

Лабораторная работа №13. Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux.”

Операционные системы

Кочарян Никита Робертович.

4 мая 2023

Российский Университет Дружбы Народов, Москва, Россия

Информация

- Кочарян Никита робертович
- студент Российского Универстите Дружбы Народов
- Российский университет дружбы народов

Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIXLinux на примере создания на языке программирования С калькулятора с простейшими функциями.

Выполнение лабораторной работы

1. В домашнем каталог создал подкаталог ~/work/os/lab_prog.

```
nrkocharyan@dk4n62 ~ $ mkdir work/os/  
nrkocharyan@dk4n62 ~ $ cd work/os  
nrkocharyan@dk4n62 ~/work/os $ mkdir lab_prog  
nrkocharyan@dk4n62 ~/work/os $ cd lab_prog
```

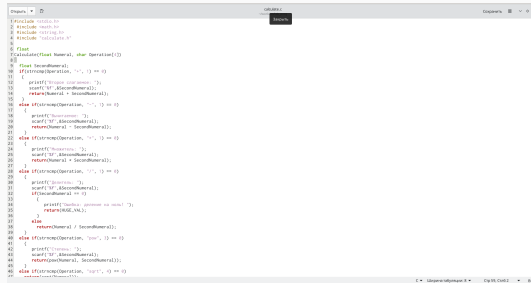
Рис. 1: рис1

2. Создаю в нем файлы calculate.h, calculate.c, main.c. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять \sin , \cos , \tan . При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.

```
nrkocharyan@dk4n62 ~/work/os/lab_prog $ touch calculate.h
nrkocharyan@dk4n62 ~/work/os/lab_prog $ touch calculate.c
nrkocharyan@dk4n62 ~/work/os/lab_prog $ touch main.c
nrkocharyan@dk4n62 ~/work/os/lab_prog $ ls
calculate.c  calculate.h  main.c
nrkocharyan@dk4n62 ~/work/os/lab_prog $
```

Рис. 2: рис2

3. Реализация функций калькулятора в файле calculate.h.



```
1 #include <stdio.h>
2 #include <math.h>
3 #include <string.h>
4 #include "calculate.h"
5
6 float
7 calculate(float Numeral, char Operation){
8     float SecondNumeral;
9     if(strlen(Operation) == 0)
10     {
11         printf("Введите второе число: ");
12         scanf("%f", &SecondNumeral);
13         return(Numeral + SecondNumeral);
14     }
15     else if(strlen(Operation) == 1)
16     {
17         printf("Введите второе число: ");
18         scanf("%f", &SecondNumeral);
19         return(Numeral - SecondNumeral);
20     }
21     else if(strlen(Operation) == 2)
22     {
23         printf("Введите второе число: ");
24         scanf("%f", &SecondNumeral);
25         return(Numeral * SecondNumeral);
26     }
27     else if(strlen(Operation) == 3)
28     {
29         printf("Введите второе число: ");
30         scanf("%f", &SecondNumeral);
31         if(SecondNumeral == 0)
32         {
33             printf("Ошибка: деление на ноль!");
34             return(HUGE_VAL);
35         }
36         else
37             return(Numeral / SecondNumeral);
38     }
39     else if(strlen(Operation) == 4)
40     {
41         printf("Введите степень: ");
42         scanf("%f", &SecondNumeral);
43         return(pow(Numeral, SecondNumeral));
44     }
45     else if(strlen(Operation) == 5)
46     {
47         return(sqrt(Numeral));
48     }
49     else if(strlen(Operation) == 3)
50     {
51         return(sin(Numeral));
52     }
53     else if(strlen(Operation) == 3)
54     {
55         return(cos(Numeral));
56     }
57     else if(strlen(Operation) == 3)
58     {
59         return(tan(Numeral));
60     }
61     else
62     {
63         printf("Неправильно введено действие ");
64         return(HUGE_VAL);
65     }
66 }
```

Рис. 3: рис3

```
45 }
46 else if(strlen(Operation) == 4)
47 {
48     return(sqrt(Numeral));
49 }
50 else if(strlen(Operation) == 3)
51 {
52     return(sin(Numeral));
53 }
54 else if(strlen(Operation) == 3)
55 {
56     return(cos(Numeral));
57 }
58 else if(strlen(Operation) == 3)
59 {
60     return(tan(Numeral));
61 }
62 else
63 {
64     printf("Неправильно введено действие ");
65     return(HUGE_VAL);
66 }
```


4. Реализация интерфейсного файла calculate.h, описывающий формат вызова функции-калькулятор

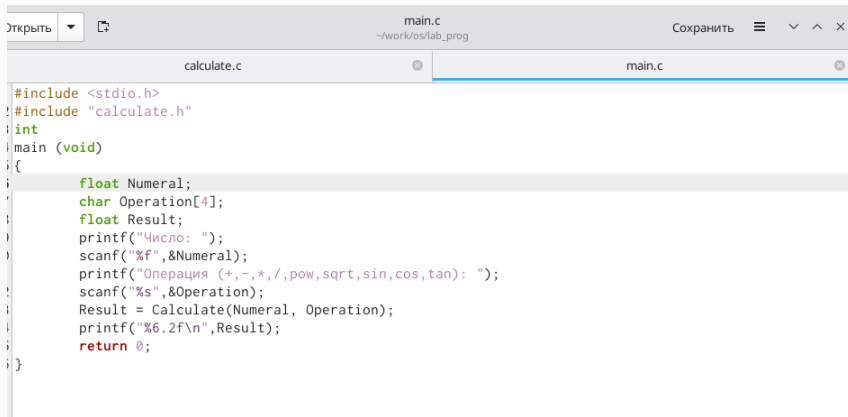


The screenshot shows a code editor with a tab labeled 'calculate.h' at the top right. Below the tab bar, the file 'calculate.c' is open. The code in the editor is as follows:

```
1 #ifndef CALCULATE_H_
2 #define CALCULATE_H_
3
4 float Calculate(float Numeral, char Operation[4]);
5
6 #endif /*CALCULATE_H_*/
```

Рис. 5: рис5

5. Реализация основного файла main.c, реализующая интерфейс пользователя к калькулятору



```
#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;
}
```

Рис. 6: рис6

6. Выполняю компиляцию программы посредством gcc

```
nrkocharyan@dk4n62 ~/work/os/lab_prog $ gcc -c calculate.c
nrkocharyan@dk4n62 ~/work/os/lab_prog $ gcc -c main.c
nrkocharyan@dk4n62 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
nrkocharyan@dk4n62 ~/work/os/lab_prog $
```

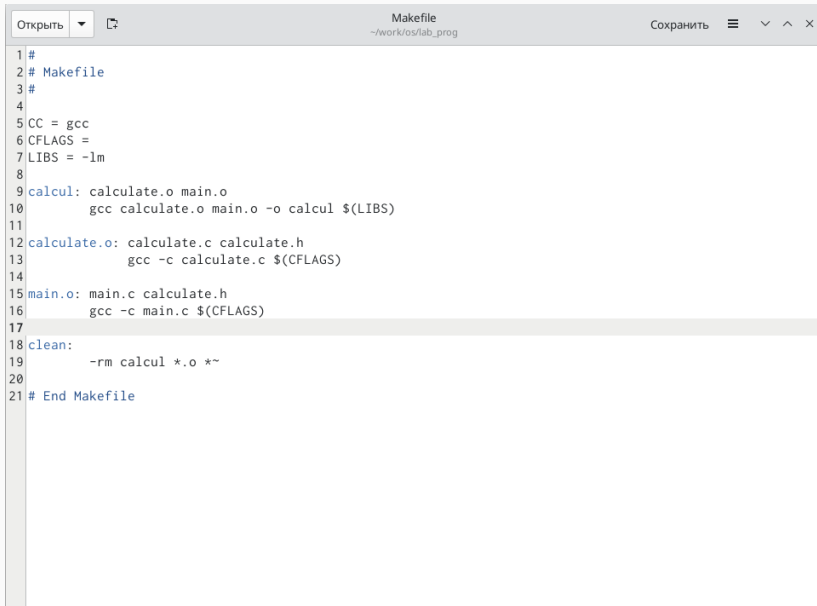
Рис. 7: рис7

```
nrkocharyan@dk4n62 ~/work/os/lab_prog $ ls
calcul calculate.c calculate.h calculate.o main.c main.o
nrkocharyan@dk4n62 ~/work/os/lab_prog $
```

Рис. 8: рис8

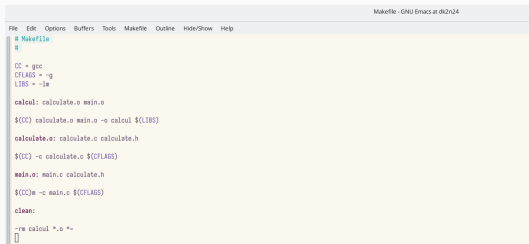
7. Исправил синтаксические ошибки

8. Создаю Makefile



```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10     gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13     gcc -c calculate.c $(CFLAGS)
14
15 main.o: main.c calculate.h
16     gcc -c main.c $(CFLAGS)
17
18 clean:
19     -rm calcul *.o *~
20
21 # End Makefile
```

9. С помощью gdb выполняю отладку программы calcul (перед использованием gdb исправляю Makefile): Запускаю отладчик GDB, загрузив в него программу для отладки `gdb ./calcul`



```
Makefile - GNU Emacs at dk2n24
File Edit Options Buffers Tools Makefile Outline Hide/Show Help
# 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)

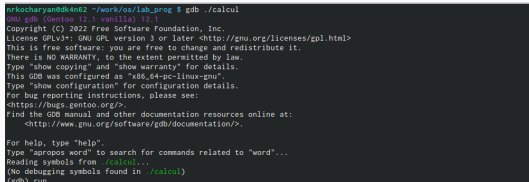
main.o: main.c calculate.h

$(CC) -c main.c $(CFLAGS)

clean:

-rm calcul *.o *
```

Рис. 10: рис10



```
nrkocharyan@dk4n62 ~/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...
(No debugging symbols found in ./calcul)
(gdb) cup
```

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