

Лабораторная работа №13

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

Аскеров А.Э.

05 мая 2023

Российский университет дружбы народов, Москва, Россия

Вступление

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

1. В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`.

```
[aeaskerov@fedora os]$ mkdir ~/work/os/lab_prog
```

Рис. 1: Создание подкаталога `lab_prog`

2. Создадим в нём файлы: `calculate.h`, `calculate.c`, `main.c` и запишем в них код, приведённый в инструкции.

```
[aeaskerov@fedora lab_prog]$ touch calculate.h calculate.c main.c
```

Рис. 2: Создаём файлы для калькулятора

3. Выполним компиляцию программы посредством gcc.

```
[aeaskerov@fedora lab_prog]$ gcc -c calculate.c -g  
[aeaskerov@fedora lab_prog]$ gcc -c main.c -g  
[aeaskerov@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

Рис. 3: Компиляция программы

4. Исправим ошибки. Допишем `-g`.

Задание 5

5. Создадим Makefile.

Первые три строки указывают компилятор, флаг и библиотеки для создания файлов в последующих разделах кода. Clean отвечает за удаление объектных файлов.

```
#
# Makefile
#

CC = gcc
CFLAGS = -g
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. 4: Makefile

6. С помощью gdb выполним отладку программы calcul.

- Запустим отладчик GDB, загрузив в него программу для отладки

```
[aeaskerov@fedora lab_prog]$ gdb ./calcul
```

Рис. 5: Запуск отладчика

- Для запуска программы внутри отладчика введём команду run

```
(gdb) run
```

Рис. 6: Запуск программы внутри отладчика

- Для постраничного (по 9 строк) просмотра исходного код используем команду list

```
(gdb) list
1  //////////////////////////////////////////////////
2  // main.c
3
4  #include <stdio.h>
5  #include "calculate.h"
6
7  int
8  main (void)
9  {
10     float Numeral;
```

Рис. 7: Постраничный просмотр исходного кода

Задание 6

- Для просмотра строк с 12 по 15 основного файла используем list с параметрами

```
(gdb) list 12,15
12      float Result;
13      printf("Введите: ");
14      scanf("%f",&Numeral);
15      printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb)
```

Рис. 8: Просмотр строк с 12 по 15 основного файла

- Для просмотра определённых строк не основного файла используем list с параметрами

```
(gdb) list calculate.c:20,27
20      {
21          printf("Вычитаемое: ");
22          scanf("%f",&SecondNumeral);
23          return Numeral - SecondNumeral;
24      }
25      else if (strcmp Operation, "+" < 1) == 0)
26      {
27          printf("Множитель: ");
(gdb)
```

Рис. 9: Просмотр определённых строк не основного файла

- Установим точку останова в файле calculate.c на строке номер 21

```
(gdb) break 21  
Breakpoint 1 at 0x40120f: file calculate.c, line 21.
```

Рис. 10: Точка останова на строке 21

- Выведем информацию об имеющихся в проекте точках останова

```
(gdb) info breakpoints  
Num   Type       Disp Enb Address            What  
1      breakpoint keep y  0x000000000040120f in calculate  
                                at calculate.c:21
```

Рис. 11: Информация об имеющихся в проекте точках останова

Задание 6

- Запустим программу внутри отладчика и убедимся, что программа остановится в момент прохождения точки останова

```
(gdb) run
Starting program: /home/aeaskerov/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): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf24 "--") at calculate.c:21
21      printf("Вычисляемое: ");
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffdf24 "--") at calculate.c:21
#1 0x000000000040140b in main () at main.c:17
(gdb)
```

Рис. 12: Проверка работы точки останова

- Посмотрим, чему равно на этом этапе значение переменной Numeral.

```
(gdb) print Numeral
$1 = 5
```

Рис. 13: Значение переменной Numeral

- Сравним с результатом вывода на экран после использования команды

```
(gdb) display Numeral  
1: Numeral = 5
```

Рис. 14: Другой способ отображения

- Уберём точки останова

```
(gdb) info breakpoints  
Num   Type             Disp Enb Address            What  
1      breakpoint       keep y   0x000000000040119f in calculate  
                                     at calculate.c:21  
breakpoint already hit 1 time  
(gdb) delete 1
```

Рис. 15: Удаление точек останова

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

```
[aeaskerov@fedora lab_prog]$ splint calculate.c
Splint 3.1.2 --- 22 Jan 2022

calculate.h:7:37: Function parameter Operation 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:10:31: Function parameter Operation declared as manifest array
                    (size constant is meaningless)
```

Рис. 16: Анализ программы calculate.c

```
[aeaskerov@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 22 Jan 2022

calculate.h:7:37: Function parameter Operation 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)
main.c: (in function main)
main.c:14:3: Return value (type int) ignored: scanf("%f", &Num...
```

Рис. 17: Анализ программы main.c

Заключение

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