

Лабораторная работа №13

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

Матвеева Анастасия Сергеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
	Список литературы	12

Список иллюстраций

2.1	...	6
2.2	создание	6
2.3	calculate.c	7
2.4	calculate.h	8
2.5	main.c	8
2.6	...	8
2.7	Makefile	9
2.8	...	10

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. В домашнем каталоге создайте подкаталог ~/work/os/lab_prog.

```
asmatveeva1@dk8n78 ~/work/os $ mkdir lab_prog
asmatveeva1@dk8n78 ~/work/os $ cd lab_prog
asmatveeva1@dk8n78 ~/work/os/lab_prog $
```

Рис. 2.1: ...

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. Реализация функций калькулятора в файле calculate.c

```
asmatveeva1@dk8n78 ~/work/os/lab_prog $ touch calculate.h
asmatveeva1@dk8n78 ~/work/os/lab_prog $ touch calculate.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $ touch main.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $ ls
calculate.c calculate.h main.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $
```

Рис. 2.2: создание

```

////////////////////////////////////
// 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);
    }
    else if(strncmp(Operation, "*", 1) == 0)
    {
        printf("Множитель: ");
        scanf("%f",&SecondNumeral);
        return(Numeral * SecondNumeral);
    }
    else if(strncmp(Operation, "/", 1) == 0)
    {
        printf("Делитель: ");
        scanf("%f",&SecondNumeral);
        if(SecondNumeral == 0)
        {
            printf("Ошибка: деление на ноль! ");
            return(HUGE_VAL);
        }
        else
            return(Numeral / SecondNumeral);
    }
    else if(strncmp(Operation, "pow", 3) == 0)
    {
        printf("Степень: ");
        scanf("%f" &SecondNumeral);
    }
}

```

Рис. 2.3: calculate.c

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

```

#ifndef CALCULATE_H_
#define CALCULATE_H_

float Calculate(float Numeral, char Operation[4]);

#endif /*CALCULATE_H_*/

```

Рис. 2.4: calculate.h

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

```

#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;
}

```

Рис. 2.5: main.c

5. Выполните компиляцию программы посредством gcc:

```

asmatveeva1@dk8n78 ~/work/os/lab_prog $ gcc -c calculate.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $ gcc -c main.c
main.c: В функции «main»:
main.c:12:10: предупреждение: формат «%s» ожидает аргумент типа «char *», но аргумент 2 имеет тип «char (*)[4]» [-Wformat=]
   12 |     scanf("%s",&Operation);
      |           ^~ ~~~~~
      |           | |
      |           | char (*)[4]
      |           char *
asmatveeva1@dk8n78 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
asmatveeva1@dk8n78 ~/work/os/lab_prog $

```

Рис. 2.6: ...

6. Создайте Makefile со следующим содержанием. С помощью gdb выполните отладку программы calcul: – Запустите отладчик GDB, загрузив в него программу для отладки – Для запуска программы внутри отладчика введите команду run:

```
asmatveeva1@dk8n78 ~/work/os/lab_prog $ touch Makefile
asmatveeva1@dk8n78 ~/work/os/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 12.1 vanilla) 12.1
Copyright (C) 2022 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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"...
Reading symbols from ./calcul...
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/a/s/asmatveeva1/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib64/libthread_db.so.1".
Число: 23
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 55
-32.00
[Inferior 1 (process 7290) exited normally]
(gdb) list
No symbol table is loaded.  Use the "file" command.
(gdb)
No symbol table is loaded.  Use the "file" command.
(gdb) Quit
(gdb)
[1]+  Остановлен   gdb ./calcul
asmatveeva1@dk8n78 ~/work/os/lab_prog $
```

Рис. 2.7: Makefile

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

```
asmatveeva1@dk8n78 ~/work/os/lab_prog $ chmod +x calculate.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $ chmod +x calculate.h
asmatveeva1@dk8n78 ~/work/os/lab_prog $ chmod +x main.c
asmatveeva1@dk8n78 ~/work/os/lab_prog $ Makefile
bash: Makefile: команда не найдена
asmatveeva1@dk8n78 ~/work/os/lab_prog $ chmod +x Makefile
asmatveeva1@dk8n78 ~/work/os/lab_prog $
```

Рис. 2.8: ...

3 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

Список литературы