

Лабораторная работа №14

Выполнил Гамаюнов Никита, 1032201719, НПМбд-01-20

Прагматика выполнения работы

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.

Задания

1. Создать файлы для корректной работы калькулятора
2. Изучить Makefile, исправить имеющийся шаблон
3. На практике воспользоваться gcc для отладки приложения
4. Проверить код с помощью splint

Процесс выполнения работы

1. Создал файлы: `calculate.h`, `calculate.c`, `main.c` (рисунки 2)

```
[negamayunov@negamayunov lab_prog]$ touch calculate.h; touch calculate.c; touch  
calculate.c  
[negamayunov@negamayunov lab_prog]$ touch main.c
```

Заполнил из готовыми текстами программ.

2. Выполнил компиляцию посредством gcc

```
[negamayunov@negamayunov ~]$ cd work/2020-2021/OperatingSystems/lab_prog  
[negamayunov@negamayunov lab_prog]$ gcc -c calculate.c  
[negamayunov@negamayunov lab_prog]$ gcc -c main.c  
[negamayunov@negamayunov lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

3. Создал и исправил текст Makefile

```
#  
## 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
```

4. Выполнил отладку в gdb

```
(gdb) run
Starting program: /home/negamayunov/work/2020-2021/OperatingSystems/lab_
alcul
Число: 4
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 5
-1.00
[Inferior 1 (process 3817) exited normally]
Missing separate debuginfos, use: debuginfo-install glibc-2.17-317.el7.x
(gdb) list
2      // main.c
3      #include <stdio.h>
4      #include "calculate.h"
5      int
6      main (void)
7      {
8      float Numeral;
9      char Operation[4];
10     float Result;
11     printf("Число: ");
(gdb) █
```

```
(gdb) list 12,15
12     scanf("%f",&Numeral);
13     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
14     scanf("%s",&Operation);
15     Result = Calculate(Numeral, Operation);
(gdb) list calculate.c:20,29
20     scanf("%f",&SecondNumeral);
21     return(Numeral - SecondNumeral);
22 }
23 else if(strncmp(Operation, "+", 1) == 0)
24 {
25     printf("Множитель: ");
26     scanf("%f",&SecondNumeral);
27     return(Numeral * SecondNumeral);
28 }
29 _ else if(strncmp(Operation, "/", 1) == 0)
```

5. С помощью утилиты splint проанализировал коды файлов calculate.c и main.c

```
SecondNumeral == 0
Two real (float, double, or long double) values are compared directly
== or != primitive. This may produce unexpected results since floating
representations are inexact. Instead, compare the difference to FLT_EPSILON
or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:36:7: Return value type double does not match declared type
(HUGE_VAL)
To allow all numeric types to match, use +relaxtypes.
calculate.c:44:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:45:7: Return value type double does not match declared type
(pow(Numeral, SecondNumeral))
calculate.c:48:7: Return value type double does not match declared type
(sqrt(Numeral))
calculate.c:50:7: Return value type double does not match declared type
(sin(Numeral))
calculate.c:52:7: Return value type double does not match declared type
(cos(Numeral))
calculate.c:54:7: Return value type double does not match declared type
(tan(Numeral))
calculate.c:58:7: Return value type double does not match declared type
(HUGE_VAL)

Finished checking --- 15 code warnings
[negamayunov@negamayunov lab_prog]$ splint main.c
```

Выводы

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