

Лабораторная работа №13

Курс “Операционные Системы”

Лихтенштейн А.А., НКАбд-03-22

6 мая 2023

Российский университет дружбы народов, Москва, Россия

- Лихтенштейн Алина Алексеевна
- студент группы НКАбд-03-22
- кафедры Компьютерные и информационные науки
- Российский университет дружбы народов
- 1132229533@pfur.ru

- 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.

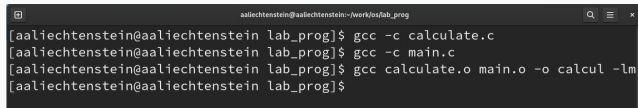
Выполнение заданий.

В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`



Рис. 1: Создание файлов

Выполните компиляцию программы посредством gcc. При необходимости исправьте синтаксические ошибки

A terminal window with a dark background and light text. The title bar at the top reads 'aaliechtenstein@aaliechtenstein:~/work/os/lab_prog'. The terminal contains four lines of text, each preceded by a prompt '[aaliechtenstein@aaliechtenstein lab_prog]\$'. The commands are: 'gcc -c calculate.c', 'gcc -c main.c', 'gcc calculate.o main.o -o calcul -lm', and an empty line.

```
aaliechtenstein@aaliechtenstein:~/work/os/lab_prog
[aaliechtenstein@aaliechtenstein lab_prog]$ gcc -c calculate.c
[aaliechtenstein@aaliechtenstein lab_prog]$ gcc -c main.c
[aaliechtenstein@aaliechtenstein lab_prog]$ gcc calculate.o main.o -o calcul -lm
[aaliechtenstein@aaliechtenstein lab_prog]$
```

Рис. 2: Компиляция

Создайте Makefile

```
Открыть ▾  Makefile
~\work\os\lab_prog

#
# 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 *~

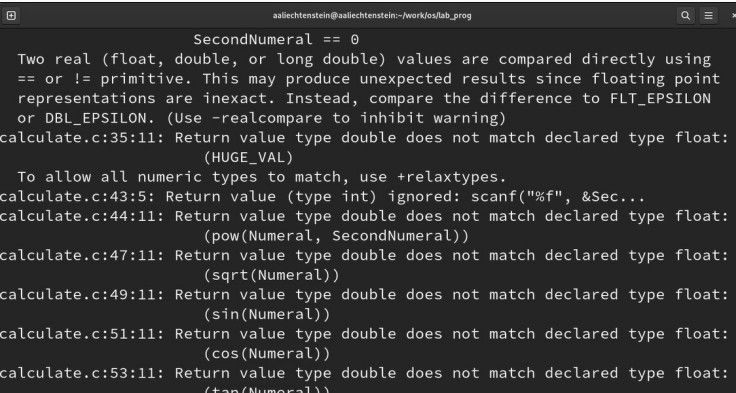
# End Makefile
```

С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile)

```
24     }
25     else if(strncmp(Operation, "*", 1) == 0)
26     {
27         printf("Множитель: ");
28         scanf("%f",&SecondNumeral);
29         return(Numeral * SecondNumeral);
(gdb) list calculate.c:20,27
20     {
21         printf("Вычитаемое: ");
22         scanf("%f",&SecondNumeral);
23         return(Numeral - SecondNumeral);
24     }
25     else if(strncmp(Operation, "*", 1) == 0)
26     {
27         printf("Множитель: ");
(gdb) break 21
Breakpoint 1 at 0x40120f: file calculate.c, line 21.
(gdb) info breakpoints
Num      Type             Disp Enb Address                  What
1        breakpoint       keep y   0x000000000040120f in Calculate at calculate.c:21
(gdb) run
Starting program: /home/dvshilonosov/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("Вычитаемое: ");
(gdb) █
```


С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c



```
aaliechtenstein@aaliechtenstein:~/work/os/lab_prog

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:35:11: Return value type double does not match declared type float:
(HUGE_VAL)
To allow all numeric types to match, use +relaxtypes.
calculate.c:43:5: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:11: Return value type double does not match declared type float:
(pow(Numeral, SecondNumeral))
calculate.c:47:11: Return value type double does not match declared type float:
(sqrt(Numeral))
calculate.c:49:11: Return value type double does not match declared type float:
(sin(Numeral))
calculate.c:51:11: Return value type double does not match declared type float:
(cos(Numeral))
calculate.c:53:11: Return value type double does not match declared type float:
(tan(Numeral))
```

Рис. 5: splint calculate.c

```
aallechtenstein@aallechtenstein:~/work/os/lab_prog — less
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:13:1: 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:15:12: Format argument 1 to scanf (%s) expects char * gets char [4] *:
    &Operation
    Type of parameter is not consistent with corresponding code in format string.
```

Рис. 6: splint main.c

Результаты

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