

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №14
дисциплина: Операционные системы

Студент: Адхамжонов Жасурбек Илхамжонович

Группа: НФИбд-02-20

МОСКВА

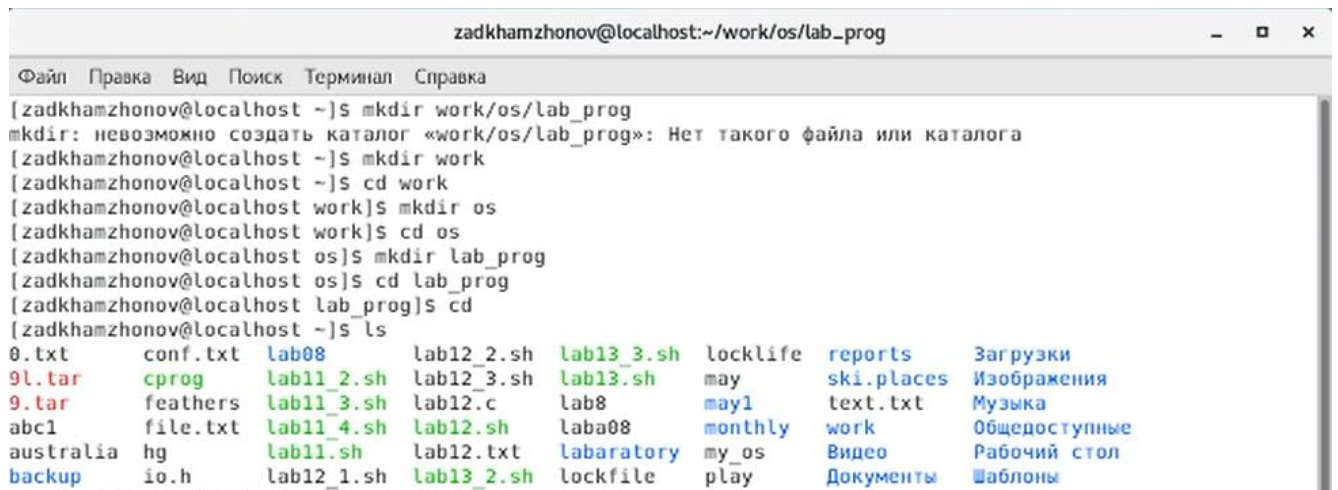
2021 г.

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.

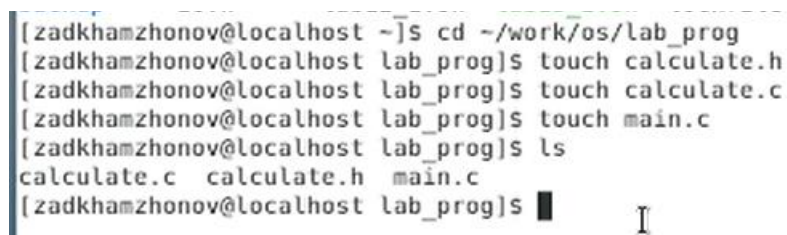
Ход работы:

1. В домашнем каталоге создал подкаталог ~/work/os/lab_prog.



```
zadkhamzhonov@localhost:~/work/os/lab_prog
Файл  Правка  Вид  Поиск  Терминал  Справка
[zadkhamzhonov@localhost ~]$ mkdir work/os/lab_prog
mkdir: невозможно создать каталог «work/os/lab_prog»: Нет такого файла или каталога
[zadkhamzhonov@localhost ~]$ mkdir work
[zadkhamzhonov@localhost ~]$ cd work
[zadkhamzhonov@localhost work]$ mkdir os
[zadkhamzhonov@localhost work]$ cd os
[zadkhamzhonov@localhost os]$ mkdir lab_prog
[zadkhamzhonov@localhost os]$ cd lab_prog
[zadkhamzhonov@localhost lab_prog]$ cd
[zadkhamzhonov@localhost ~]$ ls
0.txt      conf.txt  lab08     lab12_2.sh  lab13_3.sh  locklife  reports  Загрузки
9l.tar     cprog    lab11_2.sh  lab12_3.sh  lab13.sh    may       ski.places  Изображения
9.tar      feathers  lab11_3.sh  lab12.c     lab8         may1      text.txt   Музыка
abc1       file.txt  lab11_4.sh  lab12.sh    laba08       monthly   work       Общедоступные
australia  hg        lab11.sh    lab12.txt   laboratory   my_os     Видео     Рабочий стол
backup     io.h      lab12_1.sh  lab13_2.sh  lockfile     play      Документы  Шаблоны
```

2. Создал в нём файлы: calculate.h, calculate.c, main.c. Это примитивнейший калькулятор, способный складывать, вычитать, умножать, делить, возводить число в степень, вычислять квадратный корень, вычислять sin, cos, tan. При запуске он запрашивает первое число, операцию, второе число. После этого программа выводит результат и останавливается.



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

Реализация функций калькулятора в файле calculate.h:

Открыть



*calculate.c
~/work/os/lab_prog

Сохранить









```
#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("Second term: ");
        scanf("%f", &SecondNumeral);
        return(Numeral + SecondNumeral);
    }
    else if(strncmp(Operation, "-", 1) == 0)
    {
        printf("Subtrahend: ");
        scanf("%f", &SecondNumeral);
        return(Numeral - SecondNumeral);
    }
    else if(strncmp(Operation, "*", 1) == 0)
    {
        printf("Factor: ");
        scanf("%f", &SecondNumeral);
        return(Numeral * SecondNumeral);
    }
    else if(strncmp(Operation, "/", 1) == 0)
    {
        printf("Divisor: ");
        scanf("%f", &SecondNumeral);
        if(SecondNumeral == 0)
        {
            printf("Error: division entered by zero!");
            return(HUGE_VAL);
        }
    }

    printf("Error: division entered by zero!");
    return(HUGE_VAL);
}
else
    return(Numeral / SecondNumeral);
}
else if(strncmp(Operation, "pow", 3) == 0)
{
    printf("Degree: ");
    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
{
    printf("Incorrectly entered action");
    return(HUGE_VAL);
}
}
```

С ▾ Ширина табуляции: 8

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

```
Открыть  *calculate.h  Сохранить  -  □  ×
~/work/os/lab_prog

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

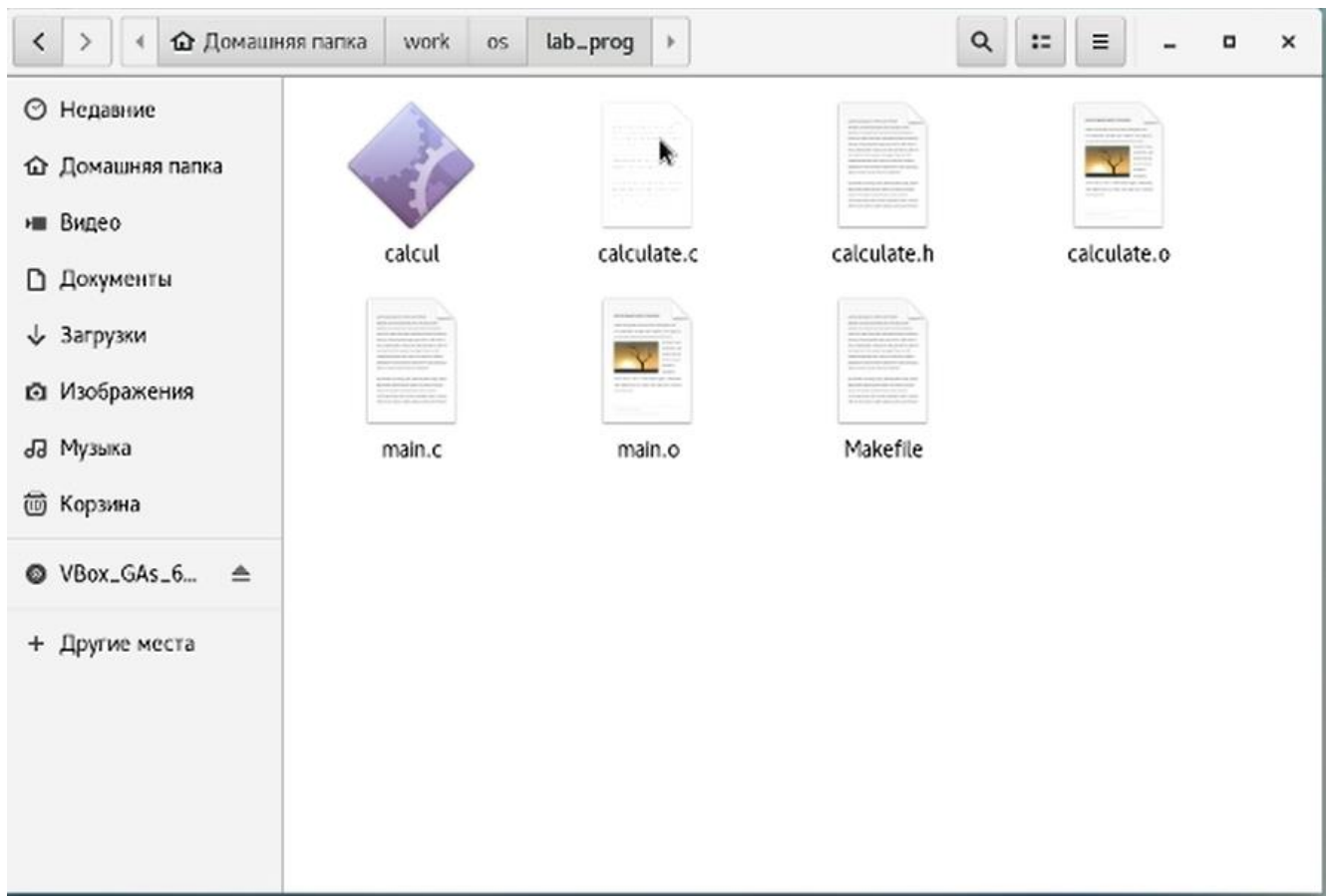
Основной файл main.c, реализующий интерфейс пользователя к калькулятору:

```
Открыть  *main.c  Сохранить  -  □  ×
~/work/os/lab_prog

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

3.Выполнил компиляцию программы посредством gcc:

```
[zadkhamzhonov@localhost lab_prog]$ gcc -c calculate.c
calculate.c: В функции «Calculate»:
calculate.c:56:14: ошибка: «HIGE_VAL» undeclared (first use in this function)
    return(HIGE_VAL);
               ^
calculate.c:56:14: замечание: each undeclared identifier is reported only once for each function it appears in
[zadkhamzhonov@localhost lab_prog]$ gcc -c calculate.c
[zadkhamzhonov@localhost lab_prog]$ gcc -c main.c
main.c:2:23: фатальная ошибка: calculate.h: Нет такого файла или каталога
#include <calculate.h>
                        ^
компиляция прервана.
[zadkhamzhonov@localhost lab_prog]$ gcc -c main.c
main.c: В функции «main»:
main.c:10:16: ошибка: «Nuneral» undeclared (first use in this function)
    scanf("%f", &Nuneral);
               ^
main.c:10:16: замечание: each undeclared identifier is reported only once for each function it appears in
[zadkhamzhonov@localhost lab_prog]$ gcc -c main.c
[zadkhamzhonov@localhost lab_prog]$ gcc calculate.o main.o -o calcul -lm
[zadkhamzhonov@localhost lab_prog]$
```



4. Исправил синтаксические ошибки.

5. Создал Makefile.

```
Makefile
~/work/os/lab_prog

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 *~
```

В содержании файла указаны флаги компиляции, тип компилятора и файлы, которые должен собрать сборщик.

6. С помощью gdb выполнил отладку программы calcul (перед использованием gdb исправил Makefile):

– запустил отладчик GDB, загрузив в него программу для отладки: `gdb ./calcul`

– для запуска программы внутри отладчика ввел команду `run`


```

[zadkhamzhonov@localhost lab_prog]$ gdb ./calcul
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-120.el7
Copyright (C) 2013 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/zadkhamzhonov/work/os/lab_prog/calcul...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/zadkhamzhonov/work/os/lab_prog/./calcul
Numeral: 3
Operation (+,-,*,/,pow,sqrt,sin,cos,tan): *
Factor: 4
&6.2f
[Inferior 1 (process 4698) exited normally]
Missing separate debuginfos, use: debuginfo-install glibc-2.17-317.el7.x86_64
(gdb) list
No symbol table is loaded. Use the "file" command.
(gdb) run
Starting program: /home/zadkhamzhonov/work/os/lab_prog/./calcul
Numeral: 3
Operation (+,-,*,/,pow,sqrt,sin,cos,tan): *
Factor: 4
&6.2f
[Inferior 1 (process 4710) exited normally]
(gdb)

```

7. С помощью утилиты splint проанализировал коды файлов calculate.c и main.c (так как у меня не сработал splint, сделал это в другом компьютере).

```

calculate.h:4:30: Function parameter op 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:6:30: Function parameter op declared as manifest array (size
        constant is meaningless)
calculate.c: (in function solve)
calculate.c:12:3: Return value (type int) ignored: scanf("%f", &num2)
    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:16:3: Return value (type int) ignored: scanf("%f", &num2)
calculate.c:20:3: Return value (type int) ignored: scanf("%f", &num2)
calculate.c:24:3: Return value (type int) ignored: scanf("%f", &num2)
calculate.c:25:6: Dangerous equality comparison involving float types:
    num2 != 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:29:11: Return value type double does not match declared type float:
    HUGE_VAL
    To allow all numeric types to match, use +relaxtypes.
calculate.c:32:46: Return value type double does not match declared type float:
    (sqrt(num1))
calculate.c:33:45: Return value type double does not match declared type float:
    (sin(num1))
calculate.c:34:45: Return value type double does not match declared type float:
    (cos(num1))
calculate.c:35:45: Return value type double does not match declared type float:
    (tan(num1))
calculate.c:39:10: Return value type double does not match declared type float:
    HUGE_VAL
Finished checking --- 13 code warnings

```

Вывод:

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