

Лабораторная работа

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

Панченко Д. Д.

1 мая 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Панченко Денис Дмитриевич
- Студент 1 курса факультета физико-математических наук.
- Российский университет дружбы народов
- derenchikde@gmail.com

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 создадим подкаталог ~/work/os/lab_prog.

```
[ddpanchenko@ddpanchenko ~]$ mkdir ~/work/os/lab_prog
```

Рис. 1: Создание подкаталога

Создадим в нём файлы: calculate.h, calculate.c, main.c.

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

Рис. 2: Создание файлов

Реализуем функцию калькулятора в файле calculate.c.

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

Рис. 3: Реализация функции

Реализуем интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора.

```
// //////////////////////////////////////  
// 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: ");
```

Реализуем основной файл main.c, реализующий интерфейс пользователя к калькулятору.

```
////////////////////////////////////  
// 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("%.2f\n",Result);  
    return 0;  
}
```

Рис. 5: Реализация основного файла

Выполним компиляцию программы посредством gcc.

```
[ddpanchenko@ddpanchenko lab_prog]$ gcc -c calculate.c  
[ddpanchenko@ddpanchenko lab_prog]$ gcc -c main.c  
[ddpanchenko@ddpanchenko lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

Рис. 6: Компиляция программы

Создадим Makefile со следующим содержанием.

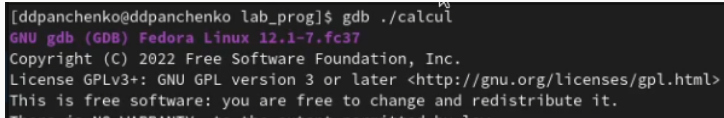
```
[ddpanchenko@ddpanchenko lab_prog]$ touch Makefile
```

Рис. 7: Создание файла

```
#  
# Makefile  
#  
  
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 *~  
  
# End Makefile
```

Рис. 8: Содержание файла

С помощью gdb выполним отладку программы calcul.

A terminal window with a dark background. The prompt is [ddpanchenko@ddpanchenko lab_prog]\$ and the command entered is gdb ./calcul. The output shows the GNU gdb version (12.1-7.fc37), the copyright notice for the Free Software Foundation, Inc. (2022), the license (GPLv3+), and a disclaimer of warranty.

```
[ddpanchenko@ddpanchenko lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora Linux 12.1-7.fc37
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.
```

Рис. 9: Запуск

```
(gdb) run
Starting program: /home/ddpanchenko/work/os/lab_prog/calcul
[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: 10
    2.00
[Inferior 1 (process 5571) exited normally]
```

Рис. 10: Запуск программы

```
(gdb) list
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__[1]
6          __attribute__ ((used, section (".eh_frame")))
7          = { 0 };
```

Рис. 11: Просмотр кода


```
(gdb) list 12,15  
Line number 12 out of range; sofini.c has 7 lines.
```

Рис. 12: Просмотр строк кода

```
(gdb) list calculate.c:20,29  
No source file named calculate.c
```

Рис. 13: Просмотр строк неосновного кода

```
(gdb) list calculate.c:20,27  
No source file named calculate.c.  
(gdb) break  
No default breakpoint address now.
```

Рис. 14: Установка точки останова

```
(gdb) info breakpoints
```

Рис. 15: Вывод информации о точке

```
(gdb) run
Starting program: /home/ddpanchenko/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: backtrace
    5.00
[Inferior 1 (process 5600) exited normally]
```

Рис. 16: Запуск программы

```
(gdb) print Numeral
```

Рис. 17: Значение переменной

```
(gdb) display Numeral
```

Рис. 18: Сравнение

```
(gdb) info breakpoints  
No breakpoints or watchpoints.  
(gdb) delete 1  
No breakpoint number 1.
```

Рис. 19: Удаление точки останова

Вывод

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