

Средства для создания приложений в ОС UNIX.

Егина Ангелина НБИбд-01-21¹

2 июня, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

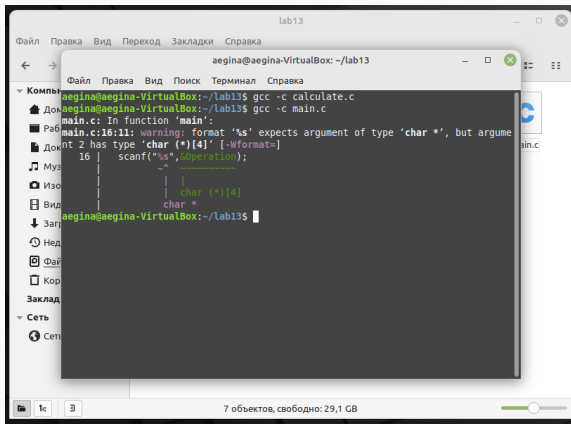
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

Задачи лабораторной работы

- 1 Написать код приложения
- 2 Выполнить компиляцию
- 3 Подготовить Makefile
- 4 Выполнить отладку в GDB
- 5 Проанализировать код при помощи splint

Процесс выполнения лабораторной работы

Выполнение работы



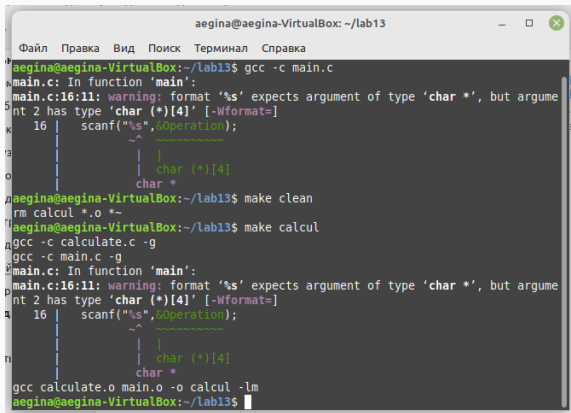
The screenshot shows a terminal window titled 'lab13' within a virtual machine environment. The terminal displays the following commands and output:

```
aegina@aegina-VirtualBox: ~/lab13
aegina@aegina-VirtualBox:~/lab13$ gcc -c calculate.c
aegina@aegina-VirtualBox:~/lab13$ gcc -c main.c
main.c: In function 'main':
main.c:16:11: warning: format '%s' expects argument of type 'char *', but argume
nt 2 has type 'char (*)[4]' [-Wformat=]
   16 |     scanf("%s", &operation);
      |             ^~
      |             |
      |             | char (*)[4]
      |             char *
aegina@aegina-VirtualBox:~/lab13$
```

The terminal window is overlaid on a file manager window showing the contents of the 'lab13' directory. The file manager window has a sidebar with a file tree and a main pane showing the files 'calculate.c' and 'main.c'. The status bar at the bottom of the file manager window indicates '7 объектов, свободно: 29,1 GB'.

Figure 1: Компиляция

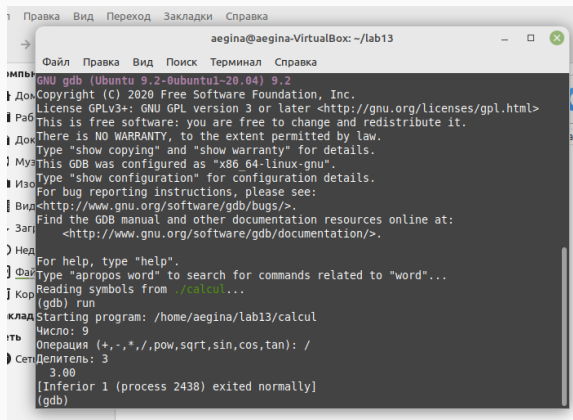
Выполнение работы



```
aegina@aegina-VirtualBox: ~/lab13
Файл  Правка  Вид  Поиск  Терминал  Справка
aegina@aegina-VirtualBox:~/lab13$ gcc -c main.c
main.c: In function 'main':
main.c:16:11: warning: format '%s' expects argument of type 'char *', but argume
nt 2 has type 'char (*)[4]' [-Wformat=]
   16 |     scanf("%s",&operation);
      |           ^~
      |           |
      |         char (*)[4]
      |         char *
aegina@aegina-VirtualBox:~/lab13$ make clean
rm calcul *.o *~
aegina@aegina-VirtualBox:~/lab13$ make calcul
gcc -c calculate.c -g
gcc -c main.c -g
main.c: In function 'main':
main.c:16:11: warning: format '%s' expects argument of type 'char *', but argume
nt 2 has type 'char (*)[4]' [-Wformat=]
   16 |     scanf("%s",&operation);
      |           ^~
      |           |
      |         char (*)[4]
      |         char *
gcc calculate.o main.o -o calcul -lm
aegina@aegina-VirtualBox:~/lab13$
```

Figure 2: Использование make

Выполнение работы

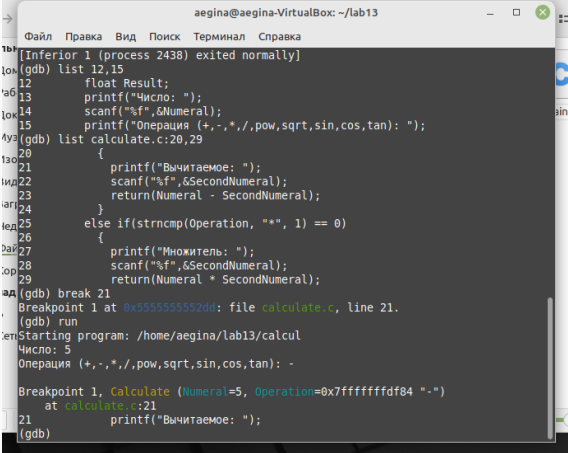


The screenshot shows a terminal window titled 'aegina@aegina-VirtualBox: ~/lab13'. The terminal displays the output of the 'gdb' command, showing the GNU Debugger version 9.2 and its license. The user then runs the program 'calcul' using the 'run' command. The program outputs the number '9' and the result of a division operation, '3.00'. The terminal also shows the program exiting normally.

```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 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-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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) run
Starting program: /home/aegina/lab13/calcul
Число: 9
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): /
Делитель: 3
3.00
[Inferior 1 (process 2438) exited normally]
(gdb)
```

Figure 3: Использование отладчика

Выполнение работы

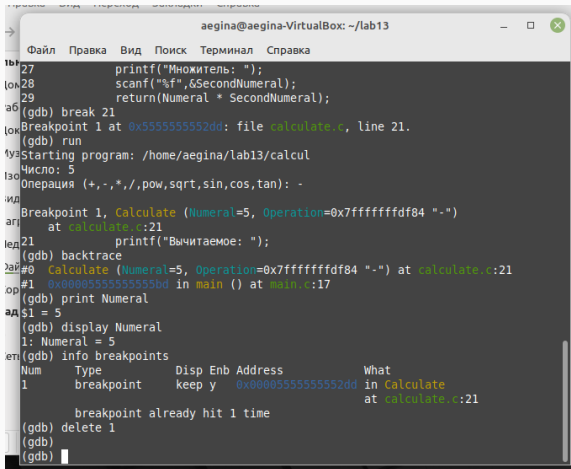


```
aegina@aegina-VirtualBox: ~/lab13
Файл  Правка  Вид  Поиск  Терминал  Справка
[Inferior 1 (process 2438) exited normally]
(gdb) list 12,15
12      float Result;
13      printf("Число: ");
14      scanf("%f",&Numeral);
15      printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb) list calculate.c:20,29
20      {
21          printf("Вычитаемое: ");
22          scanf("%f",&SecondNumeral);
23          return(Numeral - SecondNumeral);
24      }
25      else if(strncmp(Operation, "-", 1) == 0)
26      {
27          printf("Множитель: ");
28          scanf("%f",&SecondNumeral);
29          return(Numeral * SecondNumeral);
(gdb) break 21
Breakpoint 1 at 0x555555552dd: file calculate.c, line 21.
(gdb) run
Starting program: /home/aegina/lab13/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf84 "-")
at calculate.c:21
21      printf("Вычитаемое: ");
(gdb)
```

Figure 4: Использование отладчика

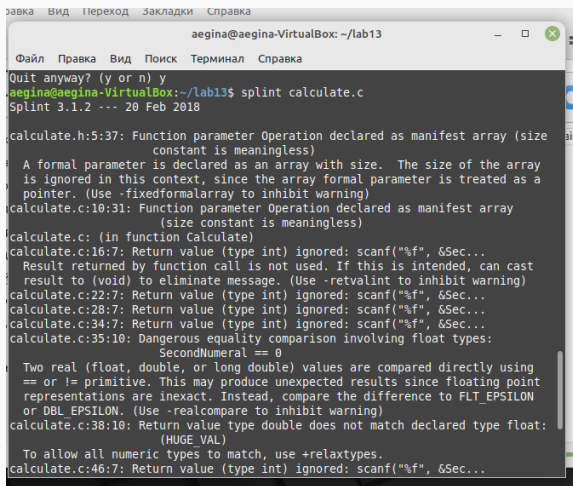
Выполнение работы



```
aegina@aegina-VirtualBox: ~/lab13
Файл  Правка  Вид  Поиск  Терминал  Справка
27     printf("Множитель: ");
28     scanf("%f",&SecondNumeral);
29     return(Numeral * SecondNumeral);
(gdb) break 21
Breakpoint 1 at 0x555555552dd: file calculate.c, line 21.
(gdb) run
Starting program: /home/aegina/lab13/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf84 "-")
  at calculate.c:21
21     printf("Вычитаемое: ");
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffdf84 "-") at calculate.c:21
#1 0x0000555555555bd in main () at main.c:17
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) info breakpoints
Num      Type             Disp Enb Address              What
1        breakpoint        keep y  0x000055555552dd    in calculate
                                     at calculate.c:21
breakpoint already hit 1 time
(gdb) delete 1
(gdb)
(gdb)
```

Figure 5: Использование отладчика

Выполнение работы



```
авка Вид Переход Закладки Справка
aegina@aegina-VirtualBox: ~/lab13
Файл Правка Вид Поиск Терминал Справка
Quit anyway? (y or n) y
aegina@aegina-VirtualBox:~/lab13$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2018

calculate.h:5:37: 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...
```

Figure 6: Использование splint

Выводы по проделанной работе

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.