

Средства для создания приложений в ОС UNIX.

Захаров Данил Алексеевич НБИбд-01-21¹

2 июня, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

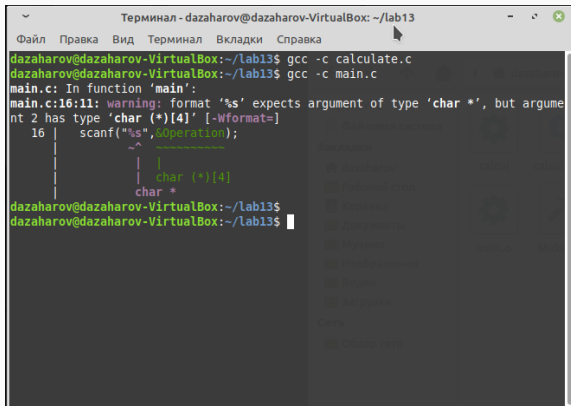
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

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

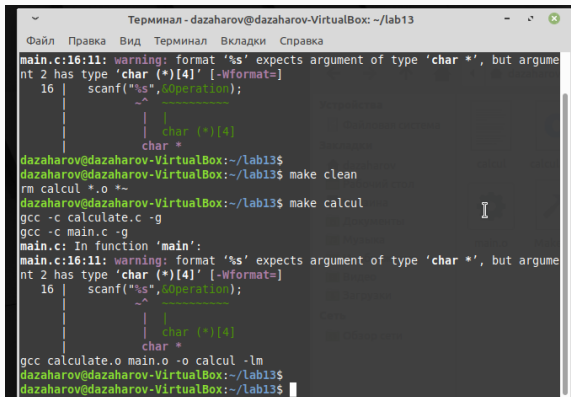
Выполнение работы



```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл  Правка  Вид  Терминал  Вкладки  Справка
dazaharov@dazaharov-VirtualBox:~/lab13$ gcc -c calculate.c
dazaharov@dazaharov-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 *
dazaharov@dazaharov-VirtualBox:~/lab13$
dazaharov@dazaharov-VirtualBox:~/lab13$
```

Figure 1: Компиляция

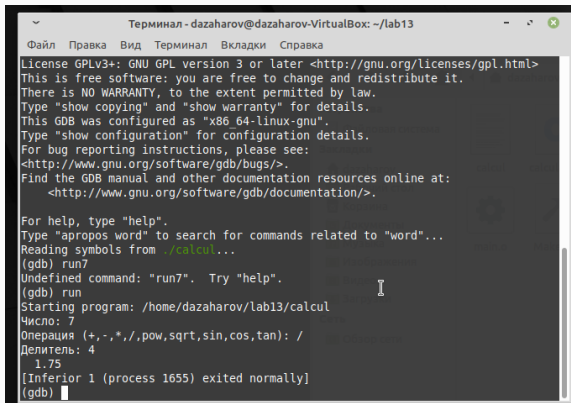
Выполнение работы



```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл  Правка  Вид  Терминал  Вкладки  Справка
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 *
dazaharov@dazaharov-VirtualBox:~/lab13$
dazaharov@dazaharov-VirtualBox:~/lab13$ make clean
rm calcul *.o *~
dazaharov@dazaharov-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 calcul.o main.o -o calcul -lm
dazaharov@dazaharov-VirtualBox:~/lab13$
dazaharov@dazaharov-VirtualBox:~/lab13$
```

Figure 2: Использование make

Выполнение работы

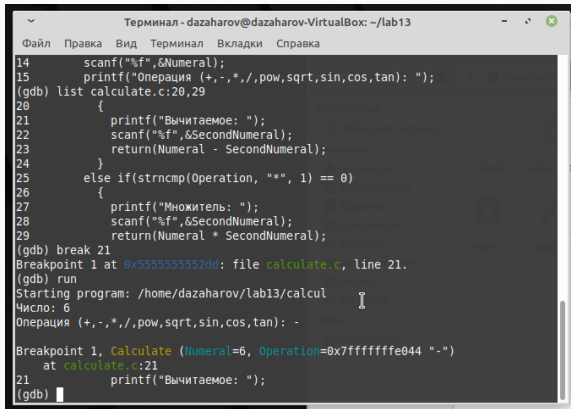
A screenshot of a terminal window titled "Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13". The terminal shows the output of the GDB debugger. It starts with the GNU GPL license notice, followed by instructions on how to use GDB. The user enters "run7", which results in an "Undefined command: 'run7'. Try 'help'." error. Then, the user enters "run", and the program "/home/dazaharov/lab13/calcul" starts. The program prompts for a number (7) and an operation (+, -, *, /, pow, sqrt, sin, cos, tan), with "/" being entered. The program then prompts for a divisor (4), and the result "1.75" is displayed. Finally, the program exits normally, and the user enters "q" to quit GDB.

```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл  Правка  Вид  Терминал  Вкладки  Справка
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) run7
Undefined command: "run7".  Try "help".
(gdb) run
Starting program: /home/dazaharov/lab13/calcul
Число: 7
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): /
Делитель: 4
1.75
[Inferior 1 (process 1655) exited normally]
(gdb) q
```

Figure 3: Использование отладчика

Выполнение работы

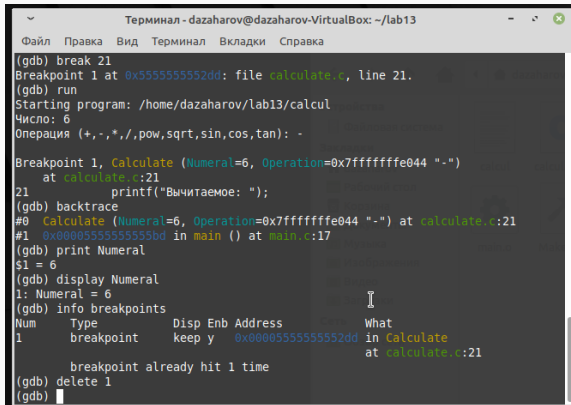


```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл  Правка  Вид  Терминал  Вкладки  Справка
14      scanf("%f",&Numeral);
15      printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb) list calculate.c:20,29
20      {
21          printf("Вычитаемое: ");
22          scanf("%f",&SecondNumeral);
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/dazaharov/lab13/calcul
Число: 6
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=6, Operation=0x7fffffff044 "-")
at calculate.c:21
21      printf("Вычитаемое: ");
(gdb)
```

Figure 4: Использование отладчика

Выполнение работы



```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл  Правка  Вид  Терминал  Вкладки  Справка

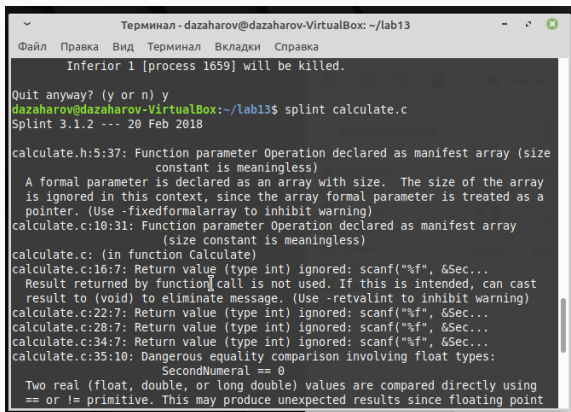
(gdb) break 21
Breakpoint 1 at 0x555555552dd: file calculate.c, line 21.
(gdb) run
Starting program: /home/dazaharov/lab13/calcul
Число: 6
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=6, Operation=0x7fffffff044 "-")
at calculate.c:21
21      printf("Вычитаемое: ");
(gdb) backtrace
#0  Calculate (Numeral=6, Operation=0x7fffffff044 "-") at calculate.c:21
#1  0x0000555555555bd in main () at main.c:17
(gdb) print Numeral
$1 = 6
(gdb) display Numeral
1: Numeral = 6
(gdb) info breakpoints
Num      Type             Disp Enb Address            What
1        breakpoint        keep y  0x0000555555552dd in Calculate
                                           at calculate.c:21

breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

Figure 5: Использование отладчика

Выполнение работы



```
Терминал - dazaharov@dazaharov-VirtualBox: ~/lab13
Файл Правка Вид Терминал Вкладки Справка
Inferior 1 [process 1659] will be killed.
Quit anyway? (y or n) y
dazaharov@dazaharov-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
```

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.