

Лабораторная работа №14

Операционные системы

Александрова Ульяна

06 мая 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Александрова Ульяна
- студентка 1-го курса
- Российский университет дружбы народов
- kulyabov-ds@rudn.ru
- <https://AleksandrovaUV.github.io/ru/>

Цель работы

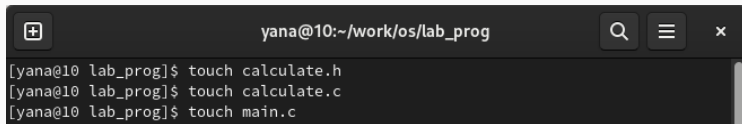
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

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 следующим содержанием.
6. С помощью `gdb` выполните отладку программы `calcul` (перед использованием `gdb` исправьте `Makefile`).
7. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`.

Выполнение лабораторной работы

Создала требуемый репозиторий и файлы, необходимые для выполнения работы.

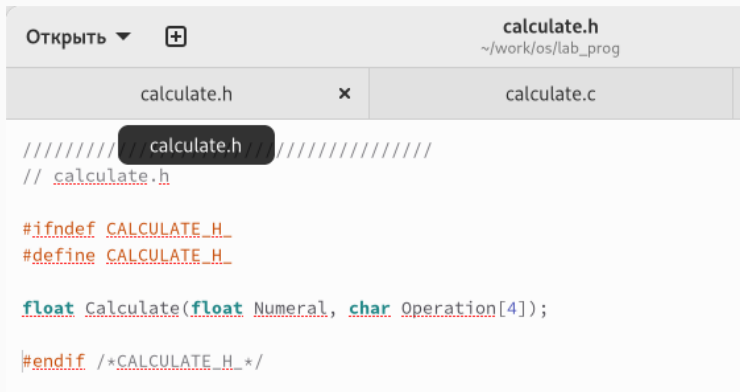
A terminal window with a dark background. The title bar shows a plus icon, the text 'yana@10:~/work/os/lab_prog', a search icon, a menu icon, and a close icon. The terminal content shows three lines of commands and their output: '[yana@10 lab_prog]\$ touch calculate.h', '[yana@10 lab_prog]\$ touch calculate.c', and '[yana@10 lab_prog]\$ touch main.c'.

```
yana@10:~/work/os/lab_prog  
[yana@10 lab_prog]$ touch calculate.h  
[yana@10 lab_prog]$ touch calculate.c  
[yana@10 lab_prog]$ touch main.c
```

Figure 1: Создание файлов

Выполнение лабораторной работы

Записываю реализацию функций калькулятора сначала в файле calculate.h.




```
Открыть ▾  calculate.h  
~/work/os/lab_prog  
calculate.h x calculate.c  
////////// calculate.h //////////  
// calculate.h  
  
#ifndef CALCULATE_H_  
#define CALCULATE_H_  
  
float Calculate(float Numeral, char Operation[4]);  
  
#endif /*CALCULATE_H_*/
```

Figure 2: calculate.h

Выполнение лабораторной работы

Затем в calculate.c.

```
calculate.c
~/work/os/lab_prog

calculate.h calculate.c x

////////////////////////////////////
// calculate.c

#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);
    }
}
```

Выполнение лабораторной работы

И в main.c.

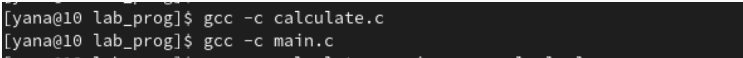
```
main.c
~/work/os/lab_prog

calculate.h    calculate.c    r

////////////////////////////////////
// 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;
}
```

Выполняю компиляцию программы посредством gcc.

A terminal window with a dark background and light-colored text. It shows two lines of commands being entered at a prompt. The first line is '[yana@10 lab_prog]\$ gcc -c calculate.c' and the second line is '[yana@10 lab_prog]\$ gcc -c main.c'.

```
[yana@10 lab_prog]$ gcc -c calculate.c  
[yana@10 lab_prog]$ gcc -c main.c
```

Figure 5: Компиляция

Выполнение лабораторной работы

Создаю и заполняю файл Makefile.



```
#
# 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)

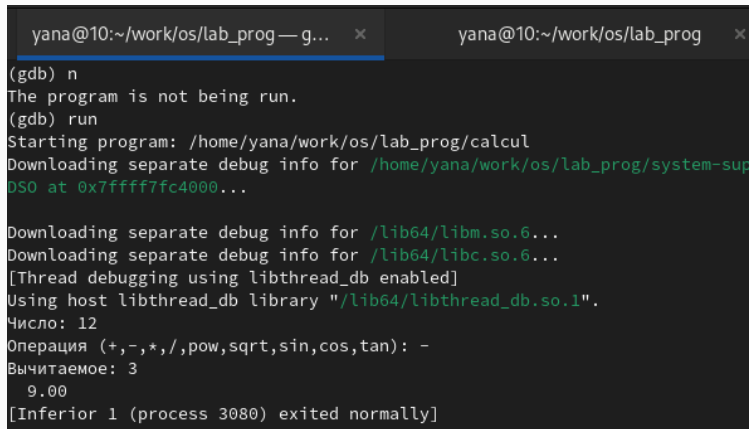
main.o: main.c calculate.h
    gcc -c main.c $(CFLAGS)

clean:
    -rm calcul *.o *~
```

Figure 6: Makefile

Выполнение лабораторной работы

Запустила отладчик gdb, программа работает.

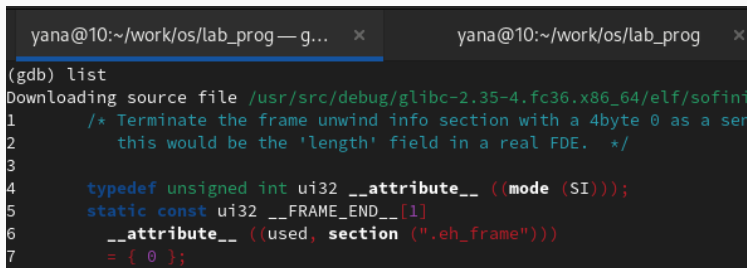


```
yana@10:~/work/os/lab_prog — g... x yana@10:~/work/os/lab_prog x
(gdb) n
The program is not being run.
(gdb) run
Starting program: /home/yana/work/os/lab_prog/calcul
Downloading separate debug info for /home/yana/work/os/lab_prog/system-sup
DSO at 0x7ffff7fc4000...

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): -
Вычитаемое: 3
9.00
[Inferior 1 (process 3080) exited normally]
```

Figure 7: Отладка

Для постраничного просмотра исходного код использовала команду list.



```
yana@10:~/work/os/lab_prog — g... x yana@10:~/work/os/lab_prog x
(gdb) list
Downloading source file /usr/src/debug/glibc-2.35-4.fc36.x86_64/elf/sofini
1      /* Terminate the frame unwind info section with a 4byte 0 as a ser
2      this would be the 'length' field in a real FDE. */
3
4      typedef unsigned int ui32 __attribute__ ((mode (SI)));
5      static const ui32 __FRAME_END__[1]
6          __attribute__ ((used, section (".eh_frame")))
7          = { 0 };
```

Figure 8: Просмотр кода

Выводы

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