

Лабораторная работа 13

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

Извекова Мария

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	создание файлов	6
2.2	calculate.c	7
2.3	calculate.h	7
2.4	main.c	8
2.5	компиляция	8
2.6	makefile	8
2.7	запуск	9
2.8	list	9

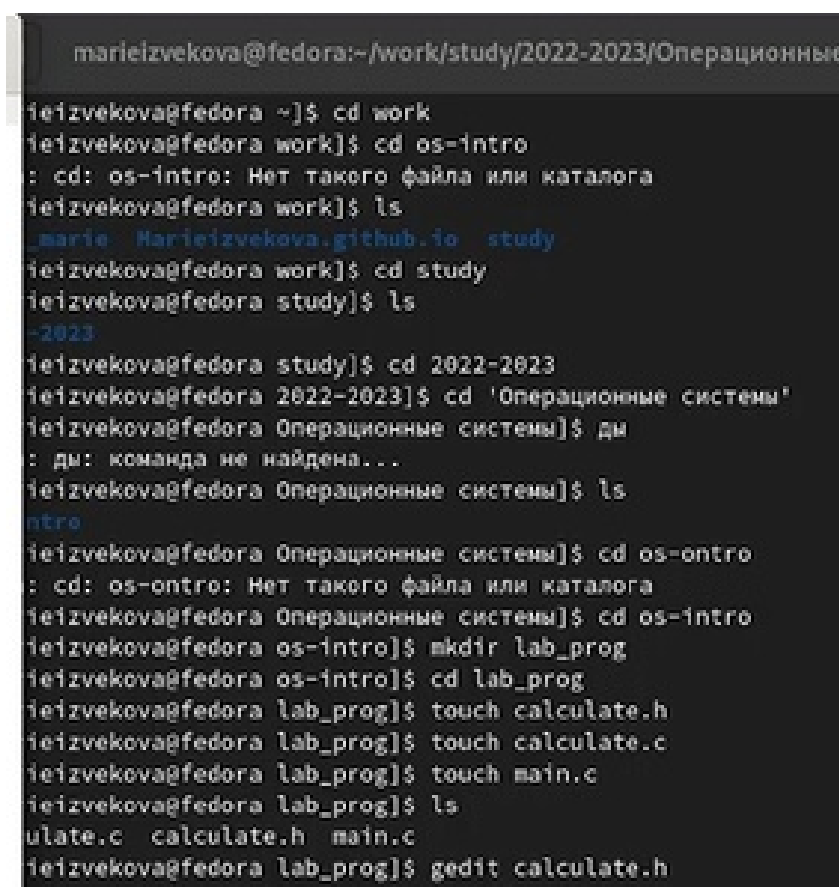
Список таблиц

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 Выполнение лабораторной работы

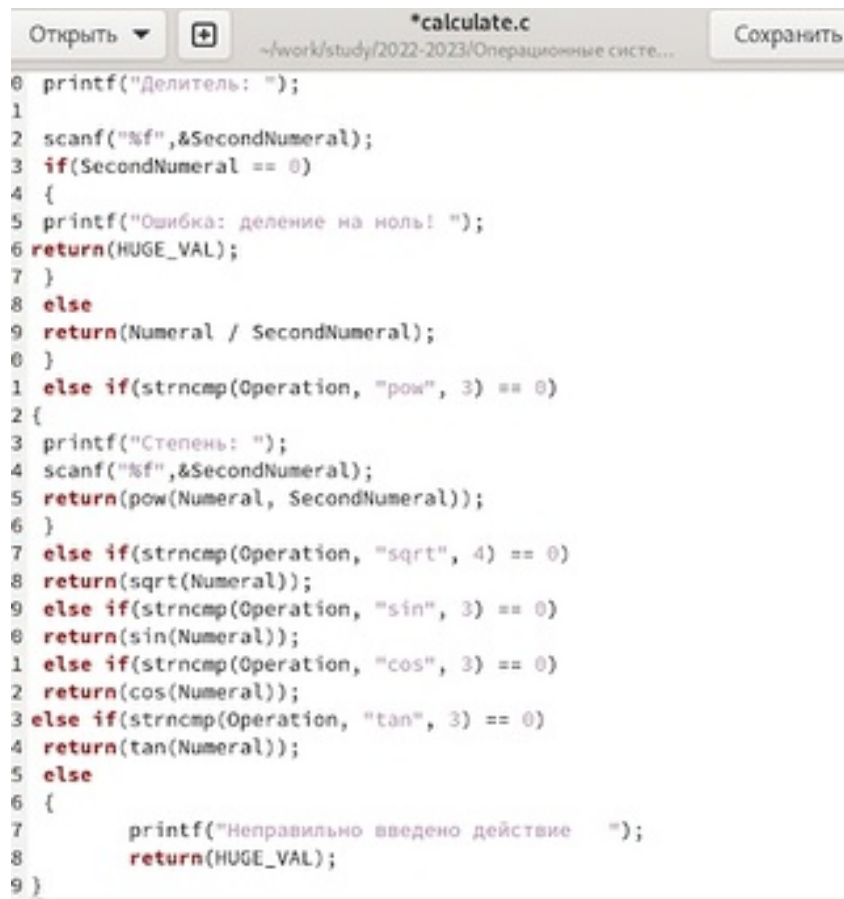
1. создаем новую папку lab_prog и в нем три файла: calculate.h, calculate.c, main.c.



```
marieizvekova@fedora:~/work/study/2022-2023/Операционные системы$  
marieizvekova@fedora ~]$ cd work  
marieizvekova@fedora work]$ cd os-intro  
: cd: os-intro: Нет такого файла или каталога  
marieizvekova@fedora work]$ ls  
_marie Marieizvekova.github.io study  
marieizvekova@fedora work]$ cd study  
marieizvekova@fedora study]$ ls  
-2023  
marieizvekova@fedora study]$ cd 2022-2023  
marieizvekova@fedora 2022-2023]$ cd 'Операционные системы'  
marieizvekova@fedora Операционные системы]$ дм  
: дм: команда не найдена...  
marieizvekova@fedora Операционные системы]$ ls  
ntro  
marieizvekova@fedora Операционные системы]$ cd os-ontro  
: cd: os-ontro: Нет такого файла или каталога  
marieizvekova@fedora Операционные системы]$ cd os-intro  
marieizvekova@fedora os-intro]$ mkdir lab_prog  
marieizvekova@fedora os-intro]$ cd lab_prog  
marieizvekova@fedora lab_prog]$ touch calculate.h  
marieizvekova@fedora lab_prog]$ touch calculate.c  
marieizvekova@fedora lab_prog]$ touch main.c  
marieizvekova@fedora lab_prog]$ ls  
ulate.c calculate.h main.c  
marieizvekova@fedora lab_prog]$ gedit calculate.h
```

Рис. 2.1: создание файлов

2. в первый файл мы вставляем программу для реализаций функций калькулятора



```

0 printf("Делитель: ");
1
2 scanf("%f",&SecondNumeral);
3 if(SecondNumeral == 0)
4 {
5     printf("Ошибка: деление на ноль! ");
6     return(HUGE_VAL);
7 }
8 else
9     return(Numeral / SecondNumeral);
0 }
1 else if(strncmp(Operation, "pow", 3) == 0)
2 {
3     printf("Степень: ");
4     scanf("%f",&SecondNumeral);
5     return(pow(Numeral, SecondNumeral));
6 }
7 else if(strncmp(Operation, "sqrt", 4) == 0)
8     return(sqrt(Numeral));
9 else if(strncmp(Operation, "sin", 3) == 0)
0     return(sin(Numeral));
1 else if(strncmp(Operation, "cos", 3) == 0)
2     return(cos(Numeral));
3 else if(strncmp(Operation, "tan", 3) == 0)
4     return(tan(Numeral));
5 else
6 {
7     printf("Неправильно введено действие ");
8     return(HUGE_VAL);
9 }

```

Рис. 2.2: calculate.c

3. во второй файл мы вставляем программу для формата вызова функции



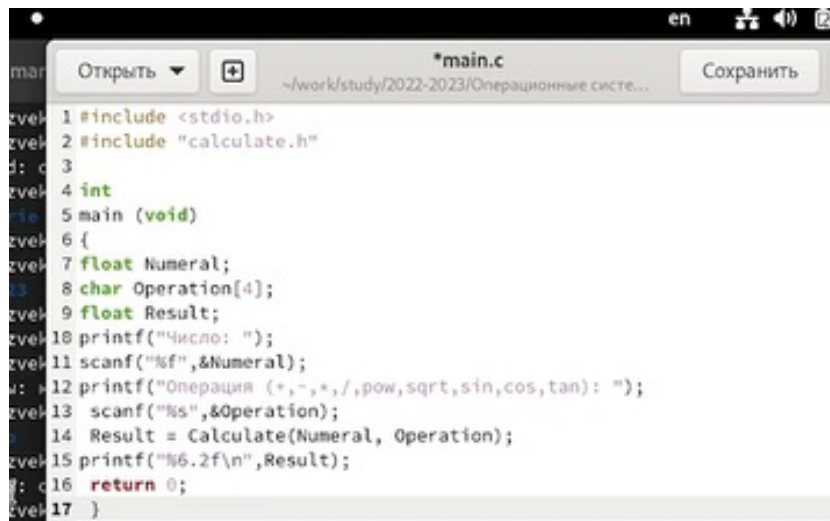
```

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

```

Рис. 2.3: calculate.h

4. в третий файл мы вставляем программу для реализации интерфейса



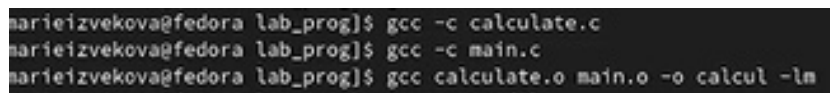
```

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("Число: ");
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. 2.4: main.c

5. далее с помощью следующих команд мы проводим компиляцию



```

marieizvekova@fedora lab_prog]$ gcc -c calculate.c
marieizvekova@fedora lab_prog]$ gcc -c main.c
marieizvekova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm

```

Рис. 2.5: компиляция

6. далее создаем новый файл makefile и вставляем следующую команду



```

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

```

Рис. 2.6: makefile

7. запускаем программу с помощью отладчика gdb ./calcul и команды run

```
Using host libthread_db library "/lib64/libthread_db.so.1"
Число: 4
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Множитель: 6
24.00
[Inferior 1 (process 3844) exited normally]
(gdb) run
Starting program: /home/marieizvekova/work/study/2022-intro/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1"
Число: 9
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): sin
0.41
```

Рис. 2.7: запуск

8. с помощью команды list можем посмотреть весь код программы в терминале

```
[Inferior 1 (process 3844) exited normally]
(gdb) list
Downloading source file /usr/src/debug/glibc-2.35-4.fc36.x86_64/elf/sofini.c...
1      /* Terminate the frame unwind info section with a 4byte 0 as a sentinel;
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__ = 0;
6      __attribute__((used, section(".eh_frame")))
7      ui32 __FRAME_END__ = 0;
(gdb) list 13-15
```

Рис. 2.8: list

3 Выводы

Мы 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

Список литературы