

Отчет по лабораторной работе №14

Перелыгин Сергей Викторович

Цель работы

Цель лабораторной 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

Выполнение лабораторной работы

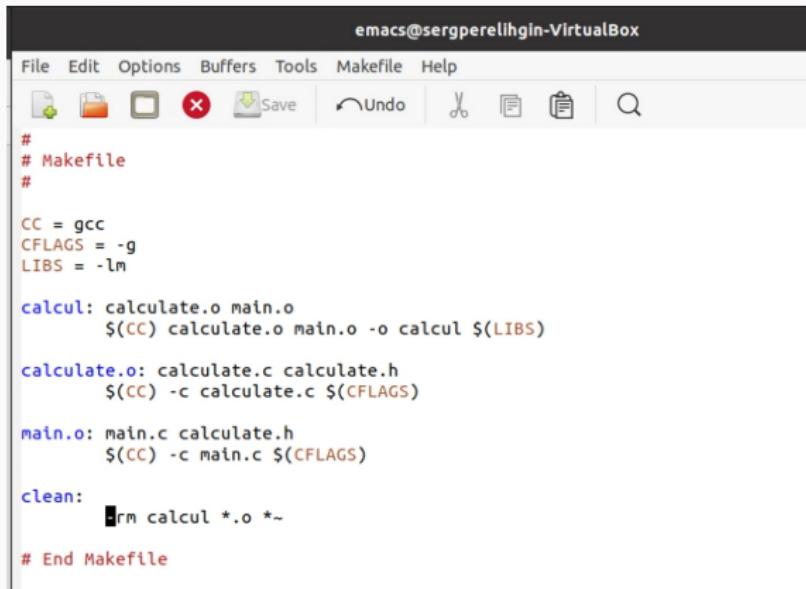
В ходе выполнения ЛР я создал примитивный калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. Для этого понадобилось заранее создать три файла (calculate.h, calculate.c, main.c)

```
bash: cd: /work/os/lab_prog: no such file or directory
sergperelthgin@sergperelthgin-VirtualBox:~/work/os/lab_prog$ touch calculate.h calculate.c main.c
sergperelthgin@sergperelthgin-VirtualBox:~/work/os/lab_prog$ ls
calculate.c calculate.h main.c
sergperelthgin@sergperelthgin-VirtualBox:~/work/os/lab_prog$
```

Рис. 1: Создание 3 файлов для калькулятора

Выполнение лабораторной работы

Затем я скомпилировал программу с помощью gcc. После этого я создал Makefile и при помощи отладчика GDB выполнил отладку программы.



The screenshot shows an Emacs window titled "emacs@sergperelihgin-VirtualBox". The buffer contains a Makefile with the following content:

```
#  
# Makefile  
  
CC = gcc  
CFLAGS = -g  
LIBS = -lm  
  
calcul: calculate.o main.o  
        $(CC) calculate.o main.o -o calcul $(LIBS)  
  
calculate.o: calculate.c calculate.h  
        $(CC) -c calculate.c $(CFLAGS)  
  
main.o: main.c calculate.h  
        $(CC) -c main.c $(CFLAGS)  
  
clean:  
        rm calcul *.o ~  
  
# End Makefile
```

Рис. 2: Makefile

```
(gdb) run
Starting program: /home/sergperelhgin/work/os/lab_prog/calcul
Число: 1
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 2
      3.00
[Inferior 1 (process 25187) exited normally]
(gdb)
```

Рис. 3: Запуск программы

Выполнение лабораторной работы

Также я научился устанавливать точки останова и удалять их.

```
(gdb) list calculate.c:20,27
20      {
21          printf("Вычитаемое: ");
22          scanf("%f", &SecondNumeral);
23          return(Numerical - SecondNumeral);
24      }
25      else if(strncmp(Operation, "*", 1) == 0)
26      {
27          printf("Множитель: ");
(gdb) break 21
Breakpoint 1 at 0x555555552dd: file calculate.c, line 21.
(gdb)
```

Рис. 4: Точка останова

```
(gdb) info breakpoints
Num      Type            Disp Enb Address          What
1      breakpoint    keep y  0x0000555555552dd in Calculate at calculate.c:21
      breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

Рис. 5: Удаление точки останова

Выполнение лабораторной работы

С помощью утилиты `splint` проанализировал коды файлов `calculate.c` и `main.c`, предварительно установив данную утилиту с помощью команд «`sudo apt update`» и «`sudo apt install splint`».

```
sergei@lhp15:~/sergei$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2018

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)
calculate.c: (in function Calculate)
calculate.c:16:7: Return value (type int) ignored: scanf("%f", &Sec...
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalint to inhibit warning)
calculate.c:22:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:10: Dangerous equality comparison involving float types:
    secondNumeral == 0
    Two real (float, double, long double) values are compared directly using
    == or != primitive. This may produce unexpected results since floating point
    representations are inexact. Instead, compare the difference to FLT_EPSILON
    or DBL_EPSILON. (Use -reallcompare to inhibit warning)
calculate.c:38:10: Return value type double does not match declared type float:
    (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:45:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:46:13: Return value type double does not match declared type float:
    (pow(Numerical, secondNumeral))
calculate.c:49:11: Return value type double does not match declared type float:
    (sqrt(Numerical))
calculate.c:51:11: Return value type double does not match declared type float:
    (sin(Numerical))
calculate.c:53:11: Return value type double does not match declared type float:
    (cos(Numerical))
calculate.c:55:11: Return value type double does not match declared type float:
    (tan(Numerical))
calculate.c:59:13: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings
```

Рис. 6: `splint calculate.c`

```
sergperelhgln@sergperelhgln-VirtualBox:~/work/os/lab_prog$ splint main.c
Splint 3.1.2 --- 20 Feb 2018

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...
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalint to inhibit warning)
main.c:16:3: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 3 code warnings
sergperelhgln@sergperelhgln-VirtualBox:~/work/os/lab_prog$
```

Рис. 7: splint main.c

Выводы

Выводы

В ходе выполнения данной лабораторной работы я приобрел простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

Спасибо за внимание!