Комитет по образованию Правительства Санкт-Петербурга  
Государственное бюджетное учреждение  
среднего профессионального образования  
«Колледж информационных технологий»

МДК 03.01

«Технология разработки программного обеспечения»

**отчёт**

по практической работе №11

Выполнил студент 474 гр.:

Черных Артём

Преподаватель:

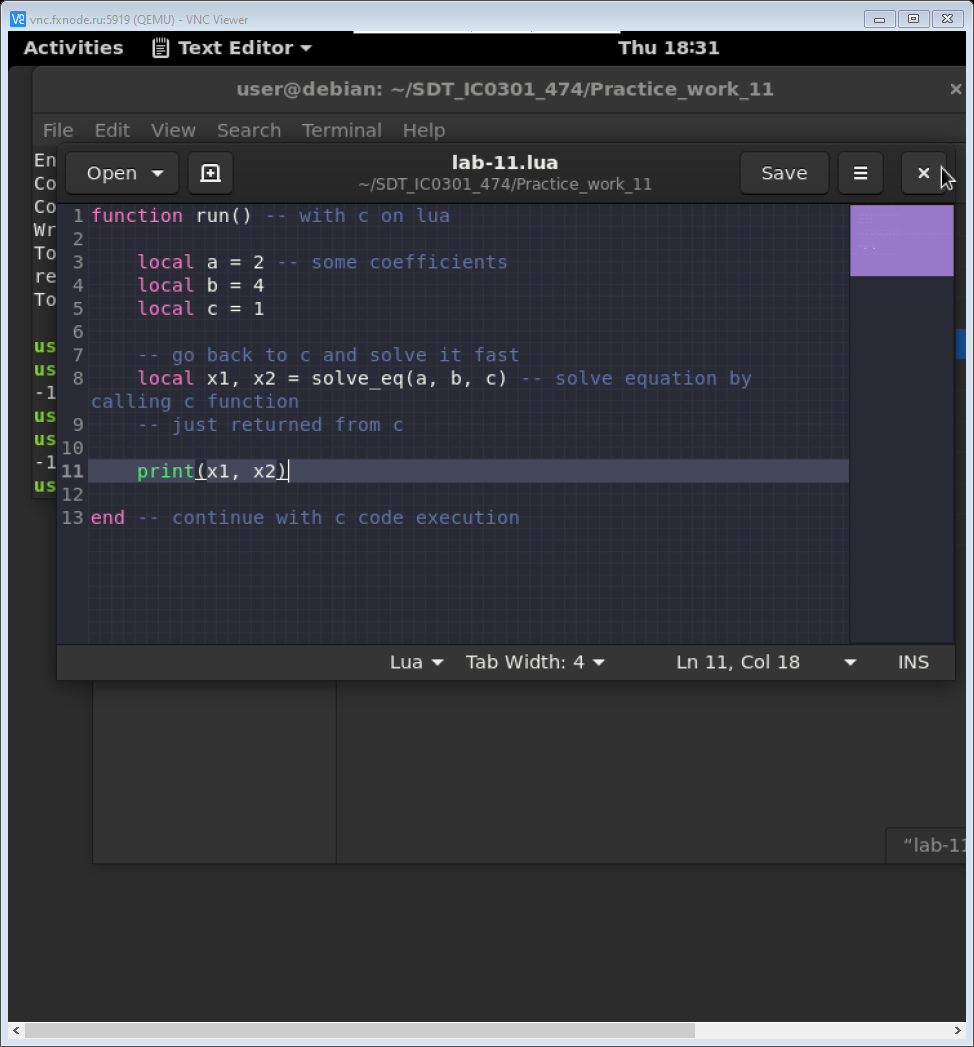
Фомин А.В.

Санкт-Петербург 2020

**Вызов функции lua из программы на c**

**Основное задание**

1. Создан сценарий на Lua, вычисляющий корни квадратного уравнения, при помощи внешней функции solve\_eq, Входные данные и запуск вычисления на C обёрнут в функцию run. Смотрите на рисунке 1.



Черных Артём

Рисунок 1 – Файл «lab-11.lua»

1. Создан код на C для решения квадратного уравнения (функция my\_solve\_e q), на входе подаются 3 числа, на выходе – 2 число. Функция lua\_pushnil( ) используется, если уравнение имеет мене 2 корней.

Главная функция main( ) создаёт новую виртуальную машину функцией luaL\_newstate( ). Загружает текст сценария из файла функцией luaL\_loadfile( ).

Запускает на исполнение файла lua\_pcall( ), только после этого появятся в виртуальной машине, глобальные переменные из файла.

Чтобы вычислить корни, выполнено деление C функцией my\_solve\_eq по именем solve\_eq с помощь последовательного вызова lua\_pushfunction( ) и lua\_setglobal( ). И наконец запуск функции run( ) в сценарии на Lua, вызванный функцией lua\_get\_gloabal( ) и lua\_pcall( ).

После окончания работы со сценарием, уничтожается виртуальная машина функцией lua\_close( ).

Файл «lab-11.c»:

#include <math.h>

#include <lua.h>

#include <lualib.h>

#include <lauxlib.h>

static int my\_solve\_eq(lua\_State \*l)// solve equation

{

float a = (float) luaL\_checknumber(l, 1);// get 3 arguments

float b = (float) luaL\_checknumber(l, 2);

float c = (float) luaL\_checknumber(l, 3);

float d = b \* b - 4.0f \* a \* c;

if(d < 0)// no roots

{

lua\_pushnil(l);// returnt 2 nil

lua\_pushnil(l);

}

else if (d > 0)// 2 roots

{

float x1 = (-b - sqrtf(d)) / (2.0f \* a);

float x2 = (-b + sqrtf(d)) / (2.0f \* a);

lua\_pushnumber(l, x1);// return 2 number

lua\_pushnumber(l, x2);

}

else// 1 roots

{

float x = (-b + sqrtf(d)) / (2.0f \* a);

lua\_pushnumber(l, x);

lua\_pushnil(l);// 2nd is nil

}

return 2;// return 2 variables

}

int main()

{

lua\_State \*l = luaL\_newstate();// create new virtual machine

luaL\_openlibs(l);// load standard libraries

luaL\_loadfile(l, "lab-11.lua");// load script file into vm

lua\_pcall(l, 0, 0, 0);// execute entire file to create global variables

lua\_pushcfunction(l, my\_solve\_eq);// share a C function to lua script

lua\_setglobal(l, "solve\_eq");// name it solve\_eq, make it globally available

lua\_getglobal(l, "run");// prepare to execute lua function

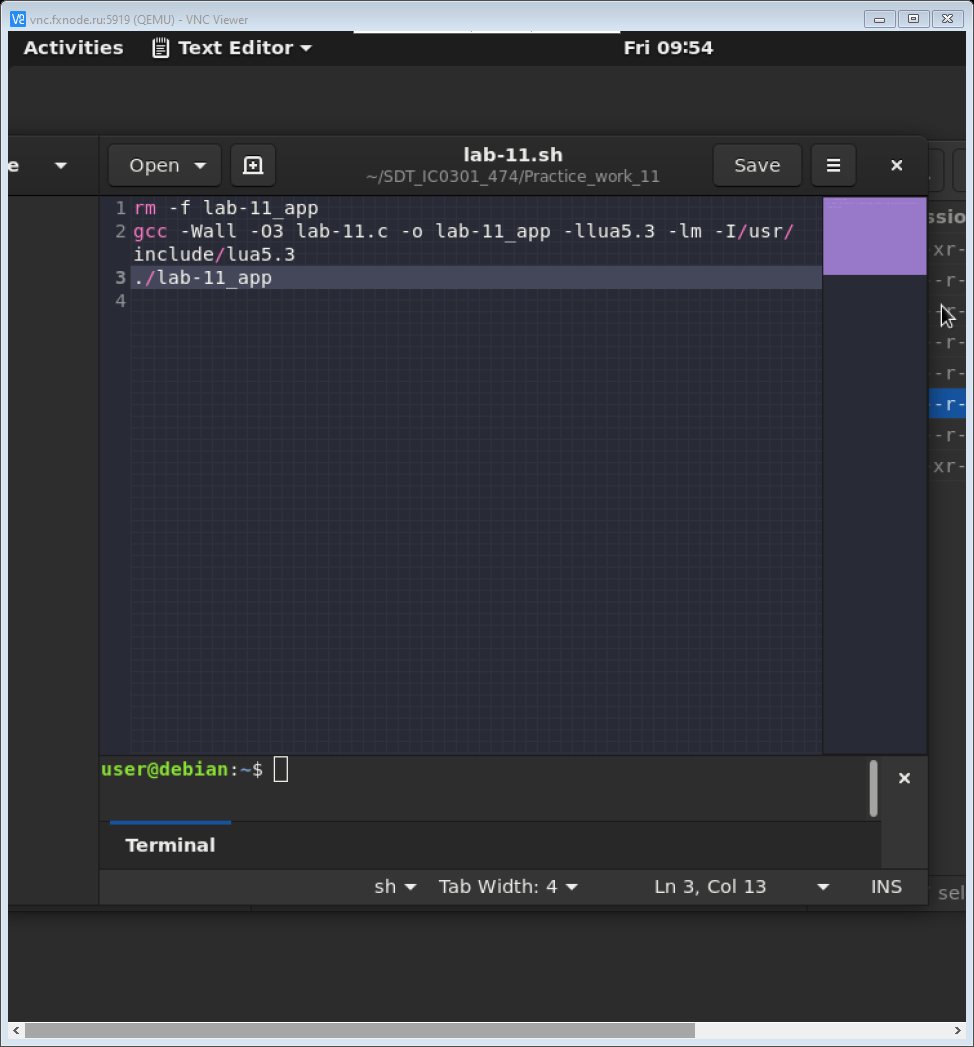
lua\_pcall(l, 0, 0, 0);// execute a function within lua

lua\_close(l);// delete lua vm

return 0;

}

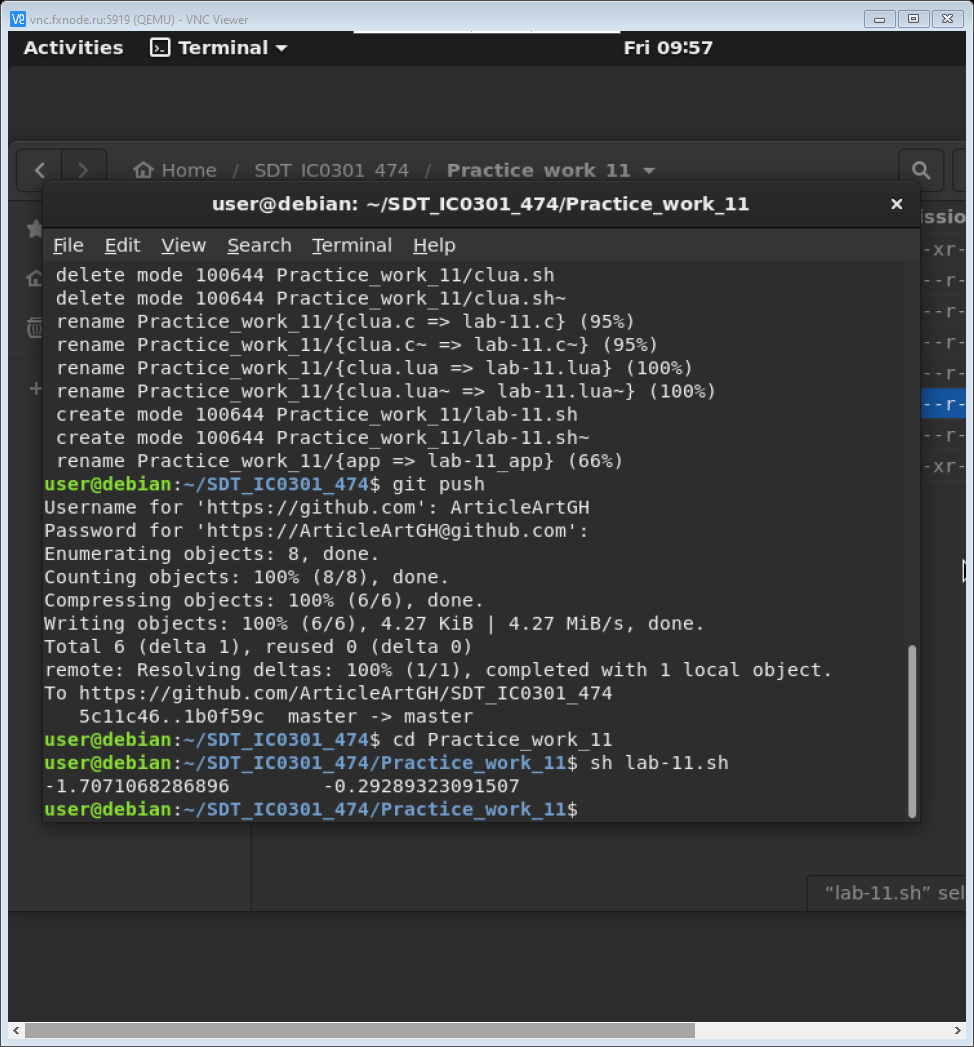
1. Сборка и запуск приложения на C, используя shell-скрипт на рисунке 2.



Черных Артём

Рисунок 2 – Файл «lab-11.sh»

1. Пример работы программы на рисунке 3.



Черных Артём

**Дополнительно**

Перед началом работы, нужно чтобы были установлены следующие пакеты:

* gcc – компилятор языка C
* libc6-dev – стандартная библиотека языка C (пакет разработчика)
* liblua-dev – библиотека языка Lua (пакет разработчика)

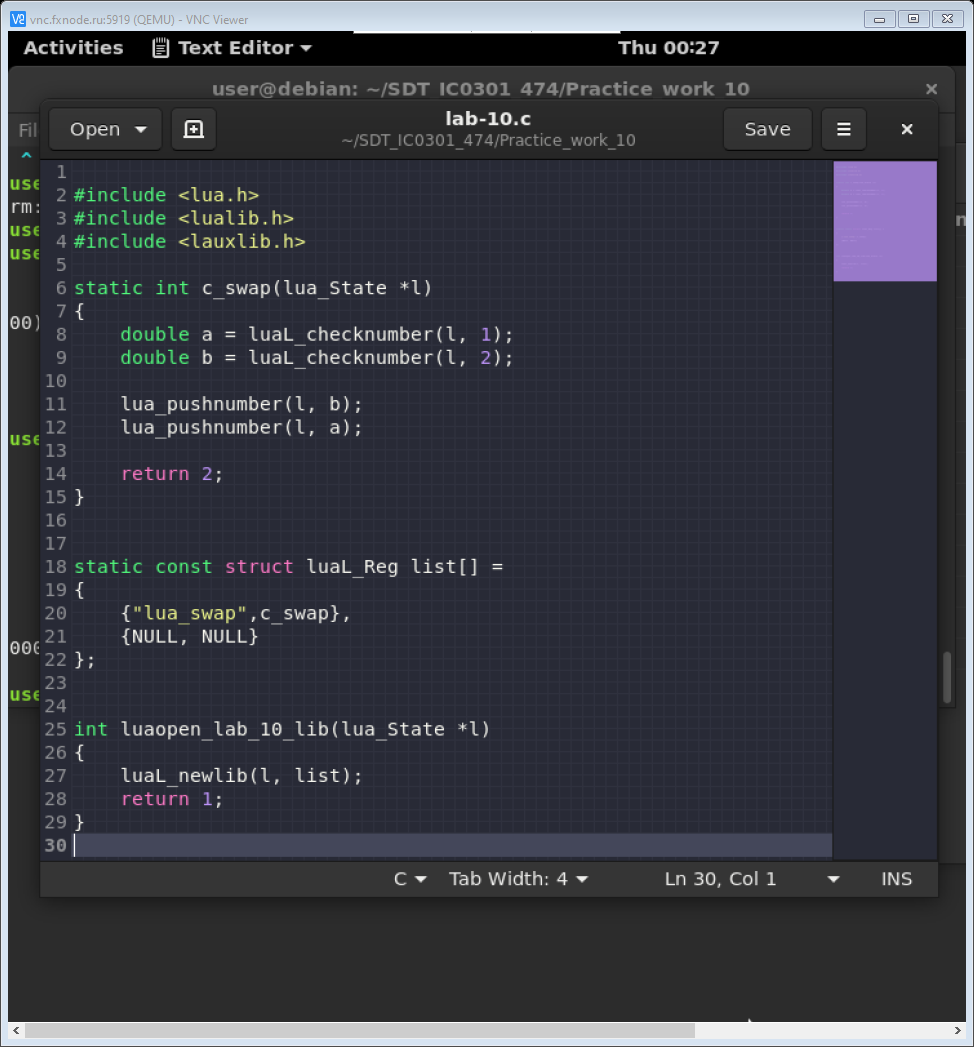
1. Написание кода библиотеки на языке C. Реализация функции обмена значений двух переменных и экспорт в Lua. Для этого подключены заголовочные файлы: lua.h, lualib.h и lauxlib.h.

Массив со списком экспортируемых функций типа luaL\_Reg. Последний элемент {NULL, NULL} для обозначения конца списка. Каждый элемент списка имеет 2 поля – имя функции в языке Lua и указатель на функцию C.

Описание функции luaopen\_имя\_библиотеки, в которой зарегистрирован список функции luaL\_newlib( ). Во всех вызываемых функциях из Lua - параметр luaState – это указатель на структуру с описанием конкретной виртуальной машины языка Lua, которая вызывает C.

Экспортируемая функция возвращает количество добавленных ею в стек значений (в данном примере – 2 шт). Функция lua\_pushnumber( ) добавляет значение в стек, а luaL\_checknumber( ) извлекает значение из стека и проверяет их тип на соответствие числовому. При вызове функции из сценария на Lua, передача параметров и результатов осуществляется через стек.

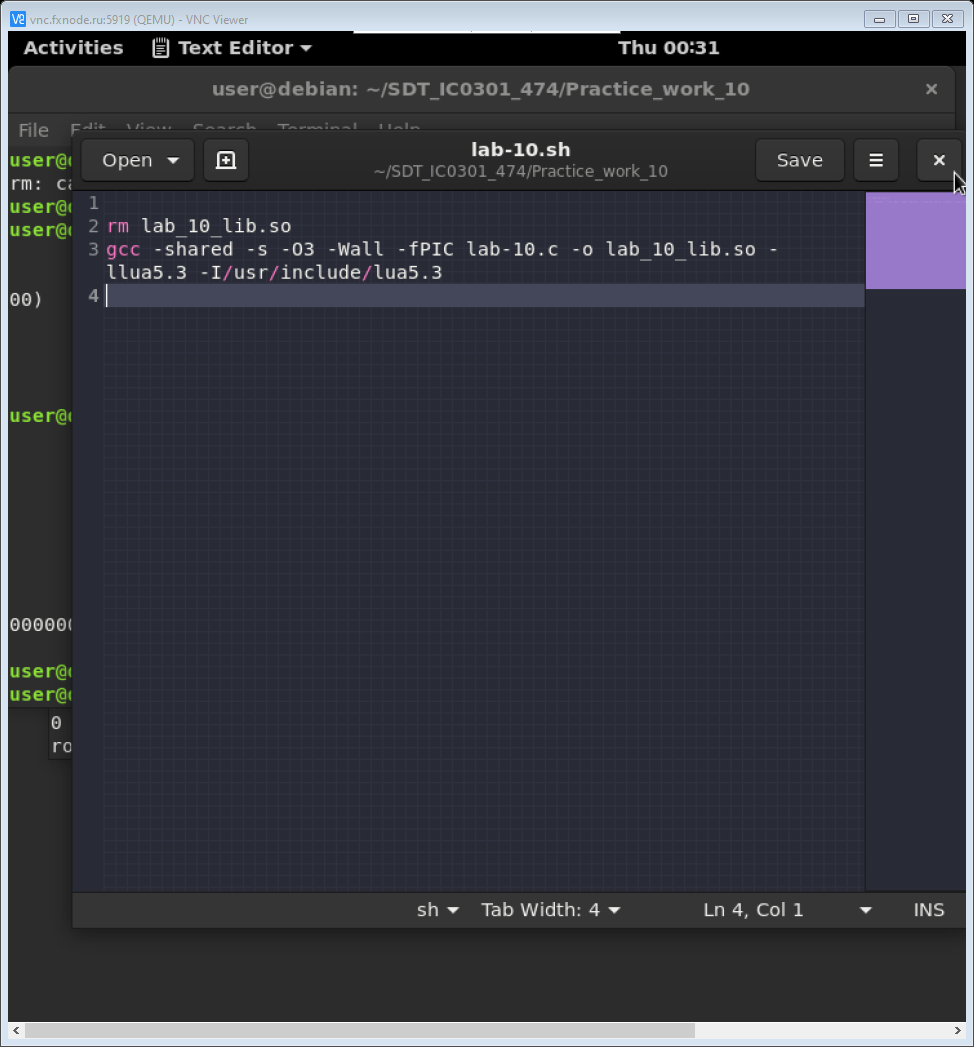
Смотрите на рисунке 4.



Черных Артём

Рисунок 4 – Файл «lab-10.c»

1. Сборка библиотеки через shell-скрипт на рисунке 5.

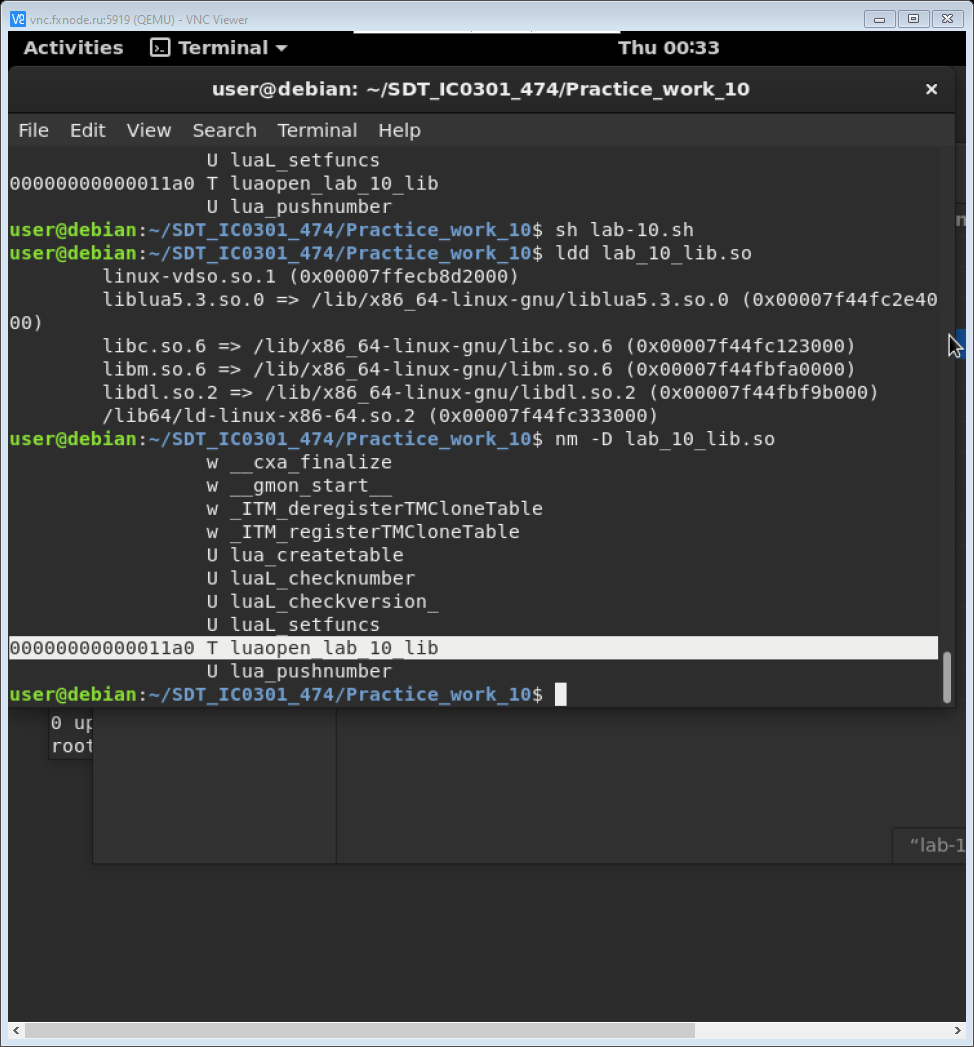


Черных Артём

Рисунок 5 – Файл «lab-10.sh»

После сборки появится файл библиотеки с расширением .so (shared object).

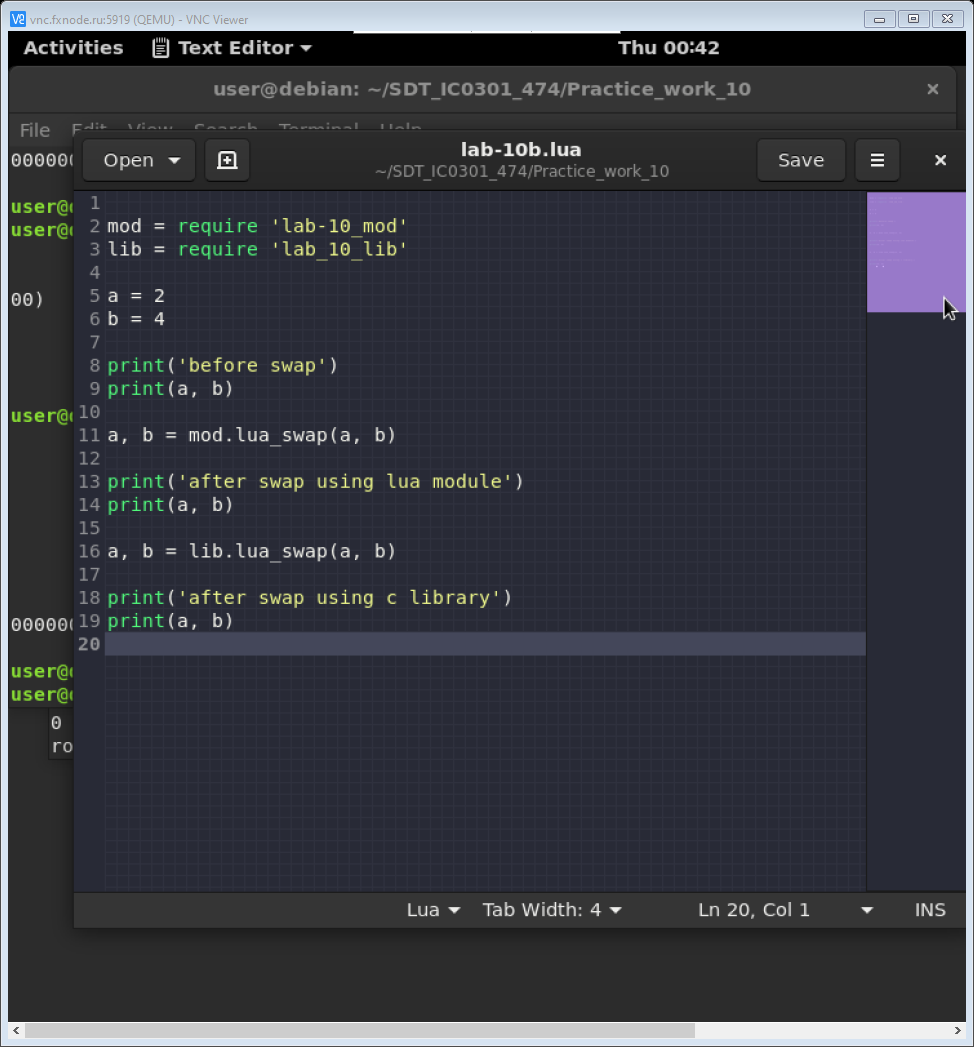
1. Используя команды ldd и nm можно получить список внешних зависимостей и содержание таблиц импорта и экспорта исполнимого файла, смотрите на рисунке 6.



Черных Артём

Рисунок 6 – Работа с файлом «lab\_10\_lib.so»

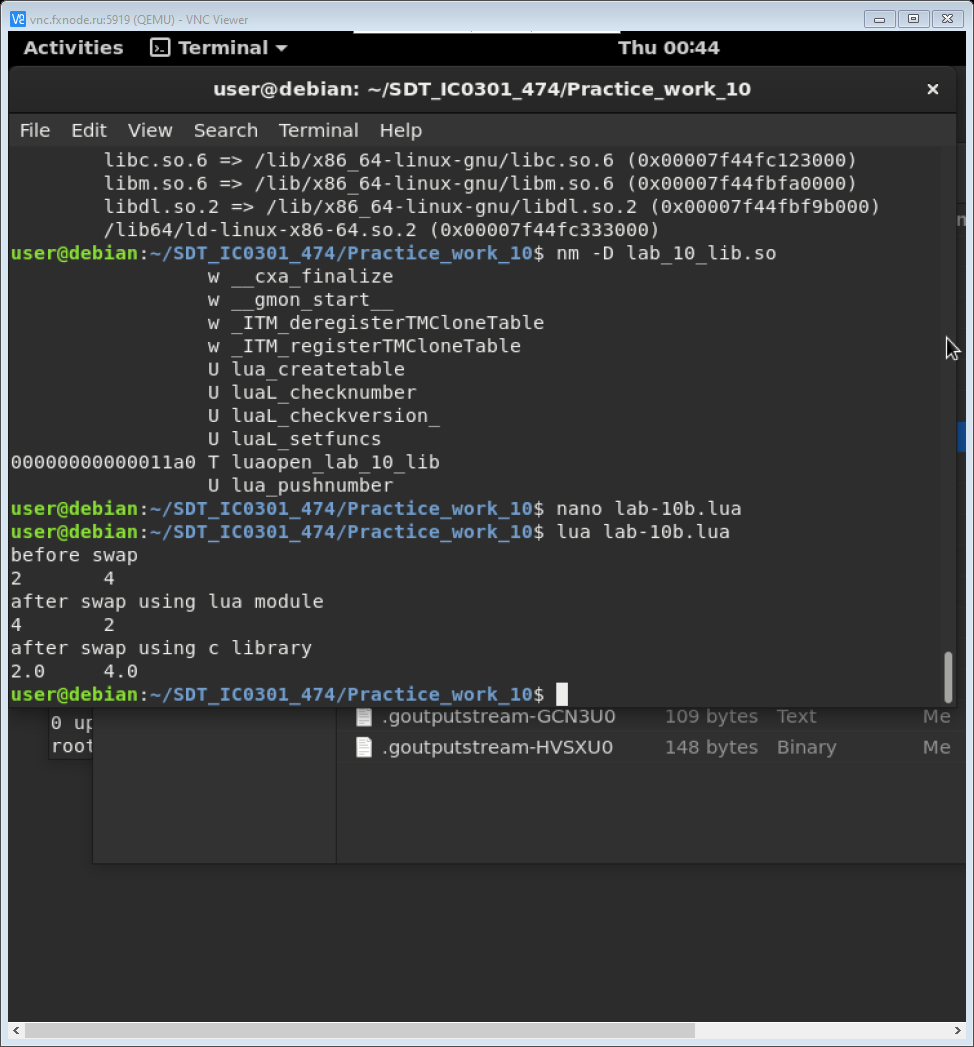
1. Для использование полученной библиотеки, добавлен в изначальный сценарий на Lua директива require. Смотрите на рисунке 7.



Черных Артём

Рисунок 7 – Файл «lab-10b.lua»

1. Пример работы приложения на рисунке 8.



Черных Артём

Рисунок 8 – Приложение работает