

Лабораторная работа №13

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

Рытов Алексей Константинович НФИбд-02-21

Список иллюстраций

2.1	Подкаталог + файлы(создание)	4
2.2	Файл calculate.h	5
2.3	Файл calculate.c	6
2.4	Файл main.c	6
2.5	Компиляция программы	7
2.6	Makefile	7
2.7	Результат работы Calcul	8
2.8	breakpoint	9
2.9	Работа со Splint	10

1 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.

2 Выполнение лабораторной работы

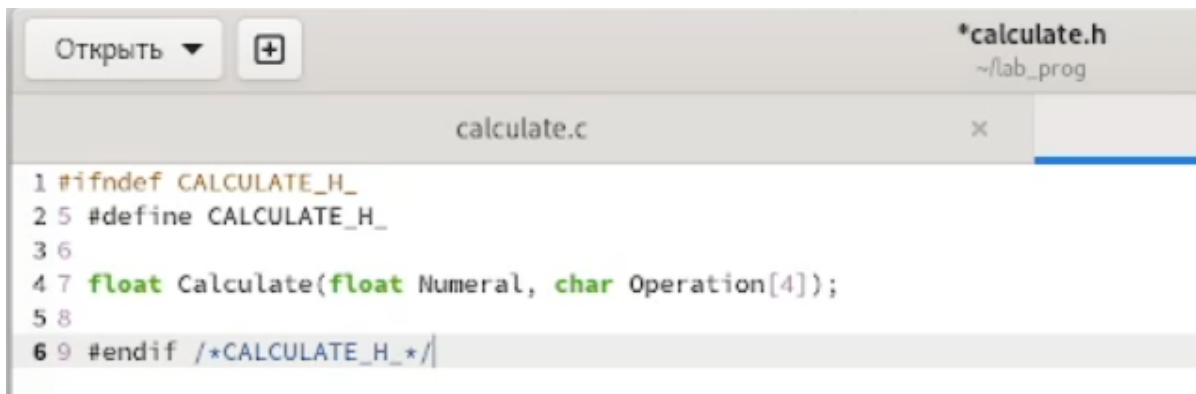
1. В домашнем каталоге создали подкаталог ~/lab_prog.
2. Создали в нём файлы: calculate.h, calculate.c, main.c.

```
[akrihtov@fedora ~]$ cd ~
[akrihtov@fedora ~]$ ls
1~ 2.txt '#4#' bin lab09.sh logfile OS study Документы Изображения Общедоступные 'Рабочий стол'
1.txt '#3#' backup '#lab09.sh#' lab09.sh~ my_os play Видео Загрузки Музыка 'Операционные системы' Шаблоны
[akrihtov@fedora ~]$ mkdir lab_prog
[akrihtov@fedora ~]$ ls
1~ '#3#' bin lab09.sh~ my_os study Загрузки Общедоступные Шаблоны
1.txt '#4#' '#lab09.sh#' lab_prog OS Видео Изображения 'Операционные системы'
2.txt backup lab09.sh logfile play Документы Музыка 'Рабочий стол'
[akrihtov@fedora ~]$ cd lab_prog
[akrihtov@fedora lab_prog]$ touch calculate.h calculate.c main.c
[akrihtov@fedora lab_prog]$
```

Рис. 2.1: Подкаталог + файлы(создание)

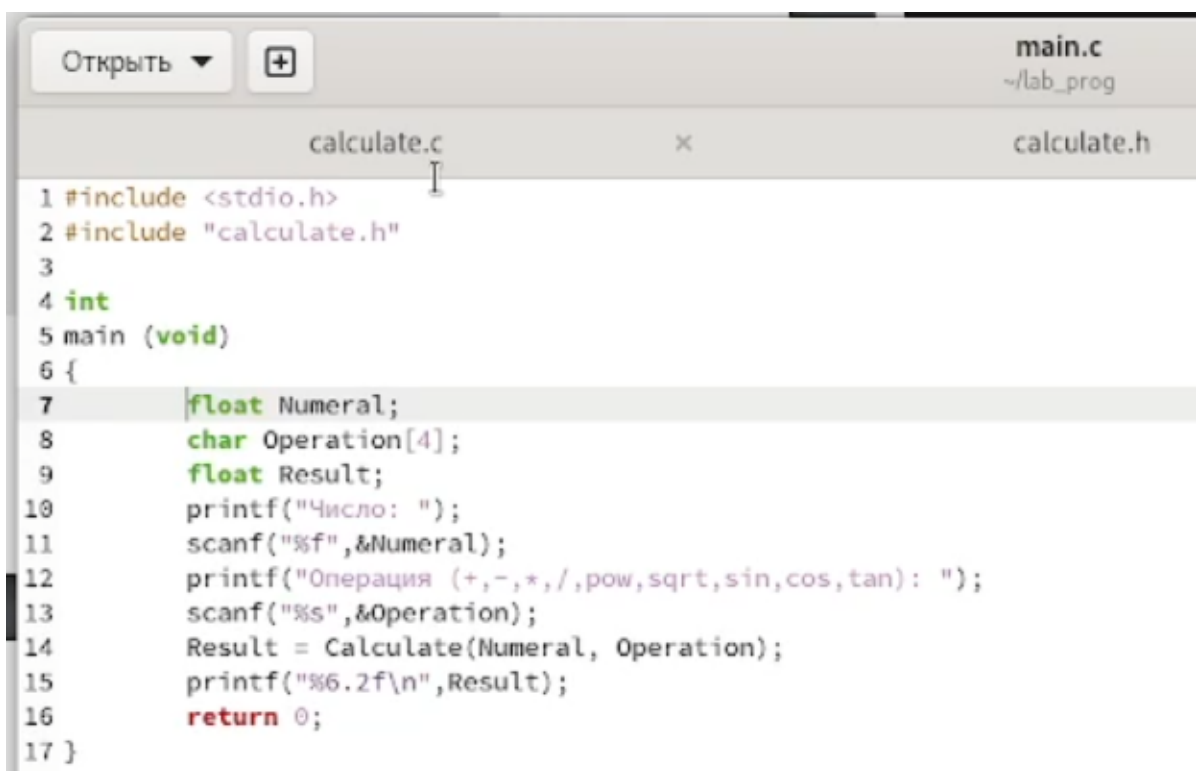
```
22  else if(strncmp(operation, "*", 1) == 0)
23  {
24      printf("Множитель: ");
25      scanf("%f",&SecondNumeral);
26      return(Numeral * SecondNumeral);
27  }
28  else if(strncmp(operation, "/", 1) == 0)
29  {
30      printf("Делитель: ");
31      scanf("%f",&SecondNumeral);
32      if(SecondNumeral == 0)
33      {
34          printf("Ошибка: деление на ноль! ");
35          return(HUGE_VAL);
36      }
37      else
38          return(Numeral / SecondNumeral);
39  }
40  else if(strncmp(operation, "pow", 3) == 0)
41  {
42      printf("Степень: ");
43      scanf("%f",&SecondNumeral);
44      return(pow(Numeral, SecondNumeral));
45  }
46  else if(strncmp(operation, "sqrt", 4) == 0)
47      return(sqrt(Numeral));
48  else if(strncmp(operation, "sin", 3) == 0)
49      return(sin(Numeral));
50  else if(strncmp(operation, "cos", 3) == 0)
51      return(cos(Numeral));
52  else if(strncmp(operation, "tan", 3) == 0)
53      return(tan(Numeral));
54  else
55  {
56      printf("Неправильно введено действие ");
57      return(HUGE_VAL);
58  }
59 }
```

Рис. 2.2: Файл calculate.h



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

Рис. 2.3: Файл calculate.c



```
1 #include <stdio.h>
2 #include "calculate.h"
3
4 int
5 main (void)
6 {
7     float Numeral;
8     char Operation[4];
9     float Result;
10    printf("Число: ");
11    scanf("%f",&Numeral);
12    printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
13    scanf("%s",&Operation);
14    Result = Calculate(Numeral, Operation);
15    printf("%6.2f\n",Result);
16    return 0;
17 }
```

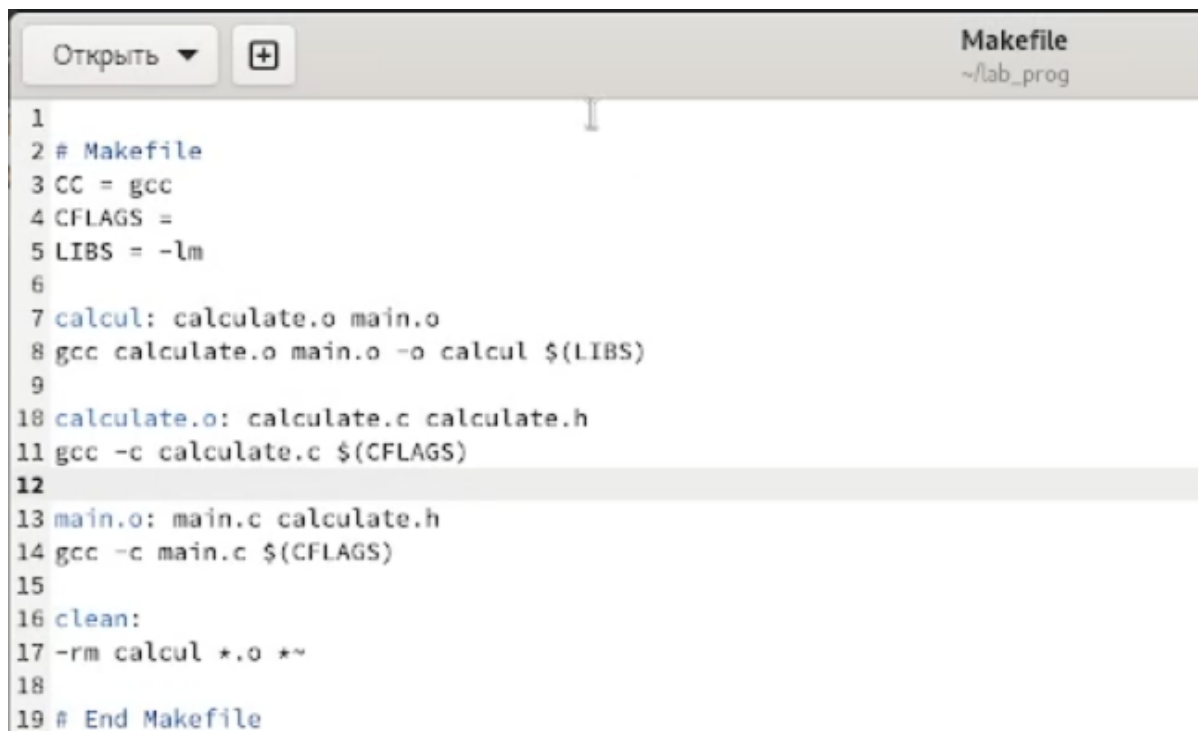
Рис. 2.4: Файл main.c

3. Выполнили компиляцию программы посредством gcc.

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

Рис. 2.5: Компиляция программы

5. Создали Makefile.



The screenshot shows a text editor window titled 'Makefile' with the path '~ /lab_prog'. The editor contains the following text:

```
1
2 # Makefile
3 CC = gcc
4 CFLAGS =
5 LIBS = -lm
6
7 calcul: calculate.o main.o
8 gcc calculate.o main.o -o calcul $(LIBS)
9
10 calculate.o: calculate.c calculate.h
11 gcc -c calculate.c $(CFLAGS)
12
13 main.o: main.c calculate.h
14 gcc -c main.c $(CFLAGS)
15
16 clean:
17 -rm calcul *.o *~
18
19 # End Makefile
```

Рис. 2.6: Makefile

6. С помощью gdb выполнили отладку программы calcul, запустили её.

```

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...

This GDB supports auto-downloading debuginfo from the following URLs:
https://debuginfod.fedoraproject.org/
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/akrihtov/lab_prog/calcul
Downloading separate debug info for /lib64/libm.so.6...
Downloading separate debug info for /lib64/libc.so.6...
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 1
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): sqrt
1.00
[Inferior 1 (process 276403) exited normally]

```

Рис. 2.7: Результат работы Calcul

7. Создание, удаление и работа с breakpoint.


```

(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) break 1
Breakpoint 3 at 0x7ffff7fce0a0: file dl-call-libc-early-init.c, line 27.
(gdb) break 2
Note: breakpoint 3 also set at pc 0x7ffff7fce0a0.
Breakpoint 4 at 0x7ffff7fce0a0: file dl-call-libc-early-init.c, line 27.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/akrihtov/lab_prog/calcul

Breakpoint 3, _dl_call_libc_early_init (libc_map=0x7ffff7fac500, initial=initial@entry=true) at dl-call-libc-early-init.c:27
27 |
(gdb) 3
Undefined command: "3". Try "help".
(gdb) backtrace
#0 _dl_call_libc_early_init (libc_map=0x7ffff7fac500, initial=initial@entry=true) at dl-call-libc-early-init.c:27
#1 0x00007ffff7feb980 in dl_main (phdr=<optimized out>, phnum=<optimized out>, user_entry=<optimized out>, auxv=<optimized out>) at
#2 0x00007ffff7fe712c in _dl_sysdep_start (start_argptr=start_argptr@entry=0x7ffff7ffe010, dl_main=dl_main@entry=0x7ffff7fe8fc0 <dl_
at ../elf/dl-sysdep.c:264
#3 0x00007ffff7fe8a88 in _dl_start_final (arg=0x7ffff7ffe010) at rtld.c:502
#4 _dl_start (arg=0x7ffff7ffe010) at rtld.c:587
#5 0x00007ffff7fe7b58 in _start () from /lib64/ld-linux-x86-64.so.2
#6 0x0000000000000001 in ?? ()
#7 0x00007ffff7ffe338 in ?? ()
#8 0x0000000000000000 in ?? ()
(gdb) delete 3
(gdb) delete 4
(gdb)

```

Рис. 2.8: breakpoint

8. С помощью утилиты `splint` проанализировали коды файлов `calculate.c` и `main.c`.

```

calculate.c:13:2: 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:19:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:25:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:31:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:5: 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:35:9: Return value type double does not match declared type float:
    (HUGE_VAL)
To allow all numeric types to match, use +relaxtypes.
calculate.c:43:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:8: Return value type double does not match declared type float:
    (pow(Numeral, SecondNumeral))
calculate.c:47:8: Return value type double does not match declared type float:
    (sqrt(Numeral))
calculate.c:49:15: 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))
calculate.c:57:8: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings
[akrihtov@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2021

calculate.h:4:37: 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:11:2: 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:13:13: 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:13:10: Corresponding format code
main.c:13:2: Return value (type int) ignored: scanf("%s", &Ope...

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

```

Рис. 2.9: Работа со Splint

3 Выводы

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.

4 Контрольные вопросы

1. Как получить информацию о возможностях программ gcc, make, gdb и др.?
2. Назовите и дайте краткую характеристику основным этапам разработки приложений в UNIX.
3. Что такое суффикс в контексте языка программирования? Приведите примеры использования.
4. Каково основное назначение компилятора языка C в UNIX?
5. Для чего предназначена утилита make?
6. Приведите пример структуры Makefile. Дайте характеристику основным элементам этого файла.
7. Назовите основное свойство, присущее всем программам отладки. Что необходимо сделать, чтобы его можно было использовать?
8. Назовите и дайте основную характеристику основным командам отладчика gdb.
9. Опишите по шагам схему отладки программы, которую Вы использовали при выполнении лабораторной работы.
10. Прокомментируйте реакцию компилятора на синтаксические ошибки в программе при его первом запуске.
11. Назовите основные средства, повышающие понимание исходного кода программы.
12. Каковы основные задачи, решаемые программой splint?