

# **Лабораторная работа №13**

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

Салькова Кристина Михайловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>11</b>
	<b>Список литературы</b>	<b>12</b>

## Список иллюстраций

3.1	создание подкаталога . . . . .	7
3.2	создание файлов . . . . .	7
3.3	реализация calculate.c(1) . . . . .	7
3.4	реализация calculate.c(2) . . . . .	8
3.5	calculate.h . . . . .	8
3.6	main.c . . . . .	8
3.7	выполнение компиляции . . . . .	9
3.8	создание makefile . . . . .	9
3.9	исправление makefile . . . . .	9
3.10	запустим gdb . . . . .	10

## **Список таблиц**

# 1 Цель работы

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.

## 2 Задание

1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.
2. Создайте в нём 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.
  - Реализация функций калькулятора в файле `calculate.c`
  - Интерфейсный файл `calculate.h`, описывающий формат вызова функции-калькулятора
  - Основной файл `main.c`, реализующий интерфейс пользователя к калькулятору
3. Выполните компиляцию программы посредством `gcc`
4. При необходимости исправьте синтаксические ошибки.
5. Создайте `Makefile` со следующим содержанием
6. С помощью `gdb` выполните отладку программы `calcul` (перед использованием `gdb` исправьте `Makefile`):
  - Запустите отладчик GDB, загрузив в него программу для отладки # Теоретическое введение

## 3 Выполнение лабораторной работы

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

```
kmsaljkova@dk4n68 ~ $ cd ~/work/os
kmsaljkova@dk4n68 ~/work/os $ mkdir lab_prog
```

Рис. 3.1: создание подкаталога

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

```
kmsaljkova@dk4n68 ~/work/os $ cd lab_prog
kmsaljkova@dk4n68 ~/work/os/lab_prog $ touch calculate.h
kmsaljkova@dk4n68 ~/work/os/lab_prog $ touch calculate.c
kmsaljkova@dk4n68 ~/work/os/lab_prog $ touch main.c
kmsaljkova@dk4n68 ~/work/os/lab_prog $ ls
calculate.c calculate.h main.c
kmsaljkova@dk4n68 ~/work/os/lab_prog $
```

Рис. 3.2: создание файлов

- Реализация функций калькулятора в файле calculate.c

```
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 {
9     float SecondNumeral;
10    if(strcmp(Operation, "+") == 0)
11    {
12        printf("Введите второе число: ");
13        scanf("%f", &SecondNumeral);
14        return(Numeral + SecondNumeral);
15    }
16    else if(strcmp(Operation, "-") == 0)
17    {
18        printf("Введите второе число: ");
19        scanf("%f", &SecondNumeral);
20        return(Numeral - SecondNumeral);
21    }
22    else if(strcmp(Operation, "*") == 0)
23    {
24        printf("Введите второе число: ");
25        scanf("%f", &SecondNumeral);
26        return(Numeral * SecondNumeral);
27    }
28    else if(strcmp(Operation, "/") == 0)
29    {
30        printf("Введите второе число: ");
31        scanf("%f", &SecondNumeral);
32        if(SecondNumeral == 0)
33        {
34            printf("Ошибка: деление на ноль!");
35            return(MINUS_VAL);
36        }
37        else
38            return(Numeral / SecondNumeral);
39    }
40    else if(strcmp(Operation, "pow",) == 0)
41    {

```

Рис. 3.3: реализация calculate.c(1)

```

45     printf("Степень: ");
46     scanf("%f",&SecondNumeral);
47     return(pow(Numeral, SecondNumeral));
48 }
49 else if(strncmp(Operation, "sqrt", 4) == 0)
50     return(sqrt(Numeral));
51 else if(strncmp(Operation, "sin", 3) == 0)
52     return(sin(Numeral));
53 else if(strncmp(Operation, "cos", 3) == 0)
54     return(cos(Numeral));
55 else if(strncmp(Operation, "tan", 3) == 0)
56     return(tan(Numeral));
57 else
58 {
59     printf("Неправильно введено действие ");
60     return(HUGE_VAL);
61 }
62

```

Рис. 3.4: реализация calculate.c(2)

- Интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора

```

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

```

Рис. 3.5: calculate.h

- Основной файл main.c, реализующий интерфейс пользователя к калькулятору

```

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

```

Рис. 3.6: main.c

### 3. Выполните компиляцию программы посредством gcc



```
msys64@kali:~/work/cv/lab_prog$ gcc -c calculate.c
msys64@kali:~/work/cv/lab_prog$ gcc -o calc.o calculate.c
main.c: в функции main:
main.c:13:11: предупреждение: формат '%s' ожидает аргумент типа char *, но аргумент 2 имеет тип char (*)[4] [-Wformat=]
   13 |         scanf("%s", &response);
      |               ^~~~~~
       |             |
       |             +---+
       |             |   char *
       |             +---+
       |             |   char *
       |             +---+
msys64@kali:~/work/cv/lab_prog$ gcc -c calculate.o main.o -o calcul -lm
gcc: предупреждение: calculate.o: некоторые данные компонента не использованы, поскольку компонента не выполняется
gcc: предупреждение: main.o: некоторые данные компоненты не использованы, поскольку компонента не выполняется
```

Рис. 3.7: выполнение компиляции

4. Видим, что существуют некоторые ошибки и исправляем их
5. Создадим Makefile

```
kmsaljkova@dk4n68 ~/work/os/lab_prog $ touch mkaefile
kmsaljkova@dk4n68 ~/work/os/lab_prog $ touch makefile
```

Рис. 3.8: создание makefile

- ## 6. Исправление makefile

```
makefile - GNU Emacs at dk3n63
```

---

```
File Edit Options Buffers Tools Makefile Outline Hide/Show Help

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)

clean:
    -rm calcul *.o *~

#End makefile
```

Рис. 3.9: исправление makefile

- Запустите отладчик GDB, загрузив в него программу для отладки

```

kmsaljkova@dk3n63 ~/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.ht
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/k/m/kmsaljkova/work/os/lab_
[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): *
Множитель: 5
25.00

```

Рис. 3.10: запустим gdb

## 4 Выводы

Мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.

## **Список литературы**