

## Лабораторная работа №13

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

---

Чемоданова А.А.

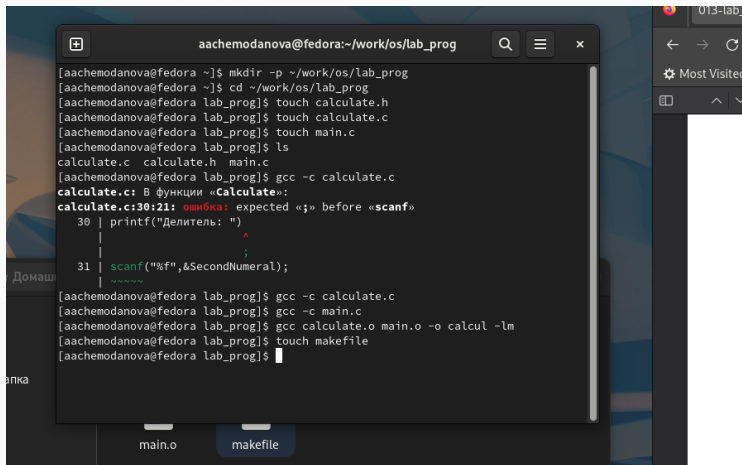
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.
2. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`. 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.
3. Выполните компиляцию программы посредством `gcc`.
4. При необходимости исправьте синтаксические ошибки.
5. Создайте `Makefile`.

6. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile): Запустите отладчик GDB, загрузив в него программу для отладки: `gdb ./calcul` Для запуска программы внутри отладчика введите команду `run: run` Для страничного (по 9 строк) просмотра исходного код используйте команду `list: 1 list` Для просмотра строк с 12 по 15 основного файла используйте `list` с параметрами: `list 12,15` Для просмотра определённых строк не основного файла используйте `list` с параметрами: `list calculate.c:20,29` Установите точку останова в файле `calculate.c` на строке номер 21: `list calculate.c:20,27 break 21` Выведите информацию об имеющихся в проекте точка останова: `info breakpoints` – Запустите программу внутри отладчика и убедитесь, что программа остановится в момент прохождения точки останова. Отладчик выдаст следующую информацию: `#0 Calculate (Numeral=5, Operation=0x7fffffff280 "-") at calculate.c:21 #1 0x000000000400b2b in main () at main.c:17` а команда `backtrace` покажет весь стек вызываемых функций от начала программы до текущего места.

Посмотрите, чему равно на этом этапе значение переменной `Numeral`, введя: `print Numeral`  
На экран должно быть выведено число 5. Сравните с результатом вывода на экран после использования команды: `display Numeral` Уберите точки останова 7. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`.

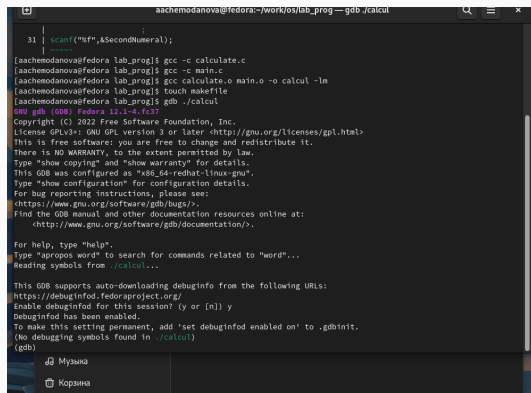
# Выполнение лабораторной работы



The screenshot shows a terminal window titled 'aachemodanova@fedora:~/work/os/lab\_prog'. The user has created a directory, navigated to it, and created three files: 'calculate.h', 'calculate.c', and 'main.c'. The 'ls' command shows these files. The user then attempts to compile 'calculate.c' with 'gcc -c calculate.c'. The compiler outputs an error: 'calculate.c:30:21: ошибка: expected «;» before «scanf»'. The error points to line 30, which contains 'printf("Делитель: ")'. Line 31 contains 'scanf("%f",&SecondNumeral);'. The user then runs 'gcc -c calculate.c' again, followed by 'gcc -c main.c', and then 'gcc calculate.o main.o -o calcul -lm'. Finally, the user runs 'touch makefile'. The terminal window has tabs at the bottom labeled 'main.o' and 'makefile'. To the right of the terminal, a browser window is partially visible with the title '013-lab\_...' and navigation buttons.

```
aachemodanova@fedora:~/work/os/lab_prog
[aachemodanova@fedora ~]$ mkdir -p ~/work/os/lab_prog
[aachemodanova@fedora ~]$ cd ~/work/os/lab_prog
[aachemodanova@fedora lab_prog]$ touch calculate.h
[aachemodanova@fedora lab_prog]$ touch calculate.c
[aachemodanova@fedora lab_prog]$ touch main.c
[aachemodanova@fedora lab_prog]$ ls
calculate.c  calculate.h  main.c
[aachemodanova@fedora lab_prog]$ gcc -c calculate.c
calculate.c: В функции «Calculate»:
calculate.c:30:21: ошибка: expected «;» before «scanf»
   30 | printf("Делитель: ")
      |                      ^
   31 | scanf("%f",&SecondNumeral);
[aachemodanova@fedora lab_prog]$ gcc -c calculate.c
[aachemodanova@fedora lab_prog]$ gcc -c main.c
[aachemodanova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[aachemodanova@fedora lab_prog]$ touch makefile
[aachemodanova@fedora lab_prog]$
```

Рис. 1: Каталог, файлы и компиляция



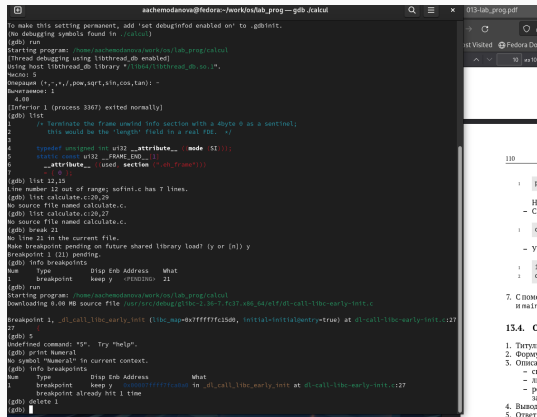
```
aachemodanova@fedora:~/work/os/lab_prog — gdb ./calcul
31 | scanf("%f",&SecondNumeral);
|
[aachemodanova@fedora lab_prog]$ gcc -c calculate.c
[aachemodanova@fedora lab_prog]$ gcc -c main.c
[aachemodanova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[aachemodanova@fedora lab_prog]$ touch makefile
[aachemodanova@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 12.1-4.fc37
Copyright (C) 2012 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...

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)
```

Рис. 2: Запуск программы calcul

# Выполнение лабораторной работы



```
aachemodanova@fedora:~/work/os/lab_prog—gdb ./calcul
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/aachemodanova/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
4.00
0x0000000000000000
Inferior 1 (process 3367) exited normally
(gdb) list
1      /* Terminate the frame unwind info section with a 4byte 0 as a sentinel;
2      this would be the 'length' field in a real FDE. */
3
4      typedef unsigned int u32 __attribute__((mode(SI)));
5      static const u32 __FRAME_END__ = 0;
6      __attribute__((used, section(".eh_frame")))
7      __asm__(".eh_frame");
(gdb) list 12,15
Line number 12 out of range; sofint.c has 7 lines.
(gdb) list calculate.c:20,29
No source file named calculate.c.
(gdb) list calculate.c:20,27
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) info breakpoints
Num Type Disposition Address What
1 breakpoint keep y *PENDING* 21
(gdb) run
Starting program: /home/aachemodanova/work/os/lab_prog/calcul
Downloading 0.00 MB source file /usr/src/debug/glibc-2.36-7.fc37.x86_64/elf/dl-call-libc-early-init.c
Breakpoint 1, _dl_call_libc_early_init (libc_map=0x7ffff7fcc500, initial_initialentry=true) at dl-call-libc-early-init.c:27
27
(gdb) s
Undefined command: "s". Try "help".
(gdb) print Numeral
No symbol "Numeral" in current context.
(gdb) info breakpoints
Num Type Disposition Address What
1 breakpoint keep y *PENDING* on _dl_call_libc_early_init at dl-call-libc-early-init.c:27
breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

013-lab\_prog.pdf

ist Visited Fedora De

110

1

H

- C

1

- y

1

2

3

7. Спом

11air

13.4. C

1. Титул

2. Форум

3. Опаса

- ci

- л

- p

3

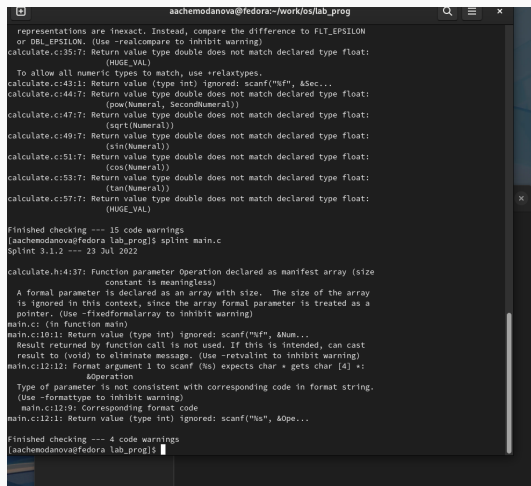
4. Вывоз

5. Ответ

Рис. 3: Работа в gdb



# Выполнение лабораторной работы



```
aachemodanova@fedora:~/work/os/lab_prog

representations are inexact. Instead, compare the difference to FLT_EPSILON
or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:35:7: Return value type double does not match declared type float:
(HUGE_VAL)
To allow all numeric types to match, use +relaxtypes.
calculate.c:43:1: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:44:7: Return value type double does not match declared type float:
(pow(Numeral, SecondNumeral))
calculate.c:47:7: Return value type double does not match declared type float:
(sqrt(Numeral))
calculate.c:49:7: Return value type double does not match declared type float:
(sin(Numeral))
calculate.c:51:7: Return value type double does not match declared type float:
(cos(Numeral))
calculate.c:53:7: Return value type double does not match declared type float:
(tan(Numeral))
calculate.c:57:7: Return value type double does not match declared type float:
(HUGE_VAL)

Finished checking --- 15 code warnings
[aachemodanova@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2022

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:19:1: 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:12:12: 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:12:9: Corresponding format code
main.c:12:1: Return value (type int) ignored: scanf("%s", &Ope...

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

Рис. 4: Splint

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