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**

(2023/2024 учебный год)

Беляев Даниил Ильич

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 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики д.т.н., профессор, Зинкин С.А.

*(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**

(2023/2024 учебный год)

                                              Беляев Даниил Ильич

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 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики д.т.н., профессор, Зинкин С.А.

*(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 | 26 | 20.06.24 –  24.06.24 |  |
| 2 | Подбор и изучение материала по теме работы | 26 | 24.06.24 –  26.06.24 |  |
| 3 | Установка виртуальной машины | 26 | 26.06.24 –  28.06.24 |  |
| 4 | Установка операционной системы | 30 | 01.07.24 –  04.07.24 |  |
| 5 | Разработка программы на языке Си | 30 | 04.07.24 –  09.07.24 |  |
| 6 | Тестирование и отладка | 38 | 09.07.24 –  12.07.24 |  |
| 7 | Оформление отчёта | 40 | 12.07.24 –  17.07.24 |  |
|  | **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**

(2023/2024 учебный год)

                                             Беляев Даниил Ильич

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 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

Беляев Д.И. выполнял практическое задание «Установка Linux. Написание программы в Unix-подобной OC». На первоначальном этапе была установлена ОС Ubuntu. Далее в качестве IDE был выбран VS Code. В нем была реализована программа, выполняющая сортировку пузырьком

Бакалавр Беляев Д. И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Руководитель Зинкин С.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

                                              Беляев Даниил Ильич

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 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

В процессе выполнения практики Беляев Д. И. решал следующие задачи: написание программы на языке С. Установка дистрибутива Linux Ubuntu на виртуальную машину. Разработка и написание программы в Linux

За период выполнения практики были освоены Ubuntu, VirtualBox. Во время выполнения работы Беляев Д. И. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике.

За выполнение работы Беляев Д. И. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики д.т.н., профессор, Зинкин С.А. « » 2024 г.

**Содержание**

[Введение 9](#_Toc170158405)

[1 Установка ОС Linux на виртуальную машину 10](#_Toc170158406)

[1.1 Установка виртуальной машины VirtualBox 10](#_Toc170158407)

[1.2 Создание виртуальной машины 12](#_Toc170158408)

[1.3 Установка ОС Ubuntu на виртуальную машину 15](#_Toc170158409)

[1.4 ОС Ubuntu 25](#_Toc170158410)

[2 Интегрированная среда разработки VS Code 27](#_Toc170158411)

[2.1 Описание VS Code 27](#_Toc170158412)

[2.2 Описание установки 27](#_Toc170158413)

[3 Разработка программы на языке С в среде VS Code 33](#_Toc170158414)

[3.1 Формулировка задачи 33](#_Toc170158415)

[3.2 Описание программы 33](#_Toc170158416)

[3.3 Работа программы 33](#_Toc170158417)

[3.4 Компиляция и отладка программы в среде VS Code 34](#_Toc170158418)

[Заключение 36](#_Toc170158419)

[Список использованных источников и литературы 37](#_Toc170158420)

[Приложение А 38](#_Toc170158421)

[Приложение Б 40](#_Toc170158422)

# **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.

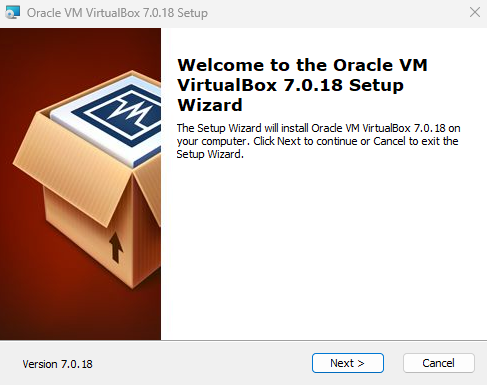


Рисунок 1 – Запуск установочного файла VirtualBox

Затем выбираем компоненты для установки и расположение.



Рисунок 2 – Выбор компонентов для установки и расположения

Потом соглашаемся на отключение Интернет-соединения на время установки VirtualBox

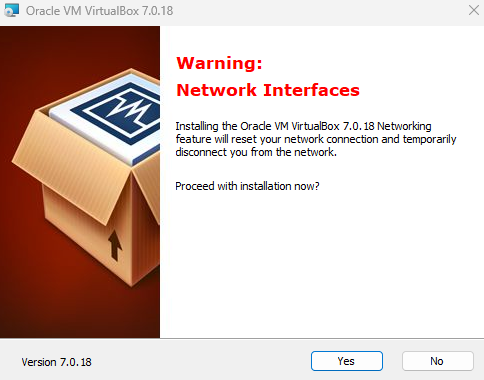


Рисунок 3 — Временное отключение Интернет-соединения

Далее идет установка ПО

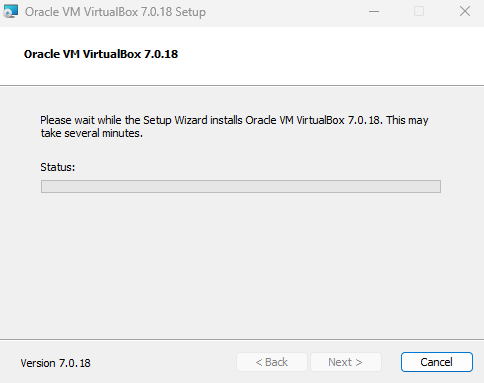


Рисунок 4 – установка VM VirtualBox

Далее нажимаем кнопку «Завершить» для выхода из Мастера установки

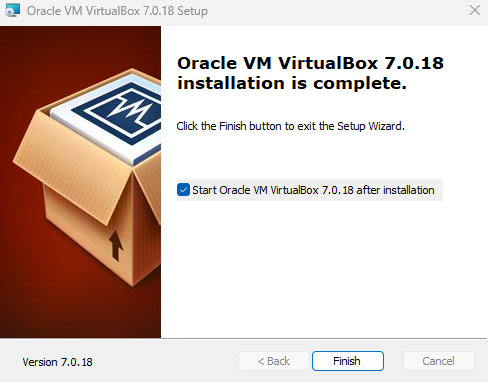


Рисунок 5 — Выход из Мастера установки

Установка VirtualBox успешно завершена, для проверки запускаем ВМ

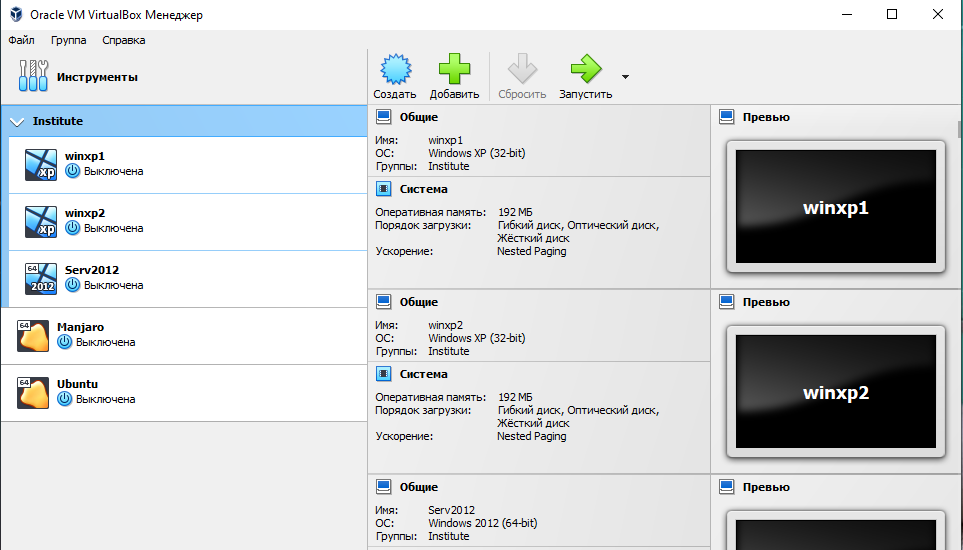


Рисунок 6 — Запуск VirtualBox

## **1.2 Создание виртуальной машины**

Для создания виртуальной машины запускаем VirtualBox и нажимаем кнопку «Создать».

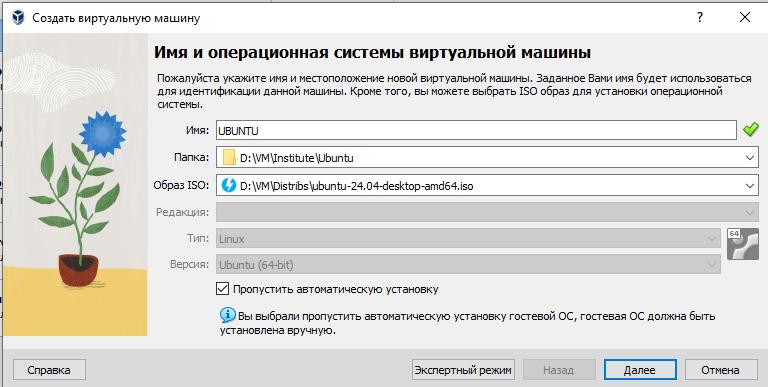


Рисунок 7 — Создание виртуальной машины

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

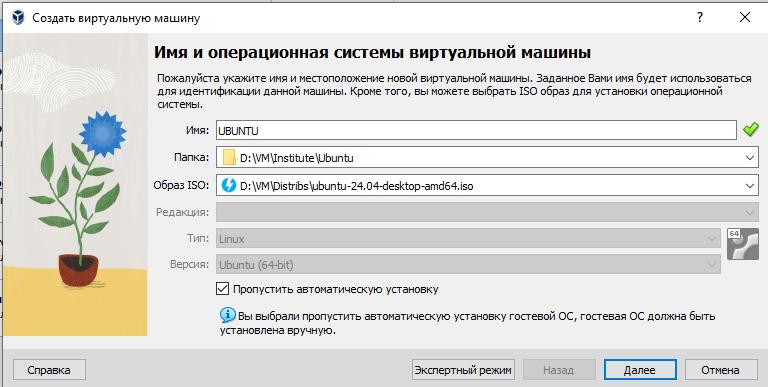


Рисунок 8 — Указание имени и типа ОС

Далее указываем объем оперативной памяти виртуальной машины и количество виртуальных процессоров.

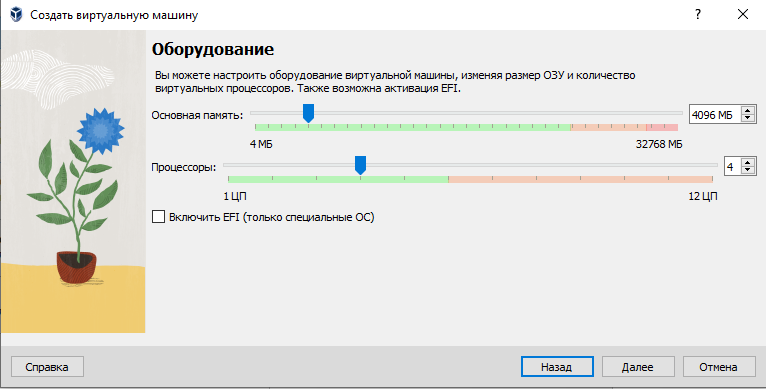


Рисунок 9 — Объем оперативной памяти и количество процессоров

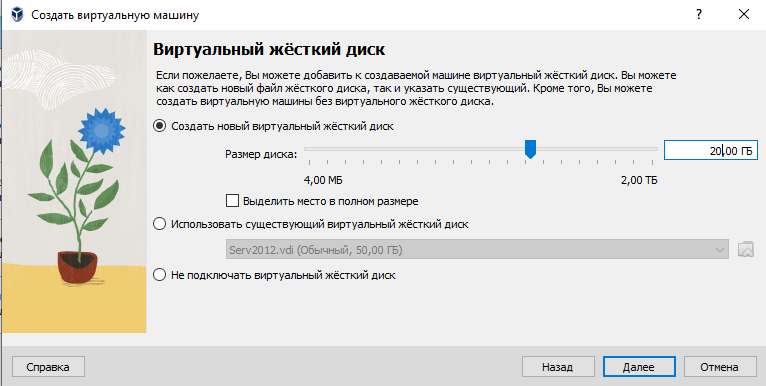
Потом создаем новый виртуальный жесткий диск. 

Рисунок 10 — Создание жесткого диска

Далее видим итоговую конфигурацию виртуальной машины.



Рисунок 11 — Итоговая конфигурация ВМ

Виртуальная машина «UBUNTU» создана.

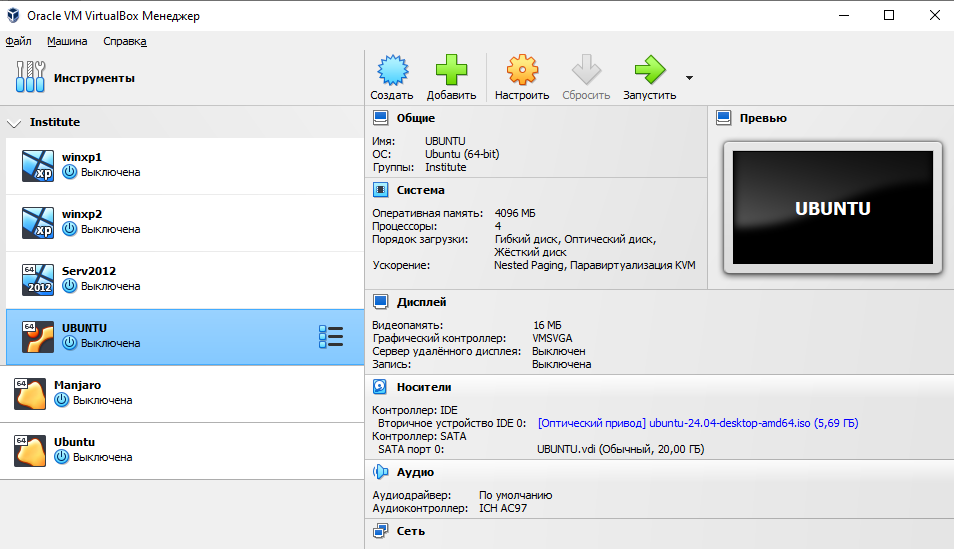


Рисунок 12 — Виртуальная машина «UBUNTU»

## **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».

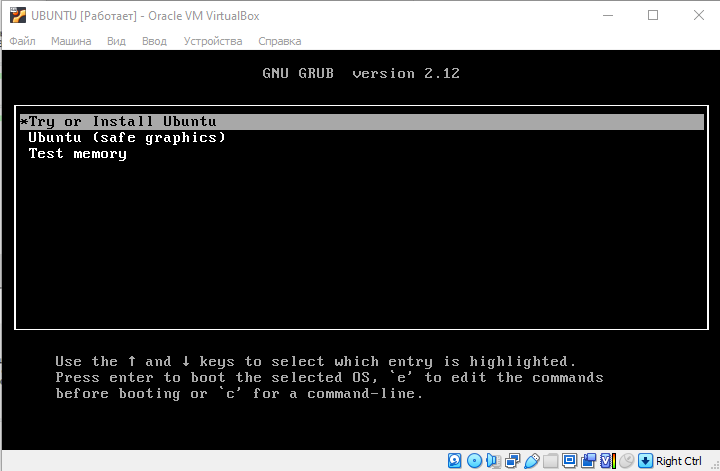


Рисунок 13 — Запуск виртуальной машины «Ubuntu»

Затем выбираем первый пункт.

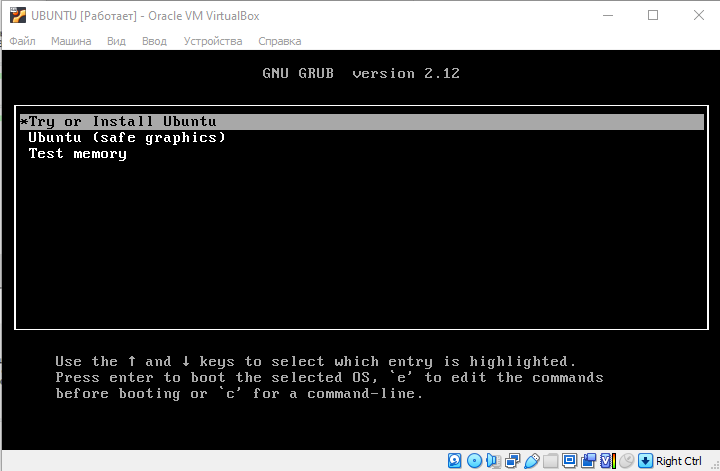


Рисунок 14 — Выбор загрузочного диска

Далее запускается установщик Ubuntu

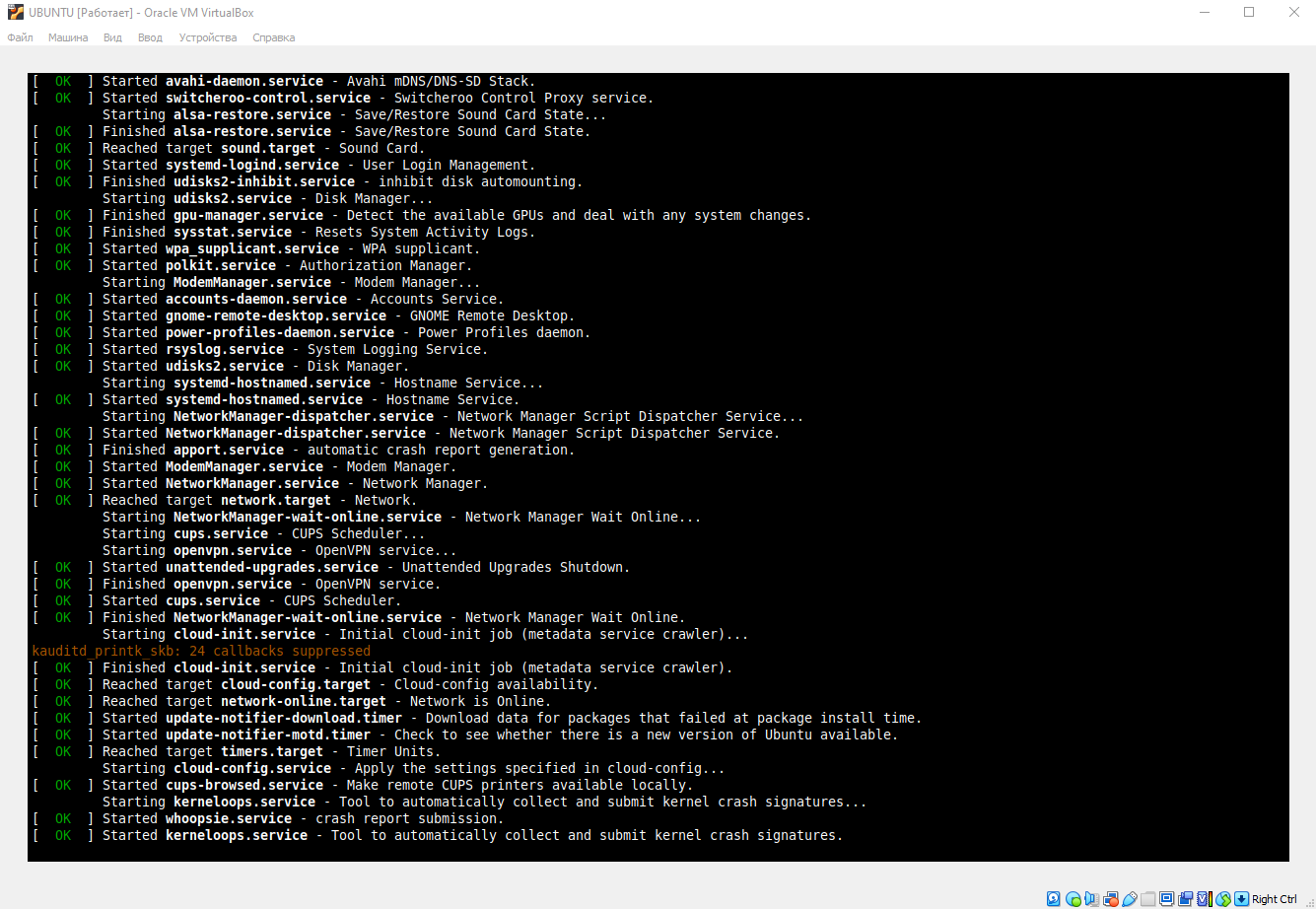


Рисунок 15 — Установщик

Далее идёт подготовка ОС к установке

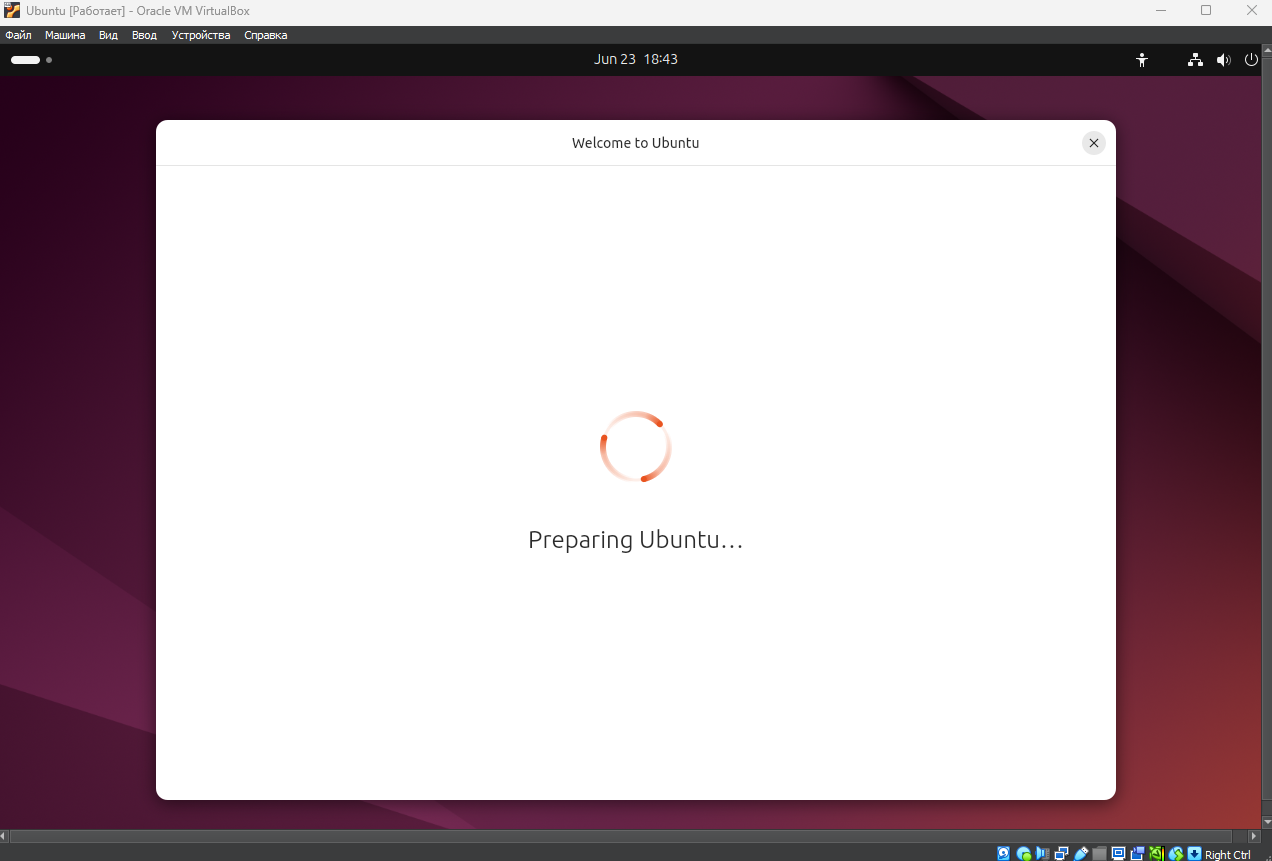


Рисунок 16 — Подготовка ОС к установке

Далее выбираем системный язык

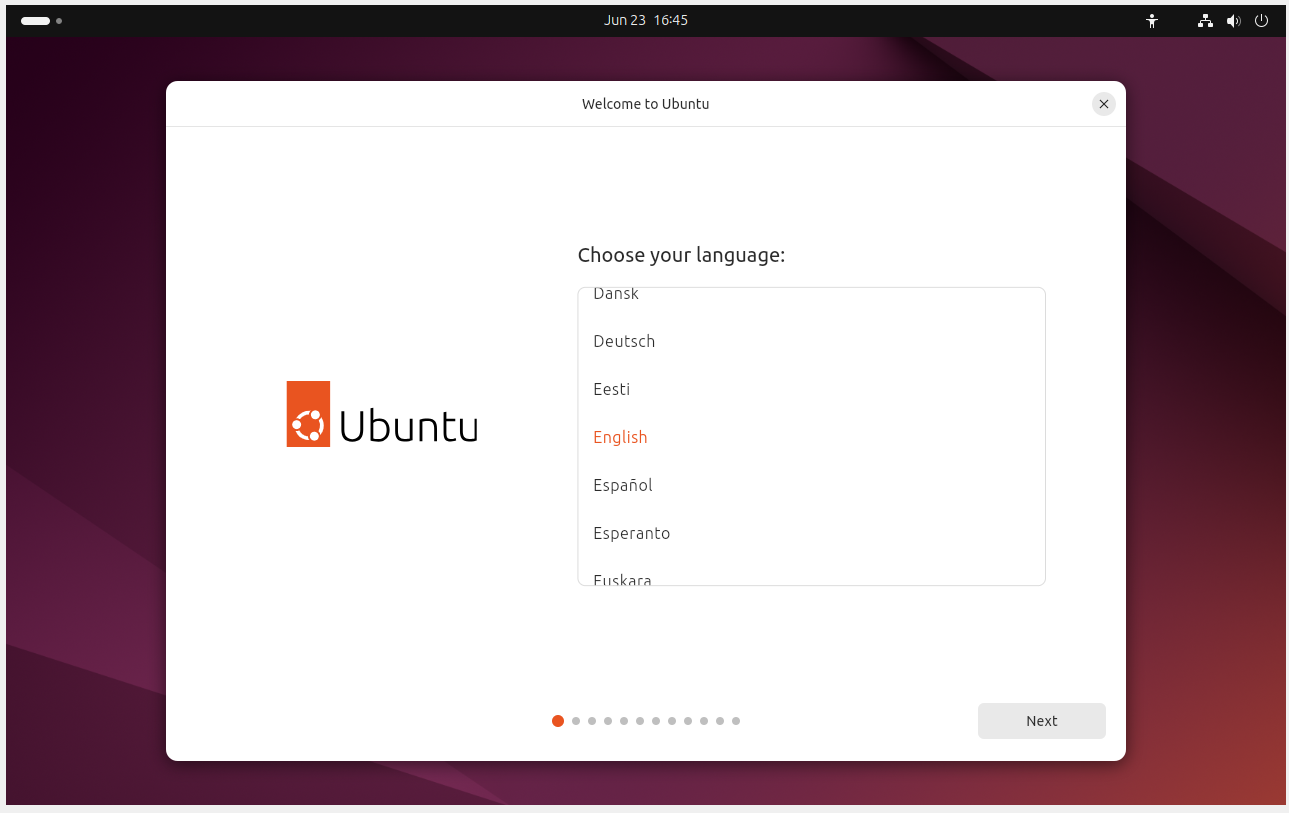


Рисунок 17 — Выбор языка системы

Далее выбираем функции для людей с ограничениями, если необходимо

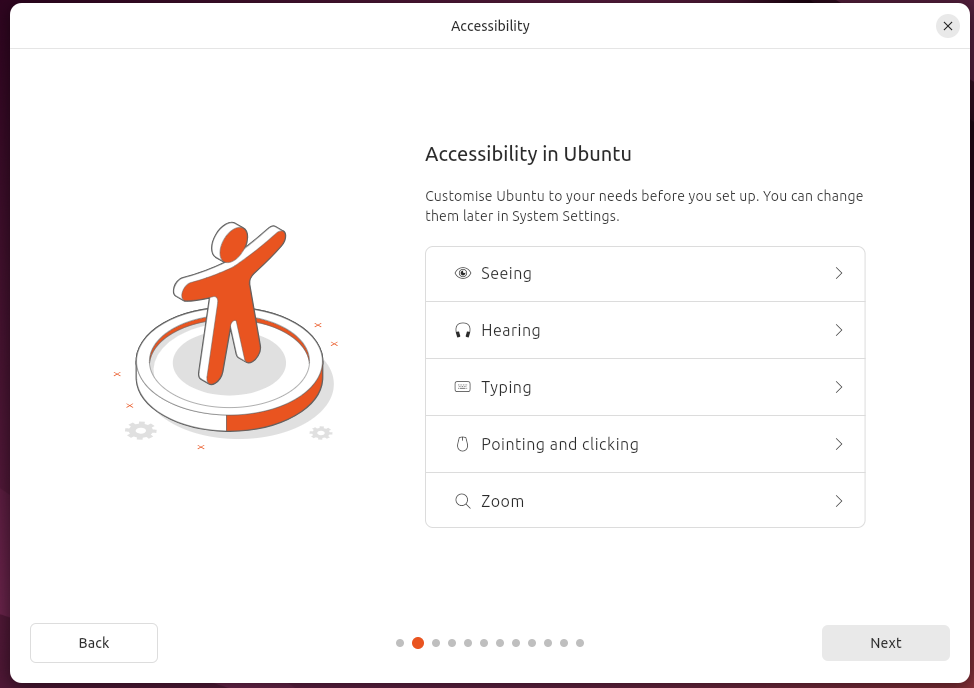


Рисунок 18 — Специальные функции

Далее настраиваем раскладку клавиатуры в системе

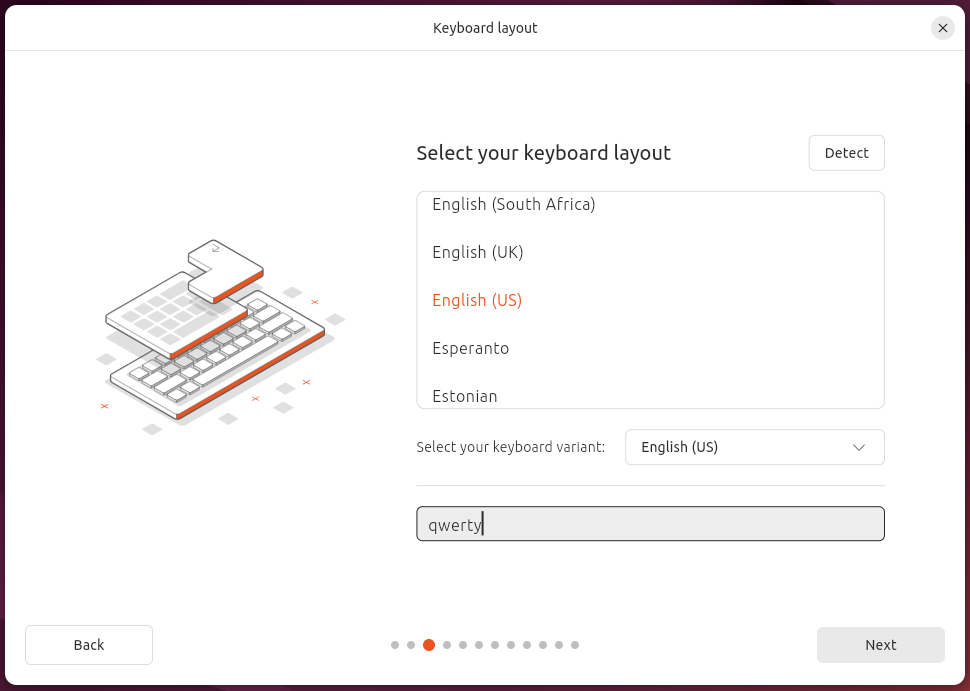


Рисунок 19 — Выбор раскладки

Далее выбираем тип подключение к интернету

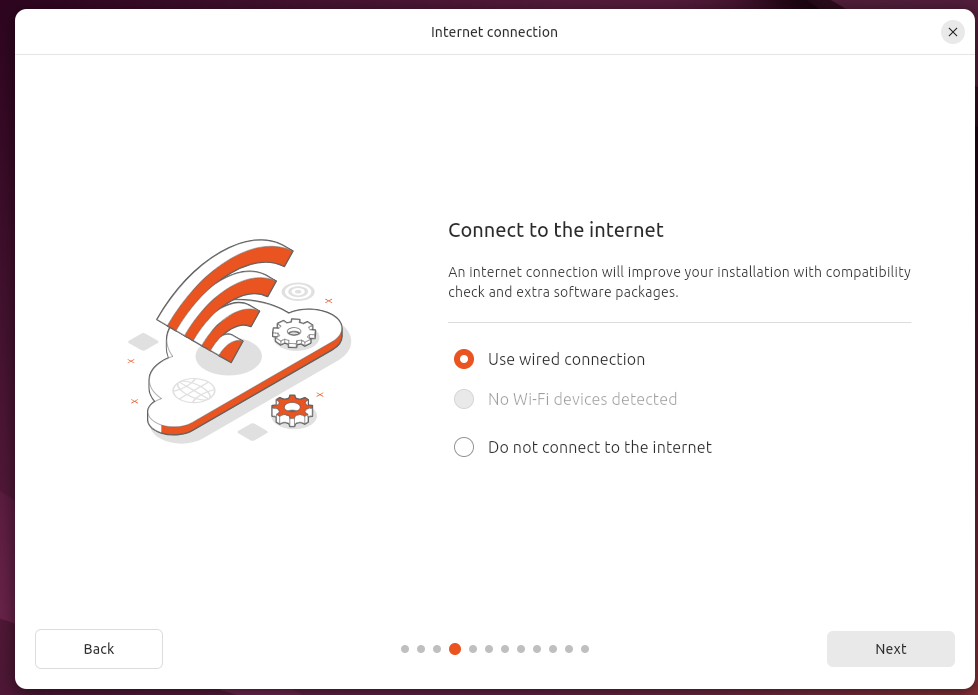


Рисунок 20 — Подключение к интернету

Обновляем установщик

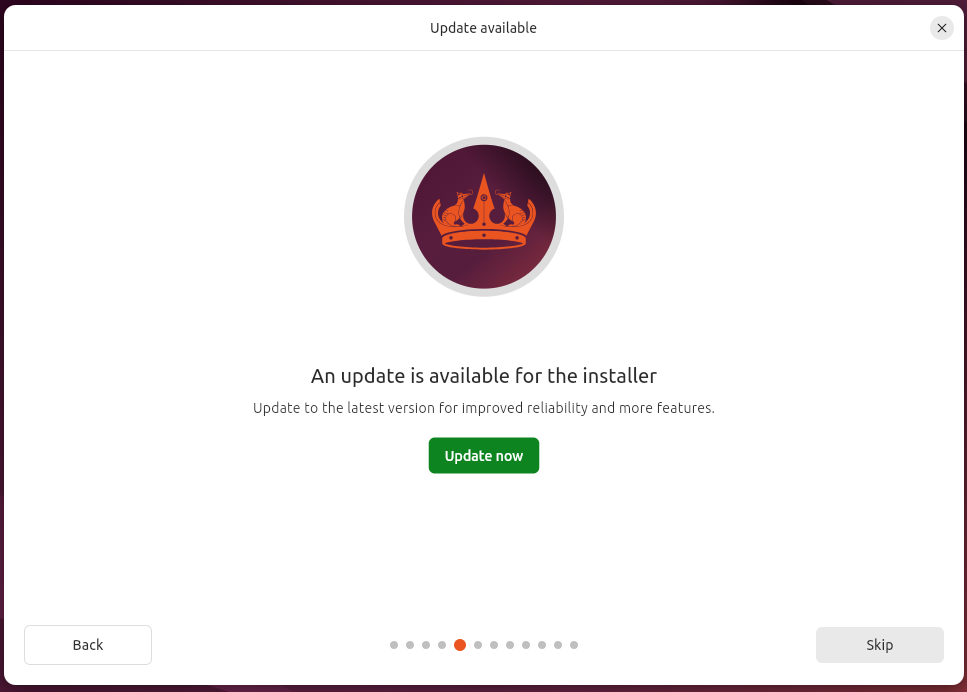


Рисунок 21 — Обновление установщика

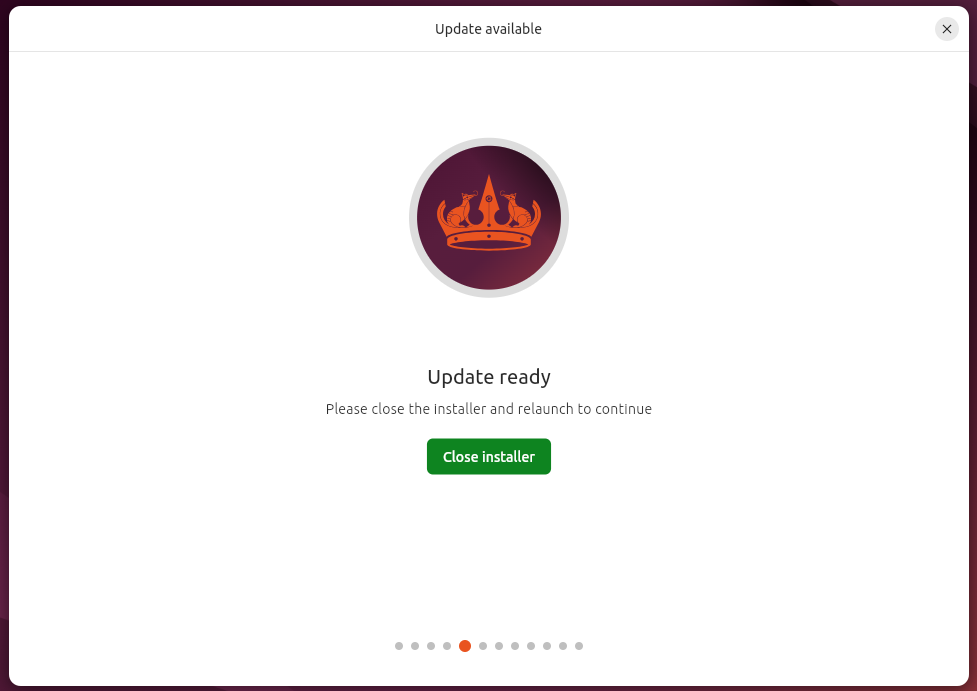


Рисунок 22 — Обновление установщика

Выбираем установку ОС



Рисунок 23 — Устанавливаем ОС

Затем указываем, какой тип установки нам необходим

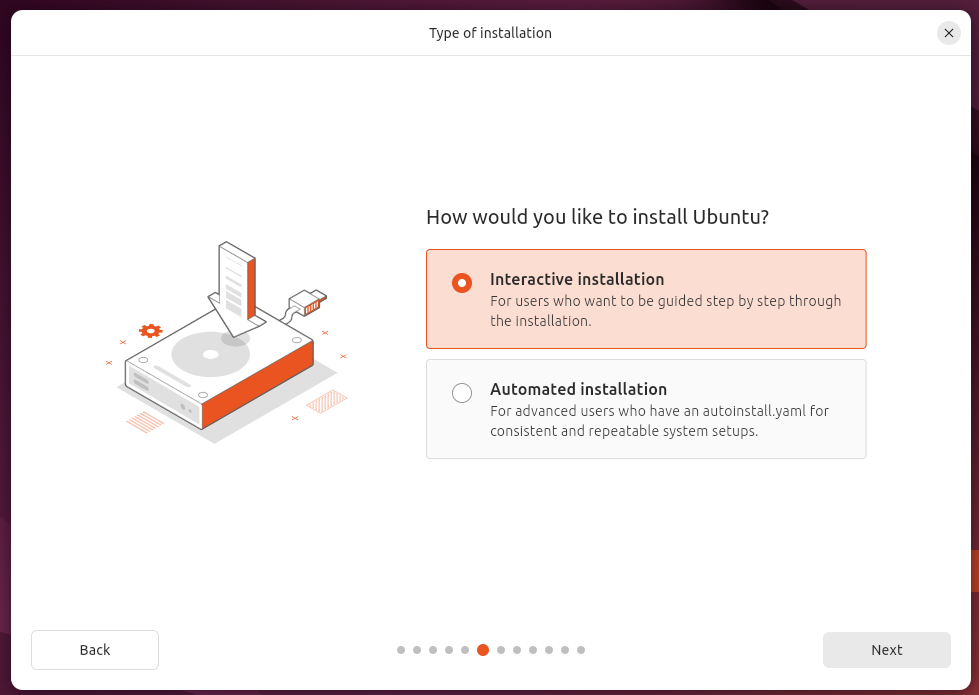


Рисунок 24 — Выбираем тип установки

Потом выбираем необходимый набор приложений, которые будут установлены

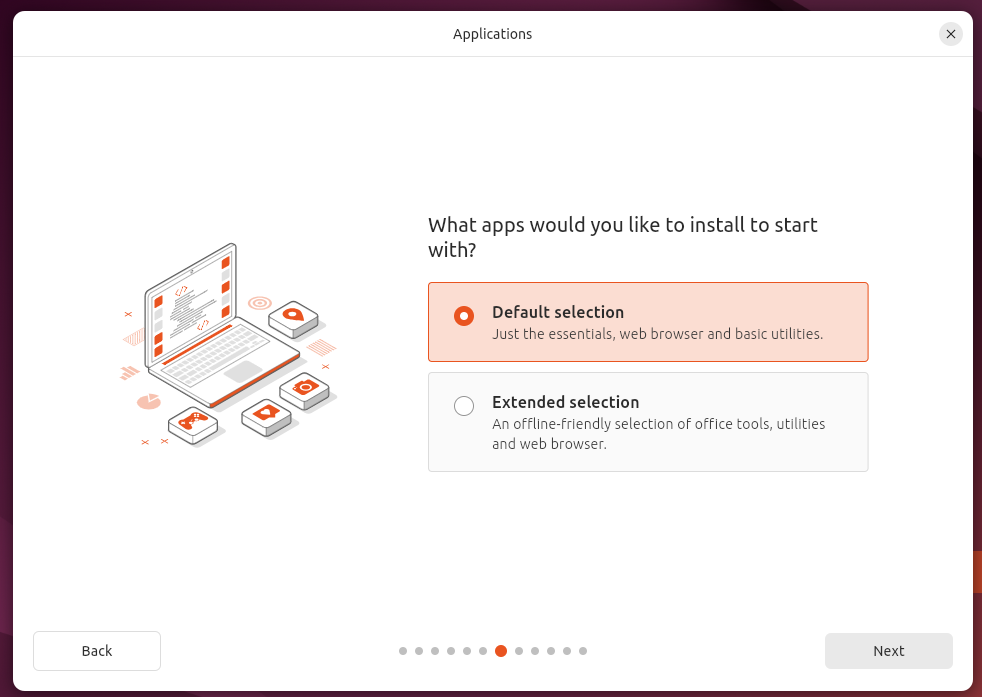


Рисунок 25 — Набор приложений при установке

Потом выбираем проприетарное программное обеспечение

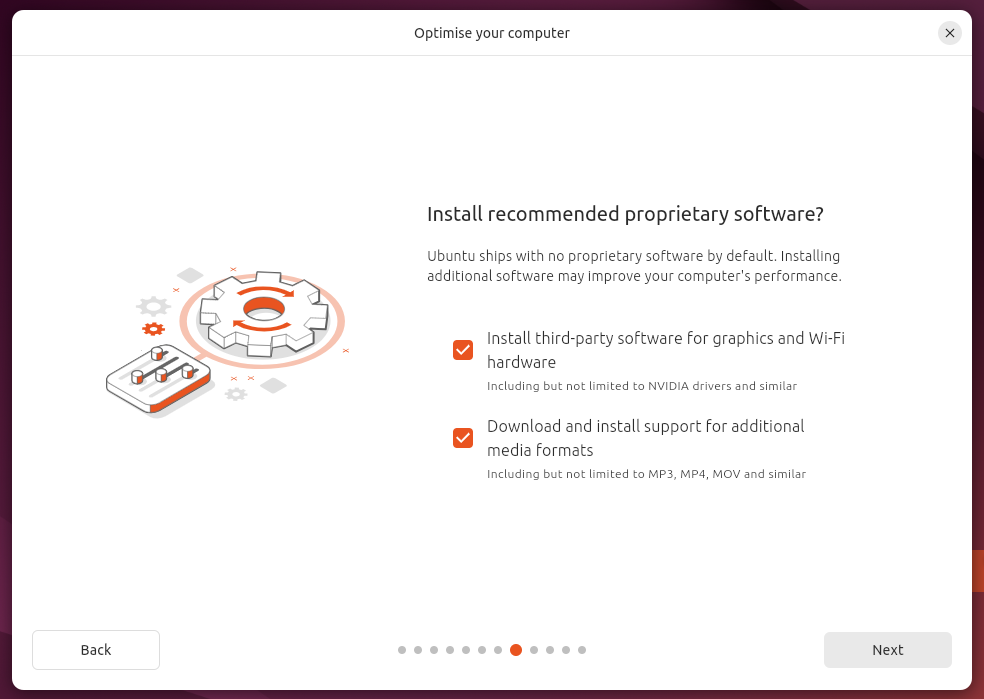


Рисунок 26 — Проприетарное программное обеспечение

Далее выбираем, что делать с носителем, на который ставится система

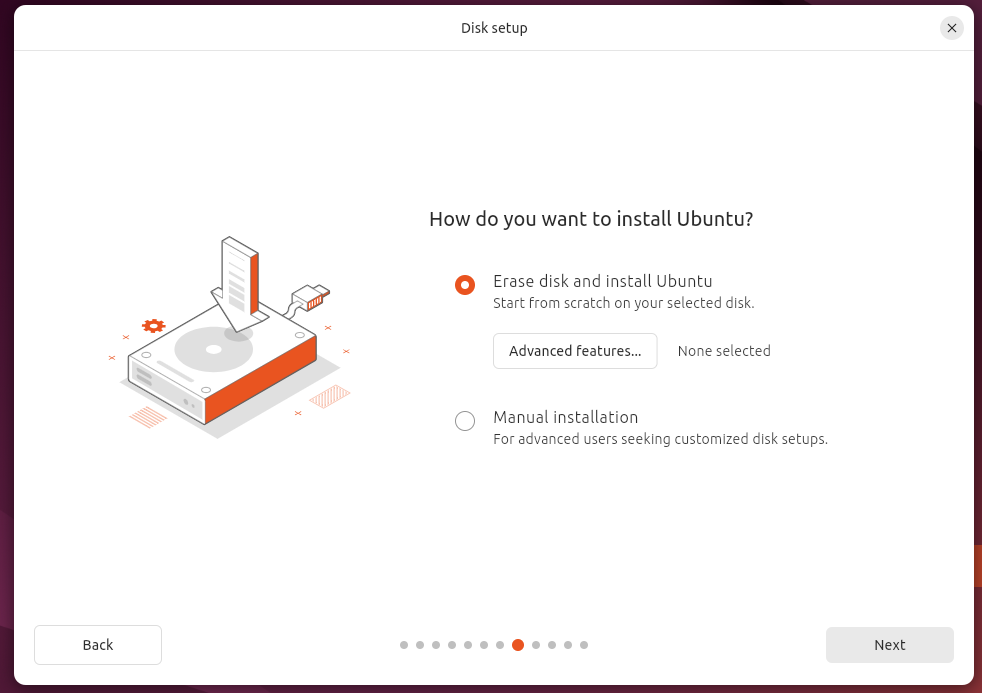


Рисунок 27 — Форматируем носитель и устанавливаем на него

Затем создаем учетную запись



Рисунок 28 — Создание учетной записи в Ubuntu

Затем выбираем временную зону

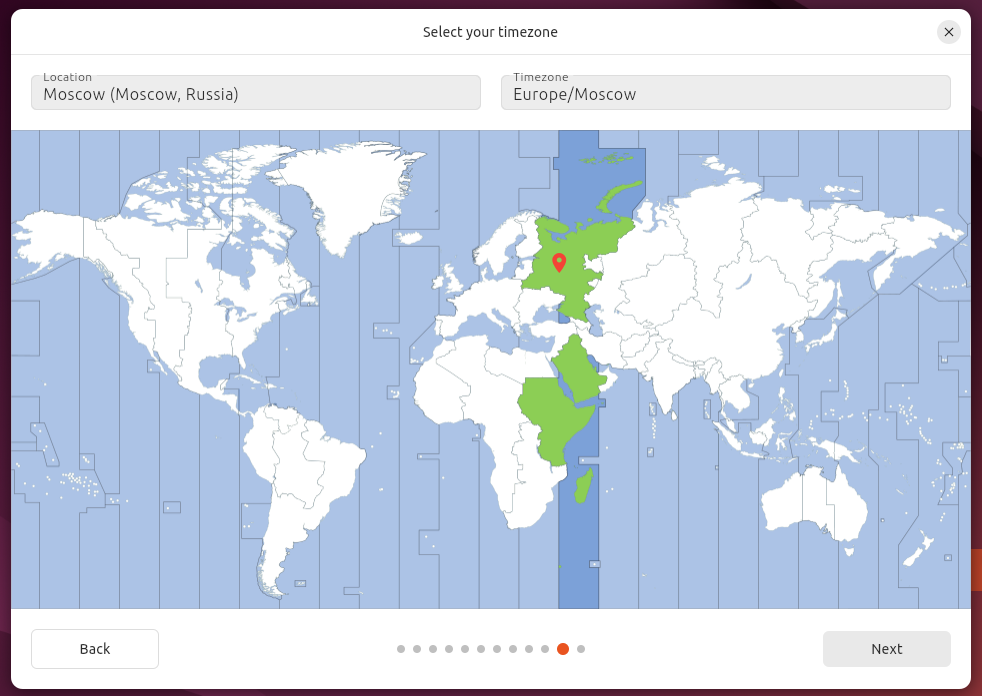


Рисунок 29 — Создание учетной записи в Ubuntu

Далее видим краткое описание нашей системы

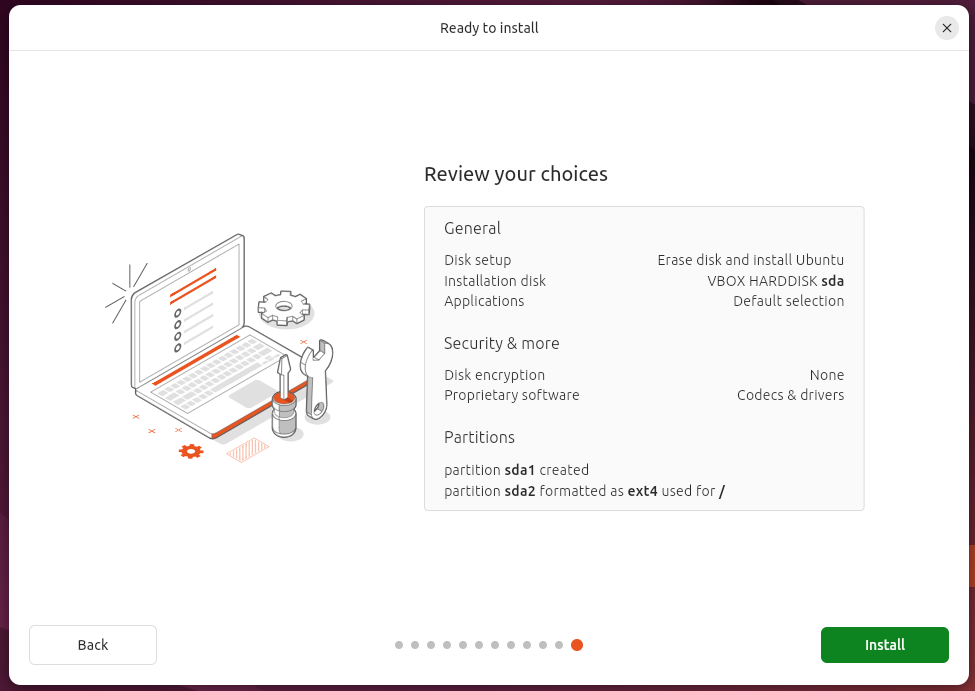


Рисунок 30 — Кратное описание выбранного ранее

Далее происходит установка ОС

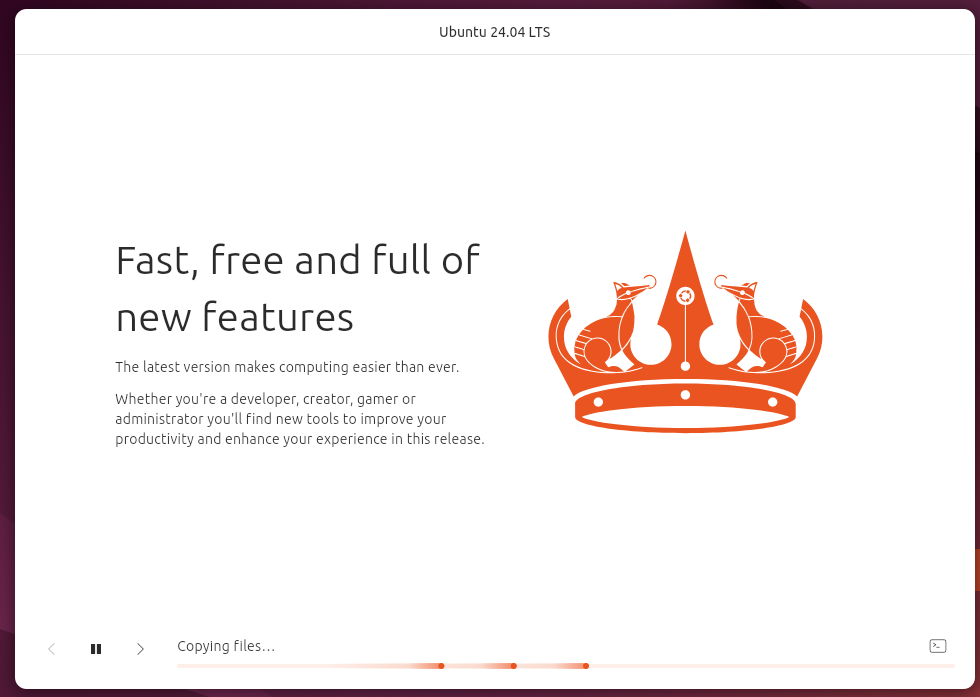


Рисунок 31 — Процесс установки

Затем требуется перезагрузить машину и извлечь образ

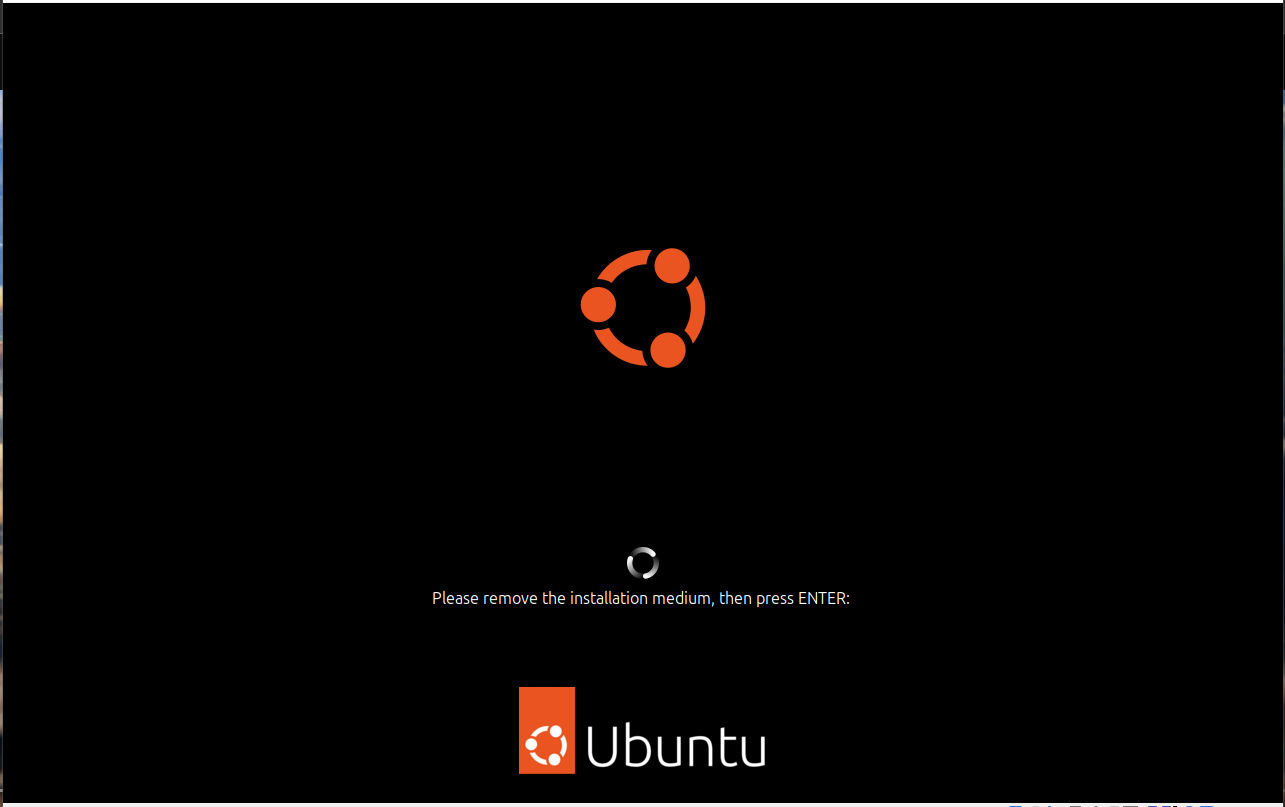


Рисунок 31 — Перезагрузка ОС

ОС Ubuntu готова к работе

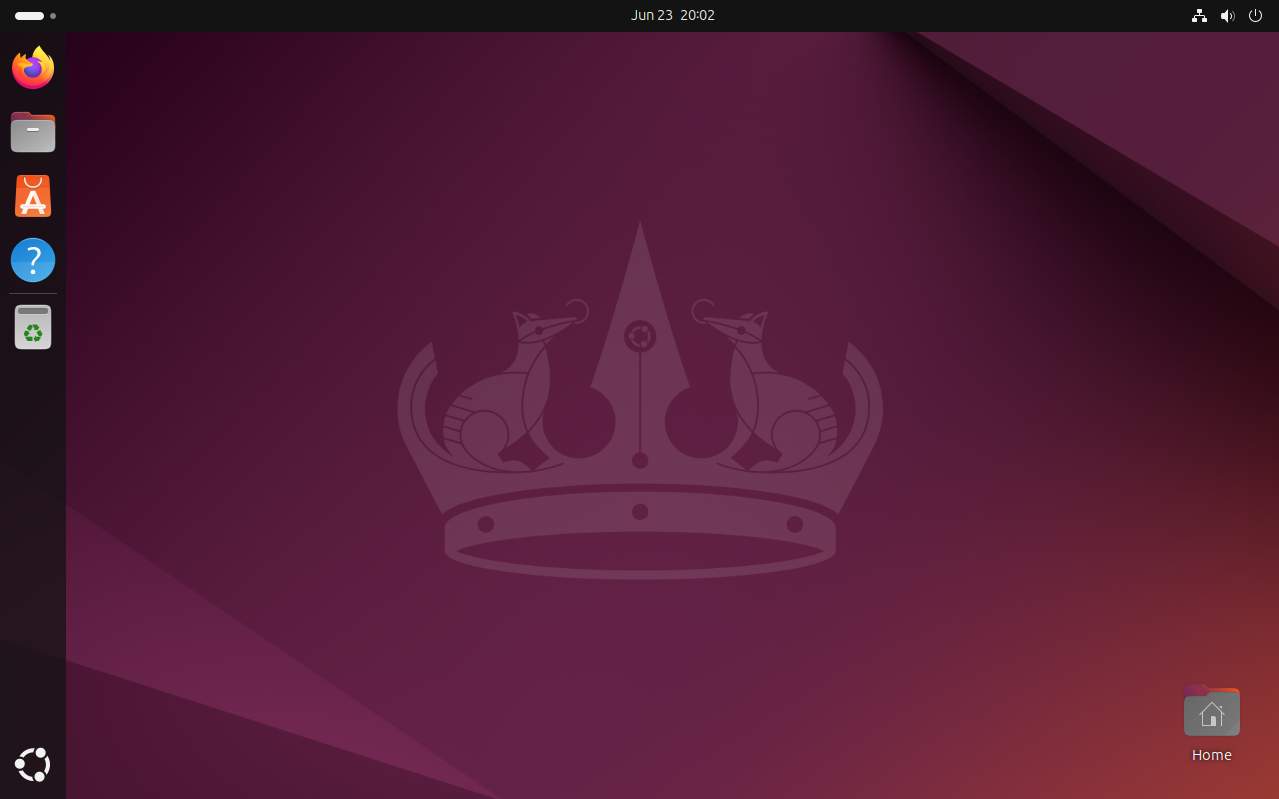


Рисунок 32 — ОС Ubuntu

## **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. К главным особенностям дистрибутива Ubuntu можно отнести:

* стабильность работы — не требует частых перезагрузок компьютера;
* безопасная система, не требующая антивирусов;
* бесплатная операционная система;
* приятный и понятный интерфейс, большое количество бесплатных тем и эффектов;
* быстрая и простая установка, даже для неопытных пользователей, основное программное обеспечение и драйвера устанавливается сразу, предполагается автоматическое обновление;
* имеется специализированный менеджер пакетов, через который устанавливается большинство программ;
* малотребовательная к ресурсам компьютера операционная система;
* модульность, пользователь может изменять структуру и компоненты операционной системы, вплоть до самого ядра.

Операционная система Ubuntu состоит из 8 основных частей:

* начальный загрузчик Grub, который обрабатывает процесс непосредственной загрузки ОС, задает параметры командной строки;
* ядро Linux является центральной частью системы, управляет процессором, памятью и устройствами ввода/вывода;
* демоны — программы, запускаемые самой системой и работающие в фоновом режиме без прямого взаимодействия с пользователем, обычно запускаются во время загрузки системы;
* командная оболочка обеспечивает интерфейс командного процессора, что позволяет управлять компьютером с помощью ввода команд в текстовом режиме;
* утилиты командной оболочки — в командной оболочке предоставляется некоторое количество основных встроенных команд;
* 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»;
* 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;
* программы рабочего стола — программы, работающие на рабочем столе, но не являющиеся частью среды рабочего стола.

# **2 Интегрированная среда разработки VS Code**

## **2.1 Описание VS Code**

Visual Studio Code – это продукт от компании Майкрософт. Инструмент, предназначенный для верстальщиков и разработчиков. Один из самых популярных в соответствующей нише. Поддерживает такие языки как: JavaScript, HTML, CSS, PHP, Go Ruby, Python, C #, TypeScript, C/C++.

Представляет собой редактор кода от Microsoft, выступающий «облегченной» интерпретацией VisualStudio. С помощью него можно не только заниматься написанием приложений. Visual Studio Code поддерживает большое количество плагинов, которые позволят «разогнать» редактор до полноценной среды программирования.

Подходит для работы на операционных системах:

* Windows;
* MacOS;
* Linux.

Распространяется на бесплатной основе, благодаря чему набрал огромную популярность. Работает одинаково хорошо как на стареньких устройствах, так и на современных компьютерах.

Рассматриваемый инструмент позволяет работать почти со всеми современными языками программирования. Он также поддерживает подключение расширений и фреймворков наиболее известных ЯП – React JS, Vue.js, LESS, SCSS.

## **2.2 Описание установки**

Для установки интегрированной среды разработки Visual Studio Code в операционной системе Ubuntu нужно открыть магазин приложений App Center и найти в нём приложение Visual Studio Code.

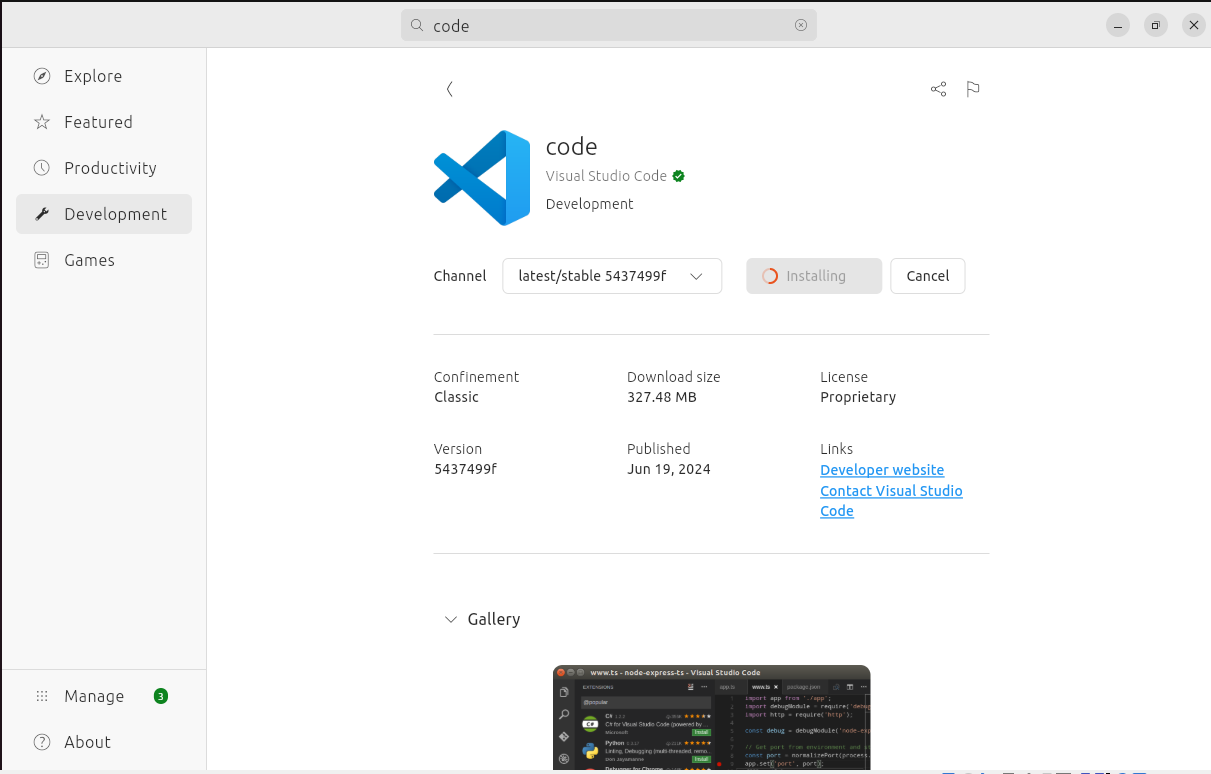


Рисунок 32 — VS Code в магазине приложений

После установки запускаем IDE

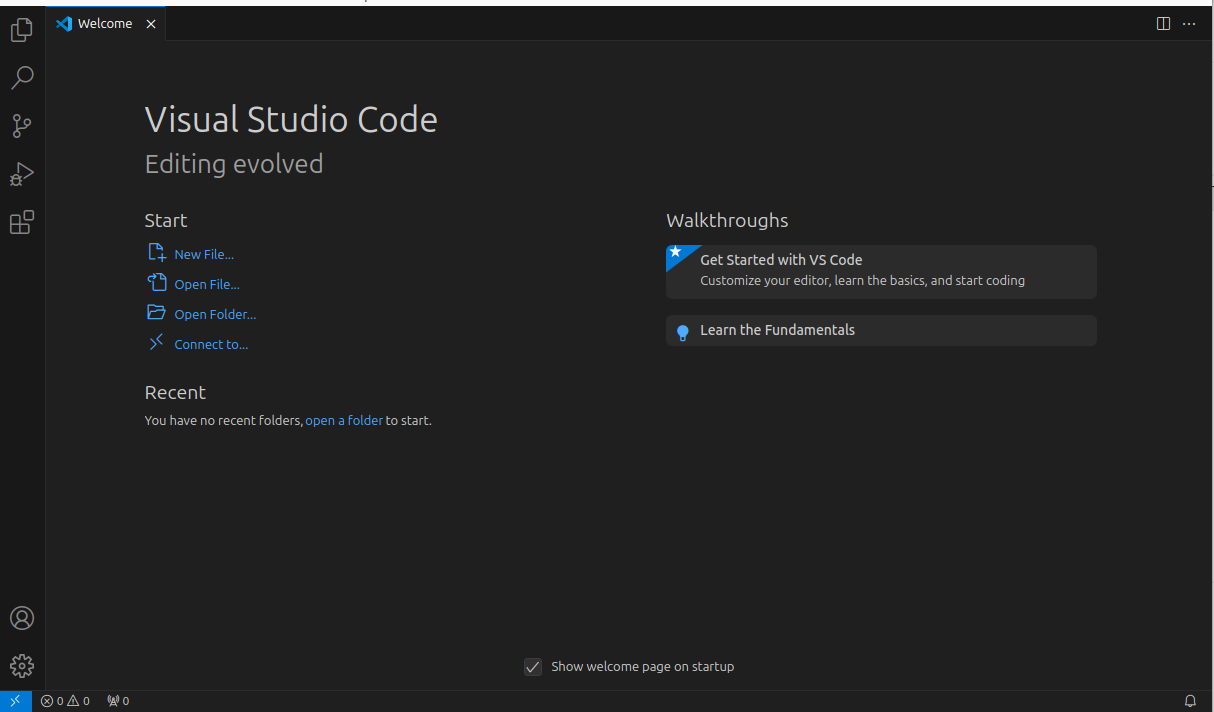


Рисунок 33 — VS Code

Далее для корректной работы необходимо загрузить расширение C/C++

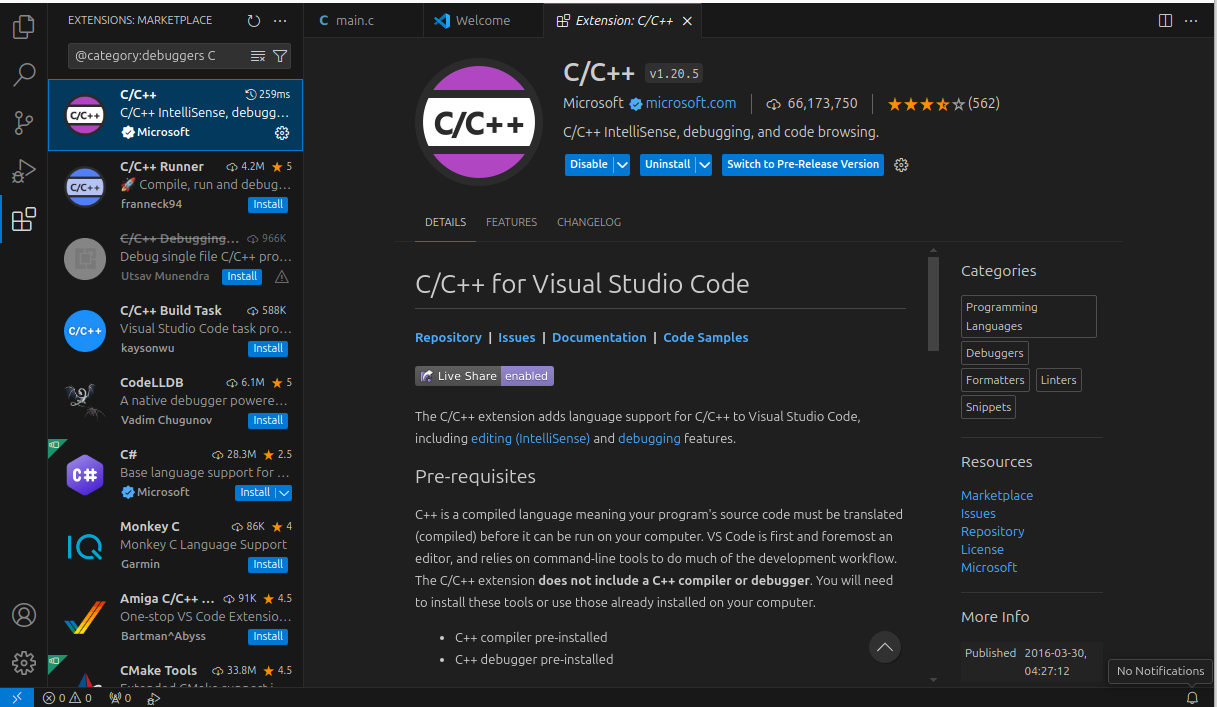


Рисунок 34 — C/C++

Далее проверяем, установлен ли GNU Compiler Collection или сокращенно GCC. Для этого используем команду *gcc -v*

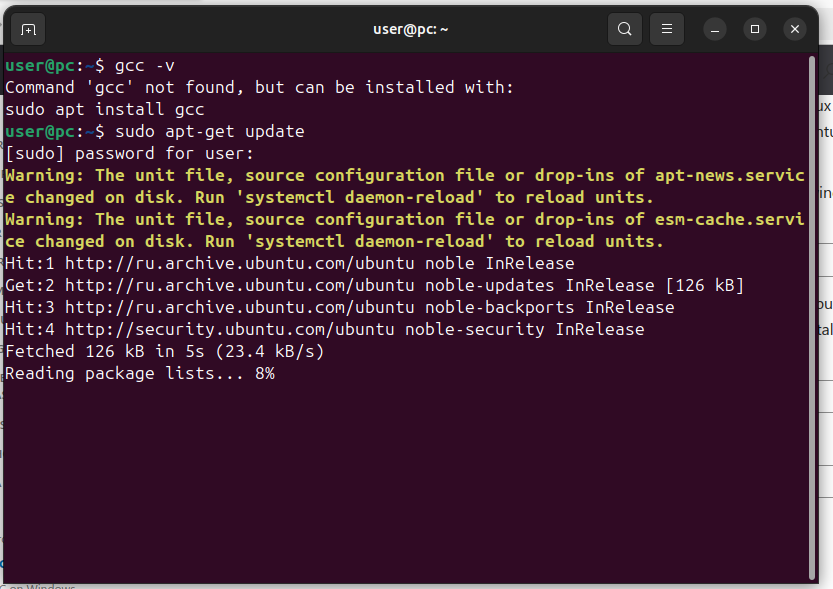


Рисунок 35 – Проверяем установлен ли GCC

Устанавливаем GNU compiler tools и GDB debugger с помощью команды *sudo apt-get install build-essential gdb*

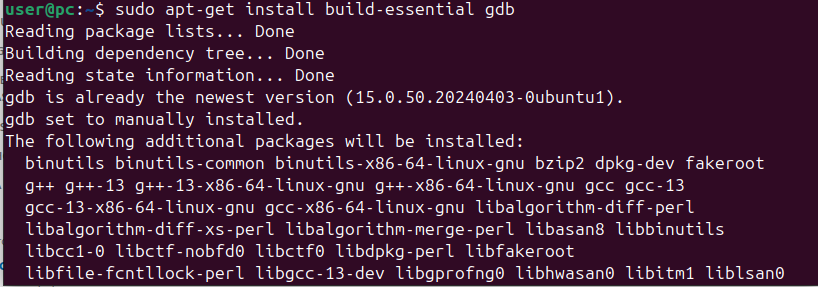


Рисунок 36 – Установка нужных компонентов

Далее создаем папку с проектами и создаем в ней подпапку с проектом. Для этого используем команды

* *mkdir* – создание директории
* *cd* – переход в указанную директорию

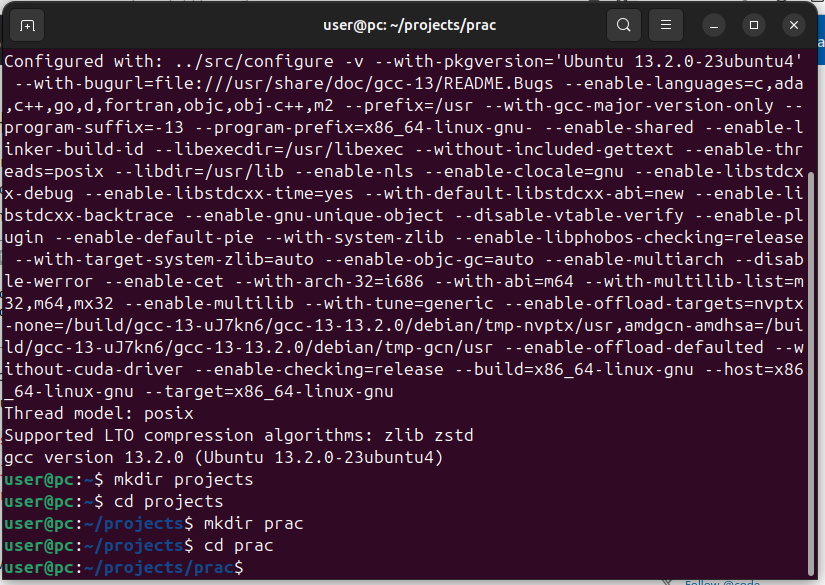


Рисунок 37 – Создаем нужные директории

Далее с помощью команды *code .* запускаем VS Code в выбранной директории



Рисунок 38 – Вводим команду для запуска IDE

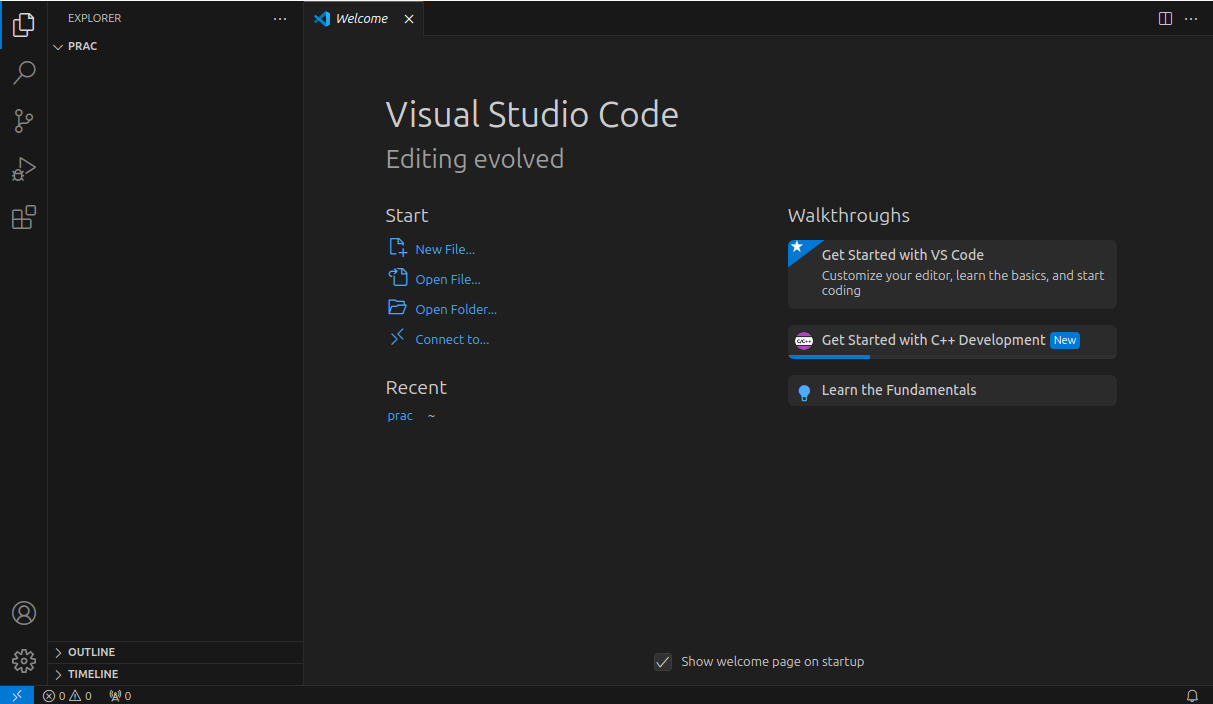


Рисунок 39 – IDE запущенная в нужной директории

Создаем пустой файл с название *prac.c*

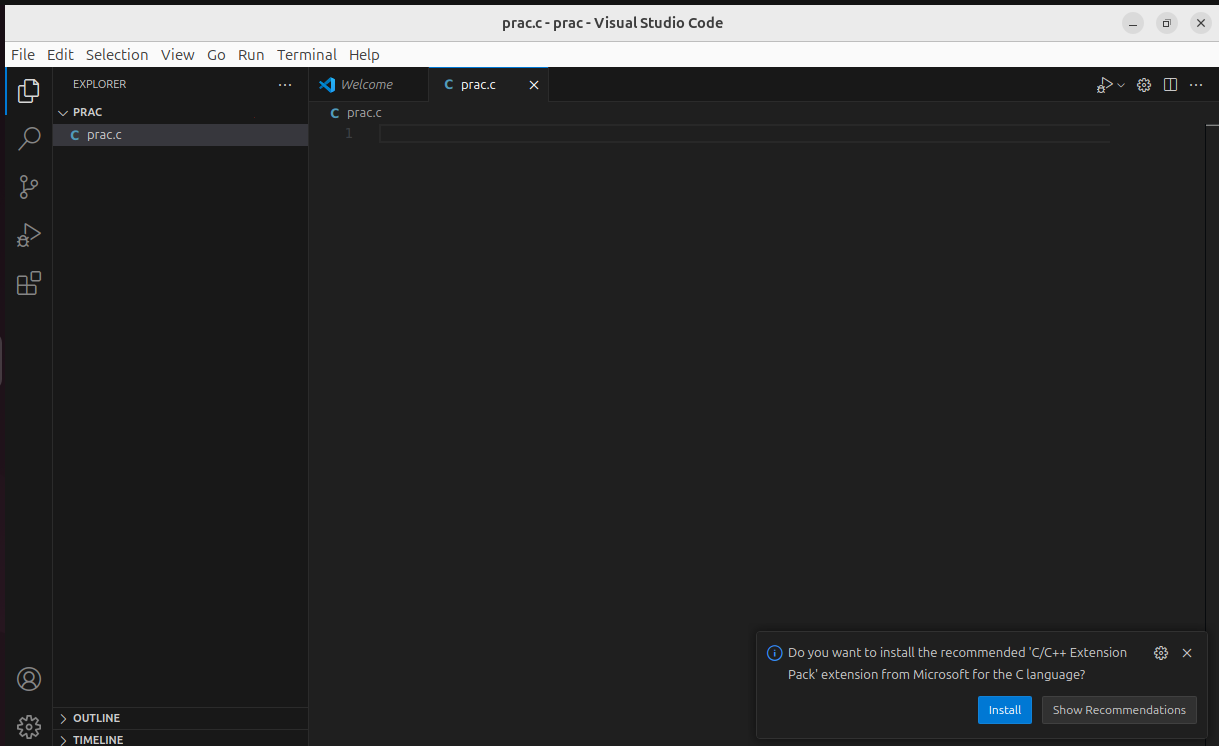


Рисунок 40 – Создали файл prac.c

Для более комфортной работы с языком C устанавливаем extension pack

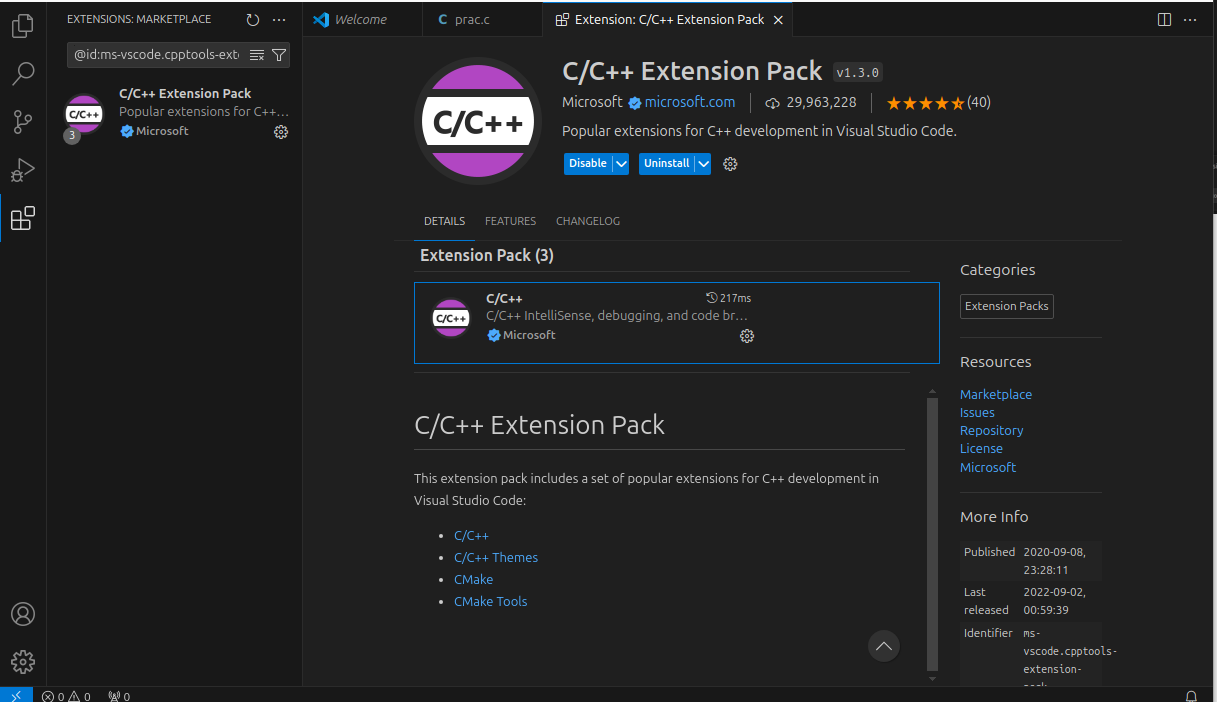


Рисунок 41 – Установка Extension pack

Для проверки работы напишем простую команду, которая выводит в консоль *«Hello World»*

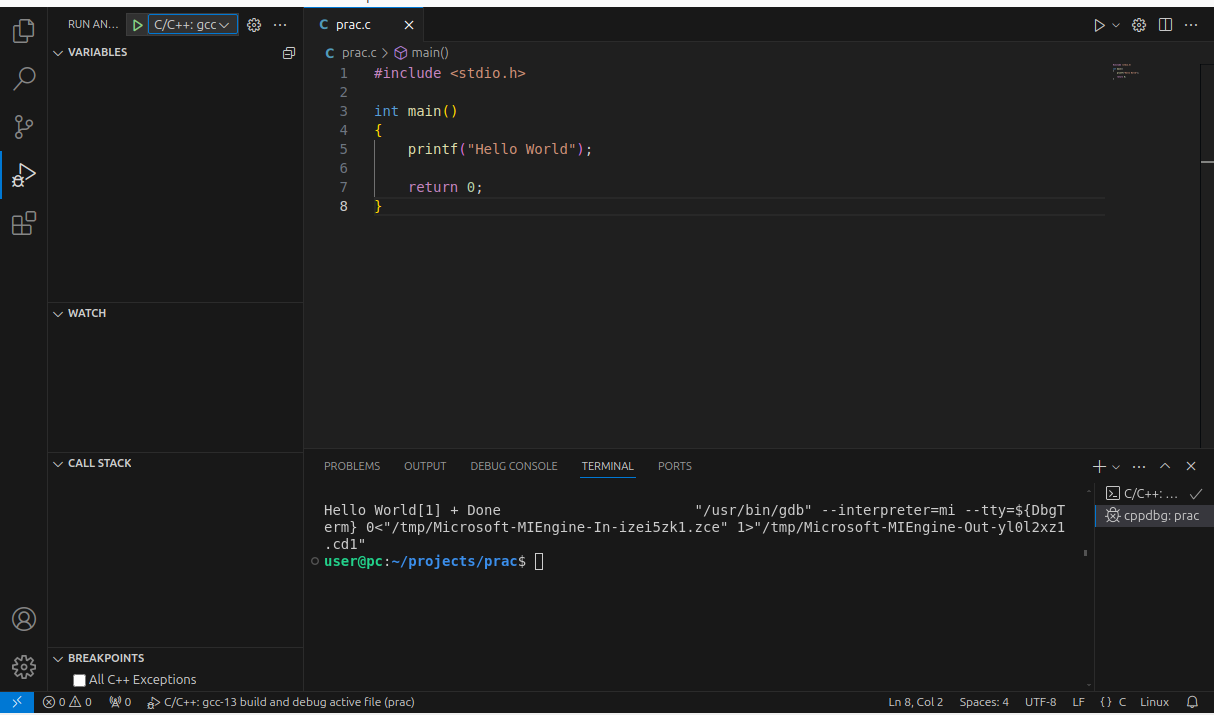


Рисунок 42 – Проверка работоспособности

# **3 Разработка программы на языке С в среде VS Code**

## **3.1 Формулировка задачи**

Разработать программу, которая сортирует массив произвольной сортировки. В качестве алгоритма выбрать сортировку «пузырьком».

## **3.2 Описание программы**

Программа принимает на вход размер массива и его элементы. Отсортированный массив выводится на экран, а также записывается в файл "res.txt".

Для сортировки используется стандартная пузырьковая сортировка. Элементы массива сравниваются между собой, пока массив не будет отсортирован полностью.

Результат работы записывается в файл. В случае ошибки программа уведомляет пользователя об этом. А если запись прошла успешно, то выводится путь до файла с результатом

## **3.3 Работа программы**

На вход подется число 4 – размер массива и исходный массив для сортировки – 4 3 2 1. После сортировки результат выводится на экран вместе с сообщением об удачной записи.

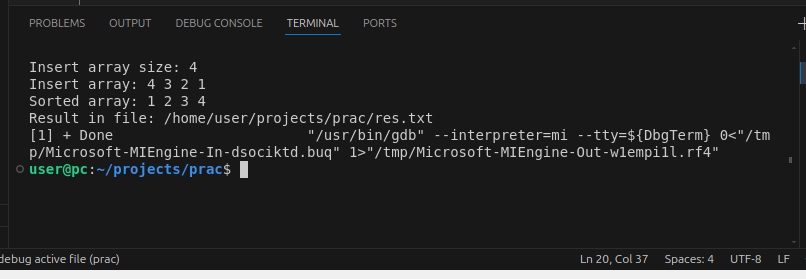


Рисунок 43 – Результат работы программы

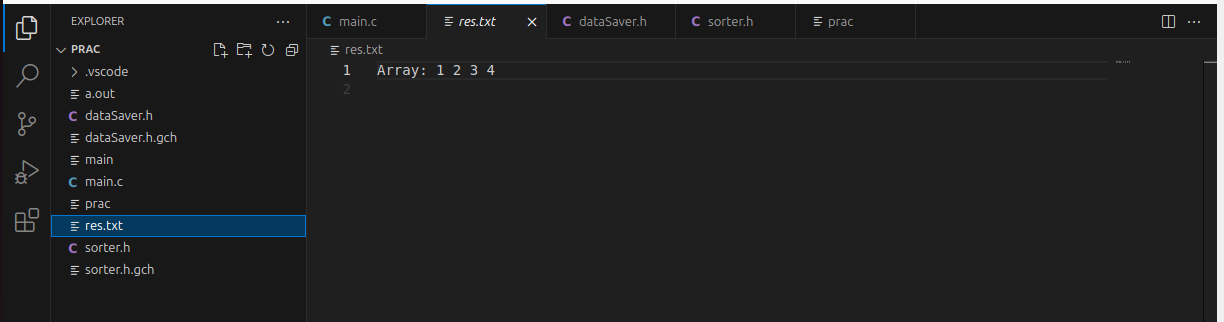


Рисунок 44 – Результат записи в файл

Программа также успешно собирается и через консоль

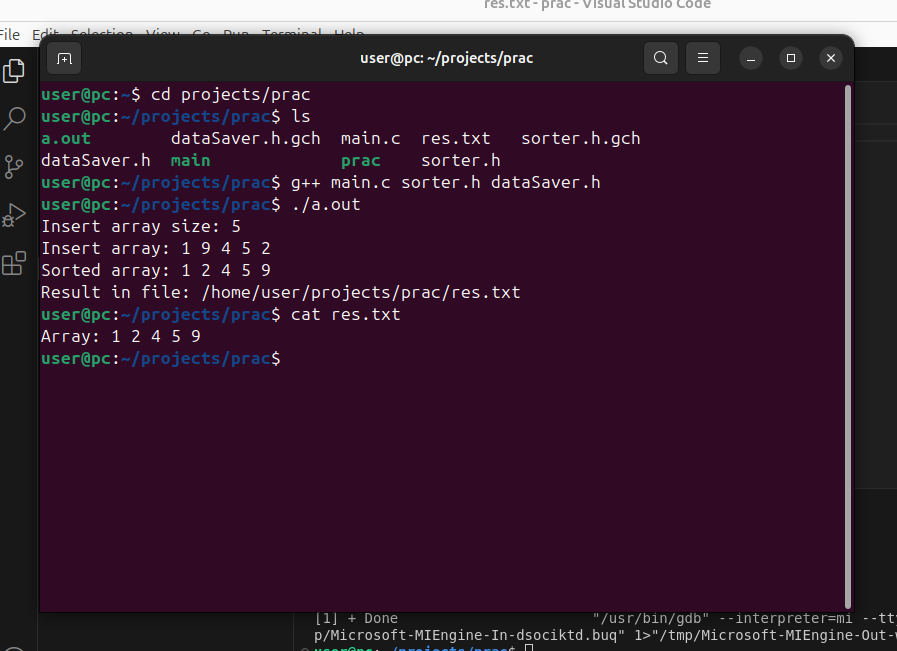


Рисунок 45 - Компиляция и запуск программы через консоль

## **3.4 Компиляция и отладка программы в среде VS Code**

Для компиляции программ в VS Code можно использовать встроенную командную строку. При каждом запуске происходит компиляция кода.

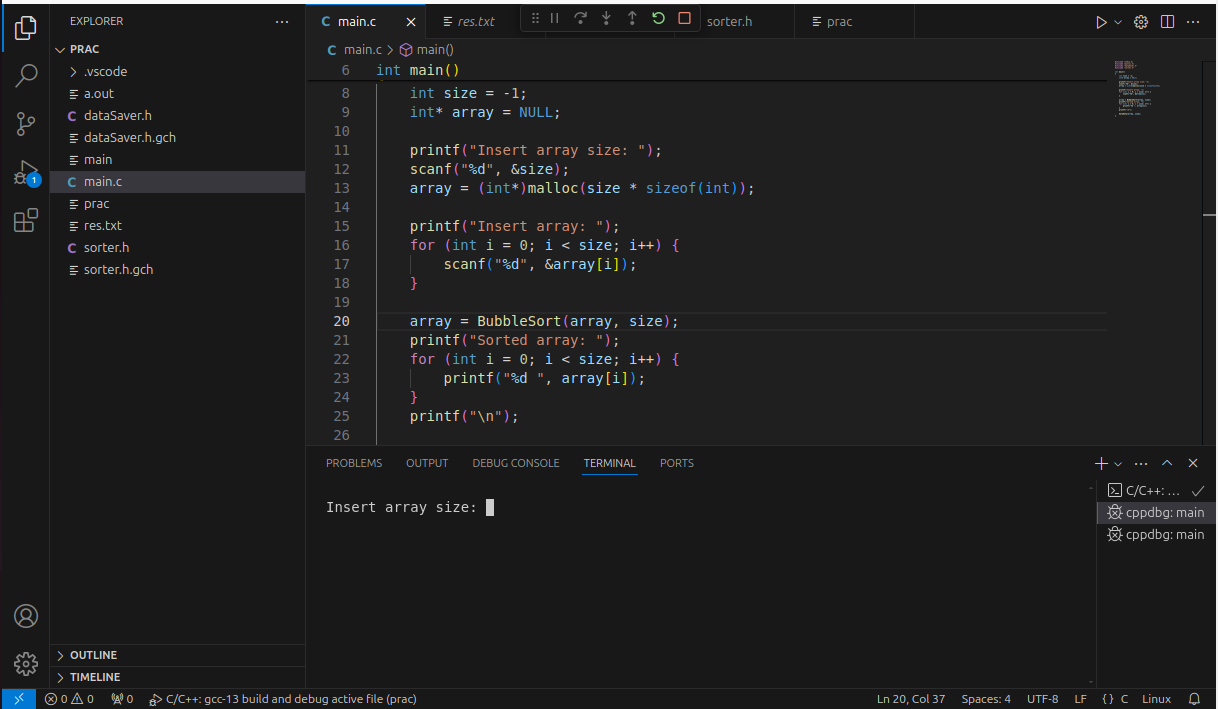


Рисунок 46 — Код скомпилирован и успешно запущен

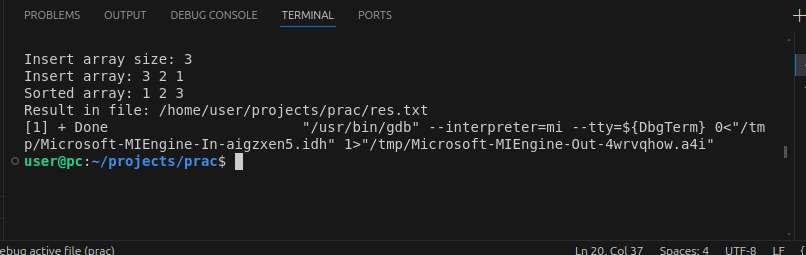


Рисунок 47— Вывод работы программы в терминале

Для отладки в среде VS Code используется встроенный отладчик GDB. Для запуска отладчика необходимо нажать кнопку «Debug C/C++ File». Также можно поставить точку останова на любую строку кода. Для завершения отладки нужно нажать кнопку «Stop» и убрать точку останова, если она была поставлена.

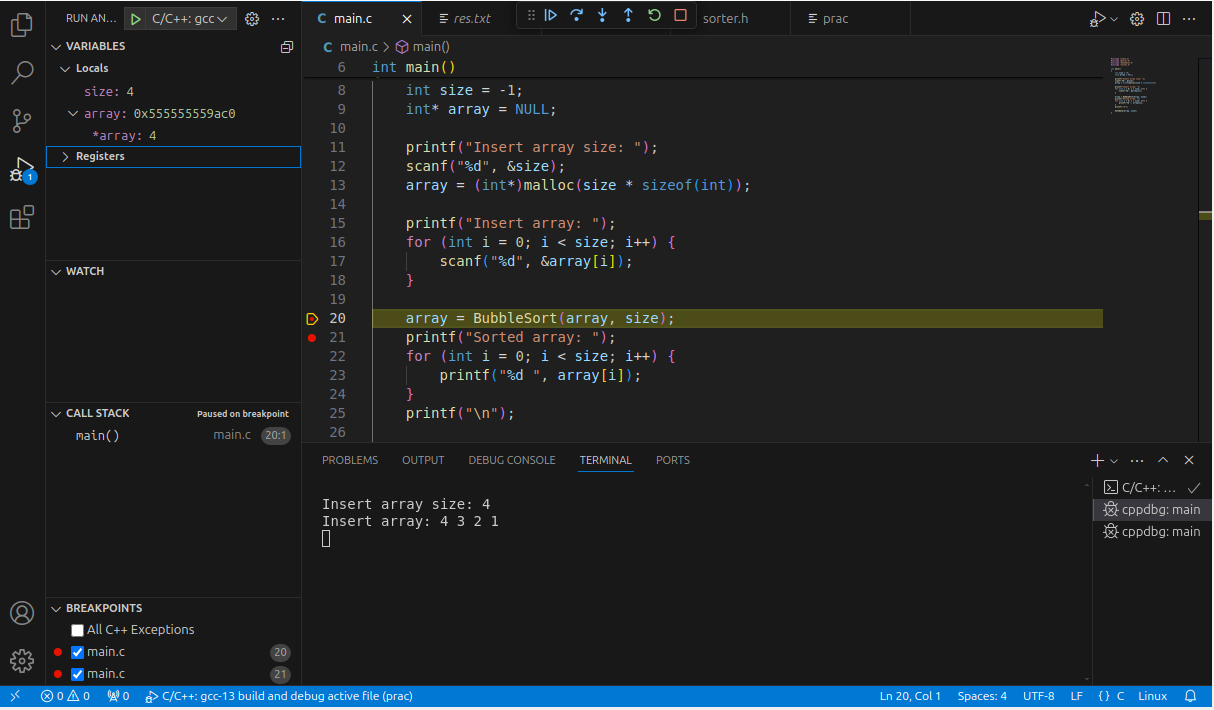


Рисунок 39 — Отладка в VS Code

# **Заключение**

В ходе выполнения практической работы была изучена и установлена операционная система Ubuntu на виртуальную машину VirtualBox, установлена среда разработки VS Code в ОС Ubuntu, были изучены основные инструменты среды и разработана программа на языке С в среде VS Code. 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. VS Code. Режим доступа: <https://code.visualstudio.com/docs/languages/cpp>

# **Приложение А**

Листинг программы

**main.c**

#include <stdio.h>

#include <malloc.h>

#include "dataSaver.h"

#include "sorter.h"

int main()

{

int size = -1;

int\* array = NULL;

printf("Insert array size: ");

scanf("%d", &size);

array = (int\*)malloc(size \* sizeof(int));

printf("Insert array: ");

for (int i = 0; i < size; i++) {

scanf("%d", &array[i]);

}

array = BubbleSort(array, size);

printf("Sorted array: ");

for (int i = 0; i < size; i++) {

printf("%d ", array[i]);

}

printf("\n", array);

SaveData(array, size);

}

**dataSaver.h**

#ifndef DATA\_SAVER

#define DATA\_SAVER

#include <stdio.h>

#define DATA\_PATH "/home/user/projects/prac/res.txt"

void SaveData(int\* array, int size)

{

FILE\* file = NULL;

file = fopen(DATA\_PATH, "w+");

if (file == NULL) {

printf("Error while writing file: %s!\n", DATA\_PATH);

return;

}

fprintf(file, "Array: ", array);

for (int i = 0; i < size; i++) {

fprintf(file, "%d ", array[i]);

}

fprintf(file, "\n", array);

fclose(file);

printf("Result in file: %s\n", DATA\_PATH);

}

#endif

**sorter.h**

#ifndef SORTER

#define SORTER

int\* BubbleSort(int\* arr, int size)

{

for (int i = 0; i < size; i++) {

for (int j = 0; j < size - 1; j++) {

if (arr[j] < arr[j + 1]) {

continue;

}

int tmp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tmp;

}

}

return arr;

}

#endif

# **Приложение Б**

Результаты испытаний

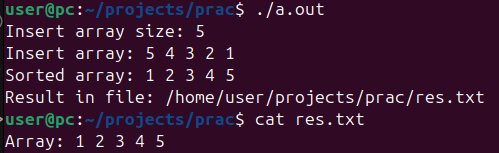


Рисунок Б.1 — Ввод 5; 5 4 3 2 1

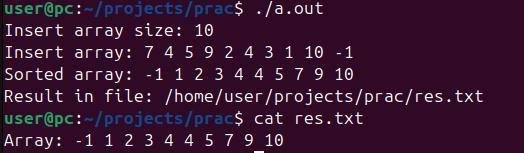


Рисунок Б.2 — Ввод a, b, c, d, e: 1 2 3 4 5