

## Лабораторная работа No 13

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

---

Акопян Сатеник

Российский университет дружбы народов, Москва, Россия

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.

1. В домашнем каталоге создайте подкаталог ~/work/os/lab\_prog.

```
[smakopyan@fedora ~]$ cd work
[smakopyan@fedora work]$ mkdir os
[smakopyan@fedora work]$ cd os
[smakopyan@fedora os]$ mkdir lab_prog
[smakopyan@fedora os]$ cd lab_prog
[smakopyan@fedora lab_prog]$
```

{##fig:0

width=50%}

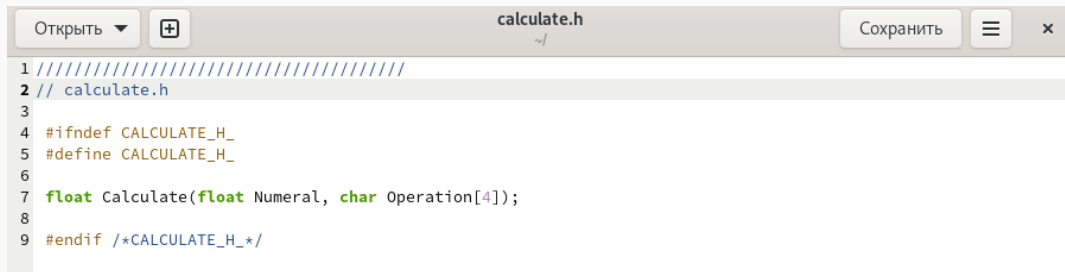
2. Создайте в нём файлы: calculate.h, calculate.c, main.c.

```
[smakopyan@fedora lab_prog]$ touch calculate.h calculate.c main.c
[smakopyan@fedora lab_prog]$ ls
calculate.c calculate.h main.c
[smakopyan@fedora lab_prog]$
```

{##fig:0

width=50%}

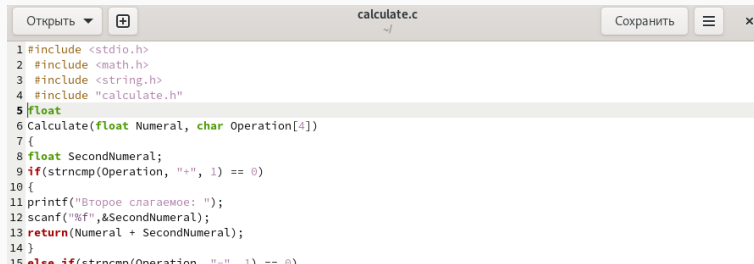
# Выполнение лабораторной работы



```
1 //////////////////////////////////////////////////
2 // calculate.h
3
4 #ifndef CALCULATE_H_
5 #define CALCULATE_H_
6
7 float Calculate(float Numeral, char Operation[4]);
8
9 #endif /*CALCULATE_H_*/
```

{##fig:0

width=50%}



```
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 {
11     printf("Второе слагаемое: ");
12     scanf("%f",&SecondNumeral);
13     return(Numeral + SecondNumeral);
14 }
15 else if(strncmp(Operation, "-", 1) == 0)
```

# Выполнение лабораторной работы

Открыть + calculate.c

```
23 printf("Множитель: ");
24 scanf("%f",&SecondNumeral);
25 return(Numeral * SecondNumeral);
26 }
27 else if(strncmp(Operation, "/", 1) == 0)
28 {
29 printf("Делитель: ");
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(strncmp(Operation, "pow", 3) == 0)
40 {
41 printf("Степень: ");
42 scanf("%f",&SecondNumeral);
43 return(pow(Numeral, SecondNumeral));
44 }
45 else if(strncmp(Operation, "sqrt", 4) == 0)
46 return(sqrt(Numeral));
47 else if(strncmp(Operation, "sin", 3) == 0)
48 return(sin(Numeral));
49 else if(strncmp(Operation, "cos", 3) == 0)
50 return(cos(Numeral));
51 else if(strncmp(Operation, "tan", 3) == 0)
52 return(tan(Numeral));
53 else
54 {
55 printf("Неправильно введено действие ");
56 return(HUGE_VAL);
57 }
58 }
```

Снимок экрана  
Вы можете вставить скриншот

### 3. Выполните компиляцию программы посредством gcc

```
[smakopyan@fedora lab_prog]$ make
gcc -c calculate.c
make: gcc: Нет такого файла или каталога
make: *** [makefile:12: calculate.o] Error 127
[smakopyan@fedora lab_prog]$ gedit makefile
[smakopyan@fedora lab_prog]$ make
gcc -c calculate.c -g
make: gcc: Нет такого файла или каталога
make: *** [makefile:12: calculate.o] Error 127
[smakopyan@fedora lab_prog]$ ^C
[smakopyan@fedora lab_prog]$ gedit makefile
[smakopyan@fedora lab_prog]$ gcc -c calculate.c
[smakopyan@fedora lab_prog]$ gcc -c main.c
[smakopyan@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[smakopyan@fedora lab_prog]$
```

{##fig:0

width=50%}

## 4. Создайте Makefile со следующим содержанием:



The screenshot shows a code editor window with the title 'makefile' and a path '~/.work/os/lab\_prog'. The editor contains a Makefile script with the following content:

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



5. Запустите отладчик GDB, загрузив в него программу для отладки:

```
[smakopyan@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora Linux 13.1-2.fc38
Copyright (C) 2023 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-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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...
```

## Выполнение лабораторной работы

6. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

```
[smakopyan@fedora lab_prog]$ splint calculate.c
```

```
Splint 3.1.2 --- 21 Jan 2023
```

```
calculate.h:7:38: 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...
```

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

