

## Лабораторная работа №13

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

---

Шуплецов А. А.

6 мая 2023

Российский университет дружбы народов, Москва, Россия

## Информация

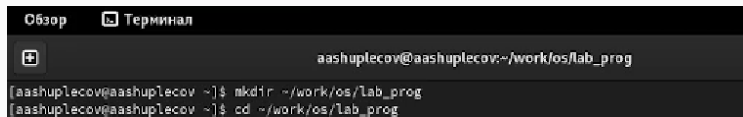
---

- Шуплецов Александр Андреевич
- студент ФФМиЕН
- Российский университет дружбы народов
- <https://github.com/winnralex>

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.



В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`.



The image shows a terminal window with a dark background. At the top, there are two tabs: 'Обзор' (Overview) and 'Терминал' (Terminal), with 'Терминал' being the active tab. Below the tabs, the terminal title bar shows a '+' icon and the text 'aashuplecov@aashuplecov:~/work/os/lab\_prog'. The terminal content shows two commands being executed: first, `[aashuplecov@aashuplecov ~]$ mkdir ~/work/os/lab_prog`, and second, `[aashuplecov@aashuplecov ~]$ cd ~/work/os/lab_prog`.

```
Обзор  Терминал
aashuplecov@aashuplecov:~/work/os/lab_prog
[aashuplecov@aashuplecov ~]$ mkdir ~/work/os/lab_prog
[aashuplecov@aashuplecov ~]$ cd ~/work/os/lab_prog
```

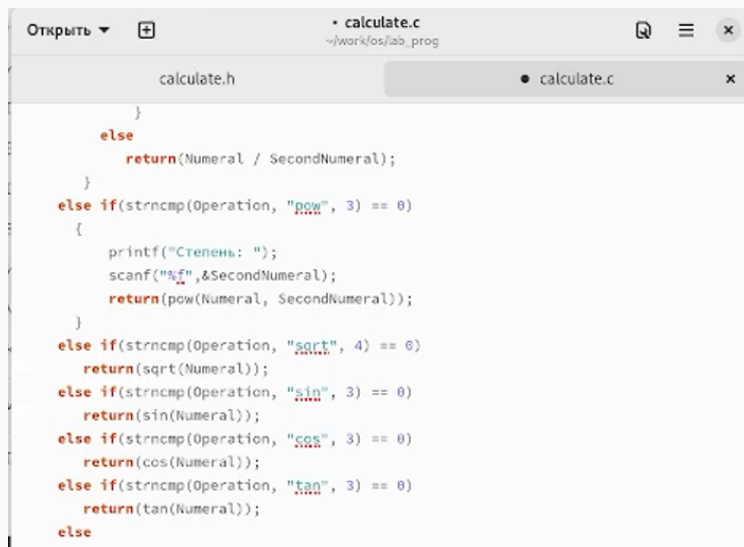
Рис. 1: создание подкаталога

Создадим в нём файлы: calculate.h, calculate.c, main.c.

```
[aashuplecov@aashuplecov ~]$ cd ~/work/os/lab_prog  
[aashuplecov@aashuplecov lab_prog]$ touch calculate.h calculate.c main.c  
[aashuplecov@aashuplecov lab_prog]$ ls  
calculate.c calculate.h main.c  
[aashuplecov@aashuplecov lab_prog]$
```

Рис. 2: создание файлов

Напишем реализацию функций калькулятора в файле calculate.c.



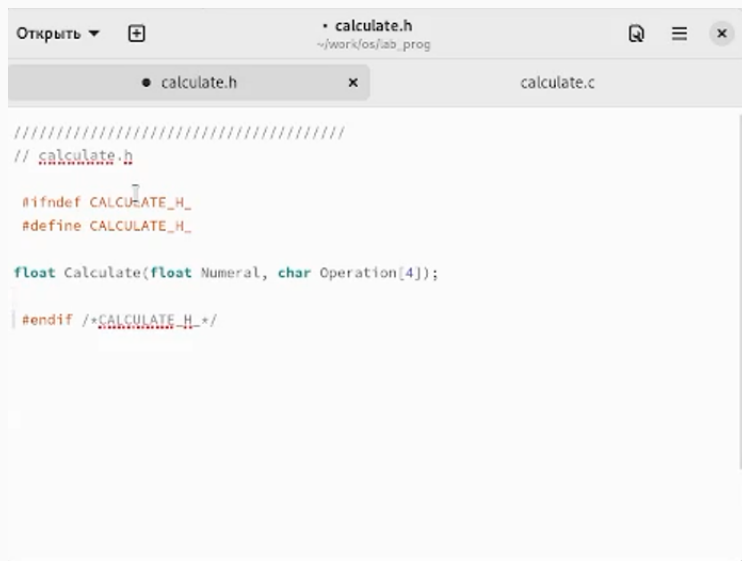
The screenshot shows a code editor with two tabs: 'calculate.h' and 'calculate.c'. The 'calculate.c' tab is active, displaying the implementation of calculator functions. The code is written in C and includes comments in Russian. It defines a function 'calculate' that takes a string 'operation' and two double values 'Numeral' and 'SecondNumeral' as arguments. The function uses a series of 'if' and 'else if' statements to handle different operations: addition, subtraction, multiplication, division, power, square root, sine, cosine, and tangent. Each operation is implemented using standard C library functions like '+', '-', '\*', '/', 'pow', 'sqrt', 'sin', 'cos', and 'tan'. The function returns the result of the operation or 0 if the operation is not recognized.

```
    }  
    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
```

Рис. 3: реализация функций калькулятора в файле calculate.c



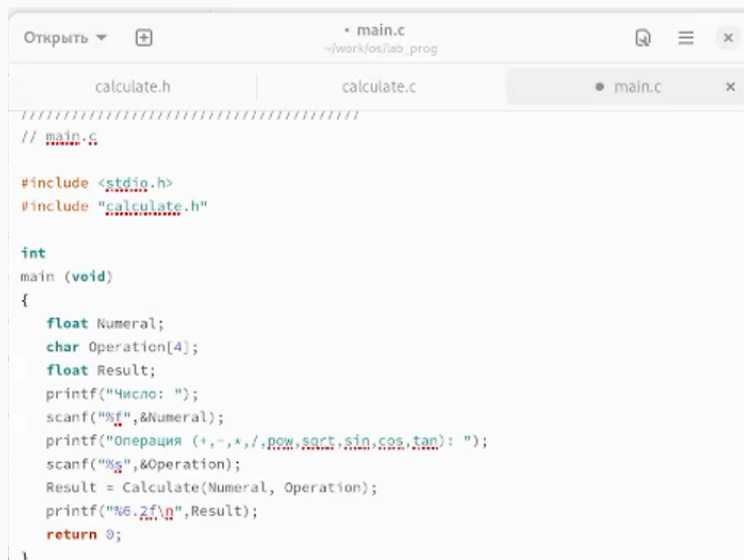
Напишем интерфейсный файл `calculate.h`, описывающий формат вызова функции калькулятора.



The screenshot shows a code editor window with a tab titled 'calculate.h'. The editor displays the following C header file code:

```
////////////////////////////////////  
// calculate.h  
  
#ifndef CALCULATE_H_  
#define CALCULATE_H_  
  
float Calculate(float Numeral, char Operation[4]);  
  
#endif /*CALCULATE_H_*/
```

Напишем основной файл main.c, реализующий интерфейс пользователя к калькулятору.



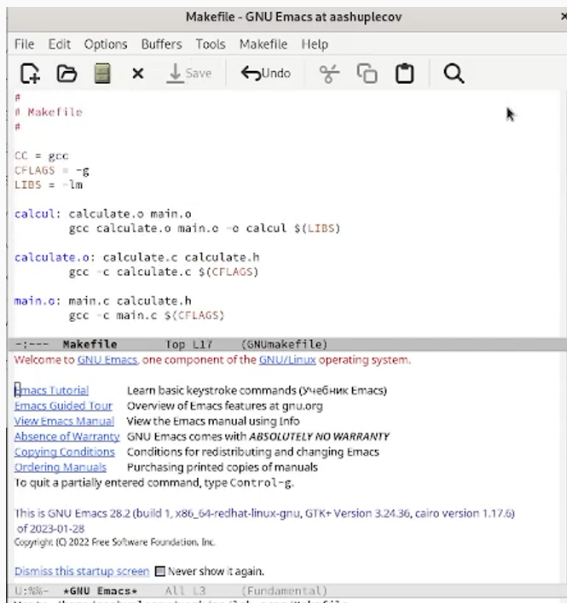
```
////////////////////////////////////  
// main.c  
  
#include <stdio.h>  
#include "calculate.h"  
  
int  
main (void)  
{  
    float Numeral;  
    char Operation[4];  
    float Result;  
    printf("Число: ");  
    scanf("%f",&Numeral);  
    printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");  
    scanf("%s",&Operation);  
    Result = Calculate(Numeral, Operation);  
    printf("%6.2f\n",Result);  
    return 0;  
}
```

Выполним компиляцию программы посредством gcc.

```
calculate.o calculate.o main.o  
[aashuplecov@aashuplecov lab_prog]$ gcc -c calculate.c  
[aashuplecov@aashuplecov lab_prog]$ gcc -c main.c  
[aashuplecov@aashuplecov lab_prog]$ gcc calculate.o main.o -o calcul -lm  
[aashuplecov@aashuplecov lab_prog]$ touch Makefile
```

Рис. 6: компиляция программы

## Создадим Makefile для нашей программы.



The screenshot shows the GNU Emacs editor window titled "Makefile - GNU Emacs at aashuplecov". The menu bar includes File, Edit, Options, Buffers, Tools, Makefile, and Help. The toolbar contains icons for file operations and editing. The main text area displays a Makefile with the following content:

```
#
# 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)
```

Below the code, a status bar shows "-- Makefile Top L17 (GNUmakefile)". The bottom of the window displays the GNU Emacs startup screen with the following text:

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

<a href="#">Emacs Tutorial</a>	Learn basic keystroke commands (Учебник Emacs)
<a href="#">Emacs Guided Tour</a>	Overview of Emacs features at gnu.org
<a href="#">View Emacs Manual</a>	View the Emacs manual using Info
<a href="#">Absence of Warranty</a>	GNU Emacs comes with <b>ABSOLUTELY NO WARRANTY</b>
<a href="#">Copying Conditions</a>	Conditions for redistributing and changing Emacs
<a href="#">Ordering Manuals</a>	Purchasing printed copies of manuals

To quit a partially entered command, type Control-g.

This is GNU Emacs 28.2 (build 1, x86\_64-redhat-linux-gnu, GTK+ Version 3.24.36, cairo version 1.17.6)  
of 2023-01-28  
Copyright (C) 2022 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

The bottom status bar shows "L:1:13 GNU Emacs+ All L3 (Fundamental)".

## Запустим отладчик программы.

```
(gdb) run
Starting program: /home/aashuplecov/work/os/lab_prog/calcul
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc6000
Downloading separate debug info for /lib64/libm.so.6
Downloading separate debug info for /lib64/libc.so.6
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 12
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 12
24.00
[Inferior 1 (process 4022) exited normally]
```

Рис. 8: отладчик программы

Для постраничного просмотра исходного кода используем 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;
```

Рис. 9: команда list

Для просмотра определённых строк не основного файла используем list с параметрами.

```
(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);
```

Рис. 10: команда list с параметрами

Установим точку останова в файле calculate.c на строке номер 21.

```
(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) break 21
Breakpoint 1 at 0x40120f: file calculate.c, line 21.
```

Рис. 11: точка останова



Выведем информацию об имеющихся в проекте точка останова.

```
(gdb) info breakpoints
Num      Type             Disp Enb Address            What
1        breakpoint      keep y   0x000000000040120f in Calculate at calculate.c:21
(gdb) run
Starting program: /home/aashuplecov/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
```

Рис. 12: вывод информации о точках останова

Запустим программу внутри отладчика и убедимся, что программа остановится в момент прохождения точки останова.

```
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf54 "=") at calculate.c:21
21      printf("Вычисляемое: ");
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffdf54 "=") at calculate.c:21
#1 0x00000000004014eb in main () at main.c:17
(gdb)
```

Рис. 13: момент прохождения точки останова

Посмотрим, чему равно на этом этапе значение переменной Numeral.

```
(gdb) print Numeral  
$1 = 5  
(gdb) display Numeral  
1: Numeral = 5
```

Рис. 14: значение Numeral

```
(gdb) info breakpoints
Num      Type             Disp Enb Address            What
1        breakpoint      keep y   0x000000000040120f in calculate at calculate.c:21
          breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

Рис. 15: удаление точек останова

## С помощью утилиты splint проанализируем код файла calculate.c.

```
aashuplecov@aashuplecov:~/work/os/lab_prog

* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
Splint 3.1.2 == 23 Jul 2022

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:16:10: 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:22:10: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:10: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:10: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:13: 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:38:22: Return value type double does not match declared type float:
        (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:46:11: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:47:17: Return value type double does not match declared type float:
        (pow(Numeral, SecondNumeral))
calculate.c:50:14: Return value type double does not match declared type float:
        (sqrt(Numeral))
calculate.c:52:14: Return value type double does not match declared type float:
        (sin(Numeral))
calculate.c:54:14: Return value type double does not match declared type float:
        (cos(Numeral))
calculate.c:56:14: Return value type double does not match declared type float:
        (tan(Numeral))
calculate.c:60:15: Return value type double does not match declared type float:
        (HUGE_VAL)
```

## С помощью утилиты splint проанализируем код файла main.c.

aashuplecov@aashuplecov:~/work/os/lab\_prog

```
calculate.c:28:10: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:10: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:13: 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:38:22: Return value type double does not match declared type float:
    (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:46:11: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:47:17: Return value type double does not match declared type float:
    (pow(Numeral, SecondNumeral))
calculate.c:50:14: Return value type double does not match declared type float:
    (sqrt(Numeral))
calculate.c:52:14: Return value type double does not match declared type float:
    (sin(Numeral))
calculate.c:54:14: Return value type double does not match declared type float:
    (cos(Numeral))
calculate.c:56:14: Return value type double does not match declared type float:
    (tan(Numeral))
calculate.c:60:15: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings

[aashuplecov@aashuplecov lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2022

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:14:4: 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:16:15: Format argument 1 to scanf (%s) expects char * gets char [4] *:
    &Operation
    Type of parameter is not consistent with corresponding code in format string.
    (Use -formattype to inhibit warning)
    main.c:16:12: Corresponding format code
main.c:16:4: Return value (type int) ignored: scanf("%s", &ope...
```

Finished checking --- 4 code warnings

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

Кулябов Д.С. “Материалы к лабораторным работам”