

Лабораторная работа № 13

Андрианова Марина Георгиевна
RUDN University, Moscow, Russian Federation
NEC–2022, 03 June

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.

Выполнение лабораторной работы

- 1). В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`(команда `"mkdir ~/work/os/lab_prog"`)(рис.1).

```
[mgandrianova@fedora ~]$ mkdir ~/work/os/lab_prog
```

Рис.1: Создание подкаталога

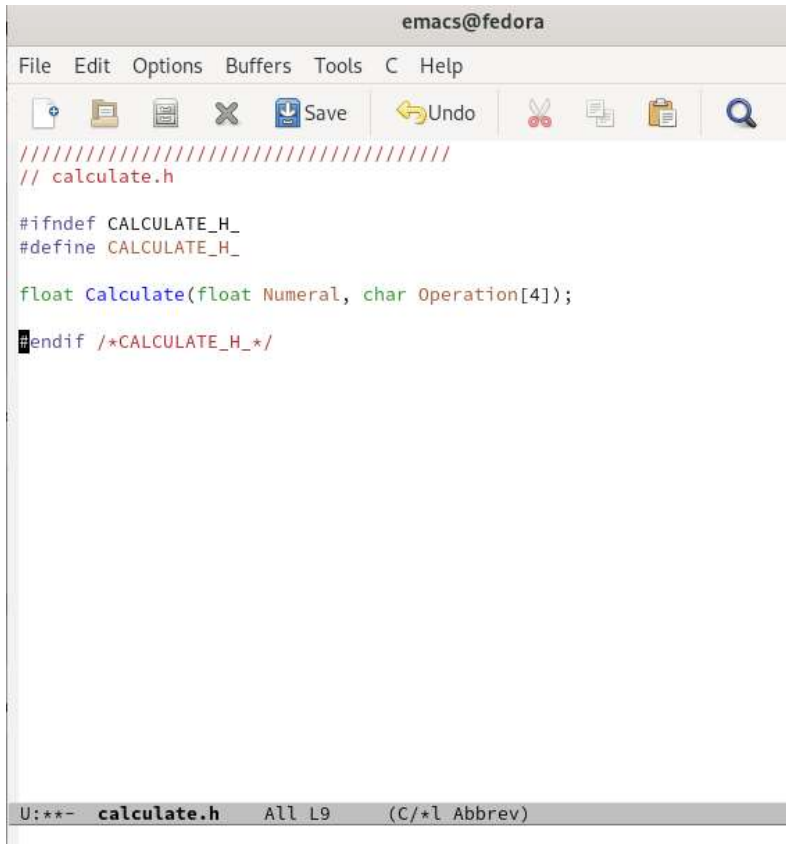
- 2). Перейдём в созданный подкаталог(команда `"cd ~/work/os/lab_prog"`) и создадим в нём файлы: `calculate.h`, `calculate.c`, `main.c`(команда `"touch calculate.h calculate.c main.c"`)(рис.2).

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

Рис.2: Создание
файлов

Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять \sin , \cos , \tan . При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится. Запустила редактор Emacs в фоновом режиме(команда “`emacs &`”) и приступила к редактированию созданных файлов. Реализация функций калькулятора в файле `calculate.c`.

Интерфейсный файл calculate.h,
описывающий формат вызова функции
калькулятора(рис.3):

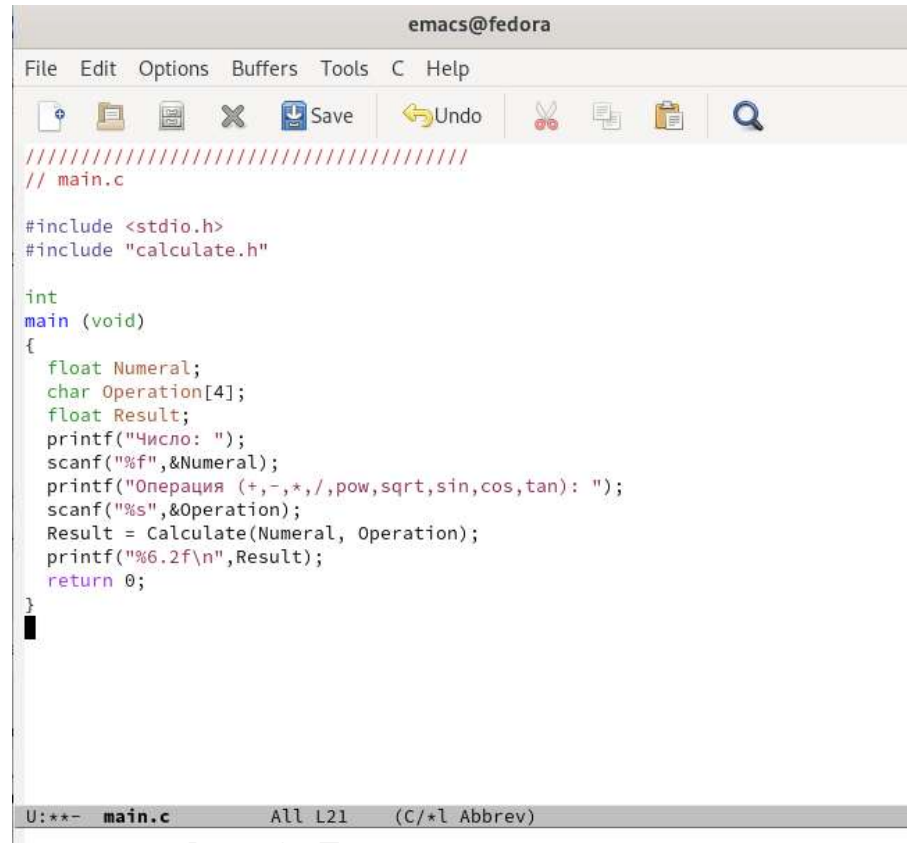
The image shows a screenshot of the Emacs text editor. The title bar at the top reads 'emacs@fedora'. Below it is a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'C', and 'Help'. A toolbar contains icons for opening a file, saving, undo, redo, and search. The main text area displays the content of a header file named 'calculate.h'. The code starts with a series of red slashes, followed by a red comment '// calculate.h'. It then contains preprocessor directives: '#ifndef CALCULATE_H_', '#define CALCULATE_H_', and '#endif /*CALCULATE_H_*/'. Between the define and the endif, there is a function declaration: 'float Calculate(float Numeral, char Operation[4]);'. The status bar at the bottom shows 'U:*** calculate.h All L9 (C/*l Abbrev)'.

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

Рис.3: Программа в
calculate.h

Файл main.c

Основной файл main.c, реализующий интерфейс пользователя к калькулятору(рис.4):



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

U:*** main.c All L21 (C/*l Abbrev)

Рис.4: Программа в main.c

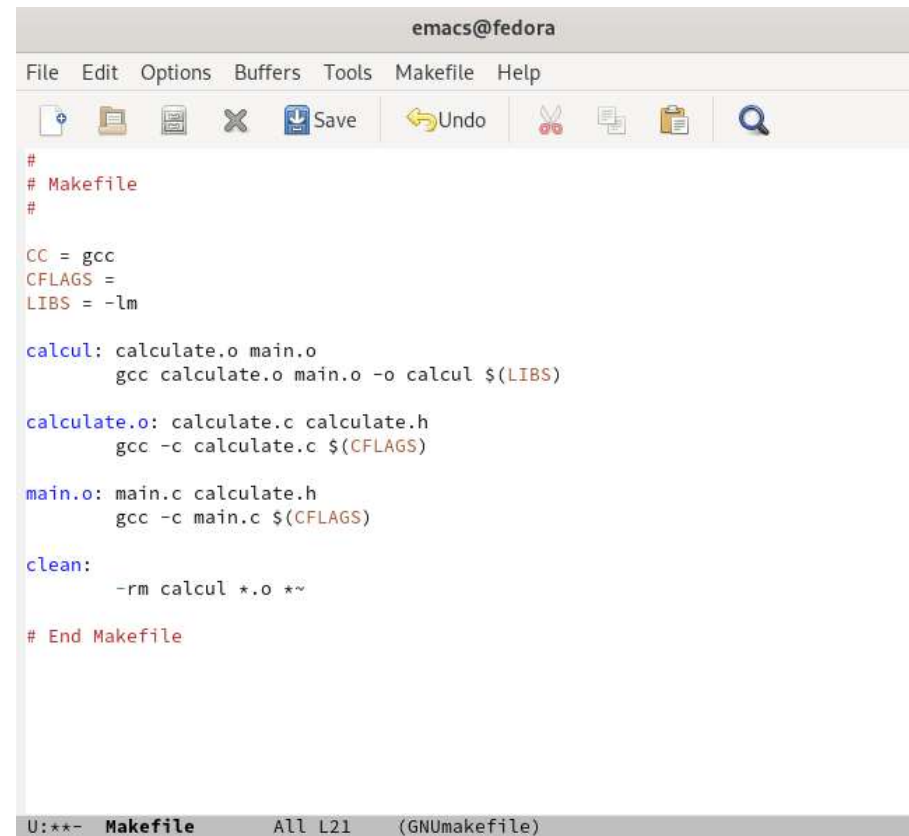
Компиляция программ

3). Выполним компиляцию программы посредством gcc, используя команды “gcc -c calculate.c”, “gcc -c main.c” и “gcc calculate.o main.o -o calcul -lm”(рис.5).

```
[mgandrianova@fedora lab_prog]$ gcc -c calculate.c  
[mgandrianova@fedora lab_prog]$ gcc -c main.c  
[mgandrianova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
```

Рис.5: Компиляция программ

- 4). В ходе компиляции программы никаких синтаксических ошибок выявлено не было.
- 5). Создала Makefile с необходимым содержанием(рис.6).



```
#
# 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)

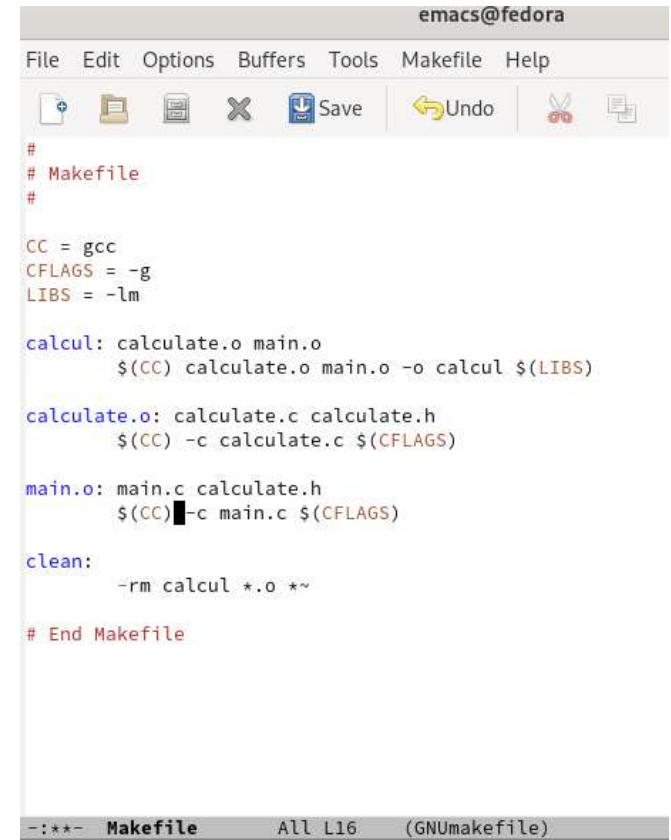
main.o: main.c calculate.h
    gcc -c main.c $(CFLAGS)

clean:
    -rm calcul *.o *~

# End Makefile
```

Рис.6: Программа в Makefile

6). Далее исправила Makefile (рис.7). В переменную CFLAGS добавила опцию -g, необходимую для компиляции объектных файлов и их использования в программе отладчика GDB. Сделала так, что утилита компиляции выбирается с помощью переменной CC.



```
#
# 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.7: Изменённая программа в Makefile

После этого я удалила исполняемые и объектные файлы из каталога с помощью команды(рис.8). Выполнила компиляцию файлов(рис.9).

```
[mgandrianova@fedora lab_prog]$ make clean  
rm calcul *.o *~  
[2]+  Завершён          emacs
```

Рис.8: Удаление файлов

```
[mgandrianova@fedora lab_prog]$ make calculate.o  
gcc -c calculate.c -g  
[mgandrianova@fedora lab_prog]$ make main.o  
gcc -c main.c -g  
[mgandrianova@fedora lab_prog]$ make calcul  
gcc calculate.o main.o -o calcul -lm
```

Рис.9: Компиляция файлов

С помощью gdb выполнила отладку программы calcul: – Запустила отладчик GDB, загрузив в него программу для отладки, используя команду:

```
[mgandrianova@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 11.2-2.fc35
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-redhat-linux-gnu".
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/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) █
```

Рис.10: Работа с
gdb

Для запуска программы внутри отладчика
ввела команду “run”(рис.11).

```
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/mgandrianova/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 4
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 1
    5.00
[Inferior 1 (process 4247) exited normally]
```

Рис.11: Работа с gdb -
run

Установила точку останова в файле calculate.c на строке номер 21, используя команды “list calculate.c:20,27” и “break 21”. Вывела информацию об имеющихся в проекте точках останова с помощью команды “info breakpoints”(рис.12).

```
(gdb) info breakpoints
Num      Type      Disp Enb Address          What
1        breakpoint keep y  0x00000000000040120f in Calculate at calculate.c:21
```

Рис.12: Работа с gdb - info breakpoints

Запустила программу внутри отладчика и убедилась, что программа остановилась в момент прохождения точки останова. Использовала команды “run”, “5”, “-” и “backtrace”. Отладчик выдал следующую информацию (рис.13).

```
#0 Calculate (Numeral=5, Operation=0x7fffffffdf14 "-") at calculate.c:21
#1 0x00000000004014eb in main () at main.c:17
```

Рис.13: Результат после команды “backtrace”

Посмотрела, чему равно на этом этапе значение переменной Numeral(рис.14).



```
(gdb) print Numeral
$1 = 5
```

Рис.14: Работа с gdb - print
Numeral

Сравнила с результатом вывода на экран после использования команды “display Numeral”(рис.15).
Значения совпадают.



```
(gdb) display Numeral
1: Numeral = 5
```

Рис.15: Работа с gdb - display
Numeral

7).Далее воспользовалась командами “splint calculate.c” и “splint main.c” .С помощью утилиты splint выяснилось, что в файлах calculate.c и main.c присутствует функция чтения scanf,возвращающая целое число (тип int), но эти числа не используются и нигде не сохраняются. Утилита вывела предупреждение о том, что в файле calculate.c происходит сравнение вещественного числа с нулем. Также возвращаемые значения (тип double) в функциях pow, sqrt, sin, cos и tan записываются в переменную типа float, что свидетельствует о потере данных.

Выводы

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.