

4 мая 2023

Российский университет дружбы народов, Москва, Россия

## Вводная часть

---

- Целью данной лабораторной работы является приобретение простейших навыков разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

## Основная часть

---

- Создал рабочее пространство для данной лабораторной работы и необходимые файлы.

```
mamishonkov@dk8n53 ~ $ mkdir -p ~/work/os/lab_prog
mamishonkov@dk8n53 ~ $ cd ~/work/os/lab_prog
mamishonkov@dk8n53 ~/work/os/lab_prog $ touch calculate.h calculate.c main.c
mamishonkov@dk8n53 ~/work/os/lab_prog $ ls
calculate.c  calculate.h  main.c
mamishonkov@dk8n53 ~/work/os/lab_prog $
```

- Написал текст программы в файле calculate.c.

```
////////////////////////////////////  
// calculate.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: ");
```

```
}  
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);  
}  
}
```

- Написал текст программы в интерфейсном файле calculate.h, описывающем формат вызова функции калькулятора.

```
////////////////////////////////////
```

```
// calculate.h
```

```
#ifndef CALCULATE_H_
```

```
#define CALCULATE_H_
```

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

```
#endif /*CALCULATE_H_*/
```



- Написал текст программы в основном файле main.c, реализующем интерфейс пользователя к калькулятору.

```
////////////////////////////////////  
// main.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);
```

- Выполнил компиляцию программы посредством gcc.

```
mamishonkov@dk8n53 ~/work/os/lab_prog $ gcc -c calculate.c
[3]+  Завершён          emacs
mamishonkov@dk8n53 ~/work/os/lab_prog $ gcc -c main.c
mamishonkov@dk8n53 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
```

- Создал makefile с необходимым содержанием, который необходим для автоматической компиляции файлов calculate.c, main.c, а также их объединения в один исполняемый файл calcul.

```
#  
# makefile  
#  
  
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)
```

- Изменил файл, добавив в переменную CEFAGS “-g”, которая необходима для компиляции объектных файлов и их использования в программе отладчика GDB.

```
#  
# makefile  
#  
  
CC = gcc  
CFLAGS = -g  
LIBS = -lm  
  
calcul: calculate.o main.o  
    $(CC) calculate.o main.o -o calcul $(LIBS)  
  
calculate.o: calculate.c calculate.h  
    $(CC) -c calculate.c $(CFLAGS)  
  
main.o: main.c calculate.h  
    $(CC) -c main.c $(CFLAGS)
```

- Удалил исполняемый и объектный файлы из каталога при помощи команды “make clean”.

```
mamishonkov@dk8n53 ~/work/os/lab_prog $ make clean
rm calcul *.o *~
[3]+  Завершён          emacs
mamishonkov@dk8n53 ~/work/os/lab_prog $
```

- Выполнил компиляцию файлов.

```
mamishonkov@dk8n53 ~/work/os/lab_prog $ make calculate.o
gcc -c calculate.c -g
mamishonkov@dk8n53 ~/work/os/lab_prog $ make main.o
gcc -c main.c -g
mamishonkov@dk8n53 ~/work/os/lab_prog $ make calcul
gcc calculate.o main.o -o calcul -lm
mamishonkov@dk8n53 ~/work/os/lab_prog $
```

- Выполнил gdb отладку программы calcul. Ввёл команду “run” для запуска программы внутри отладчика.

```
mamishonkov@dk8n53 ~/work/os/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 12.1 vanilla) 12.1
Copyright (C) 2022 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/m/a/mamishonkov/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib64/libthread_db.so.1".
Число: 6
```

## Команда “list”

- Для построчичного просмотра исходного кода использовал команду “list”.

```
(gdb) list
1  ////////////////////////////////////////////
2  // main.c
3
4  #include <stdio.h>
5  #include "calculate.h"
6
7  int
8  main (void)
9  {
10     float Numeral;
(gdb) list
11     char Operation[4];
12     float Result;
13     printf("Число: ");
14     scanf("%f",&Numeral);
15     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
16     scanf("%s",Operation);
17     Result = Calculate(Numeral, Operation);
18     printf("%6.2f\n",Result);
19     return 0;
```



## Различные команды

- Проверил команды list (для просмотра строк основного файла), list calculate.c (для просмотра определённых строк не основного файла), break (для установки точки в файле), info breakpoints (для вывода информации об имеющихся точках останова).

```
(gdb) list calculate.c:20,29
20     {
21     printf("Вычитаемое: ");
22     scanf("%f",&SecondNumeral);
23     return(Numeral - SecondNumeral);
24     }
25     else if(strncmp(Operation, "*", 1) == 0)
26     {
27     printf("Множитель: ");
28     scanf("%f",&SecondNumeral);
29     return(Numeral * SecondNumeral);
(gdb) list calculate.c:20,27
20     {
21     printf("Вычитаемое: ");
22     scanf("%f",&SecondNumeral);
23     return(Numeral - SecondNumeral);
24     }
25     else if(strncmp(Operation, "*", 1) == 0)
26     {
27     printf("Множитель: ");
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) list calculate.c:20,27
20     {
21     printf("Вычитаемое: ");
22     scanf("%f",&SecondNumeral);
23     return(Numeral - SecondNumeral);
```

## Запуск программы

- запустил программу внутри отладчика и убедился, что программа остановилась в момент прохождения точки останова. Посмотрел, чем равно на этом этапе значение переменной Numeral и сравнил его с результатом вывода на экран. Значения совпадают. Затем убрал точки останова.

```
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdb4 "-") at calculate.c:21
21     printf("Вычитаемое: ");
(gdb) backtrace
#0  Calculate (Numeral=5, Operation=0x7fffffffdb4 "-") at calculate.c:21
#1  0x00005555555555a5 in main () at main.c:17
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
```

- При помощи утилиты splint проанализировал коды файлов calculate.c и main.c.

```
mamishonkov@dk8n53 ~ $ cd work
mamishonkov@dk8n53 ~/work $ cd os
mamishonkov@dk8n53 ~/work/os $ cd lab_prog
mamishonkov@dk8n53 ~/work/os/lab_prog $ splint calculate.c
Splint 3.1.2 --- 07 Dec 2021

calculate.h:7:37: 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:10:31: Function parameter Operation declared as manifest array
        (size constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:16:1: 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:22:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:4: 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
    representations are inexact. Instead, compare the difference to FLT_EPSILON
    or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:38:7: Return value type double does not match declared type float:
        (HUGE_VAL)
```

```
calculate.c:50:7: Return value type double does not match declared type float:  
    (sqrt(Numeral))  
calculate.c:52:7: Return value type double does not match declared type float:  
    (sin(Numeral))  
calculate.c:54:7: Return value type double does not match declared type float:  
    (cos(Numeral))  
calculate.c:56:7: Return value type double does not match declared type float:  
    (tan(Numeral))  
calculate.c:60:7: Return value type double does not match declared type float:  
    (HUGE_VAL)
```

Finished checking --- 15 code warnings

mamishonkov@dk8n53 ~/work/os/lab\_prog \$

```
mamishonkov@dk8n53 ~/work/os/lab_prog $ splint main.c
Splint 3.1.2 --- 07 Dec 2021

calculate.h:7:37: 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:14:3: 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:16:3: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 3 code warnings
mamishonkov@dk8n53 ~/work/os/lab_prog $
```

## Заключение

---

В ходе выполнения данной лабораторной работы я приобрёл простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.