

## Лабораторная работа №13

---

Легиньких Галина - группа НФИбд-02-21

20.05.2022

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

---

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 создала подкаталог ~/work/os/lab\_prog.(рис. 1)

A terminal window with a dark background. The prompt is 'galeginjkikh@dk6n39' in green. The current directory is '~/work/os/lab\_prog' in blue. The prompt character is '\$' in blue. A white cursor is at the end of the line.

```
galeginjkikh@dk6n39 ~/work/os/lab_prog $
```

Figure 1: Подкаталог

2. Создала в подкаталоге файлы: `calculate.h`, `calculate.c`, `main.c`.

У меня получился примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять  $\sin$ ,  $\cos$ ,  $\tan$ .

При запуске он запрашивает первое число, операцию, второе число. После этого программа выведет результат и останавливается.

## Реализация функций калькулятора в файле calculate.c:(рис. 2)(рис. 3)

```
#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);
```

Интерфейсный файл calculate.h, описывающий формат вызова функции калькулятора:(рис. 4)

```
#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;
}
~
~
~
~
```

Figure 4: calculate.h

Основной файл main.c, реализующий интерфейс пользователя к калькулятору:(рис. 5)

```
galeglnjkhHdk6n39 ~/work/os/lab_prog $ gcc -c calculate.c
galeglnjkhHdk6n39 ~/work/os/lab_prog $ gcc -c main.c
main.c: В функции «main»:
main.c:13:10: предупреждение: формат «%s» ожидает аргумент типа «char *», но аргумент 2 имеет тип «char (*)[4]» [-Wformat=]
   13 |     scanf("%s", &operation);
      |           ^~
      |           |
      |           | char (*)[4]
      |           char *
galeglnjkhHdk6n39 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
galeglnjkhHdk6n39 ~/work/os/lab_prog $ ls
calcul calculate.c calculate.c~ calculate.h calculate.h~ calculate.o main.c main.c~ main.o Makefile Makefile~
```

Figure 5: main.c



### 3. Выполнила компиляцию программы посредством gcc:(рис. 6)

```
#
# Makefile
#

CC = gcc
CFLAGS = -g
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

# End Makefile
~
~
~
~
```

Figure 6: gcc

#### 4. Создала Makefile со следующим содержанием:(рис. 7)

```
galeginkikh@dk6n39 ~/work/os/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 10.2 vanilla) 10.2
Copyright (C) 2021 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) |
```

Figure 7: Makefile

5. С помощью gdb выполнила отладку программы calcul (перед использованием gdb исправьте Makefile):

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

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.