МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

|  |  |
| --- | --- |
|  | Утвержден на заседании кафедры  «Вычислительная техника»  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г.  Заведующий кафедрой                                              М.А. Митрохин |
|  |  |

**ОТЧЕТ ПО УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКЕ**

(2024/2025 учебный год)

                                         \_\_\_\_

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения2семестр 4

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

*(должность, ученая степень, ученое звание, Ф.И.О.)*

Руководитель практики к.т.н., доцент, Карамышева Н.С.

*(должность, ученая степень, ученое звание)*

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

|  |  |
| --- | --- |
|  | Утвержден на заседании кафедры  «Вычислительная техника»  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г.  Заведующий кафедрой                                              М.А. Митрохин |
|  |  |

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2024/2025 учебный год)

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения                  2                 семестр                 4

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

*(должность, ученая степень, ученое звание, Ф.И.О.)*

Руководитель практики к.т.н., доцент, Карамышева Н.С.

*(должность, ученая степень, ученое звание)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Планируемая форма работы во время практики | Количество часов | Календарные сроки проведения работы | Подпись  руководителя  практики от вуза |
| 1 | Выбор темы и разработка индивидуального плана проведения работ (распределенная практика) | 26 | 20.06.25 –  23.06.25 |  |
| 2 | Подбор и изучение материала по теме работы (распределенная практика) | 26 | 24.06.25 –  27.06.25 |  |
| 3 | Разработка алгоритма (распределенная практика) | 26 | 28.06.25 –  1.07.25 |  |
| 4 | Описание алгоритма и программы  (распределенная практика) | 30 | 2.07.25–  7.07.25 |  |
| 5 | Тестирование  (сосредоточенная практика) | 30 | 8.07.25 –  12.07.25 |  |
| 6 | Получение и анализ результатов  (сосредоточенная практика) | 38 | 13.07.25 –  17.07.25 |  |
| 7 | Оформление отчёта  (сосредоточенная практика) | 40 | 18.07.25 –  19.07.25 |  |
|  | **Общий объём часов** | 216 |  |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЁТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2021/2022 учебный год)

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения2семестр 4

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Заитов М.Э. выполнял практическое задание «Установка операционной системы Linux на виртуальную машину и написание простой программы». На первоначальном этапе был изучен VirtualBox, а также ознакомился с операционной системой Linux. Написал простую программу на языке Си. Также, осуществил работу с файлами. Протестировал и отладил программу. Оформил отчёт.

Бакалавр Заитов М.Э. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Руководитель Карамышева Н.С. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2022 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2021/2022 учебный год)

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения2семестр 4

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

В процессе выполнения практики Заитов М.Э. решал следующие задачи: установка операционной системы Libux Ubuntu на виртуальную машину VirtualBox, написание программы на языке Си, оформление отчета.

За период выполнения практики были получены навыки установки операционной системы Linux, а так ее настройки. Во время выполнения работы Заитов М.Э. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике и программированию.

За выполнение работы Заитов М.Э. заслуживает оценки «\_\_\_\_\_\_\_\_\_\_».

Руководитель практики к.т.н., доцент, Карамышева Н.С. « » 2025 г.

**Содержание**

Введение...................................................................................................................7

1 Установка ОС Linux на виртуальную машину..................................................8

1.1 Установка виртуальной машины VirtualBox................................................8

1.2 Создание виртуальной машины...................................................................12

1.3 Установка ОС Ubuntu на виртуальную машину........................................16

1.4 ОС Ubuntu......................................................................................................22

2 Интегрированная среда разработки Qt Creator................................................23

2.1 Описание Qt Creator......................................................................................23

2.2 Описание установки......................................................................................24

3 Разработка программы на языке С в среде Qt Creator....................................26

3.1 Формулировка задачи...................................................................................26

3.2 Создание проекта в Qt Creator......................................................................26

3.3 Описание программы....................................................................................29

3.4 Компиляция и отладка программы в среде Qt Creator...............................30

Заключение.............................................................................................................33

Список использованных источников и литературы...........................................34

Приложение А........................................................................................................35

Приложение Б........................................................................................................36

**Введение**

Linux, или GNU/Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU. В Linux есть много ответвлений, дистрибутивов таких, как Ubuntu, Linux Mint, Fedora, CentOS, Debian и другие. Большинство из них совершенно бесплатны и распространяются с открытым исходным кодом, что делает их более удобными для разработчиков программного обеспечения. Основным преимуществом ОС Linux является безопасность, отсутствие вирусов и угрозы со стороны вредоносных программ. Популярные дистрибутивы Linux очень просты в установке и не требовательны к ресурсам. Linux идеален для программистов и веб-разработчиков, поскольку имеет встроенные компиляторы, удобную командную строку, через которую можно даже устанавливать популярные приложения. Linux прекрасно работает как на обычных домашних компьютерах, так и на рабочих компьютерах и серверах.

Виртуальная машина (ВМ или VM) — это виртуальный компьютер, который использует выделенные ресурсы реального компьютера (процессор, диск, адаптер). Эти ресурсы хранятся в облаке и позволяют ВМ работать автономно. ВМ может работать в отдельном окне как программа или запускаться через панель управления. Виртуальная машина может использоваться для развертывания двух и более операционных систем на одном физическом устройстве. Таким образом, установив на виртуальную машину операционную систему Linux, можно работать в ней, не перезагружая компьютер с основной ОС Windows. К тому же ВМ дает возможность одновременной работы в нескольких операционных системах, а также взаимодействие между ними. Еще одним достоинством виртуальной машины является то, что работа в ней никак не навредит реальному компьютеру.

Задачей данной практической работы стала установка операционной системы Linux на виртуальную машину и разработка программы на языке С в установленной в данной ОС интегрированной среде разработки.

1 Установка ОС Linux на виртуальную машину

1.1 Установка виртуальной машины VirtualBox

Virtualbox — кроссплатформенное программное обеспечение, обладающее понятным и дружественным интерфейсом.

Для установки виртуальной машины нужно скачать с официального сайта установочный файл и запустить его.

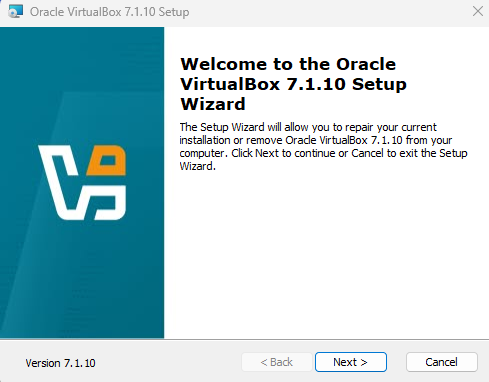


Рисунок 1 — Запуск установочного файла VirtualBox

Затем выбираем компоненты для установки и расположение.



Рисунок 2 — Выбор компонентов для установки и расположения

Далее отмечаем «Создать ярлык на рабочем столе».

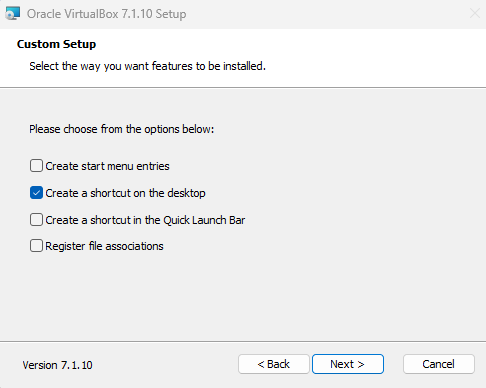


Рисунок 3 — Создание ярлыка на рабочем столе

Потом соглашаемся на отключение Интернет-соединения на время установки VirtualBox.

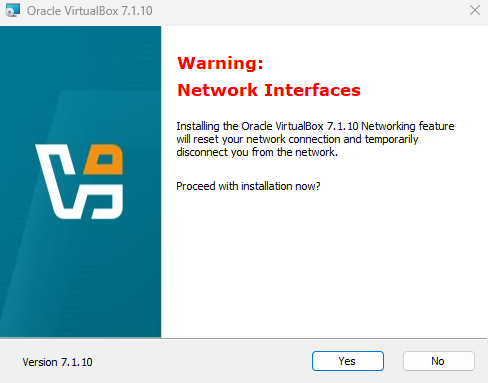


Рисунок 4 — Временное отключение Интернет-соединения

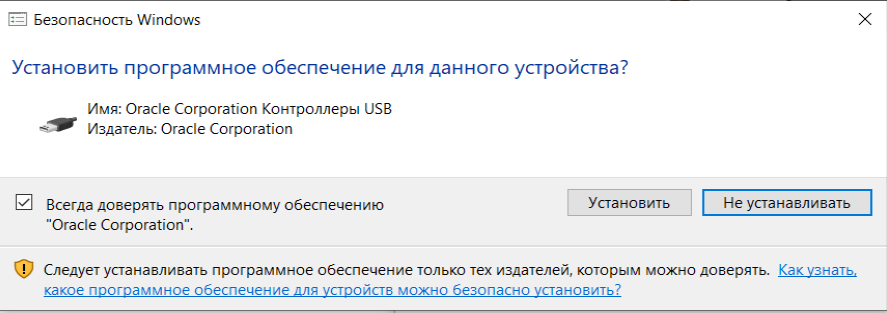


Рисунок 5 — Установка программного обеспечения для ВМ

Затем устанавливаем программное обеспечение для виртуальной машины.

Далее нажимаем кнопку «Завершить» для выхода из Мастера установки

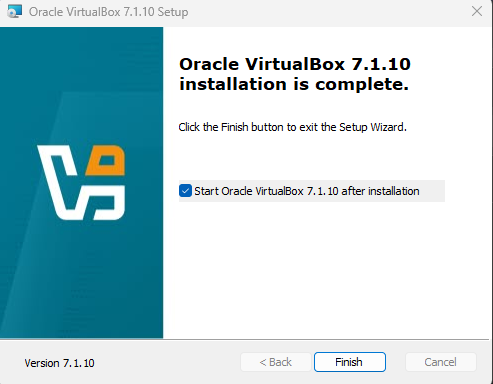


Рисунок 6 — Выход из Мастера установки

Потом устанавливаем VirtualBox.

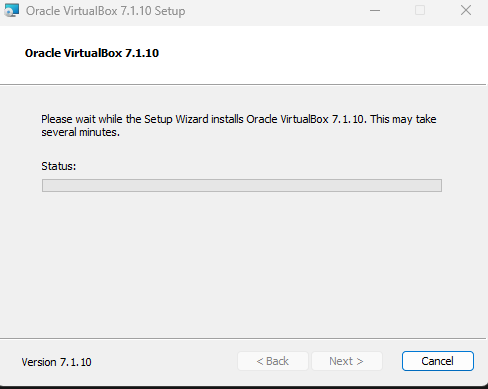


Рисунок 7 — Установка VirtualBox

Установка VirtualBox успешно завершена, для проверки запускаем ВМ.

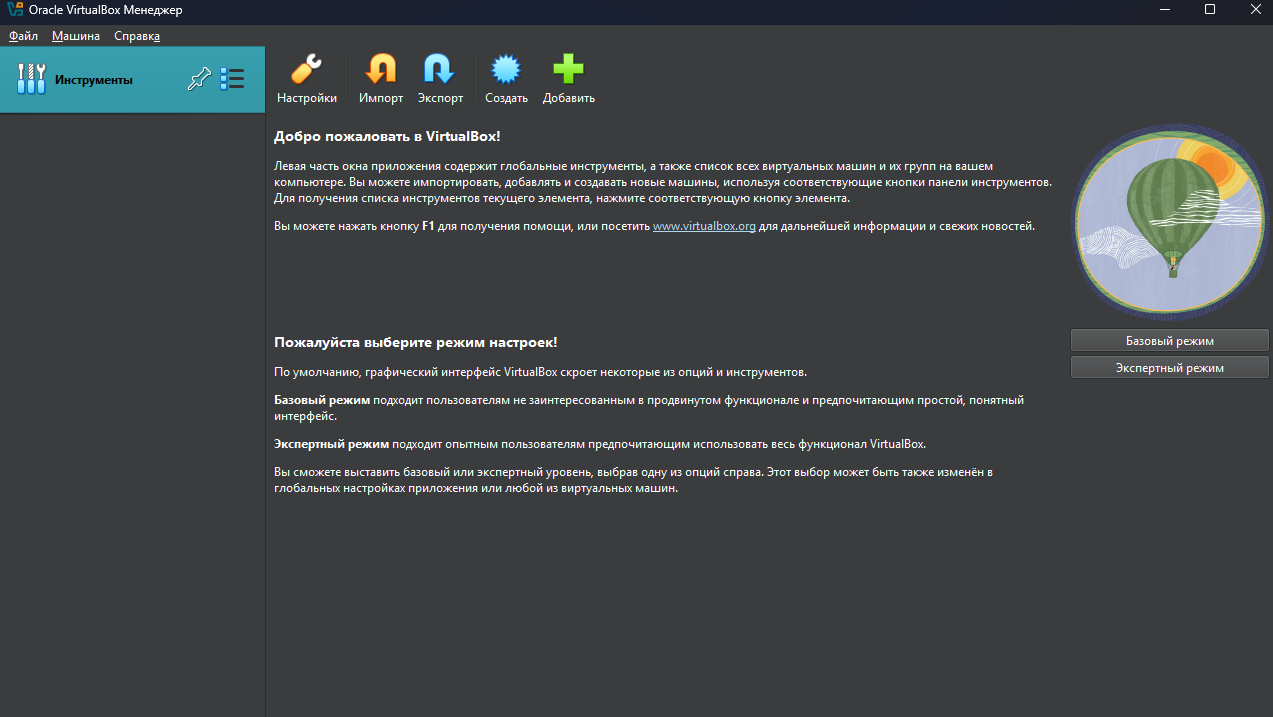


Рисунок 8 — Запуск VirtualBox

1.2 Создание виртуальной машины

Для создания виртуальной машины запускаем VirtualBox (рисунок 8) нажимаем кнопку «Создать».

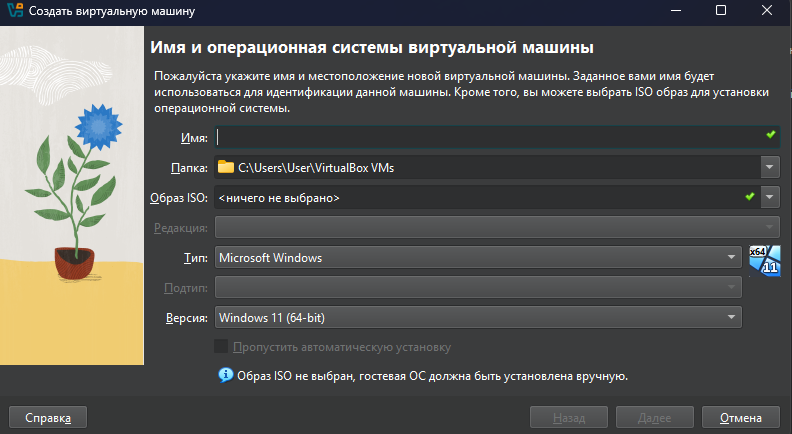


Рисунок 9 — Создание виртуальной машины

Затем указываем имя, местоположение, тип и версию операционной системы новой виртуальной машины.

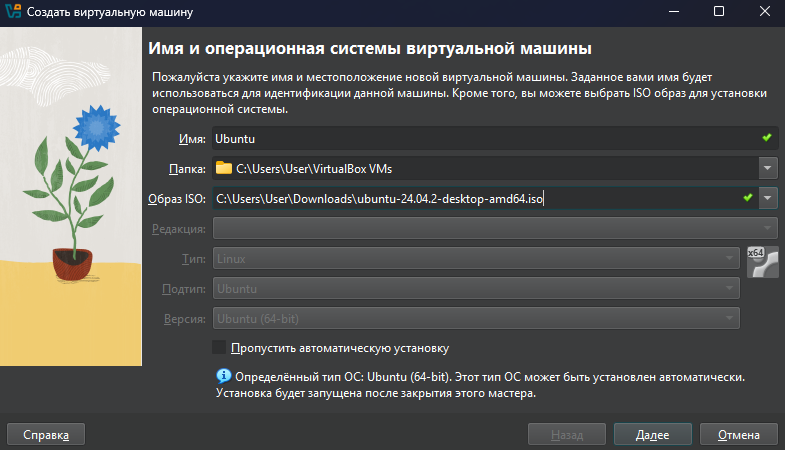


Рисунок 10 — Указание имени и типа ОС

Далее указываем объем оперативной памяти виртуальной машины.

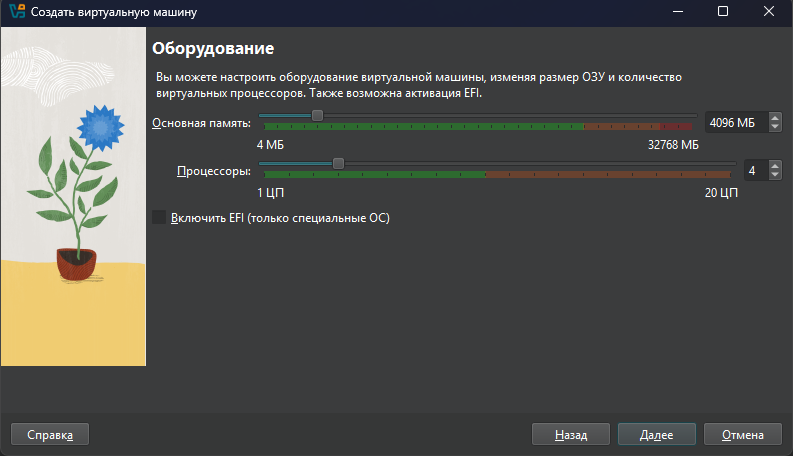


Рисунок 11 — Объем оперативной памяти и число процессоров

Потом создаем новый виртуальный жесткий диск.

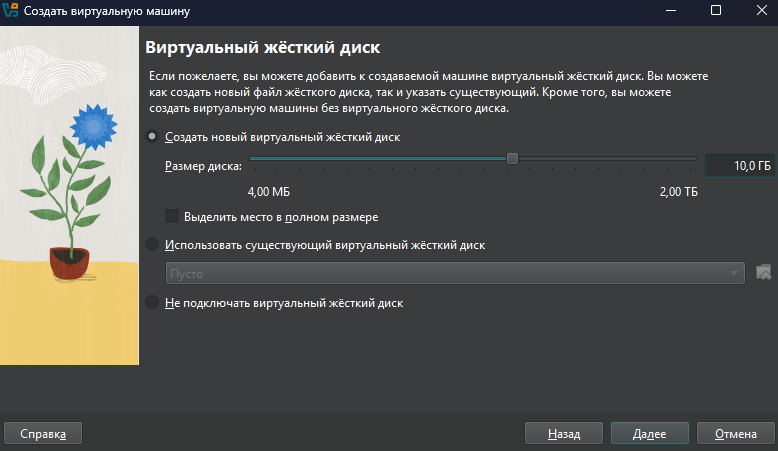


Рисунок 12 — Создание жесткого диска

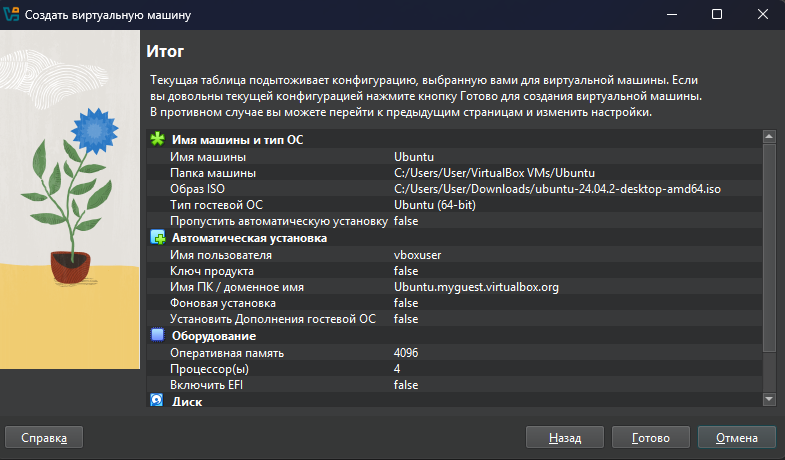


Рисунок 13 — Конечный этап создания системы «Ubuntu»

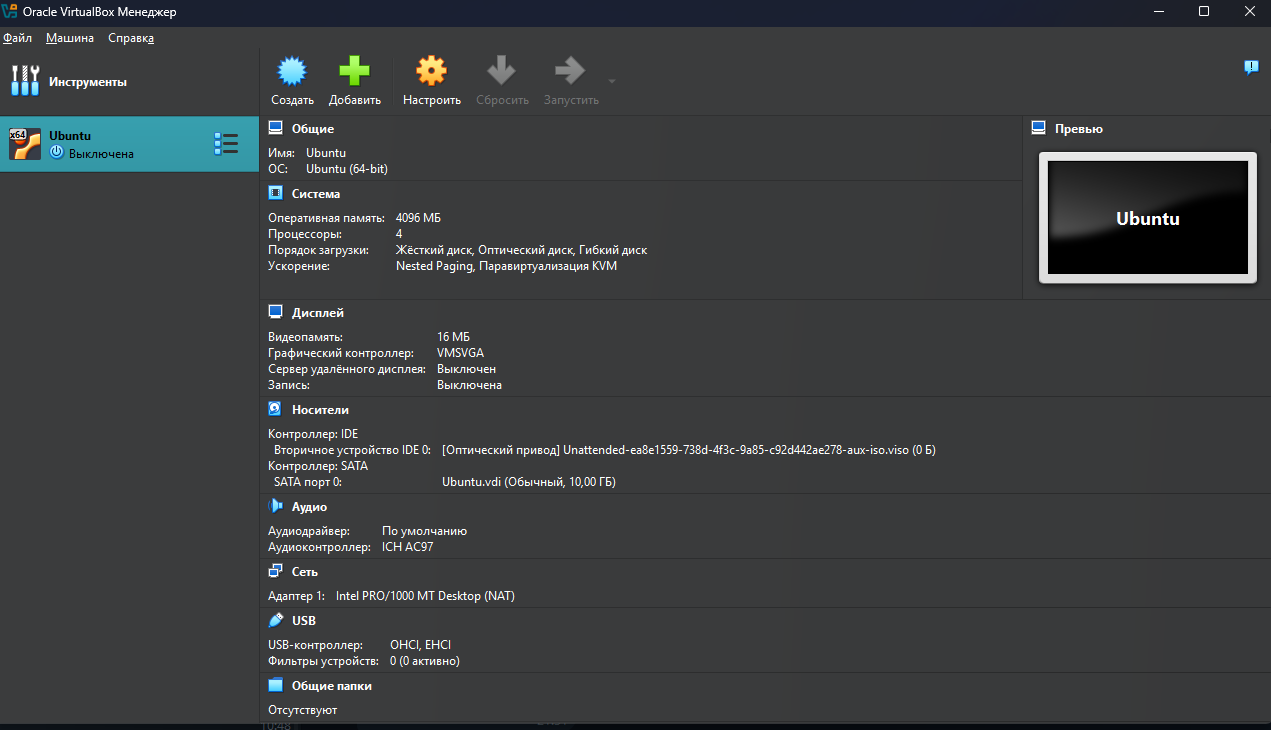


Рисунок 16 — Виртуальная машина «Ubuntu»

1.3 Установка ОС Ubuntu на виртуальную машину

Для установки ОС Ubuntu нужно скачать с официального сайта файл, содержащий дистрибутив операционной системы, и запустить созданную виртуальную машину «Ubuntu».

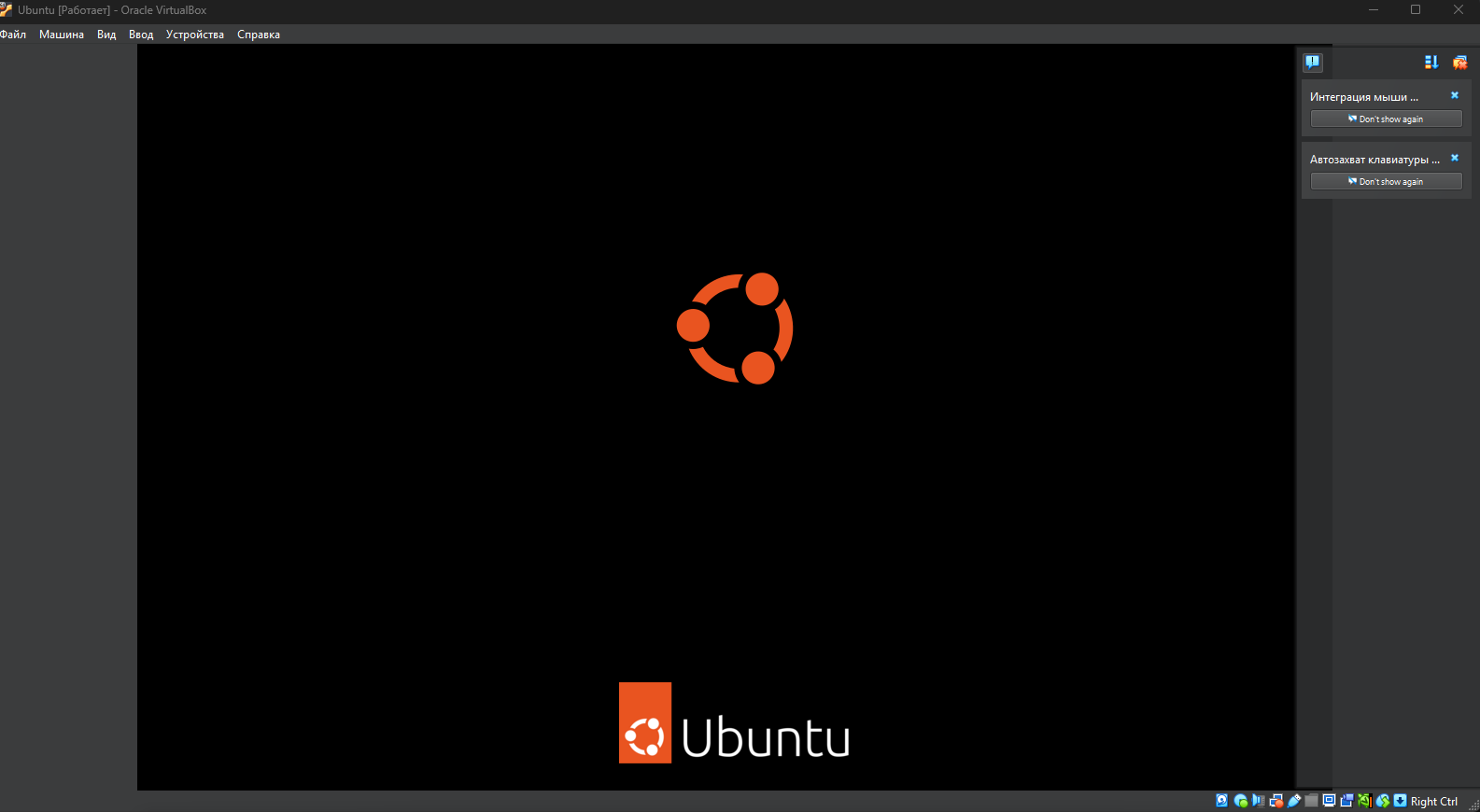


Рисунок 17 — Запуск Linux системы «Ubuntu»

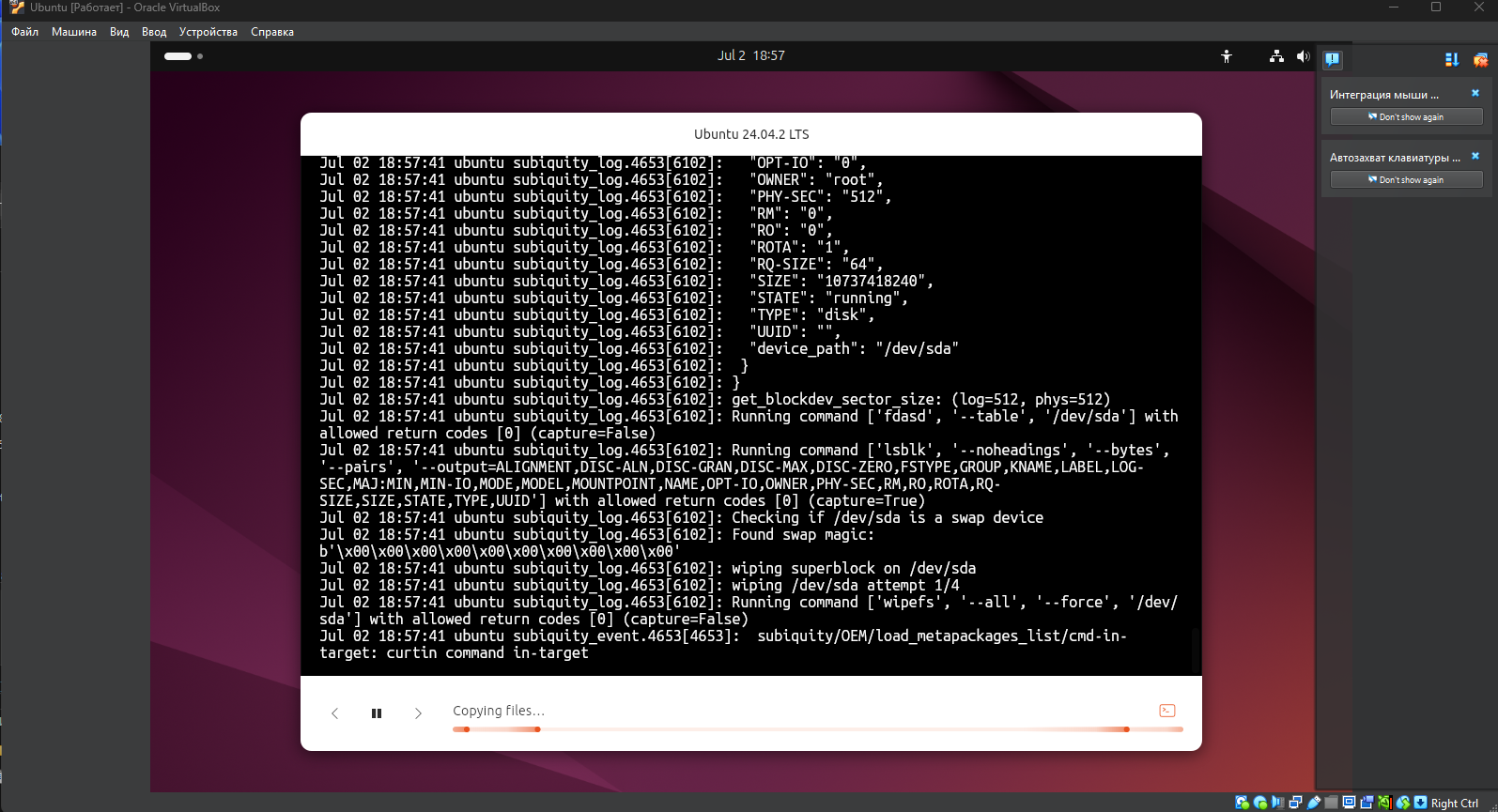


Рисунок 17 — Запуск установщика Ubuntu

Затем запускается Ubuntu, вводим пароль для входа в систему.

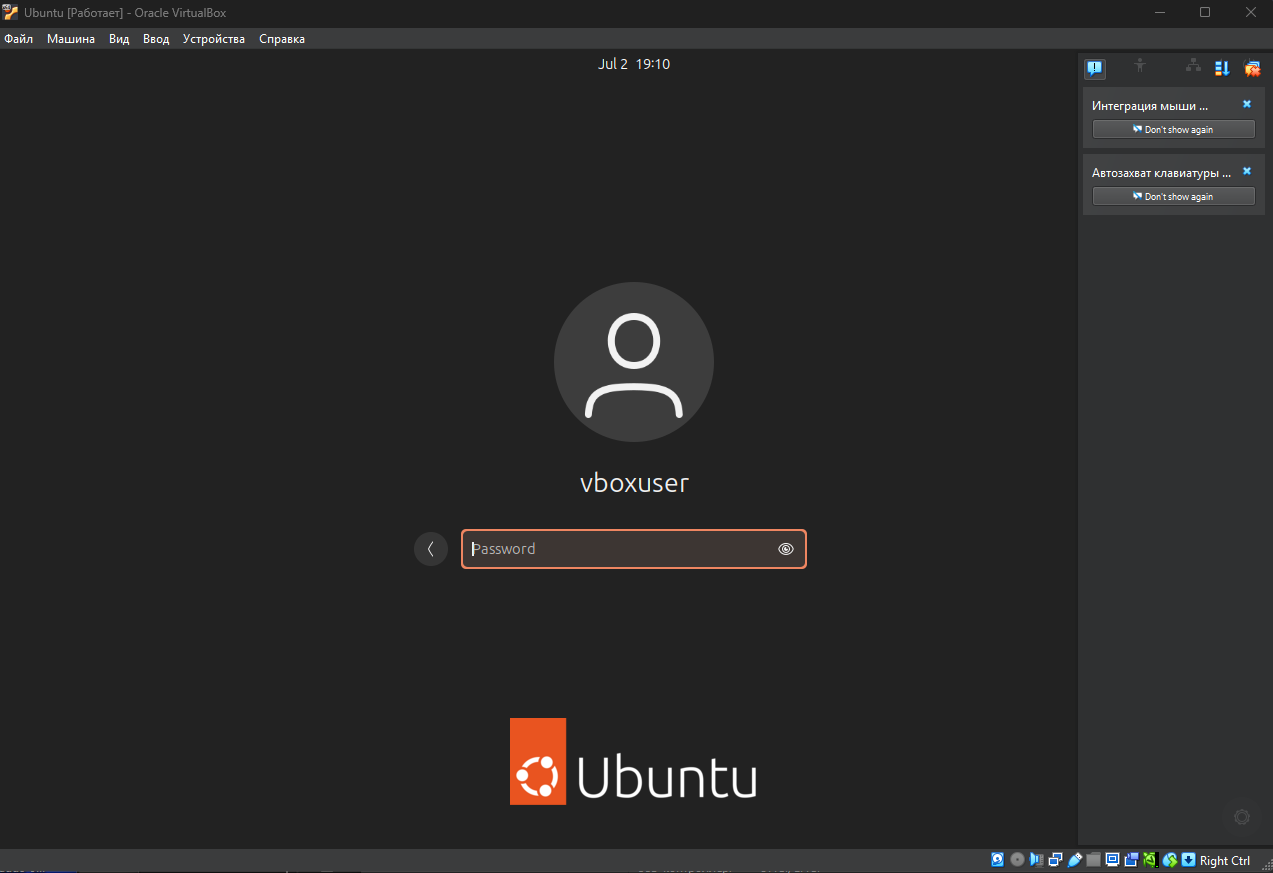


Рисунок 18 — Запуск Ubuntu

ОС Ubuntu готова к работе.

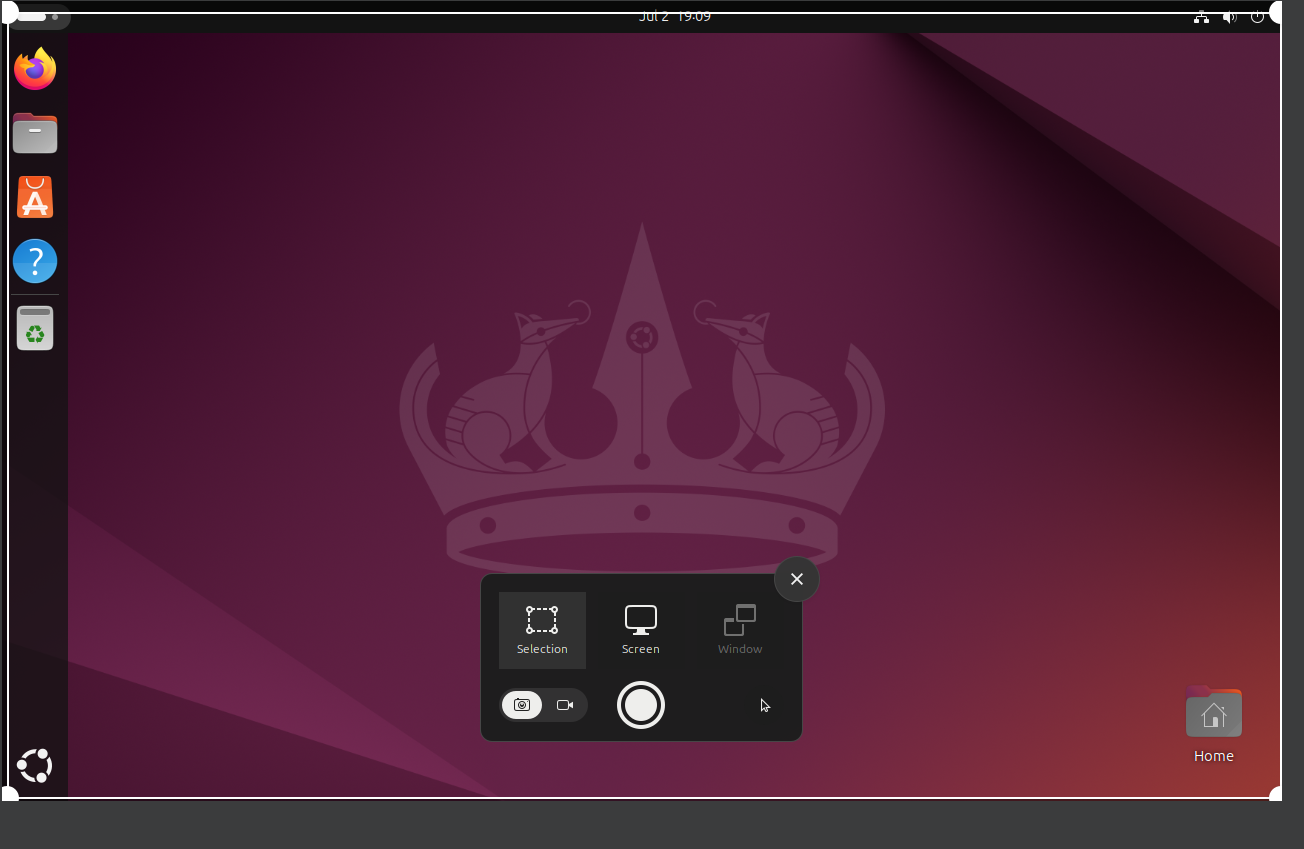


Рисунок 19 — ОС Ubuntu

**1.4 ОС Ubuntu**

Ubuntu — операционная система, основанная на Debian GNU/Linux, является одним из самых популярных дистрибутивов Linux. Она ориентирована на удобство и простоту использования, включает широко распространённое использование утилиты sudo, которая позволяет пользователям выполнять администраторские задачи. Ubuntu, кроме того, имеет развитую интернационализацию./4/ К главным особенностям дистрибутива Ubuntu можно отнести:

1) стабильность работы — не требует частых перезагрузок компьютера;

2) безопасная система, не требующая антивирусов;

3) бесплатная операционная система;

4) приятный и понятный интерфейс, большое количество бесплатных тем и эффектов;

5) быстрая и простая установка, даже для неопытных пользователей, основное программное обеспечение и драйвера устанавливается сразу, предполагается автоматическое обновление;

6) имеется специализированный менеджер пакетов, через который устанавливается большинство программ;

7) малотребовательная к ресурсам компьютера операционная система;

8) модульность, пользователь может изменять структуру и компоненты операционной системы, вплоть до самого ядра.

Операционная система Ubuntu состоит из 8 основных частей:

1) начальный загрузчик Grub, который обрабатывает процесс непосредственной загрузки ОС, задает параметры командной строки;

2) ядро Linux является центральной частью системы, управляет процессором, памятью и устройствами ввода/вывода;

3) демоны — программы, запускаемые самой системой и работающие в фоновом режиме без прямого взаимодействия с пользователем, обычно запускаются во время загрузки системы;

22

4) командная оболочка обеспечивает интерфейс командного процессора, что позволяет управлять компьютером с помощью ввода команд в текстовом режиме;

5) утилиты командной оболочки — в командной оболочке предоставляется некоторое количество основных встроенных команд;

6) графический сервер — часть Linux, представляющая собой графический рабочий стол, не является частью ядра Linux и реализуется с помощью пакета специального вида, известного как «Сервер X»;

7) среда рабочего стола — то, что действительно используется на рабочем столе Linux. В состав системы Ubuntu входит среда рабочего стола Unity, которая обеспечивает фон рабочего стола, панели, заголовки и границы окон;

8) программы рабочего стола — программы, работающие на рабочем столе, но не являющиеся частью среды рабочего стола.

2 Интегрированная среда разработки Qt Creator

2.1 Описание Qt Creator

Qt Creator — кроссплатформенная свободная интегрированная среда разработки для разработки на языках С, С++ и QML. Основная задача Qt Creator — упростить разработку приложения с помощью фреймворка Qt на разных платформах. Среда интегрирована с кроссплатформенными системами автоматизации сборки: qmake и CMake. В редакторе кода реализовано автодополнение, в том числе ключевых слов, подсветка кода, отображение сообщений об ошибках и предупреждений, также есть возможность задания стиля выравнивания, отступов и постановки скобок. Встроенный Qt Designer используется для проектирования и сборки графических интерфейсов пользователя (GUI) из виджетов Qt./5/

23

Qt Creator интегрирован с набором полезных инструментов, таких как системы управления версиями и эмулятор Qt. Среда разработки также может помочь с отладкой приложений, предоставляет интерфейсы к GNU Symbolic Debugger (GDB) и Microsoft Console Debugger (CDB) для отладки обычных приложений на C++ и внутренние отладчики для Java Script.

Qt Creator предоставляет поддержку для сборки и запуска приложений на Qt для настольных компьютеров (Windows, Linux и Mac OS) и мобильных устройств (Symbian, Maemo и MeeGo). Настройки сборки позволят быстро переключаться между целями сборки.

2.2 Описание установки

Для установки интегрированной среды разработки Qt Creator в операционной системе Ubuntu нужно открыть Терминал и написать в нем необходимые команды.

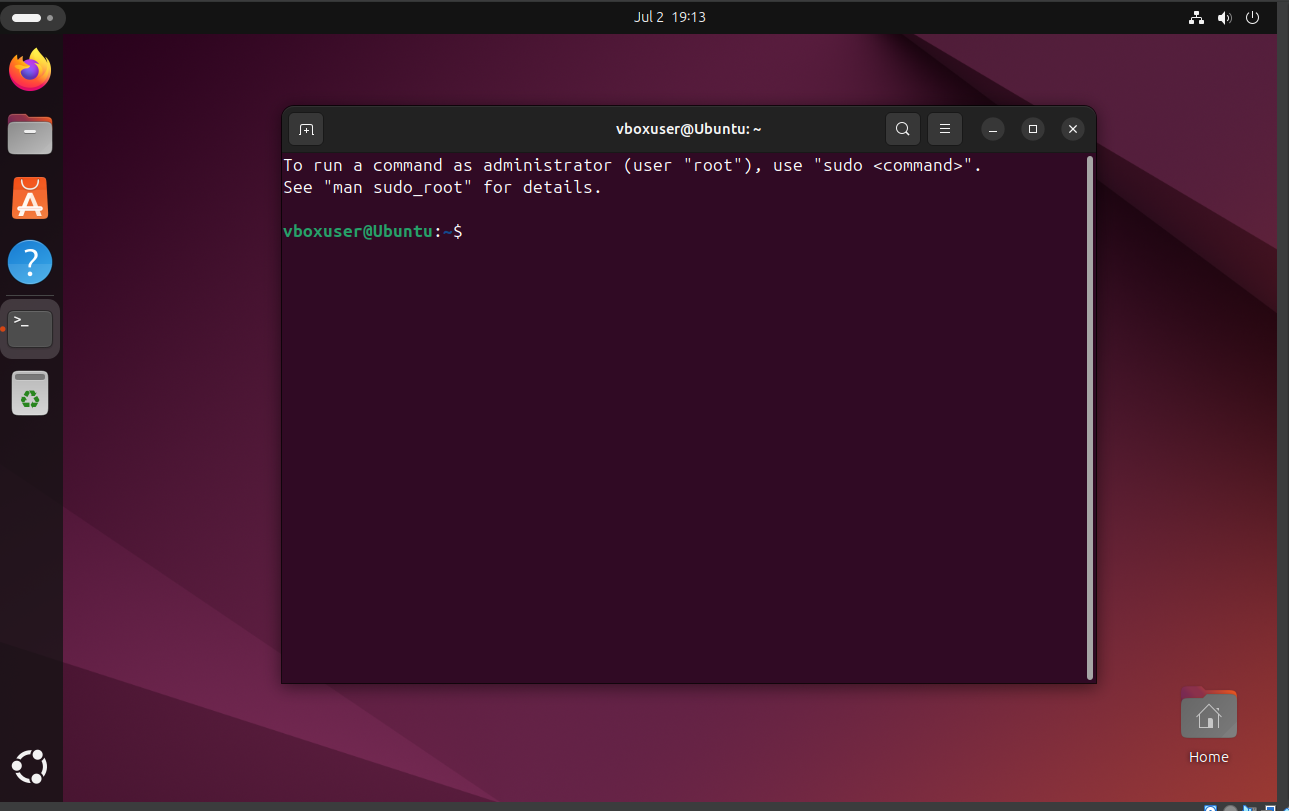


Рисунок 20— Открытие Терминала

Команда sudo apt update предназначена для установки обновлений списка пакетов.

Команда sudo apt install qt5-default предназначена для установки Qt.

Команда sudo apt install qtcreator предназначена для установки Qt Creator.

Команда sudo apt install qtbase5-examples qtdeclarative5-examples предназначена для установки примеров программ.

Установка Qt Creator завершена, далее открываем среду.

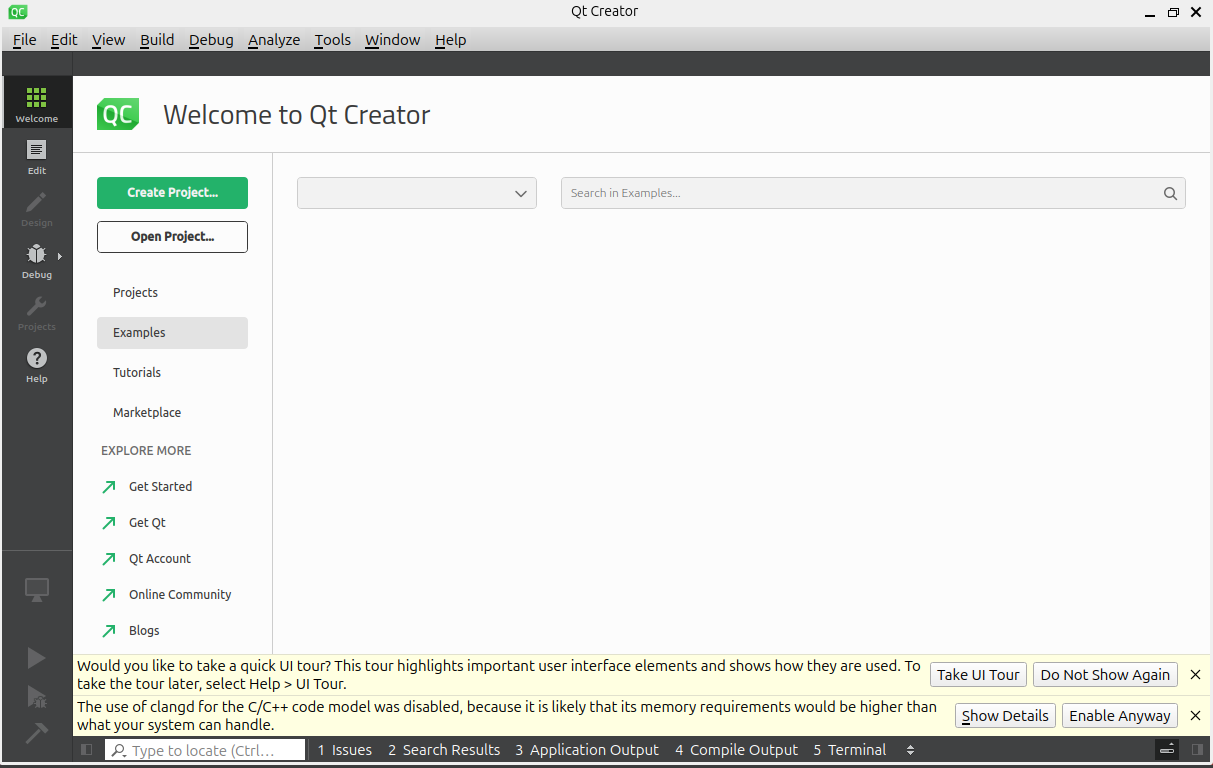


Рисунок 21 — Открытие Qt Creator

3 Разработка программы на языке С в среде Qt Creator

3.1 Формулировка задачи

Разработать программу, которая по введенному пользователем значению аргумента х вычисляет значение функции f(x), причем, при х > 0 f(x) = x^2/5 + 2x - 10, при x < 0 f(x) = 3x + 15, при x = 0 f(x) = 1. Результат вывести на экран и записать в файл.

3.2 Создание проекта в Qt Creator

Для создания проекта в Qt Creator необходимо запустить среду разработки (рисунок 20) и на начальной странице нажать кнопку «Создать». Затем выбираем «Приложение на языке С».



Рисунок 22 — Выбор создаваемого приложения

Далее указываем название и размещение проекта.

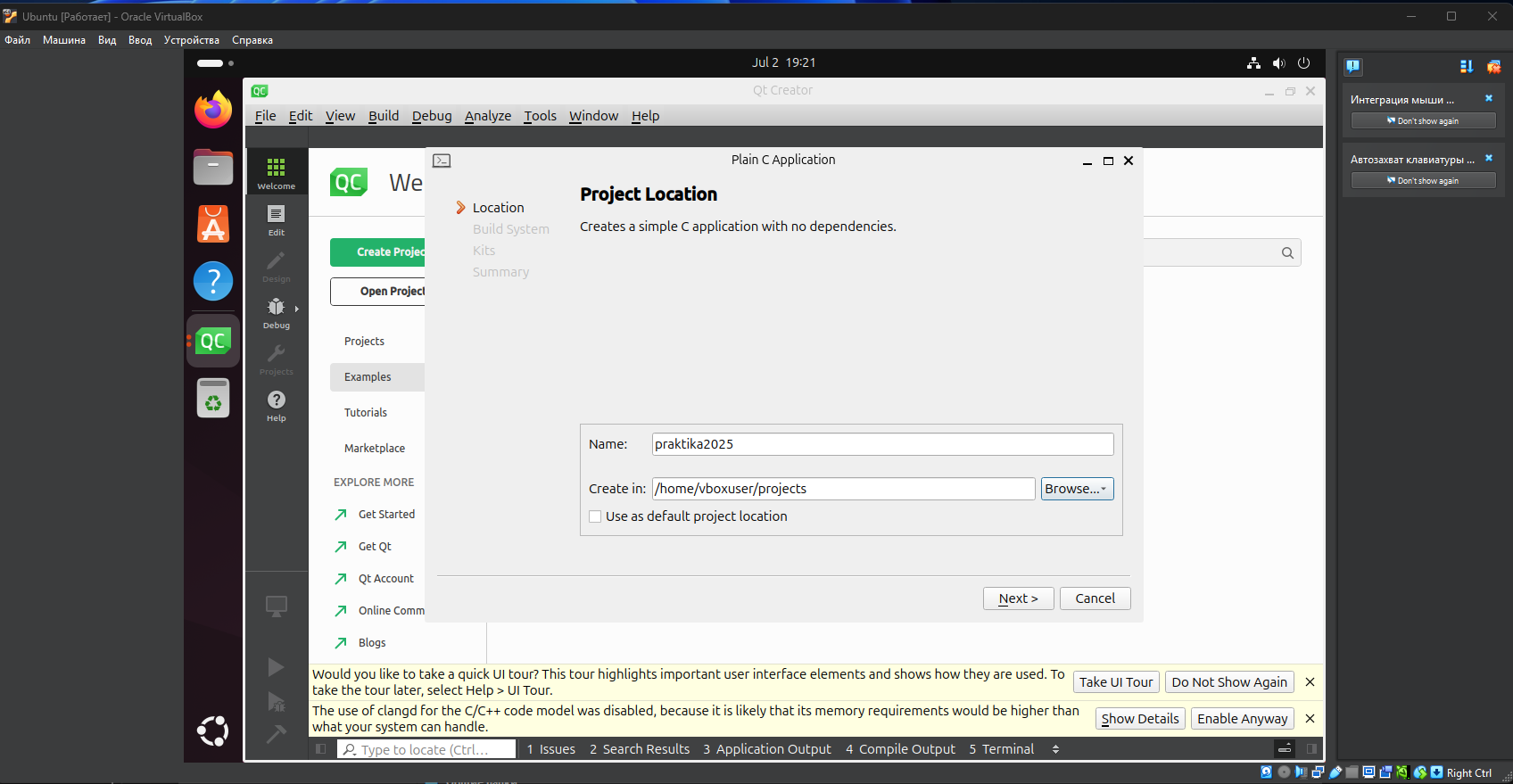


Рисунок 23 — Имя и размещение проекта

Проект создан, открывается редактор для написания кода программы.

Потом выбираем систему сборки проекта.

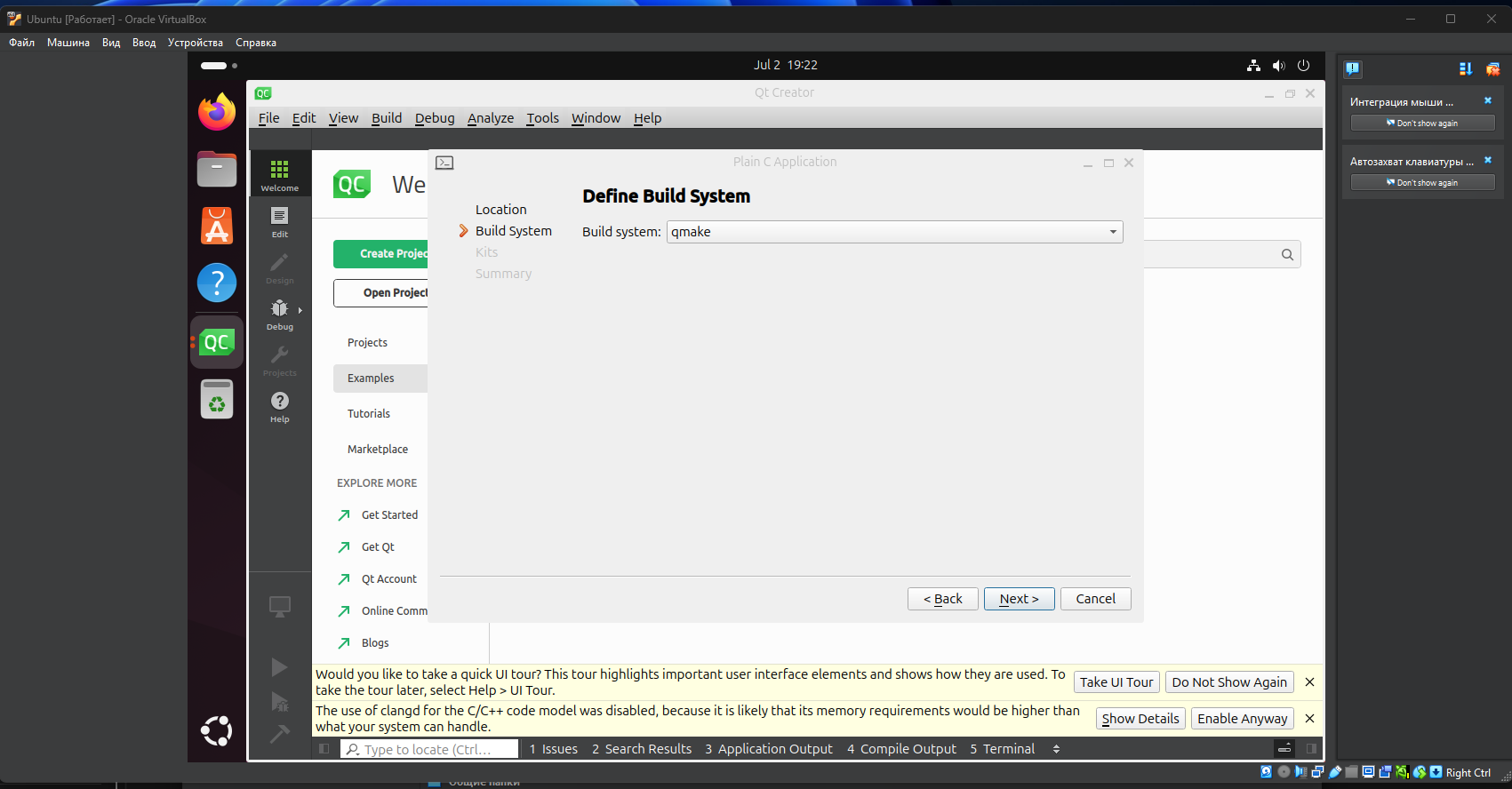


Рисунок 24 — Выбор системы сборки

Затем выбираем комплекты проекта.

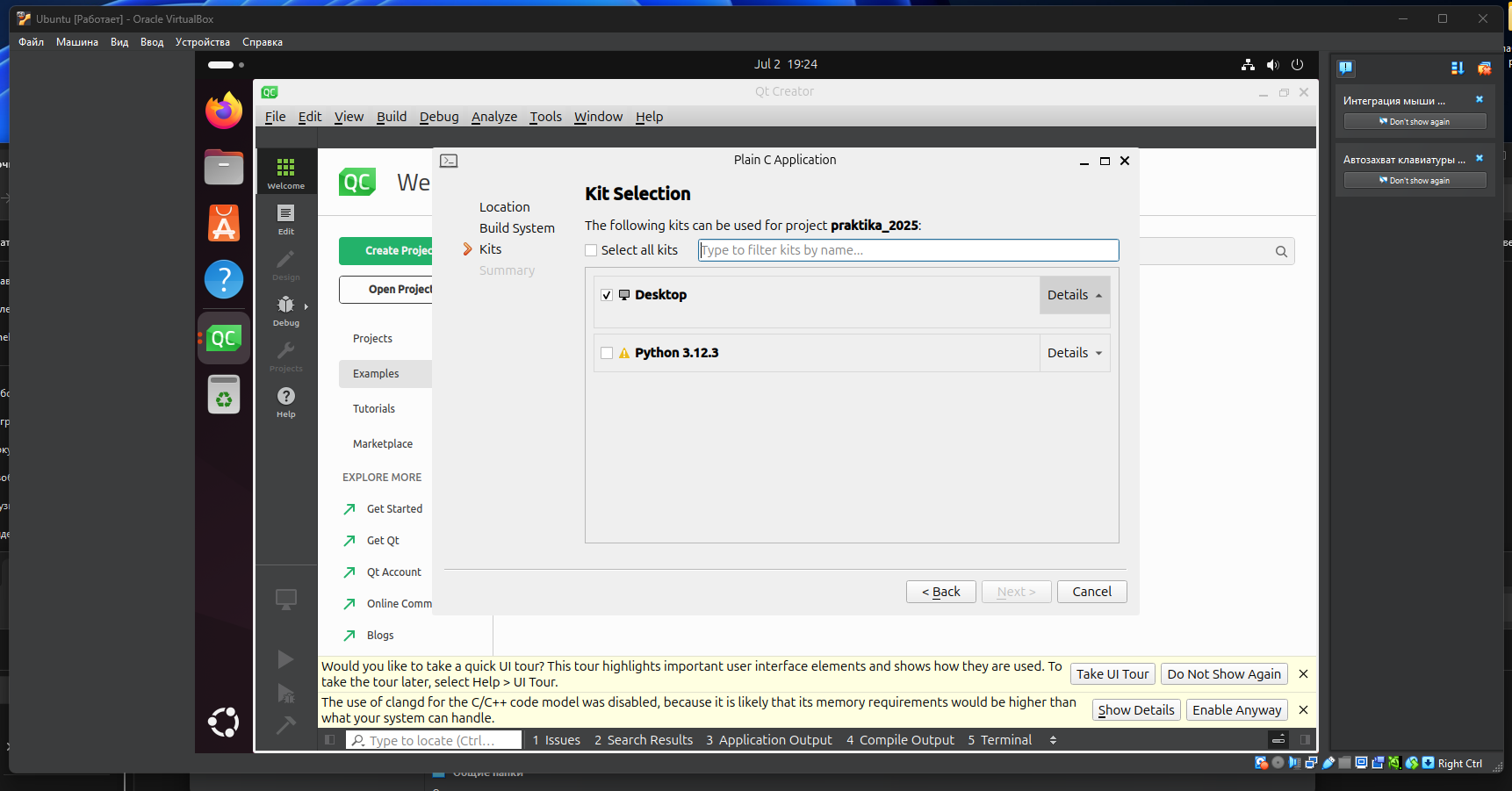


Рисунок 25 — Выбор комплекта

Далее указываем управление проектом и нажимаем кнопку «Завершить».

Проект создан, открывается редактор для написания кода программы.

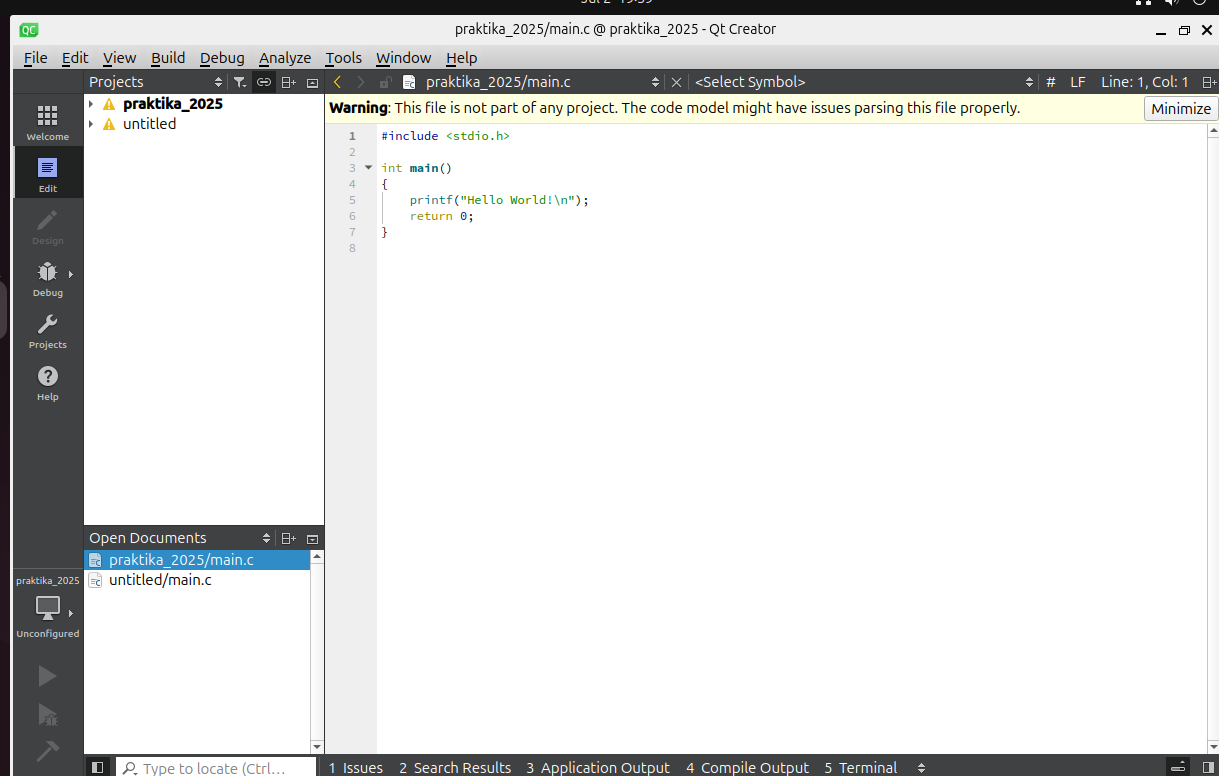


Рисунок 26 — Открытие редактора

**3.3 Описание программы**

Данная программа предназначена для демонстрации и сравнения различных алгоритмов поиска элементов в массиве целых чисел. Она предоставляет пользователю возможность загрузить массив из текстового файла, выбрать алгоритм поиска и получить результаты, как на экране, так и в отдельном файле. Программа разработана на языке C и предназначена для выполнения в консольном режиме.

Она представляет собой практическую реализацию различных алгоритмов поиска в массиве на языке C. Она демонстрирует основные принципы работы с массивами, указателями, файлами и динамической памятью. Программа может быть использована в образовательных целях для изучения и сравнения различных подходов к решению задачи поиска. Результаты работы программы могут быть легко сохранены в текстовый файл для дальнейшего анализа.

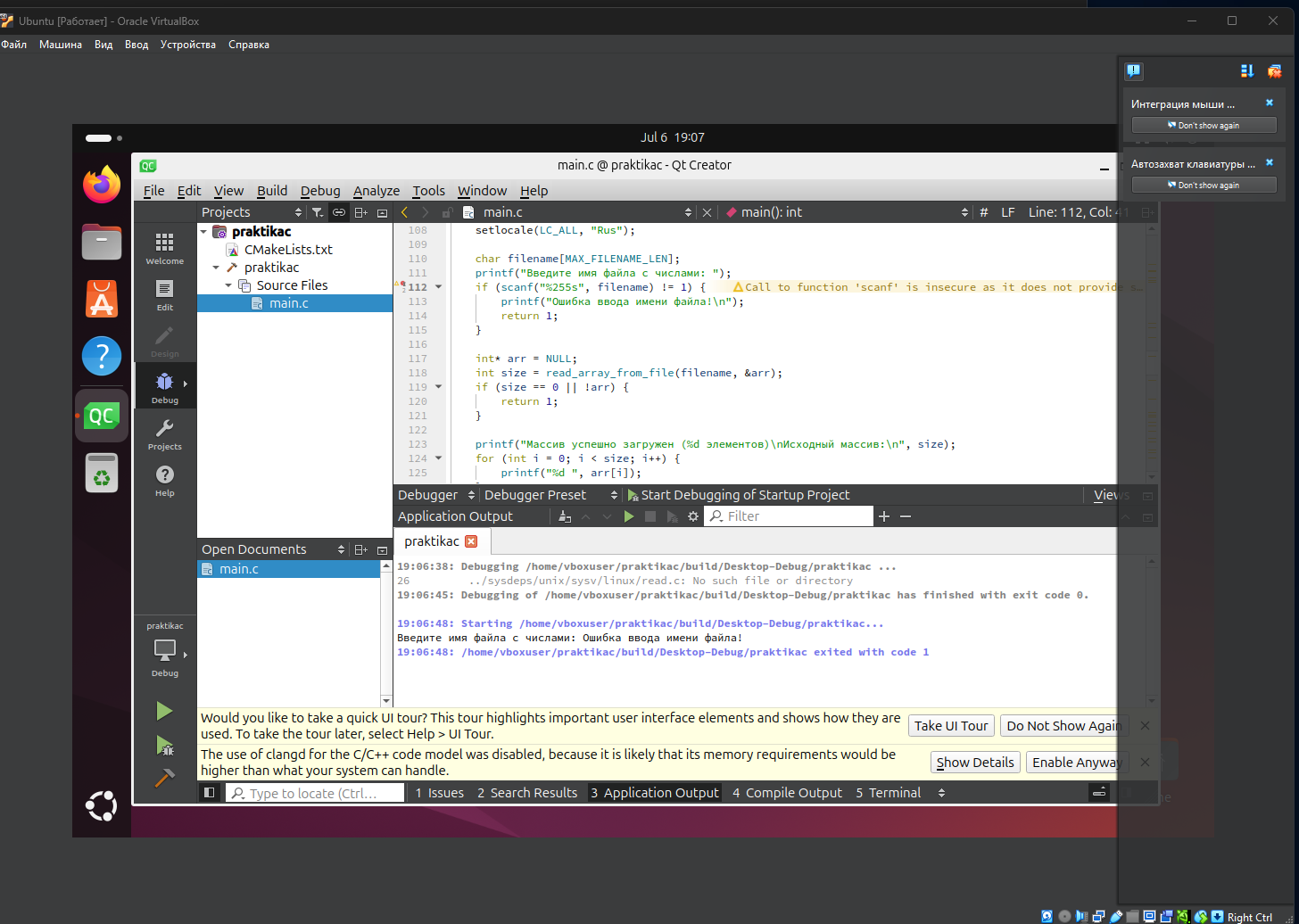


Рисунок 27 — Вывод результата сборки проекта

При запуске проекта производится его компиляция. Для запуска программы необходимо нажать кнопку «Запустить» или комбинацию клавиш Ctrl + R.

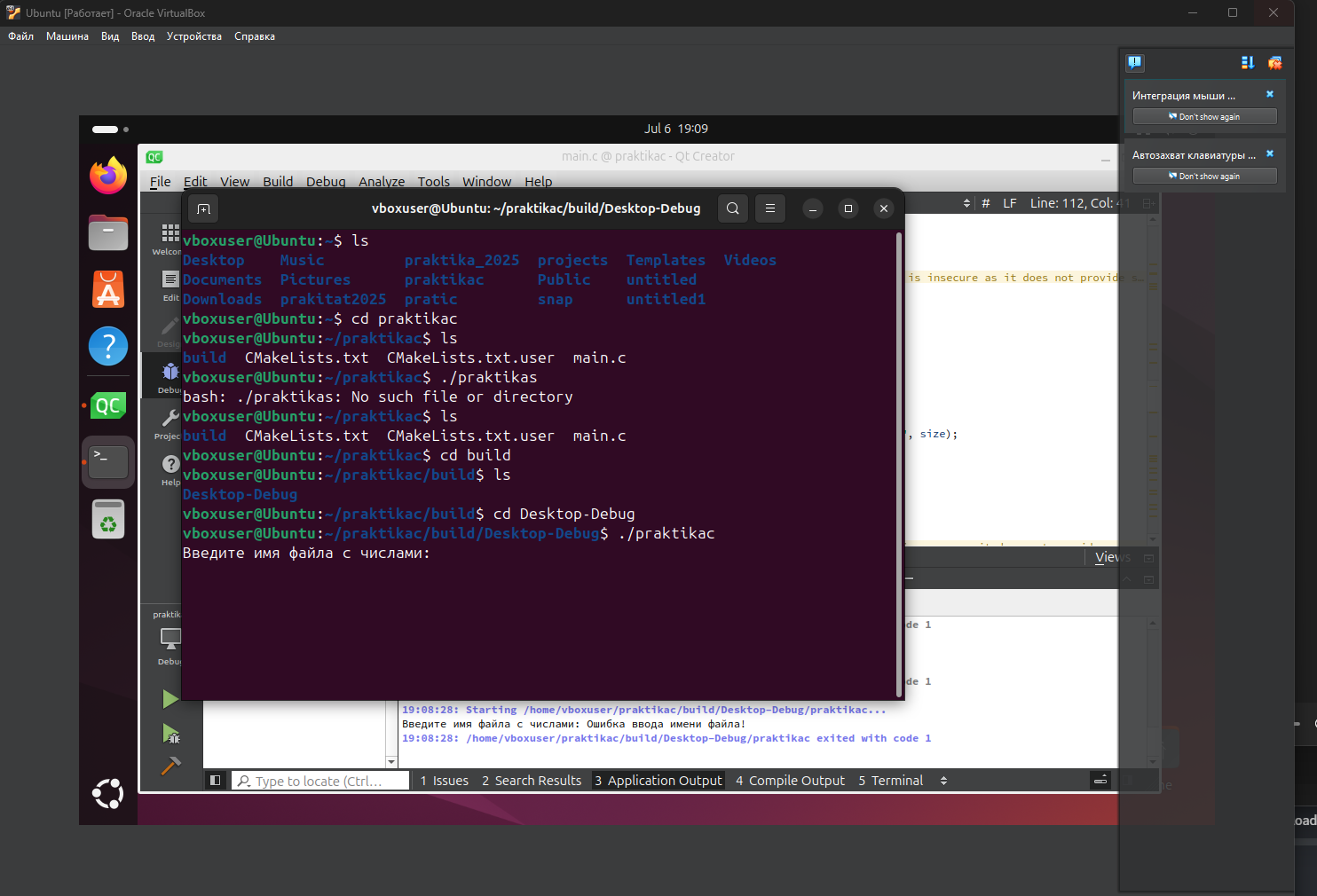


Рисунок 36 — Запуск программы

Для отладки проекта в среде Qt Creator есть встроенный отладчик GDB, расположенный в /usr/bin. Для запуска отладчика необходимо нажать кнопку «Начать отладку запускающего проекта». Также можно поставить точку останова на любую строку кода. Для завершения отладки нужно нажать кнопку «Остановить отладчик» и убрать точку останова, если она была поставлена.

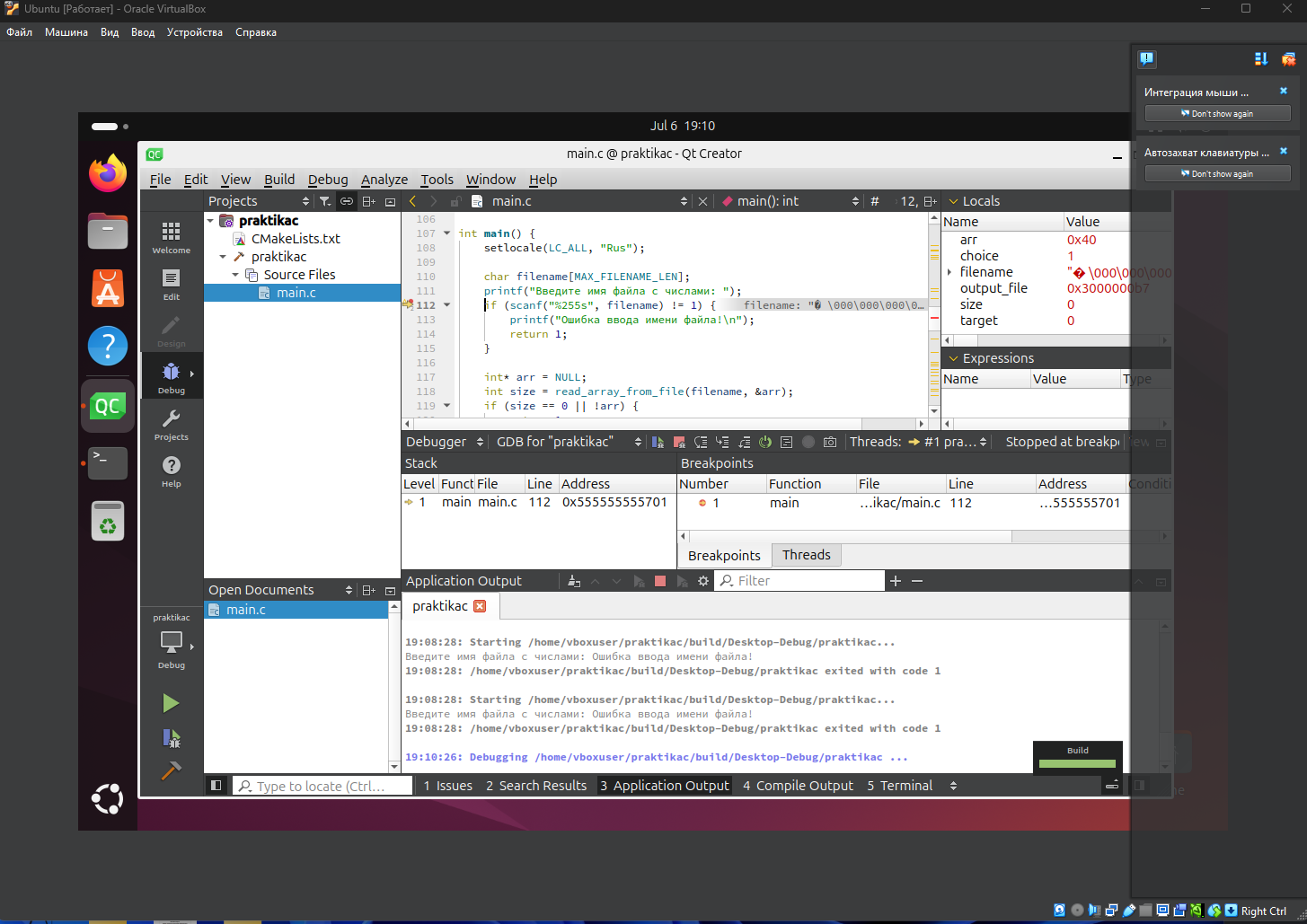


Рисунок 37 — Запуск отладчика и установка точки останова

Заключение

В ходе выполнения практической работы была изучена и установлена операционная система Ubuntu на виртуальную машину VirtualBox, установлена интегрированная среда разработки Qt Creator в ОС Ubuntu, были изучены основные инструменты среды и разработана программа на языке С в среде Qt Creator. Результаты испытаний показали, что программа работает согласно формулировке задачи.

Список использованных источников и литературы

1. Колисниченко Д.Н., Linux: Полное руководство

2. Виртуальная машина. Режим доступа:

https://ru.wikipedia.org/wiki/Виртуальная\_машина

3. VirtualBox. Режим доступа: https://ru.wikipedia.org/wiki/VirtualBox

4. Ubuntu. Режим доступа: https://hyperhost.ua/info/ru/ubuntu-opisanie-osaktualnyie-versii-plyus

5. Qt Creator. Режим доступа: http://doc.crossplatform.ru/qtcreator/2.0.1/ creatoroverview.html

Приложение А Листинг программы

[#include](https://vk.com/im/convo/516434405?search=%23include&entrypoint=list_all) <stdio.h>  
[#include](https://vk.com/im/convo/516434405?search=%23include&entrypoint=list_all) <stdlib.h>  
[#include](https://vk.com/im/convo/516434405?search=%23include&entrypoint=list_all) <string.h>  
[#include](https://vk.com/im/convo/516434405?search=%23include&entrypoint=list_all) <locale.h>  
[#define](https://vk.com/im/convo/516434405?search=%23define&entrypoint=list_all) MAX\_FILENAME\_LEN 256  
[#define](https://vk.com/im/convo/516434405?search=%23define&entrypoint=list_all) MAX\_BUFFER\_SIZE 1024  
typedef struct {  
int index;  
int value;  
} SearchResult;  
// 1. Линейный поиск  
int linear\_search(int arr[], int size, int target) {  
for (int i = 0; i < size; i++) {  
if (arr[i] == target) {  
return i;  
}  
}  
return -1;  
}  
// 2. Поиск всех вхождений  
void search\_all\_occurrences(int arr[], int size, int target, FILE\* output\_file) {  
char buffer[MAX\_BUFFER\_SIZE] = {0};  
snprintf(buffer, sizeof(buffer), "Элемент %d найден на позициях: ", target);  
int found = 0;  
for (int i = 0; i < size; i++) {  
if (arr[i] == target) {  
char temp[20];  
snprintf(temp, sizeof(temp), "%d ", i);  
strncat(buffer, temp, sizeof(buffer) - strlen(buffer) - 1);  
found = 1;  
}  
}  
if (!found) {  
strncat(buffer, "не найден.", sizeof(buffer) - strlen(buffer) - 1);  
}  
strncat(buffer, "\n", sizeof(buffer) - strlen(buffer) - 1);  
  
fprintf(output\_file, "%s", buffer);  
printf("%s", buffer);  
}  
// 3. Поиск с указателями  
int pointer\_search(int\* arr, int size, int target) {  
int\* end = arr + size;  
for (int\* ptr = arr; ptr < end; ptr++) {  
if (\*ptr == target) {  
return (int)(ptr - arr);  
}  
}  
return -1;  
}  
  
// 4. Поиск с возвратом структуры  
SearchResult search\_with\_struct(int arr[], int size, int target) {  
SearchResult res = { -1, -1 };  
for (int i = 0; i < size; i++) {  
if (arr[i] == target) {  
res.index = i;  
res.value = arr[i];  
break;  
}  
}  
return res;  
}  
// Чтение массива из файла  
int read\_array\_from\_file(const char\* filename, int\*\* arr) {  
FILE\* file = fopen(filename, "r");  
if (!file) {  
perror("Ошибка при открытии файла");  
return 0;  
}  
int count = 0;  
int num;  
while (fscanf(file, "%d", &num) == 1) {  
count++;  
}  
[\*arr](https://vk.com/id63402676) = (int\*)malloc(count \* sizeof(int));  
if (![\*arr](https://vk.com/id63402676)) {  
perror("Ошибка выделения памяти");  
fclose(file);  
return 0;  
}  
rewind(file);  
for (int i = 0; i < count; i++) {  
if (fscanf(file, "%d", &([\*arr](https://vk.com/id63402676))[i]) != 1) {  
free([\*arr](https://vk.com/id63402676));  
[\*arr](https://vk.com/id63402676) = NULL;  
fclose(file);  
return 0;  
}  
}  
fclose(file);  
return count;  
}  
int main() {  
setlocale(LC\_ALL, "Rus");  
char filename[MAX\_FILENAME\_LEN];  
printf("Введите имя файла с числами: ");  
if (scanf("%255s", filename) != 1) {  
printf("Ошибка ввода имени файла!\n");  
return 1;  
}  
int\* arr = NULL;  
int size = read\_array\_from\_file(filename, &arr);  
if (size == 0 || !arr) {  
return 1;  
}  
printf("Массив успешно загружен (%d элементов)\nИсходный массив:\n", size);  
for (int i = 0; i < size; i++) {  
printf("%d ", arr[i]);  
}  
printf("\n\n");  
int target;  
printf("Введите число для поиска: ");  
if (scanf("%d", &target) != 1) {  
printf("Ошибка ввода числа для поиска!\n");  
free(arr);  
return 1;  
}  
// Очистка буфера ввода  
while (getchar() != '\n');  
printf("\nВыберите метод поиска:\n"  
"1. Линейный поиск (первое вхождение)\n"  
"2. Поиск всех вхождений\n"  
"3. Поиск с указателями\n"  
"4. Поиск с возвратом структуры\n"  
"Ваш выбор: ");  
int choice;  
if (scanf("%d", &choice) != 1) {  
printf("Ошибка ввода выбора метода!\n");  
free(arr);  
return 1;  
}  
FILE\* output\_file = fopen("search\_results.txt", "w");  
if (!output\_file) {  
perror("Ошибка при создании файла результатов");  
free(arr);  
return 1;  
}  
fprintf(output\_file, "Исходный массив (%d элементов):\n", size);  
for (int i = 0; i < size; i++) {  
fprintf(output\_file, "%d ", arr[i]);  
}  
fprintf(output\_file, "\n\n");  
switch (choice) {  
case 1: {  
int result = linear\_search(arr, size, target);

char message[100];  
snprintf(message, sizeof(message),  
"Линейный поиск: элемент %d %sна позиции %d.\n",  
target, result == -1 ? "не найден.\n" : "найден ", result);  
fprintf(output\_file, "%s", message);  
printf("%s", message);  
break;  
}  
case 2:  
fprintf(output\_file, "Результат поиска всех вхождений:\n");  
printf("Результат поиска всех вхождений:\n");  
search\_all\_occurrences(arr, size, target, output\_file);  
break;  
case 3: {  
int result = pointer\_search(arr, size, target);  
char message[100];  
snprintf(message, sizeof(message),  
"Поиск с указателями: элемент %d %sна позиции %d.\n",  
target, result == -1 ? "не найден.\n" : "найден ", result);  
fprintf(output\_file, "%s", message);  
printf("%s", message);  
break;  
}  
case 4: {  
SearchResult result = search\_with\_struct(arr, size, target);  
char message[100];  
snprintf(message, sizeof(message),  
"Поиск с структурой: элемент %d %sна позиции %d.\n",  
target, result.index == -1 ? "не найден.\n" : "найден ", result.index);  
fprintf(output\_file, "%s", message);  
printf("%s", message);  
break;  
}  
default:  
fprintf(output\_file, "Неверный выбор метода поиска.\n");  
printf("Неверный выбор метода поиска.\n");  
}  
  
fclose(output\_file);  
free(arr);  
  
printf("\nРезультаты поиска также сохранены в файл 'search\_results.txt'.\n");  
return 0;  
}

Приложение Б Результаты испытаний

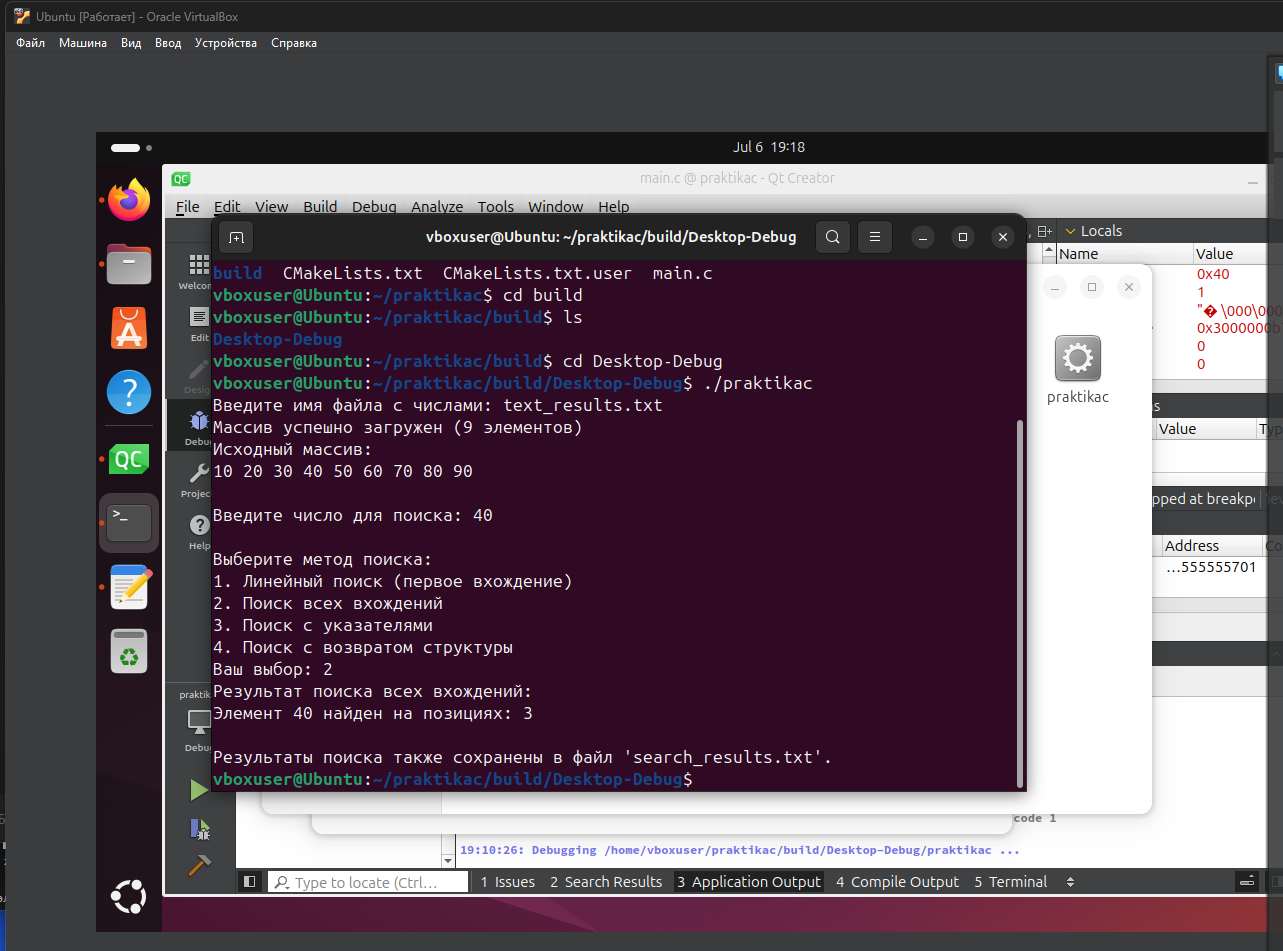


Рисунок Б.1 — Массив из 9 элементов

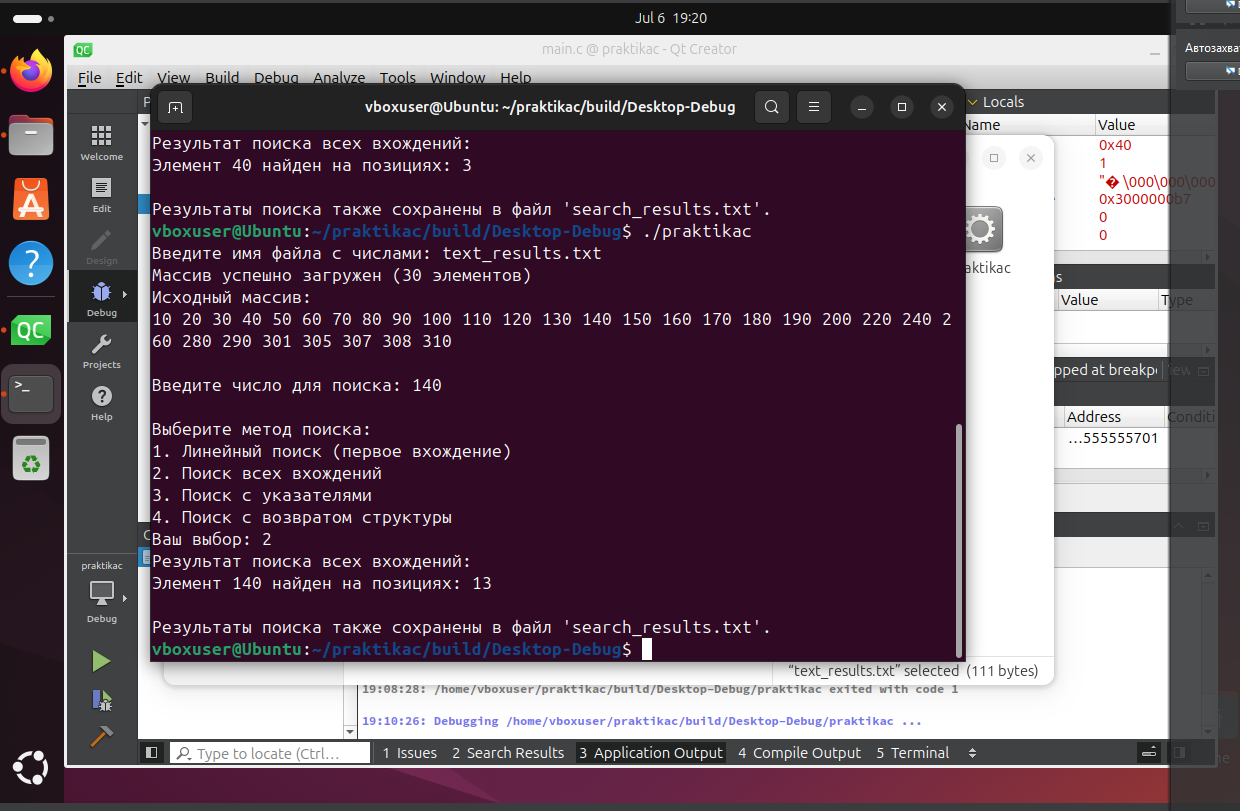


Рисунок Б.2 — Массив из 30 элементов

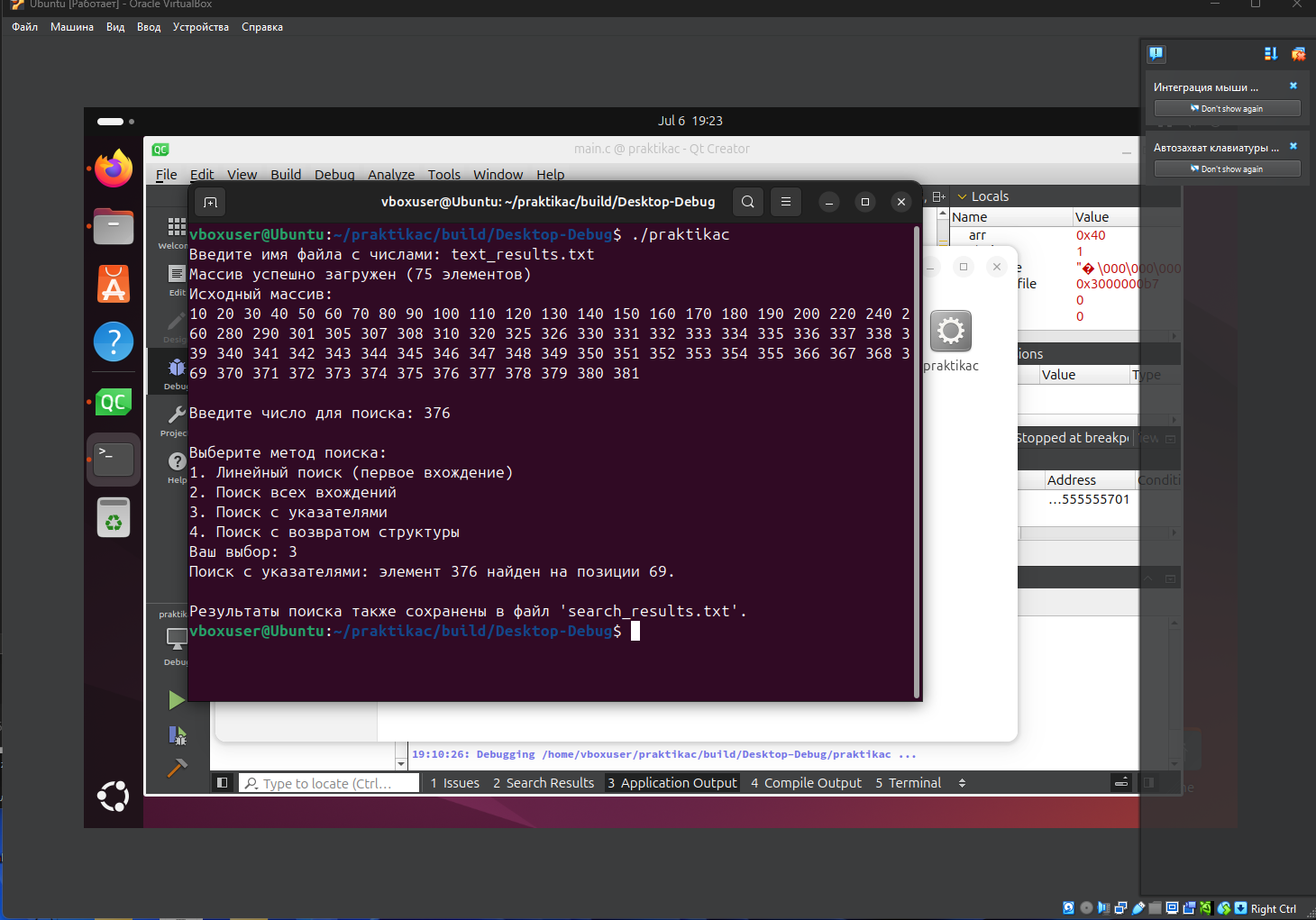


Рисунок Б.3 — Массив из 76 элементов

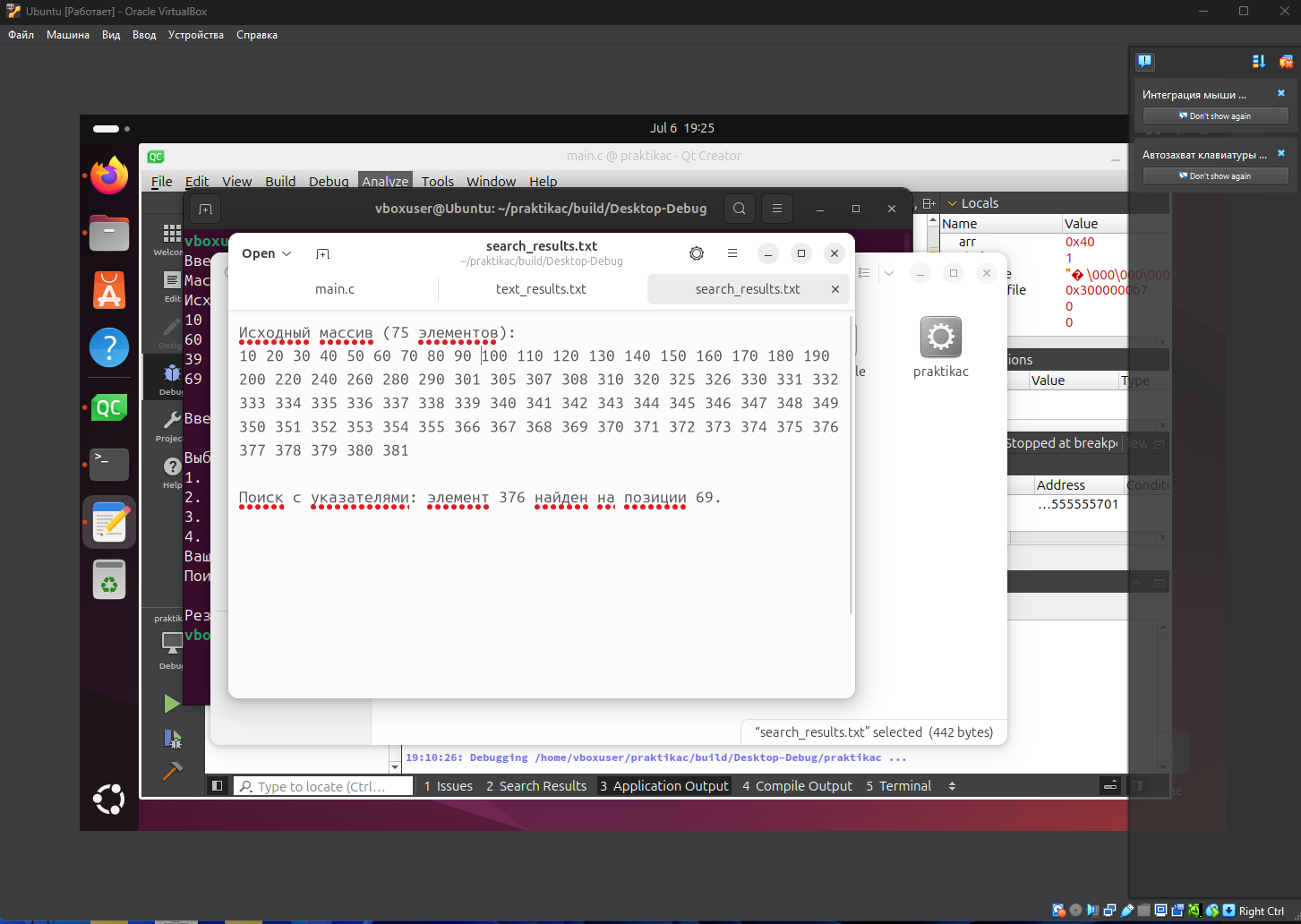


Рисунок Б.4 — Проверка записи результата в файл