Лабораторная работа №13

Кочкарев “sakochkarev” Станислав

RUDN University

Цель работы

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

Задание

Написать программу и проанализировать ее код и выполнение.

Выполнение лабораторной работы

Выполнение лабораторной работы

Первым делом мы создали в домашнем каталоге подкаталог `~/work/os/lab_prog` командой `mkdir -p ~/work/os/lab_prog`.

```
sakochkarev@sakochkarev [15:18:42] [~]
-> % mkdir -p ~/work/os/lab_prog
```

Далее в нем были созданы три файла `calculate.h, calculate.c, main.c`.

```
sakochkarev@sakochkarev [15:20:29] [~]
-> % cd work/os/lab_prog
sakochkarev@sakochkarev [15:20:33] [~/work/os/lab_prog]
-> % touch calculate.h calculate.c main.c
```

В каждый из этих файлов были написаны соответствующие коды.

```c
////////////////////////////////////
// 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);
```

```
1  ////////////////////////////////////////
2  // calculate.h
3  #ifndef CALCULATE_H_
4  #define CALCULATE_H_
5
6  float Calculate(float Numeral, char Operation[4]);
7
8  #endif /*CALCULATE_H_*/
```

```c
////////////////////////////////////////
// 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("%6.2f\n",Result);
return 0;
}
```

После этого была выполнена компиляция данных файлов.

```
sakochkarev@sakochkarev [15:23:17] [~/work/os/lab_prog]
-> % gcc -c calculate.c
gcc -c main.c
gcc calculate.o main.o -o calcul -lm
main.c:13:73: warning: format specifies type 'char *' but the argument has type 'char (*)[4]' [-Wformat]
printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): "); scanf("%s",&Operation);
                                                        ~~  ^~~~~~~~~
1 warning generated.
```

Далее был создан `Makefile`, где был написан соответствующий код.

```
sakochkarev@sakochkarev [15:20:36] [~/work/os/lab_prog]
-> % touch Makefile
```

```
#
# Makefile #
CC = gcc
CFLAGS =
LIBS = -lm

calcul: calculate.o main.o
	gcc calculate.o main.o -o calcul -g $(LIBS)

calculate.o: calculate.c calculate.h
	gcc -c calculate.c $(CFLAGS) -g

main.o: main.c calculate.h
	gcc -c main.c $(CFLAGS) -g

clean:
	-rm calcul *.o *~

# End Makefile

```

Код содержит в себе переменные с используемым компилятором, а также флагами для компиляции. В нем находится четыре цели, все из которых, кроме clean выполняют сам процесс компиляции. Цель clean позволяет удалить результаты компиляции.

Далее, используя `gdb` была выполнена отладка программы `calcul`.

```
sakochkarev@sakochkarev [15:26:41] [~/work/os/lab_prog]
-> % sudo gdb ./calcul
Password:
GNU gdb (GDB) 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-apple-darwin21.3.0".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

--Type <RET> for more, q to quit, c to continue without paging--
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) list
1       ////////////////////////////////////
2       // main.c
3       #include <stdio.h>
4       #include "calculate.h"
5
6       int
7       main (void) {
8       float Numeral;
9       char Operation[4];
10      float Result;
(gdb)
```

В конце, используя утилиту splint, были проанализированы коды файлов calculate.c и main.c.

```
sakochkarev@sakochkarev [15:05:05] [~/work/os/lab_prog]
-> % splint calculate.c
Splint 3.1.2 --- 30 Oct 2021

calculate.h:6:37: 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:7:37: Function parameter Operation declared as manifest array (size
                    constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:11:9: 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:15:43: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:18:37: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:23:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:23:32: Dangerous equality comparison involving float types:
                     SecondNumeral == 0
  Two real (float, double, or 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 -realcompare to inhibit warning)
calculate.c:25:63: Return value type double does not match declared type float:
                     (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:32:29: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:63: Return value type double does not match declared type float:
                     (pow(Numeral, SecondNumeral))
calculate.c:35:7: Return value type double does not match declared type float:
                    (sqrt(Numeral))
calculate.c:37:7: Return value type double does not match declared type float:
                    (sin(Numeral))
calculate.c:39:7: Return value type double does not match declared type float:
                    (cos(Numeral))
calculate.c:41:7: Return value type double does not match declared type float:
                    (tan(Numeral))
calculate.c:43:74: Return value type double does not match declared type float:
                     (HUGE_VAL)

Finished checking --- 15 code warnings
```

```
sakochkarev@sakochkarev [15:05:09] [~/work/os/lab_prog]
-> % splint main.c
Splint 3.1.2 --- 30 Oct 2021

calculate.h:6:37: 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:12:1: 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:13:73: Format argument 1 to scanf (%s) expects char * gets char [4] *:
                &Operation
  Type of parameter is not consistent with corresponding code in format string.
  (Use -formattype to inhibit warning)
   main.c:13:70: Corresponding format code
main.c:13:62: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
```

Выводы

По выполнении данной лабораторной работы мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.