

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

Световидова Полина Михайловна

Российский Университет Дружбы Народов

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.

Выполнение лабораторной работы

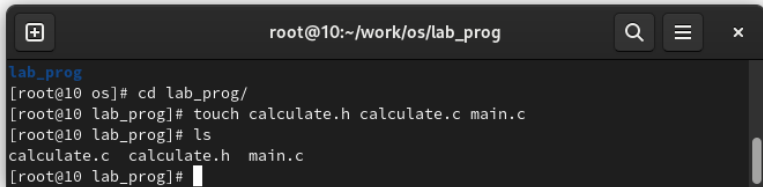
В домашнем каталоге создаю подкаталог `~/work/os/lab_prog`.

```
[root@10 ~]# cd work
[root@10 work]# cd os
-bash: cd: os: Нет такого файла или каталога
[root@10 work]#
[root@10 work]# mc

[root@10 work]# mkdir os
[root@10 work]# cd os
[root@10 os]# mkdir lab_prog
[root@10 os]# ls
lab_prog
[root@10 os]#
```

Рис. 1: lab_prog

Создаю в нём файлы `calculate.h`, `calculate.c` и `main.c`

A terminal window with a dark background. The title bar shows 'root@10:~/work/os/lab_prog' and standard window controls (search, menu, close). The terminal content shows a sequence of commands: 'cd lab_prog/', 'touch calculate.h calculate.c main.c', and 'ls', resulting in the output 'calculate.c calculate.h main.c'.

```
lab_prog
[root@10 os]# cd lab_prog/
[root@10 lab_prog]# touch calculate.h calculate.c main.c
[root@10 lab_prog]# ls
calculate.c calculate.h main.c
[root@10 lab_prog]#
```

Рис. 2: создание файлов

Реализация функций калькулятора в файле `calculate.h`

```
1 //////////////////////////////////////////////////
2 // calculate.c
3
4 #include <stdio.h>
5 #include <math.h>
6 #include <string.h>
7 #include "calculate.h"
8
9 float
10 Calculate(float Numeral, char Operation[4])
11 {
12     float SecondNumeral;
13     if(strcmp(Operation, "+", 1) == 0)
14     {
15         printf("Добавляемое: ");
16         scanf("%f", &SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strcmp(Operation, "-", 1) == 0)
20     {
21         printf("Вычитаемое: ");
22         scanf("%f", &SecondNumeral);
23         return(Numeral - SecondNumeral);
24     }
25     else if(strcmp(Operation, "*", 1) == 0)
26     {
27         printf("Умножитель: ");
28         scanf("%f", &SecondNumeral);
29         return(Numeral * SecondNumeral);
30     }
31     else if(strcmp(Operation, "/", 1) == 0)
32     {
33         printf("Делитель: ");
34         scanf("%f", &SecondNumeral);
35         if(SecondNumeral == 0)
36         {
37             printf("Ошибка: деление на ноль! ");
38             return(HUGE_VAL);
39         }
40         else
41             return(Numeral / SecondNumeral);
42     }
43     else if(strcmp(Operation, "pow", 3) == 0)
44     {
45         printf("Степень: ");
46         scanf("%f", &SecondNumeral);
47         return(pow(Numeral, SecondNumeral));
48     }
```

Рис. 3: calculate.h

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

```
1 //////////////////////////////////////////////////
2 2 // calculate.h
3 3
4 4 #ifndef CALCULATE_H_
5 5 #define CALCULATE_H_
6 6
7 7 float Calculate(float Numeral, char Operation[4]);
8 8
9 9 #endif /*CALCULATE_H_*/
```

Рис. 4: calculate.h

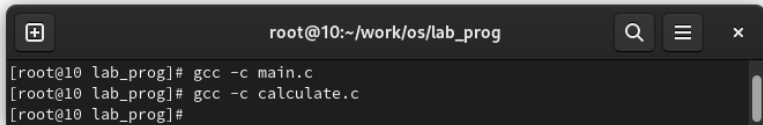
Основной файл `main.c`, реализующий интерфейс пользователя калькулятору



```
1 //////////////////////////////////////////////////
2 // main.c
3
4 #include <stdio.h>
5 #include "calculate.h"
6
7 int
8 main (void)
9 {
10     float Numeral;
11     char Operation[4];
12     float Result;
13     printf("Число: ");
14     scanf("%f",&Numeral);
15     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
16     scanf("%s",&Operation);
17     Result = Calculate(Numeral, Operation);
18     printf("%.2f\n",Result);
19     return 0;
20 }
```

Рис. 5: main.c

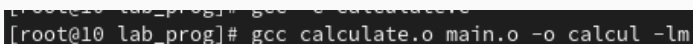
Далее выполняю компиляцию программы посредством gcc



A terminal window with a dark background. The title bar shows 'root@10:~/work/os/lab_prog' and standard window controls (search, menu, close). The terminal content shows three lines of commands and their prompts:

```
[root@10 lab_prog]# gcc -c main.c
[root@10 lab_prog]# gcc -c calculate.c
[root@10 lab_prog]#
```

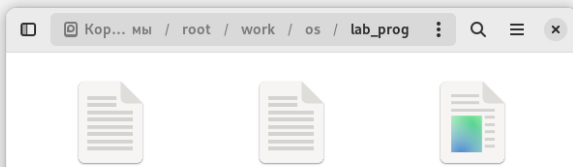
Рис. 6: компиляция программы



A terminal window showing a single line of a command and its prompt:

```
[root@10 lab_prog]# gcc calculate.o main.o -o calcul -lm
```

Рис. 7: .



Создаю Makefile



```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10 gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13 gcc -c calculate.c $(CFLAGS)
14
15 main.o: main.c calculate.h
16 gcc -c main.c $(CFLAGS)
17
18 clean:
19 rm calcul *.o *~
20
```

Рис. 9: код makefile

Далее исправляю код для Makefile

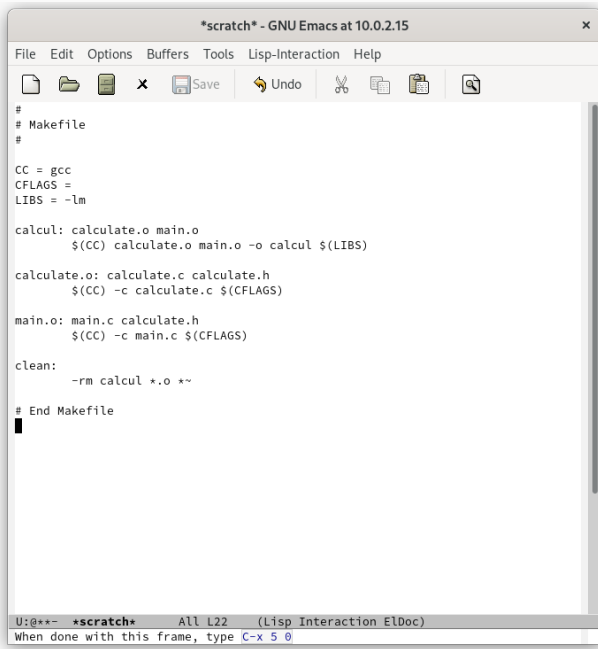
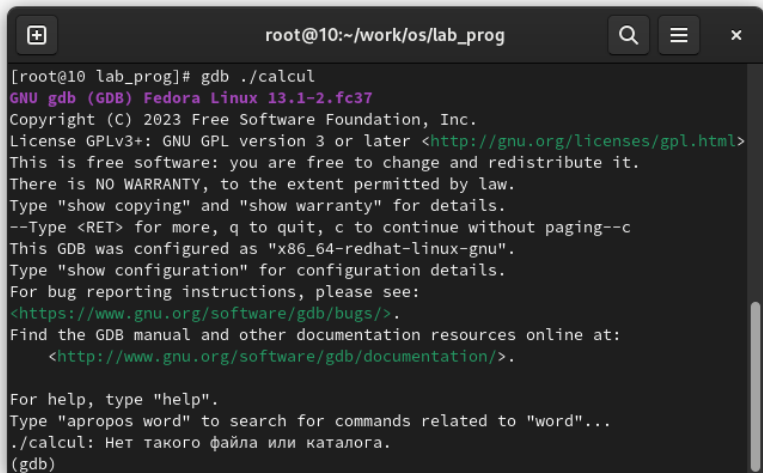


Рис. 10: исправление кода

С помощью gdb выполняю отладку программы calcul и запускаю отладчик GDB, загрузив в него программу для отладки. Для запуска ввожу команду run

A terminal window with a dark background. The title bar shows the user 'root' at host '10' in directory '~/work/os/lab_prog'. The terminal content shows the execution of 'gdb ./calcul' and the subsequent GDB startup messages, including copyright information and help instructions. The prompt '(gdb)' is visible at the bottom.

```
[root@10 lab_prog]# gdb ./calcul
GNU gdb (GDB) Fedora Linux 13.1-2.fc37
Copyright (C) 2023 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.
--Type <RET> for more, q to quit, c to continue without paging--
This GDB was configured as "x86_64-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
./calcul: Нет такого файла или каталога.
(gdb)
```

Рис. 11: запуск отладчика

Выводы

В ходе выполнения лабораторной работы №13 я приобрела простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями