

Титульный лист

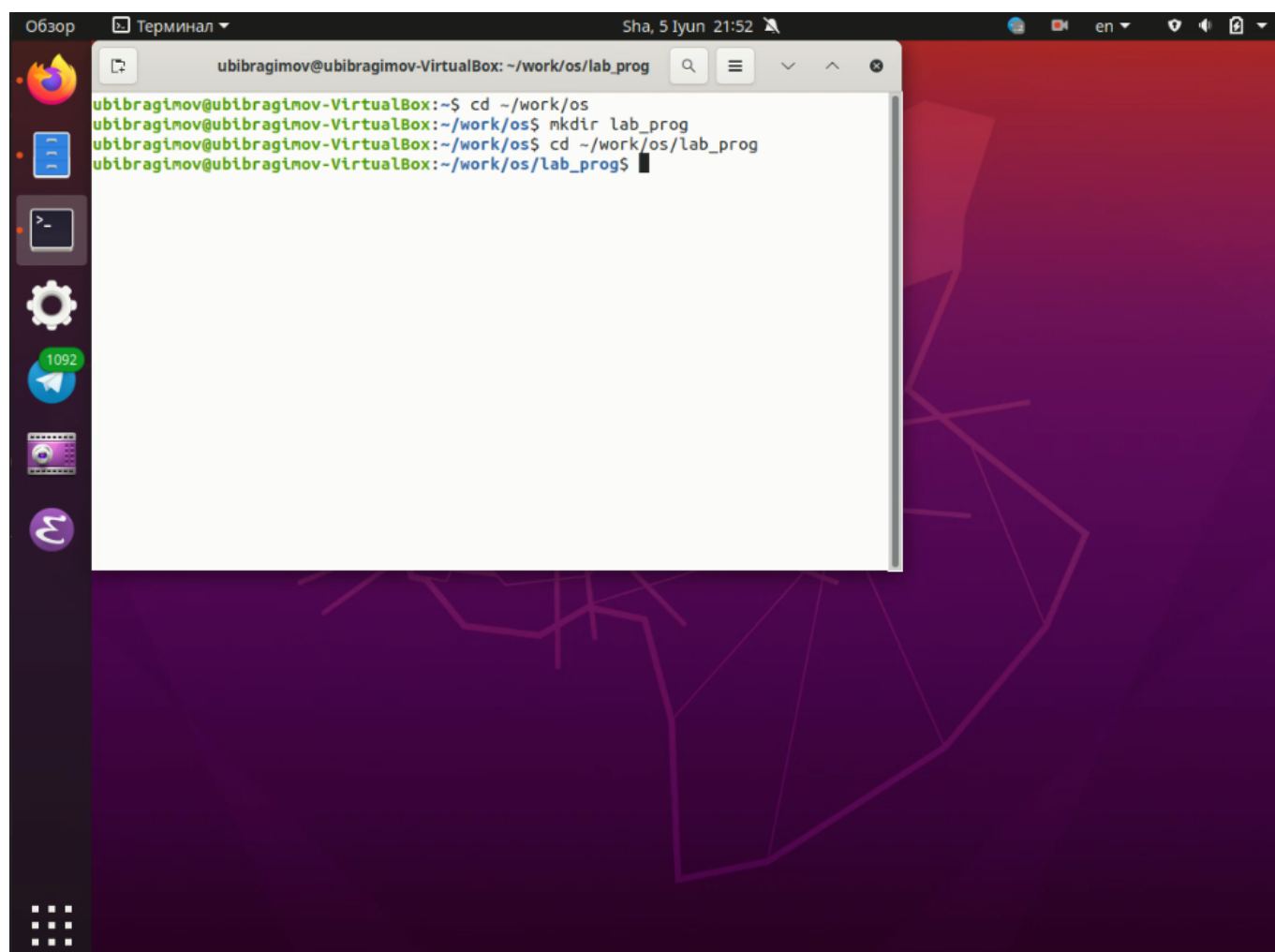
Отчёт по лабораторной работе №14 Ибрагимов Улугбек Ботырхонович

Цель работы

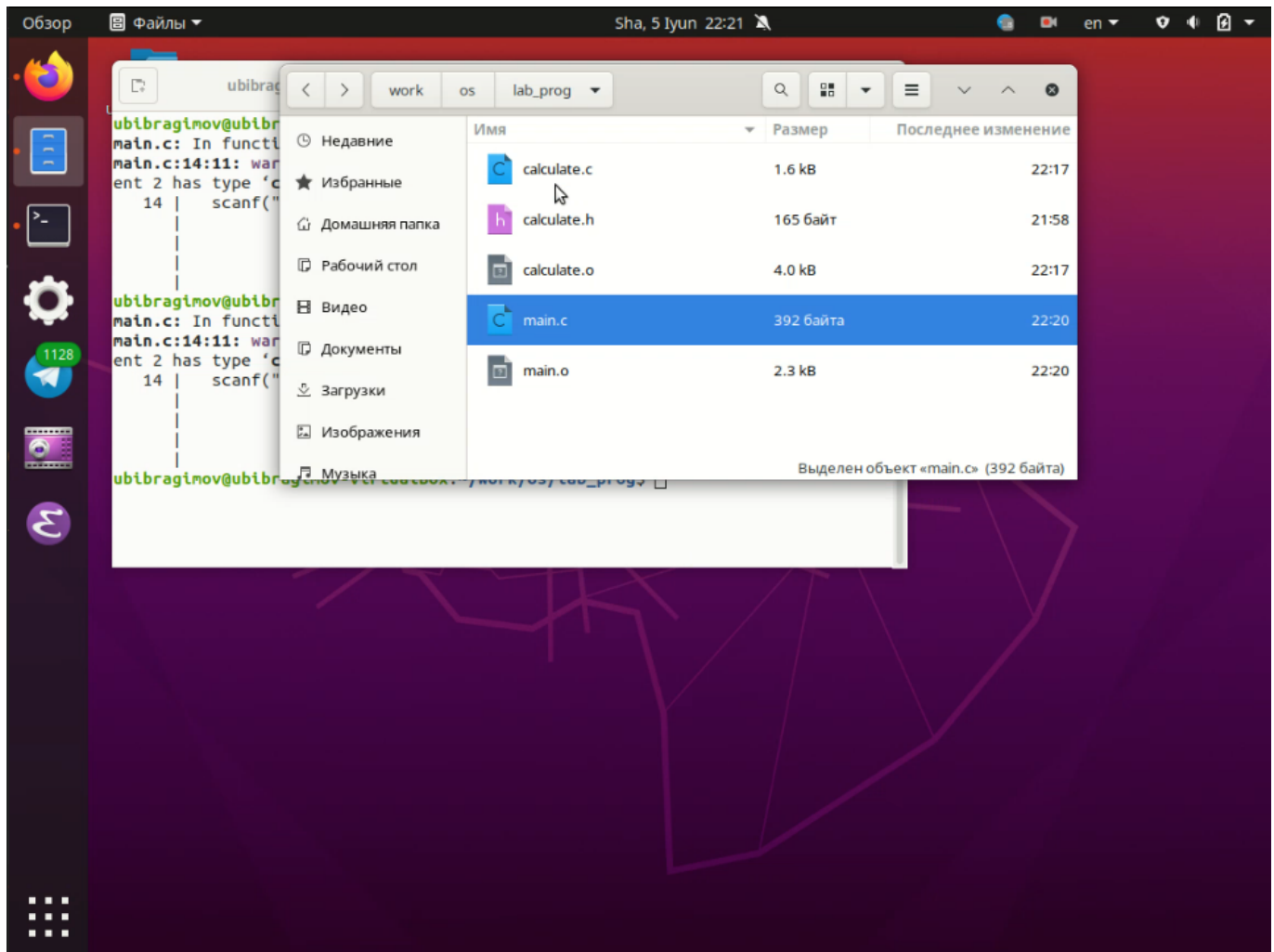
пПриобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

Ход работы:

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



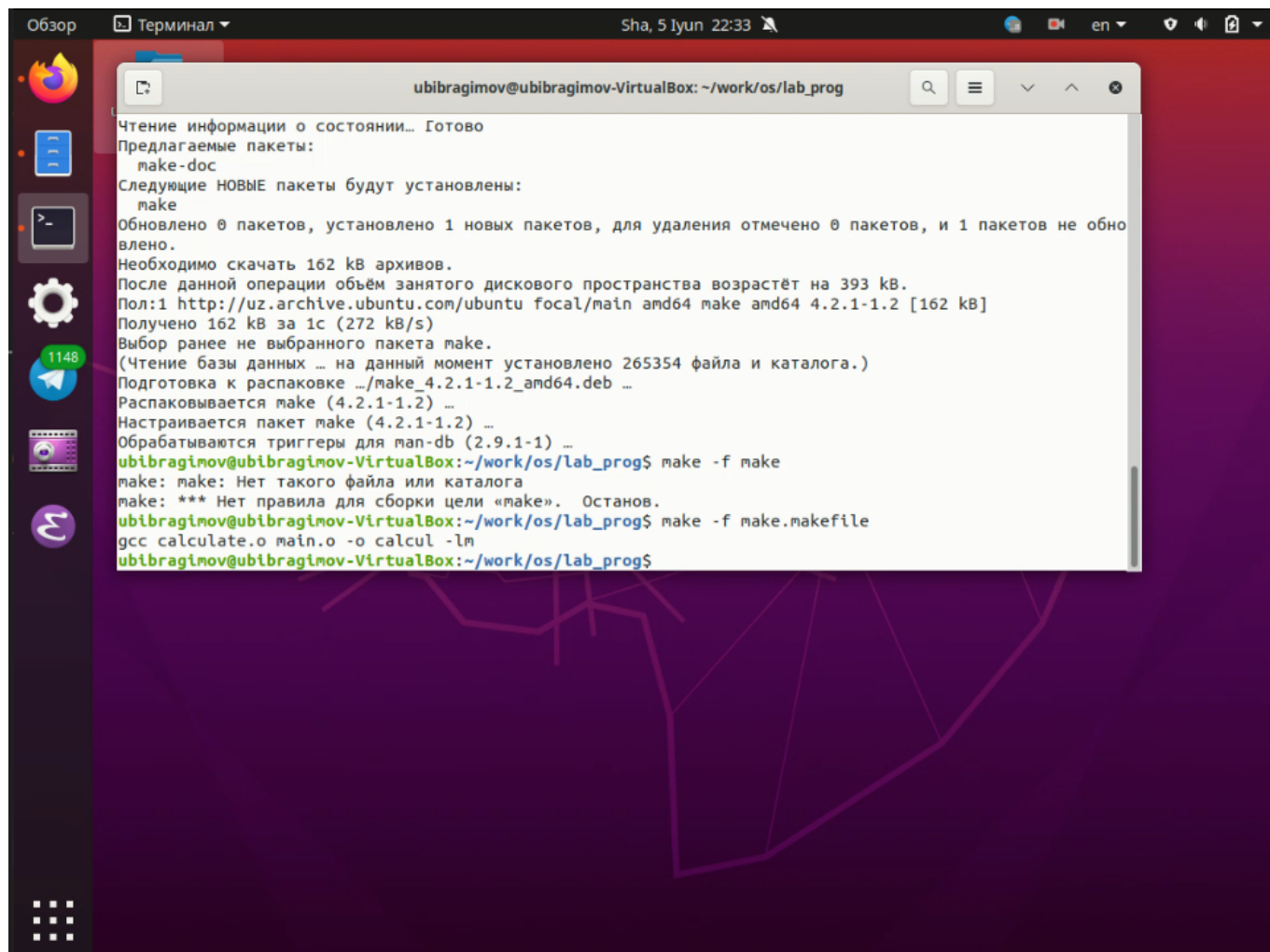
3. Выполните компиляцию программы посредством `gcc`: `gcc -c calculate.c gcc -c main.c gcc calculate.o main.o -o calculate -lm`

4. При необходимости исправьте синтаксические ошибки!

5. Создайте Makefile со следующим содержанием

В нём указаны :

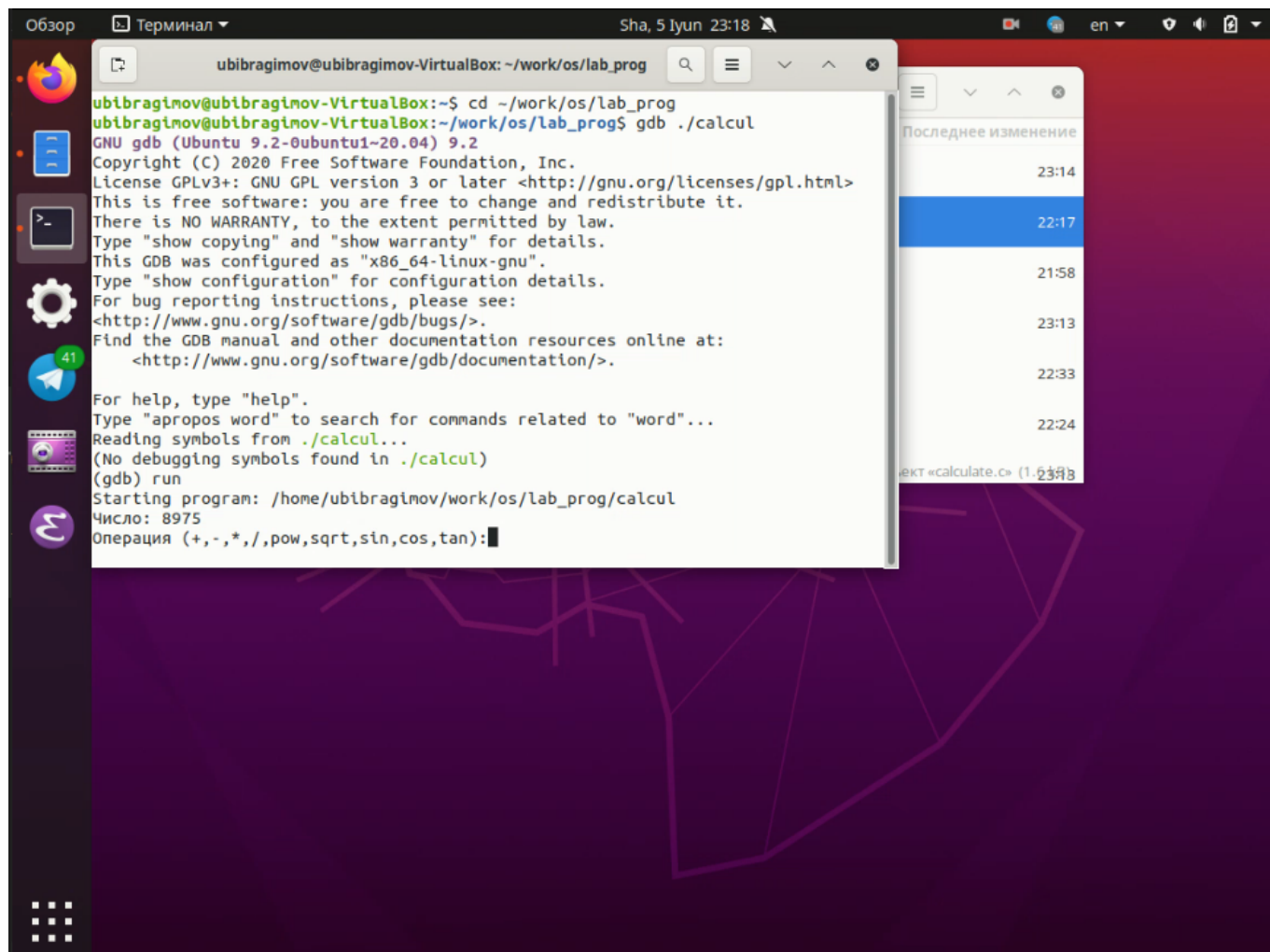
- CC - тип компилятора
- CFLAGS - опции для компиляции главного файла
- Компиляция каждого зависящего файла с расширением `%name%.o`
- Сборщик всей программы называется `"calculate"`
- Команда `clean`, для быстрого удаления всех файлов в том числе и временных



```
Обзор Терминал Sha, 5 Iyun 22:33 en
ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab_prog

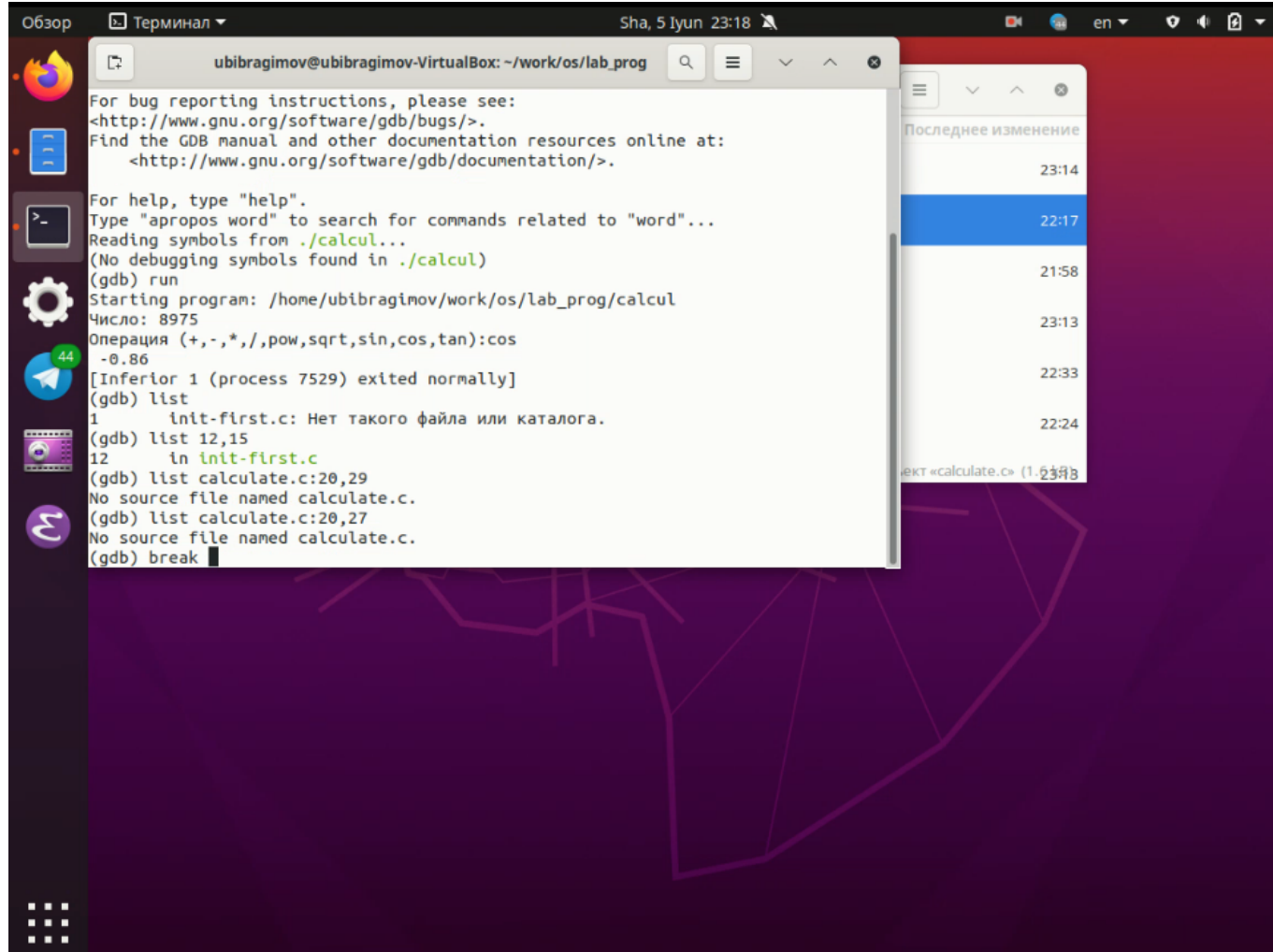
Чтение информации о состоянии... Готово
Предлагаемые пакеты:
  make-doc
Следующие НОВЫЕ пакеты будут установлены:
  make
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 1 пакетов не обно
влено.
Необходимо скачать 162 кВ архивов.
После данной операции объем занятого дискового пространства возрастёт на 393 кВ.
Пол:1 http://uz.archive.ubuntu.com/ubuntu focal/main amd64 make amd64 4.2.1-1.2 [162 kB]
Получено 162 кВ за 1с (272 кВ/с)
Выбор ранее не выбранного пакета make.
(Чтение базы данных ... на данный момент установлено 265354 файла и каталога.)
Подготовка к распаковке .../make_4.2.1-1.2_amd64.deb ...
Распаковывается make (4.2.1-1.2) ...
Настраивается пакет make (4.2.1-1.2) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$ make -f make
make: make: Нет такого файла или каталога
make: *** Нет правила для сборки цели «make». Останов.
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$ make -f make.makefile
gcc calculate.o main.o -o calcul -lm
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$
```

6. С помощью **gdb** выполните отладку программы **calcul** (перед использованием **gdb** исправьте **Makefile**):



```
Обзор Терминал Sha, 5 Iyun 23:18
ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab_prog
ubibragimov@ubibragimov-VirtualBox:~$ cd ~/work/os/lab_prog
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$ gdb ./calcul
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/>.

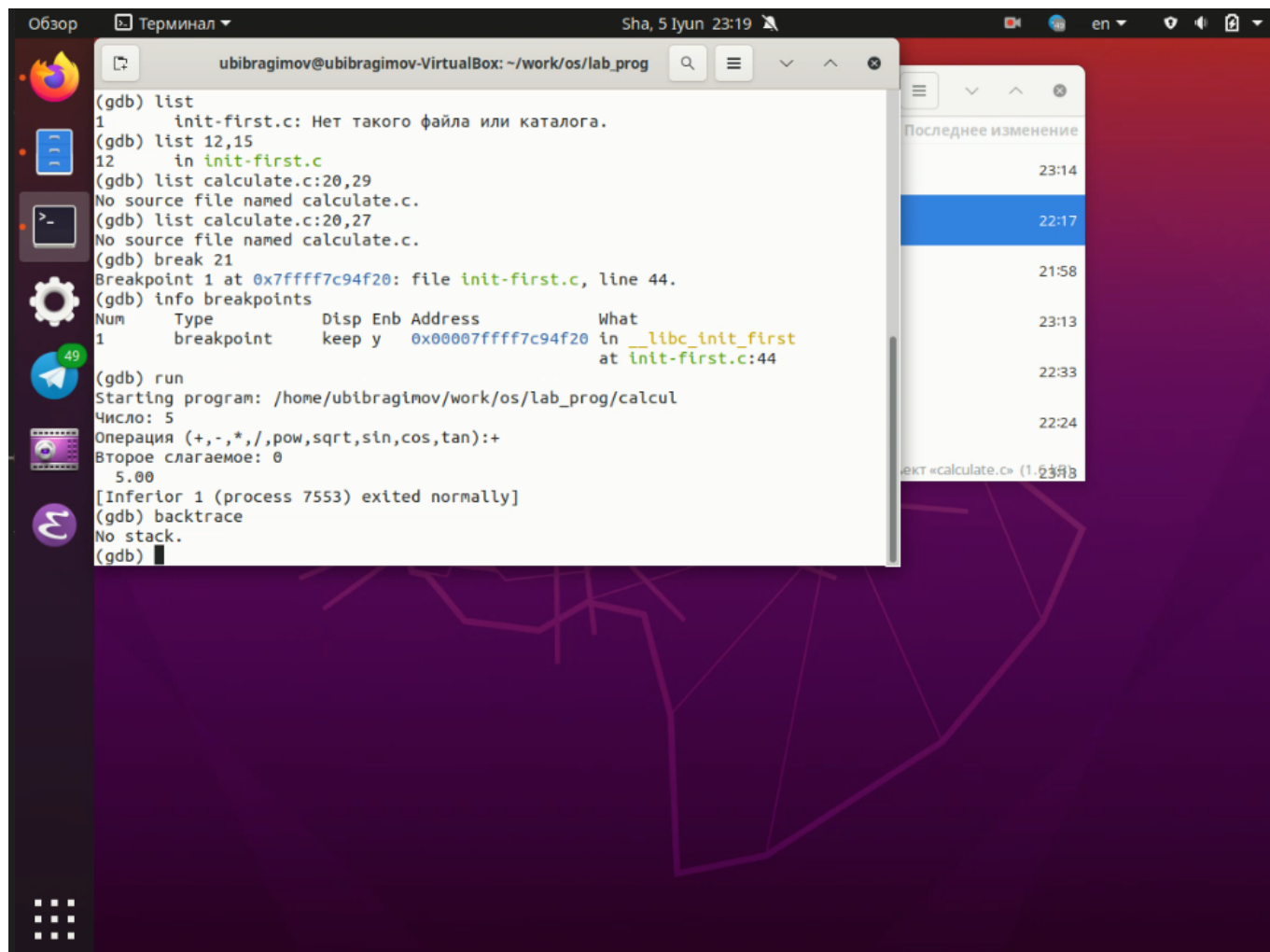
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /home/ubibragimov/work/os/lab_prog/calcul
Число: 8975
Операция (+,-,*,/,pow,sqrt,sin,cos,tan):
```



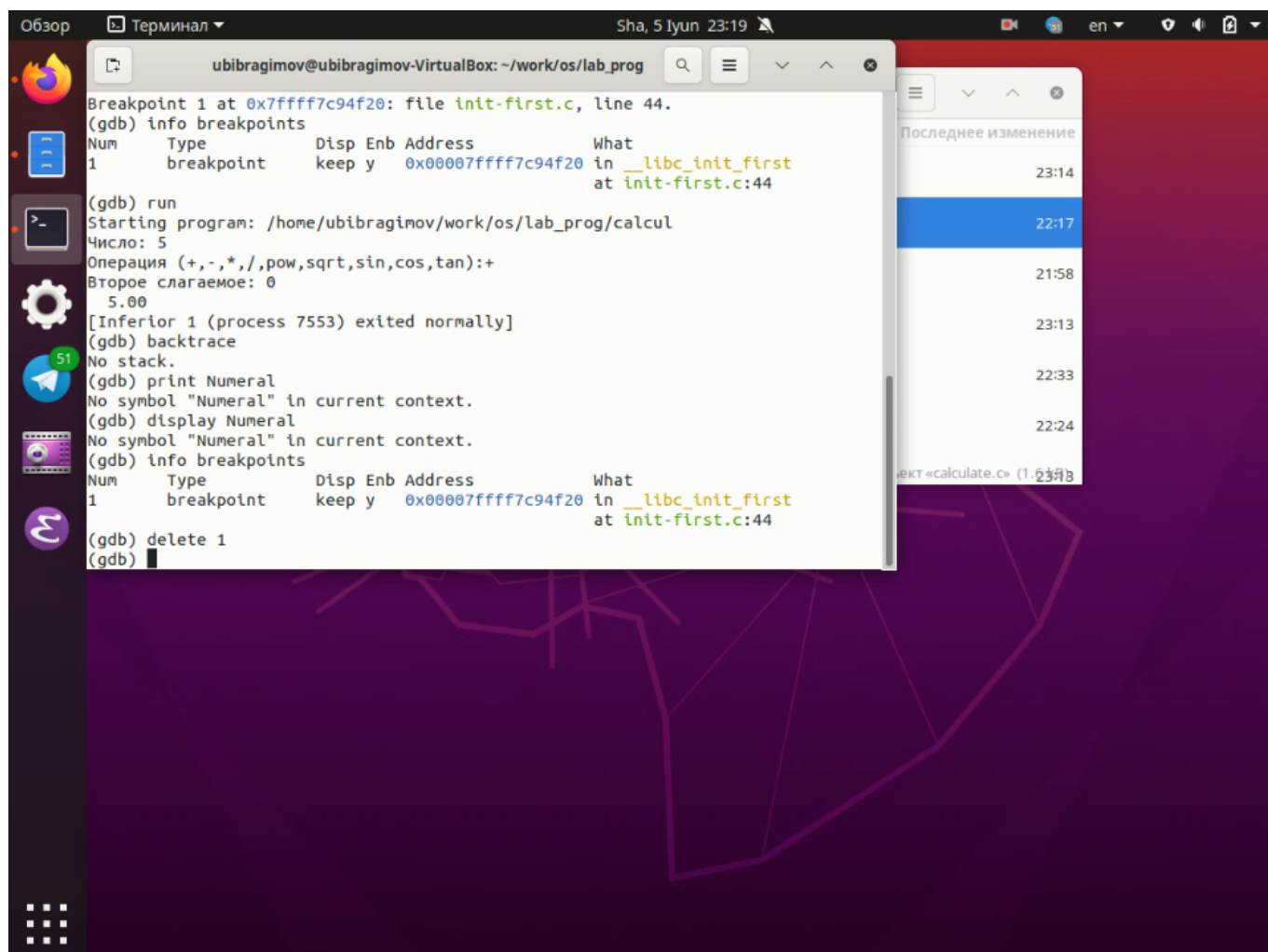
```
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(gdb) run
Starting program: /home/ubibragimov/work/os/lab_prog/calcul
Число: 8975
Операция (+,-,*,/,pow,sqrt,sin,cos,tan):cos
-0.86
[Inferior 1 (process 7529) exited normally]
(gdb) list
1      init-first.c: Нет такого файла или каталога.
(gdb) list 12,15
12      in init-first.c
(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
```

 Скриншот лабораторной работы №14



```
(gdb) list
1      init-first.c: Нет такого файла или каталога.
(gdb) list 12,15
12     in init-first.c
(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
Breakpoint 1 at 0x7ffff7c94f20: file init-first.c, line 44.
(gdb) info breakpoints
Num    Type         Disp Enb Address            What
1      breakpoint   keep y   0x00007ffff7c94f20 in __libc_init_first
                                at init-first.c:44
(gdb) run
Starting program: /home/ubibragimov/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan):+
Второе слагаемое: 0
5.00
[Inferior 1 (process 7553) exited normally]
(gdb) backtrace
No stack.
(gdb)
```



```
Breakpoint 1 at 0x7ffff7c94f20: file init-first.c, line 44.
(gdb) info breakpoints
Num    Type         Disp Enb Address            What
1      breakpoint   keep y   0x00007ffff7c94f20 in __libc_init_first
                                at init-first.c:44
(gdb) run
Starting program: /home/ubibragimov/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan):+
Второе слагаемое: 0
5.00
[Inferior 1 (process 7553) exited normally]
(gdb) backtrace
No stack.
(gdb) print Numeral
No symbol "Numeral" in current context.
(gdb) display Numeral
No symbol "Numeral" in current context.
(gdb) info breakpoints
Num    Type         Disp Enb Address            What
1      breakpoint   keep y   0x00007ffff7c94f20 in __libc_init_first
                                at init-first.c:44
(gdb) delete 1
(gdb)
```

6. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`


```

Обзор Терминал Sha, 5 Iyun 23:21
ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab_prog

Обрабатываются триггеры для man-db (2.9.1-1) ...
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2018

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:9:36: Function parameter Operation declared as manifest array (size
                    constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:13:7: 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:18:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:23:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:29:12: 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:31:19: Return value type double does not match declared type float:
    (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:36:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:37:15: Return value type double does not match declared type float:
    (pow(Numeral, SecondNumeral))
calculate.c:39:15: Return value type double does not match declared type float:
    (sqrt(Numeral))
calculate.c:41:15: Return value type double does not match declared type float:
    (sin(Numeral))
calculate.c:43:15: Return value type double does not match declared type float:
    (cos(Numeral))
calculate.c:45:15: Return value type double does not match declared type float:
    (tan(Numeral))
calculate.c:48:15: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$

```

```

Обзор Терминал Sha, 5 Iyun 23:21
ubibragimov@ubibragimov-VirtualBox: ~/work/os/lab_prog

calculate.c:18:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:23:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:29:12: 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:31:19: Return value type double does not match declared type float:
    (HUGE_VAL)
    To allow all numeric types to match, use +relaxtypes.
calculate.c:36:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:37:15: Return value type double does not match declared type float:
    (pow(Numeral, SecondNumeral))
calculate.c:39:15: Return value type double does not match declared type float:
    (sqrt(Numeral))
calculate.c:41:15: Return value type double does not match declared type float:
    (sin(Numeral))
calculate.c:43:15: Return value type double does not match declared type float:
    (cos(Numeral))
calculate.c:45:15: Return value type double does not match declared type float:
    (tan(Numeral))
calculate.c:48:15: Return value type double does not match declared type float:
    (HUGE_VAL)

Finished checking --- 15 code warnings
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$ splint main.c
Splint 3.1.2 --- 20 Feb 2018

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)
main.c: (in function main)
main.c:12:3: 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:14:3: Return value (type int) ignored: scanf("%s", Oper...

Finished checking --- 3 code warnings
ubibragimov@ubibragimov-VirtualBox:~/work/os/lab_prog$

```


Вывод

В ходе выполнения Лабораторной работы №14, были приобретены навыки по разработке, анализу, тестированию и отладки приложений в Linux

Контрольные вопросы

1. Можно использовать стандартные команды для получения справки - `man gcc`, `man make`, `man gdb` ну или вместо утилиты `man` использовать команду `--help`
2. Основные этапы разработки приложений в Unix: Создание исходного кода (написание в IDE) -> Сохранение промежуточных файлов или альтернативных веток разработки исходного кода -> Компиляция исходных файлов или их интерпретация в зависимости от выбранного языка программирования и/или системы сборки проектов -> Тестирование проекта который был собран -> Запись в соответствующую ветку разработки Git (main или dev, по-умолчанию)
3. Суффикс - нужен для определения расширения в контексте файловой системы или компилятора с помощью которого будет производиться компиляция или интерпретация исходного кода в работающую программу (например `hello1.py` компилируется только `ipython`, а вот `hello2.c` компилируется только `gcc`, `Cmake`)
4. Компилятор Си предназначен для компиляции внутренних файлов системы без полного скачивания программ, а просто скачав исходный код системных утилит и произвести с помощью встроенного компилятора компиляцию системных утилит
5. Утилита `make` - предназначена для упрощения разработки приложений, путем написания файла конфигурации который описывает пути компиляции для компилятора языка программирования
6. Можно использовать пример из лабораторной работы
7. Пошаговая отладка программ (трассировка) - её суть заключается в пошаговом выполнении каждой строки кода
8. Основные команды отладчика `gdb`:
 - `backtrace` - вывод на экран путь к текущей точке останова.
 - `break` - установить точку останова (строка или функция)
 - `clear` - удалить все точки останова в функции
 - `continue` - продолжить выполнение программы
 - `delete (n)` - удалить точку останова
 - `display` - добавить выражение в список выражений, значения которых отображаются при достижении точки останова программы
 - `finish` - выполнить программу до момента выхода из функции
 - `info breakpoints` - вывести на экран список используемых точек останова
 - `info watchpoints` - вывести на экран список используемых контрольных выражений
 - `list` - вывести на экран исходный код (в качестве параметра может быть указано название файла и через двоеточие номера начальной и конечной строк)
 - `next` - выполнить программу пошагово, но без выполнения вызываемых в программе функций
 - `print` - вывести значение указываемого в качестве параметра выражения
 - `run` - запуск программы на выполнение
 - `set[variable]` - установить новое значение переменной
 - `step` - пошаговое выполнение программы

- `watch` - установить контрольное выражение, при изменении значения которого программа будет остановлена
9. Мои действия при отладке программ: Запустил `Makefile` -> Начал отладку (`run`) -> Вывел содержимое `main` файла -> Установил точку останова в `main` файле -> Продолжил выполнение (`run`) -> Использовал команды `print & display` для вывод промежуточных данных -> Удалил точку останова -> Закончил отладку
 10. Нейтральная реакция компилятора, т.е. программных ошибок обнаружено не было
 11. `cppcheck`, `splint`, `cscope` и другие
 12. Проверка корректности аргументов и поиск ошибок и значений в программе которые могут быть улучшены, а также оценка всей программы