

---

## Front matter

title: "Отчёт по лабораторной работе №13" subtitle: " Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux " author: "Федорина Эрнест Васильевич НКНбд-01-21"

## Generic options

lang: ru-RU toc-title: "Содержание"

## Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables  
fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia  
polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true  
polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs:  
english ## Fonts mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT  
Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions:  
Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9  
## Biblatex biblatex: true biblio-style: "gost-numeric" biblatexoptions: - parenttracker=true  
- backend=biber - hyperref=auto - language=auto - autolang=other\* - citestyle=gost-  
numeric ## Pandoc-crossref LaTeX customization figureTitle: "Рис." tableTitle: "Таблица"  
listingTitle: "Листинг" lofTitle: "Список иллюстраций" lotTitle: "Список таблиц" lolTitle:  
"Листинги" ## Misc options indent: true header-includes: -

## keep figures where there are in the text

– # keep figures where there are in the text

## 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.

## Задание

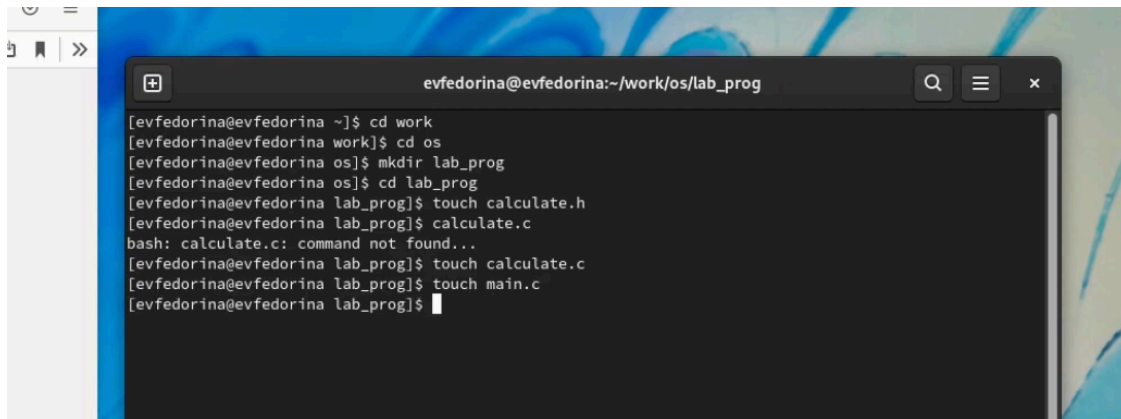
1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.
2. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять `sin`, `cos`, `tan`. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.
3. Выполните компиляцию программы посредством `gcc`
4. При необходимости исправьте синтаксические ошибки.
5. Создайте `Makefile` со некоторым содержанием
6. С помощью `gdb` выполните отладку программы `calcul` (перед использованием `gdb` исправьте `Makefile`) – Запустите отладчик `GDB`, загрузив в него программу для отладки
7. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`.

## Теоретическое введение

Процесс разработки программного обеспечения обычно разделяется на следующие этапы: – планирование, включающее сбор и анализ требований к функционалу и другим характеристикам разрабатываемого приложения; – проектирование, включающее в себя разработку базовых алгоритмов и спецификаций, определение языка программирования; – непосредственная разработка приложения: – кодирование — по сути создание исходного текста программы (возможно в нескольких вариантах); – анализ разработанного кода; – сборка, компиляция и разработка исполняемого модуля; – тестирование и отладка, сохранение произведённых изменений; – документирование. Для создания исходного текста программы разработчик может воспользоваться любым удобным для него редактором текста: `vi`, `vim`, `mceditor`, `emacs`, `geany` и др. После завершения написания исходного кода программы (возможно состоящей из нескольких файлов), необходимо её скомпилировать и получить исполняемый модуль.

## Выполнение лабораторной работы

В домашнем каталоге создали подкаталог `~/work/os/lab_prog` и создали нужные файлы.(рис.1)



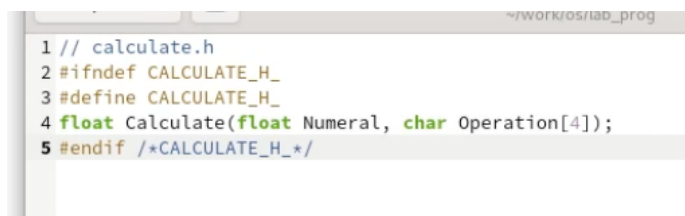
```
evfedorina@evfedorina:~/work/os/lab_prog
[evfedorina@evfedorina ~]$ cd work
[evfedorina@evfedorina work]$ cd os
[evfedorina@evfedorina os]$ mkdir lab_prog
[evfedorina@evfedorina os]$ cd lab_prog
[evfedorina@evfedorina lab_prog]$ touch calculate.h
[evfedorina@evfedorina lab_prog]$ calculate.c
bash: calculate.c: command not found...
[evfedorina@evfedorina lab_prog]$ touch calculate.c
[evfedorina@evfedorina lab_prog]$ touch main.c
[evfedorina@evfedorina lab_prog]$
```

{рис.1}

Заполнили файлы материалом из гайда(рис.2,3)

```
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[4])
8 {
9     float SecondNumeral;
10    if(strncmp(Operation, "+", 1) == 0)
11    {
12        printf("Второе слагаемое: ");
13        scanf("%f",&SecondNumeral);
14        return(Numeral + SecondNumeral);
15    }
16    else if(strncmp(Operation, "-", 1) == 0)
17    {
18        printf("Вычитаемое: ");
19        scanf("%f",&SecondNumeral);
20        return(Numeral - SecondNumeral);
21    }
22    else if(strncmp(Operation, "*", 1) == 0)
23    {
24        printf("Множитель: ");
25        scanf("%f",&SecondNumeral);
26        return(Numeral * SecondNumeral);
27    }
```

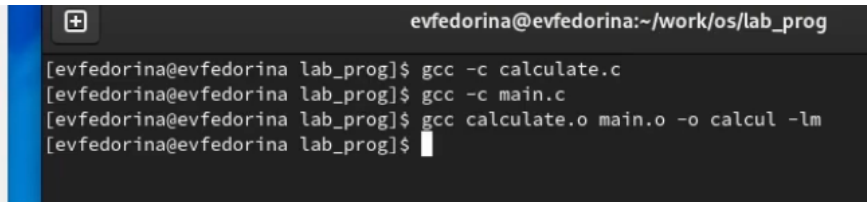
{рис.2}



```
~/work/os/lab_prog
1 // calculate.h
2 #ifndef CALCULATE_H_
3 #define CALCULATE_H_
4 float Calculate(float Numeral, char Operation[4]);
5 #endif /*CALCULATE_H_*/
```

{рис.3}

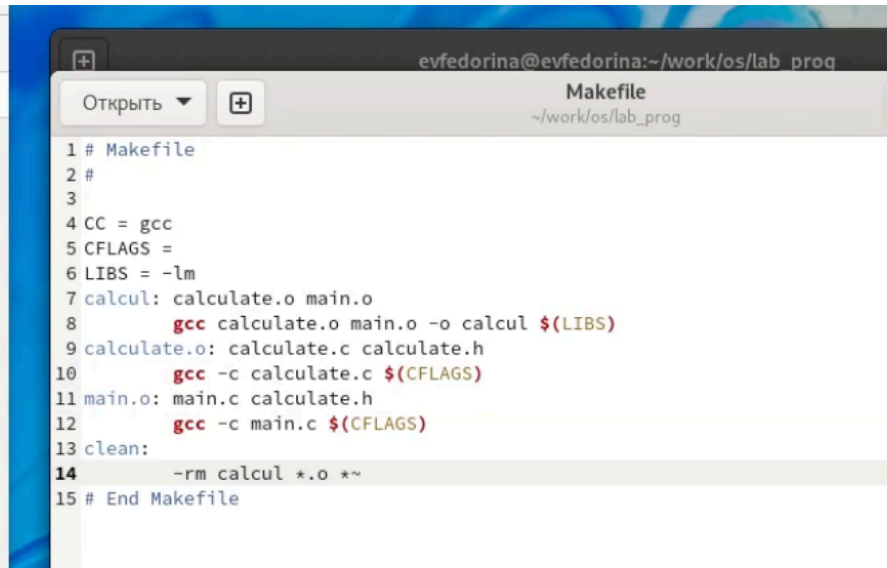
Выполнили компиляцию программы посредством gcc(рис.4)



```
evfedorina@evfedorina:~/work/os/lab_prog
[evfedorina@evfedorina lab_prog]$ gcc -c calculate.c
[evfedorina@evfedorina lab_prog]$ gcc -c main.c
[evfedorina@evfedorina lab_prog]$ gcc calculate.o main.o -o calcul -lm
[evfedorina@evfedorina lab_prog]$
```

{рис.4}

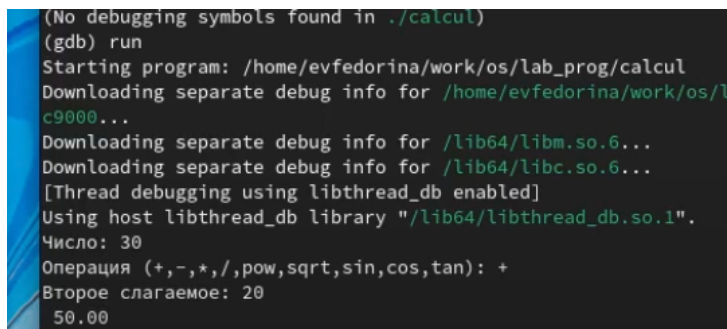
Создали Makefile со некоторым содержанием(рис.5)



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

{рис.5}

С помощью gdb выполнили отладку программы calcul (перед использованием gdb исправил Makefile) и запустили отладчик GDB(рис.6)



```
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/evfedorina/work/os/lab_prog/calcul
Downloading separate debug info for /home/evfedorina/work/os/lib64/libc.so.6...
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: 30
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 20
50.00
```

{рис.6}

С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c(рис.7)

```

        constant is meaningless)
calculate.c: (in function Calculate)
> calculate.c:13:2: 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:19:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:25:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:5: 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:35:10: Return value type double does not match declared type float:
    (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:43:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:8: Return value type double does not match declared type float:
    (pow(Numeral, SecondNumeral))
calculate.c:47:8: Return value type double does not match declared type float:
    (sqrt(Numeral))
calculate.c:49:8: Return value type double does not match declared type float:
    (sin(Numeral))
calculate.c:51:8: Return value type double does not match declared type float:
    (cos(Numeral))
calculate.c:53:8: Return value type double does not match declared type float:
    (tan(Numeral))
calculate.c:57:8: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings
[evfedorina@evfedorina lab_prog]$

```

{рис.7}

## Выводы

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