

# Лабораторная работа №13

# 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 я создала подкаталог  
~/work/os/lab\_prog.

2. Создала в нём файлы: `calculate.h`, `calculate.c`, `main.c` и в них написала код по примеру лабораторной работы (рис.1).

```
[asmaslova@fedora ~]$ cd work
[asmaslova@fedora work]$ cd os
bash: cd: os: No such file or directory
[asmaslova@fedora work]$ ls
study
[asmaslova@fedora work]$ cd study
[asmaslova@fedora study]$ ls
2021-2022
[asmaslova@fedora study]$ cd ,,
bash: cd: ,,: No such file or directory
[asmaslova@fedora study]$ cd ,,
bash: cd: ,,: No such file or directory
[asmaslova@fedora study]$ cd ..
[asmaslova@fedora work]$ ls
study
[asmaslova@fedora work]$ pwd
/home/asmaslova/work
[asmaslova@fedora work]$ mkdir os
[asmaslova@fedora work]$ cd os
[asmaslova@fedora os]$ mkdir lab_prog
[asmaslova@fedora os]$ cd lab_prog/
[asmaslova@fedora lab_prog]$ ls
[asmaslova@fedora lab_prog]$ pwd
/home/asmaslova/work/os/lab_prog
[asmaslova@fedora lab_prog]$ touch calculate.h calculate.c main.c
[asmaslova@fedora lab_prog]$ ls
calculate.c calculate.h main.c
```

рис.1 Создание подкаталога и необходимых файлов

3.Далее я выполнила компиляцию программы посредством gcc (рис.2).

```
[asmaslova@fedora lab_prog]$ gcc -c calculate.c
[asmaslova@fedora lab_prog]$ gcc -c main.c
[asmaslova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[asmaslova@fedora lab_prog]$ ./calcul
Число: 15
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): sin
0.65
[asmaslova@fedora lab_prog]$ ./calcul
Число: 16
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 24
-8.00
[asmaslova@fedora lab_prog]$ ./calcul
Число: 91
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): pow
Степень: 3
753571.00
[asmaslova@fedora lab_prog]$ ./calcul
Число: 25
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): tan
-0.13
[asmaslova@fedora lab_prog]$ touch Makefile
[asmaslova@fedora lab_prog]$ emacs Makefile &
[1] 4750
```

рис.2 Компиляция программы посредством gcc



4. Синтаксических ошибок компилятор мне не выдал, поэтому я ничего не исправляла, вместо этого я проверила работу программы и убедилась, что она работает.
5. После этого я создала Makefile с необходимым содержанием (рис.2).
6. С помощью gdb выполните отладку программы calcul (рис.3-4)

```

[asmaslova@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 10.2-9.fc35
Copyright (C) 2021 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...
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/asmaslova/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 15
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 3
12.00
[Inferior 1 (process 5688) exited normally]
(gdb) list
1      /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/<built-in>: Success.
(gdb) list 12,15
12     /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/<built-in>: Success.
(gdb) list calculate.c:20,29
No source file named calculate.c.
(gdb) break 21
No line 21 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 1 (21) pending.
(gdb) list
12     in /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/<built-in>
(gdb) list 12,21
12     in /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/<built-in>
(gdb) info breakpoints
Num   Type             Disp Enb Address      What
1     breakpoint       keep y   <PENDING> 21

```

рис.3 Отладка программы с помощью gdb

```

(gdb) info breakpoints
Num      Type           Disp Enb Address      What
1        breakpoint     keep y   <PENDING>    21
(gdb) run
Starting program: /home/asmaslova/work/os/lab_prog/calcul
Downloading source file /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/rtld.c...

Breakpoint 1, _dl_start (arg=0x7ffffffffffe070) at rtld.c:71
71      HP_TIMING_NOW (*var);
(gdb) backtrace
#0  _dl_start (arg=0x7ffffffffffe070) at rtld.c:71
#1  0x00007ffff7fcc098 in _start () from /lib64/ld-linux-x86-64.so.2
#2  0x0000000000000001 in ?? ()
#3  0x00007ffffffffffe384 in ?? ()
#4  0x0000000000000000 in ?? ()
(gdb) print Numeral
No symbol "Numeral" in current context.
(gdb) delete 1
(gdb) info breakpoints
No breakpoints or watchpoints.

```

рис.4 Отладка программы с помощью gdb

7.С помощью утилиты splint я попробовала проанализировать коды файлов calculate.c и main.c и увидела множество синтаксических ошибок, которые не выдал компилятор (рис.5).

```

[asmaslova@fedora lab_prog]$ splint calculate.c
Splint 3.1.2 --- 23 Jul 2021

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)
calculate.c:8:31: Function parameter Operation declared as manifest array (size
        constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:14:7: 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:20:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:26:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:32:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:33:10: 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:36:10: Return value type double does not match declared type float:
        (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:44:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:45:13: Return value type double does not match declared type float:
        (pow(Numeral, SecondNumeral))
calculate.c:48:11: Return value type double does not match declared type float:
        (sqrt(Numeral))
calculate.c:50:11: Return value type double does not match declared type float:
        (sin(Numeral))
calculate.c:52:11: Return value type double does not match declared type float:
        (cos(Numeral))
calculate.c:54:11: Return value type double does not match declared type float:
        (tan(Numeral))
calculate.c:58:13: Return value type double does not match declared type float:
        (HUGE_VAL)

Finished checking --- 15 code warnings

```

рис.5 Результат работы утилиты splint

# Выводы

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