

Лабораторная работа № 13

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

Шулуужук Айраана Вячеславовна НПИбд-02-22

8 апреля 2023

Российский университет дружбы народов, Москва, Россия

- Шулуужук Айраана Вячеславовна
- НПИбд-02-22
- 1132221890
- Российский университет дружбы народов

Цели и задачи

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

Создадим нужный каталог и необходимые файлы

```
[avshuluuzhuk@fedora ~]$ cd work  
[avshuluuzhuk@fedora work]$ cd os  
[avshuluuzhuk@fedora os]$ mkdir lab_prog  
[avshuluuzhuk@fedora os]$ cd lab_prog  
[avshuluuzhuk@fedora lab_prog]$ touch calculate.h  
[avshuluuzhuk@fedora lab_prog]$ touch calculate.c  
[avshuluuzhuk@fedora lab_prog]$ touch main.c  
[avshuluuzhuk@fedora lab_prog]$
```

Рис. 1: создание файлов и каталога

Внесем тексты программ в файлы

```
////////////////////////////////////  
// 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);  
    }  
}
```

Рис. 2: calculate.c

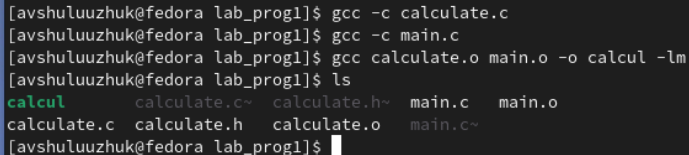
```
////////////////////////////////////  
// calculate.h  
  
#ifndef CALCULATE_H_  
#define CALCULATE_H_  
  
float Calculate(float Numeral, char Operation[4]);  
  
#endif /*CALCULATE_H_*/  
~
```

Рис. 3: calculate.h


```
////////////////////////////////////  
// 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. 4: main.c

Выполним компиляцию программы посредством gcc



```
[avshuluuzhuk@fedora lab_prog1]$ gcc -c calculate.c
[avshuluuzhuk@fedora lab_prog1]$ gcc -c main.c
[avshuluuzhuk@fedora lab_prog1]$ gcc calculate.o main.o -o calcul -lm
[avshuluuzhuk@fedora lab_prog1]$ ls
calcul      calculate.c~  calculate.h~  main.c      main.o
calculate.c calculate.h   calculate.o   main.c~
```

Рис. 5: компиляция программы

Создадим файл Makefile со следующим содержанием

```
#  
# 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)  
  
15 main.o: main.c calculate.h  
    gcc -c main.c $(CFLAGS)  
clean:  
    -rm calcul *.o *~  
  
# End Makefile
```

Рис. 6: Makefile

С помощью gdb выполним отладку программы calcul. Запустим отладчик GDB, загрузив в него программу для отладки и введем команду run

```
(gdb) run
Starting program: /home/avshuluuzhuk/work/os/lab_prog1/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): +
Второе слагаемое: 3
      8.00
[Inferior 1 (process 4867) exited normally]
(gdb)
```

Рис. 7: запуск программы

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

```
[avshuluuzhuk@fedora lab_prog]$ splint calculate.c
Splint 3.1.2 --- 23 Jul 2022

calculate.c: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, 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: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:46:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:47:13: Return value type double does not match declared type float:
(pow(Numeral, SecondNumeral))
calculate.c:50:11: Return value type double does not match declared type float:
(sqrt(Numeral))
calculate.c:52:11: Return value type double does not match declared type float:
(sin(Numeral))
calculate.c:54:11: Return value type double does not match declared type float:
(cos(Numeral))
calculate.c:56:11: Return value type double does not match declared type float:
(tan(Numeral))
calculate.c:60:13: Return value type double does not match declared type float:
(HUGE_VAL)

Finished checking --- 15 code warnings
[avshuluuzhuk@fedora lab_prog]$
```

Рис. 8: анализ файла calculate.c

```
Finished checking --- 23 code warnings
[avshuluuzhuk@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 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...
    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:16:14: 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:16:11: Corresponding format code
main.c:16:3: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
[avshuluuzhuk@fedora lab_prog]$
```

Рис. 9: анализ файла main.c

Выводы

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