**Целищев Анатолий**

**Лабораторная работа №1**

**ВИРТУАЛИЗАЦИЯ, ГИПЕРВИЗОРЫ,**

**СОЗДАНИЕ ВИРТУАЛЬНЫХ МАШИН**

**Цели работы:**

* Изучить состав инструментальных средств создания и сопровождения аппаратных конфигураций виртуальных машин (ВМ);
* Получить практические навыки выполнения типовых операций мониторинга и управления состояниями виртуальных машин;
* Изучить функциональные возможности интеграции виртуальных (гостевых) и физической (хостовой) машин;
* Ознакомиться с типами гипервизоров и их возможностями от различных разработчиков.

**Ход работы**

1. **Установка VM VirtualBox**

Пройдем на официальный сайт <https://www.virtualbox.org> и в разделе *Downloads* увидим предлагаемые способы установки гипервизора VirtualBox. Так как лабораторная работа выполняется на ОС Ubuntu 22.04, имеются два варианта: скачать deb-пакет для соответствующей версии системы либо выполнить установку через терминал. Выберем последний способ.

Следуя инструкциям, добавим в файл /etc/apt/sources.txt требуемую строку:



Рис 1. Открываем файл при помощи текстового редактора nano

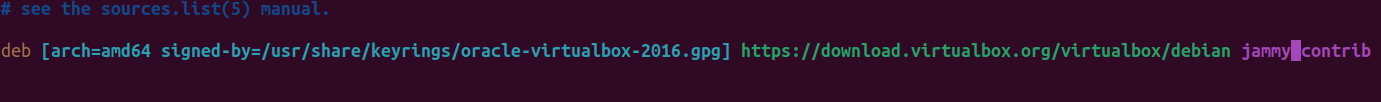


Рис 2. Вставка требуемой строки в файл /etc/apt/sources.txt

Установим и зарегистрируем публичный ключ Oracle, исполнив команду:

wget -O- https://www.virtualbox.org/download/oracle\_vbox\_2016.asc | sudo gpg --dearmor --yes --output /usr/share/keyrings/oracle-virtualbox-2016.gpg

Исполним следующую команду и дождемся установки VirtualBox:

sudo apt-get update && sudo apt-get install virtualbox-6.1

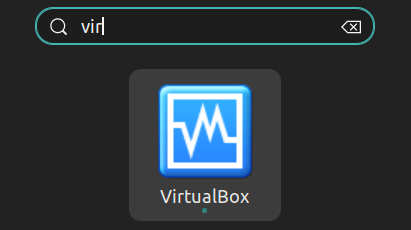


Рис 3. Успешно установленная программа.

1. **Создание виртуальной машины, ее настройка**

Для начала создадим виртуальную машину. Запустим VirtualBox, “Машина” -> “Создать”. Введем имя виртуальной машины, тип операционной системы и версию.

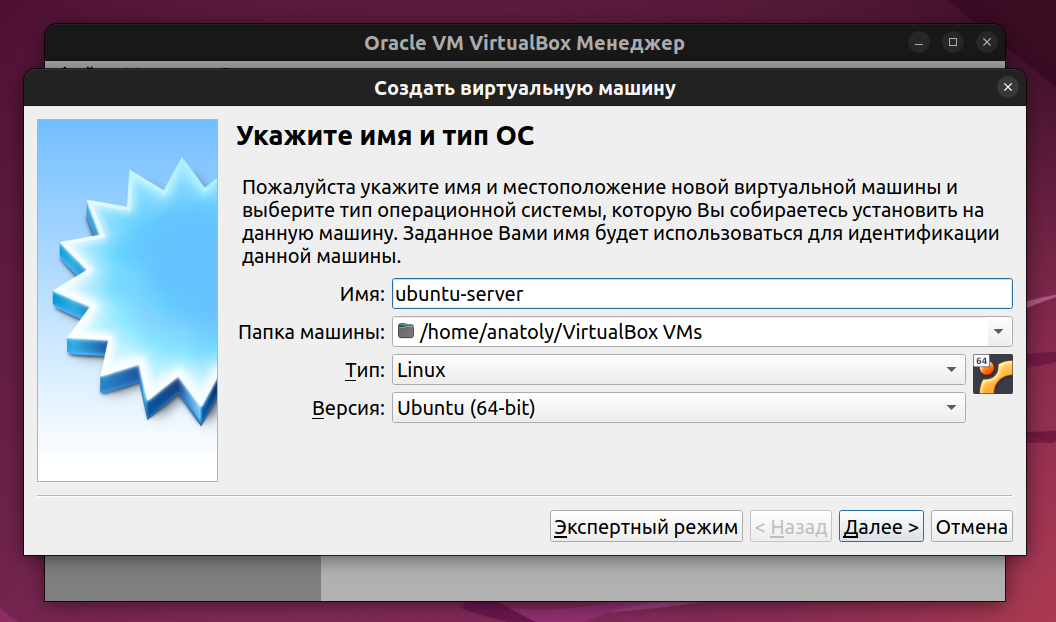


Рис 4. Указываем имя и тип ОС.

Выделим 2 гб оперативной памяти для ВМ.

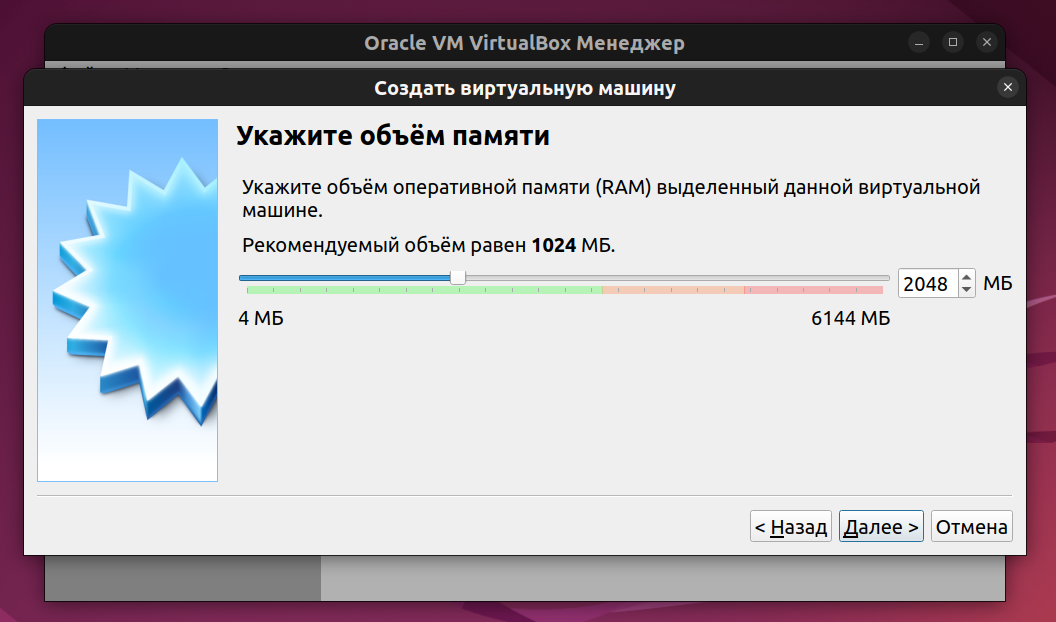


Рис 5. Выделение оперативной памяти объемом 2 гб.

Создадим новый виртуальный жесткий диск, тип VDI,

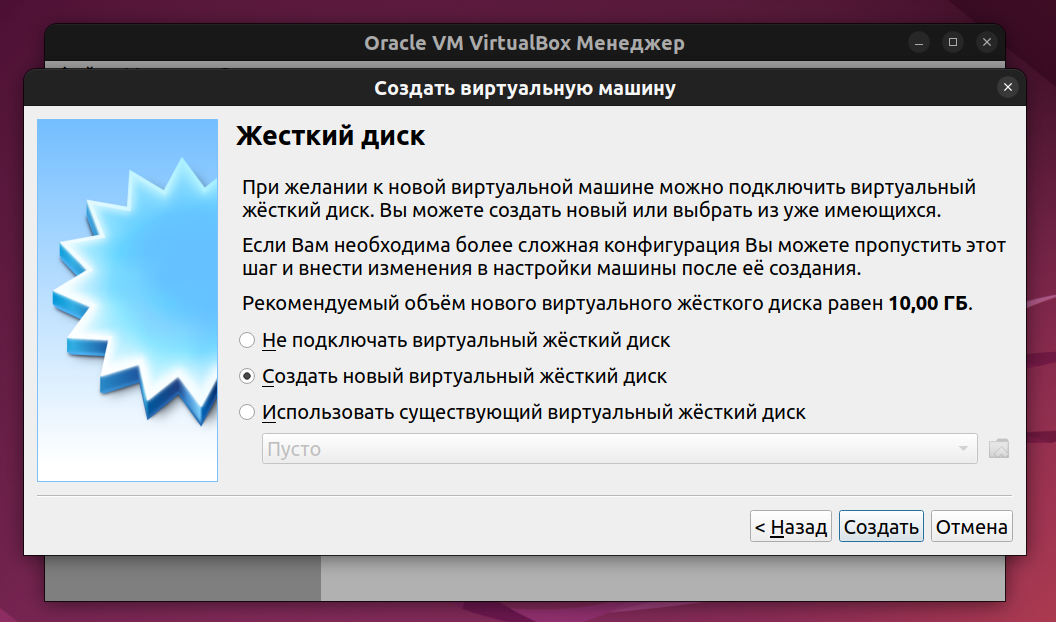


Рис 6. Создание нового виртуального жесткого диска.

Формат хранения - динамический. Хотя некоторые процессы впоследствии будут работать медленнее (такие как ввод/вывод данных), таким образом мы по возможности сэкономим память хостовой машины - в данном случае это важно, так как на хостовой машине, с которой выполняется работа, объем свободной памяти небольшой.

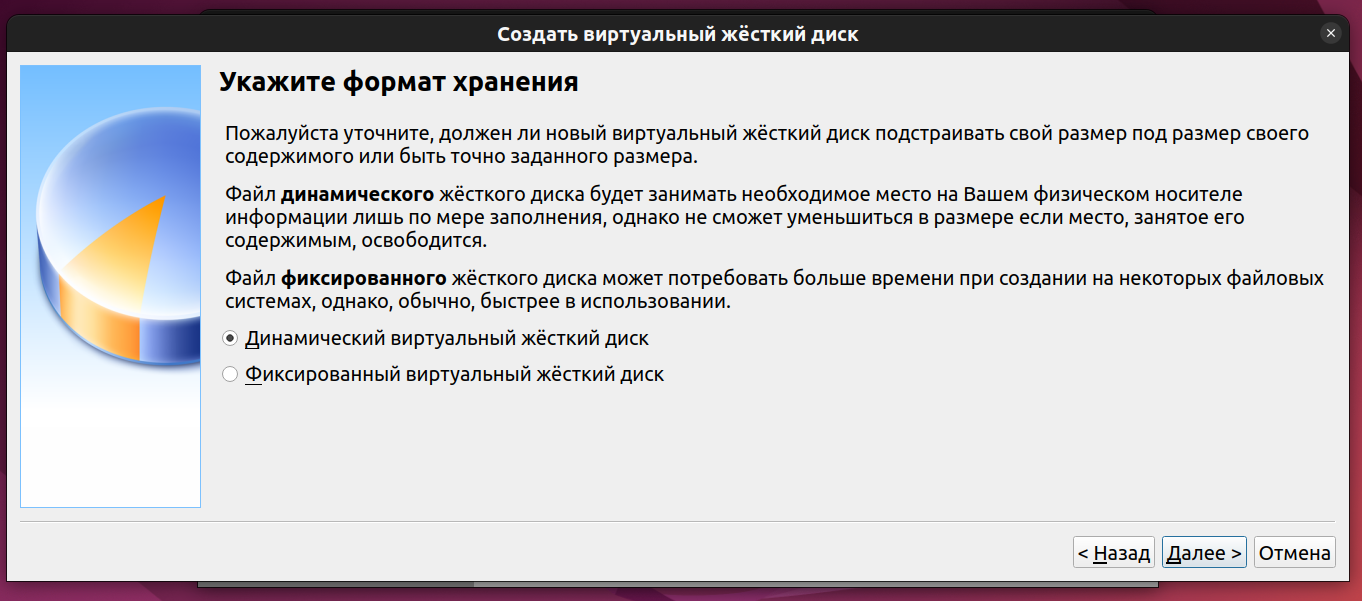


Рис 7. Выбираем динамический формат хранения.

Выделяем 6 гб для виртуального жесткого диска - этого объема памяти должно хватить для выполнения всех поставленных задач в будущем.

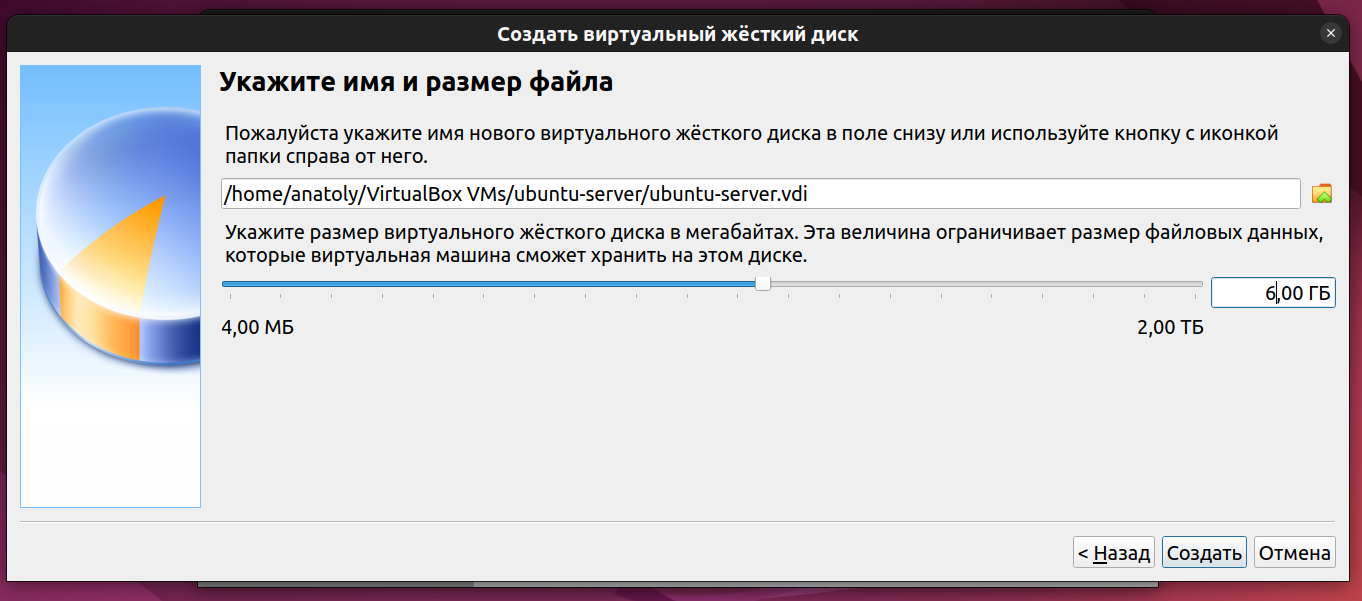


Рис 8. Обозначение максимально допустимого размера хранимых файловых данных.

Виртуальная машина создана и в данный момент выключена и не имеет конкретной операционной системы.

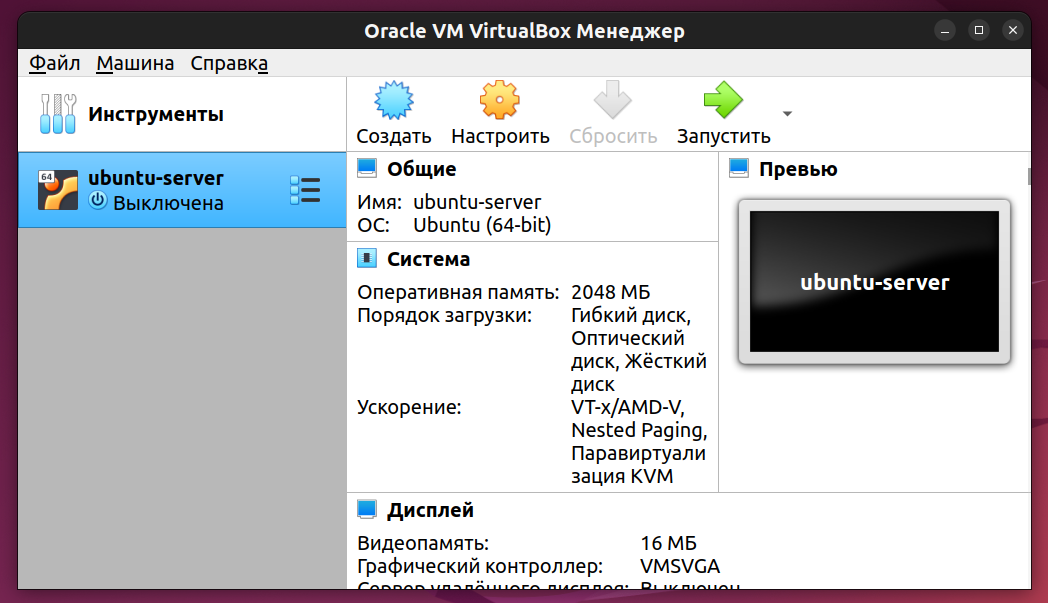


Рис 9. Результат создания виртуальной машины ubuntu-server.

Настроим процессор и видеопамять для более эффективной эмуляции гостевой машины. Выделим созданную машину и выберем пункт “Настроить”.

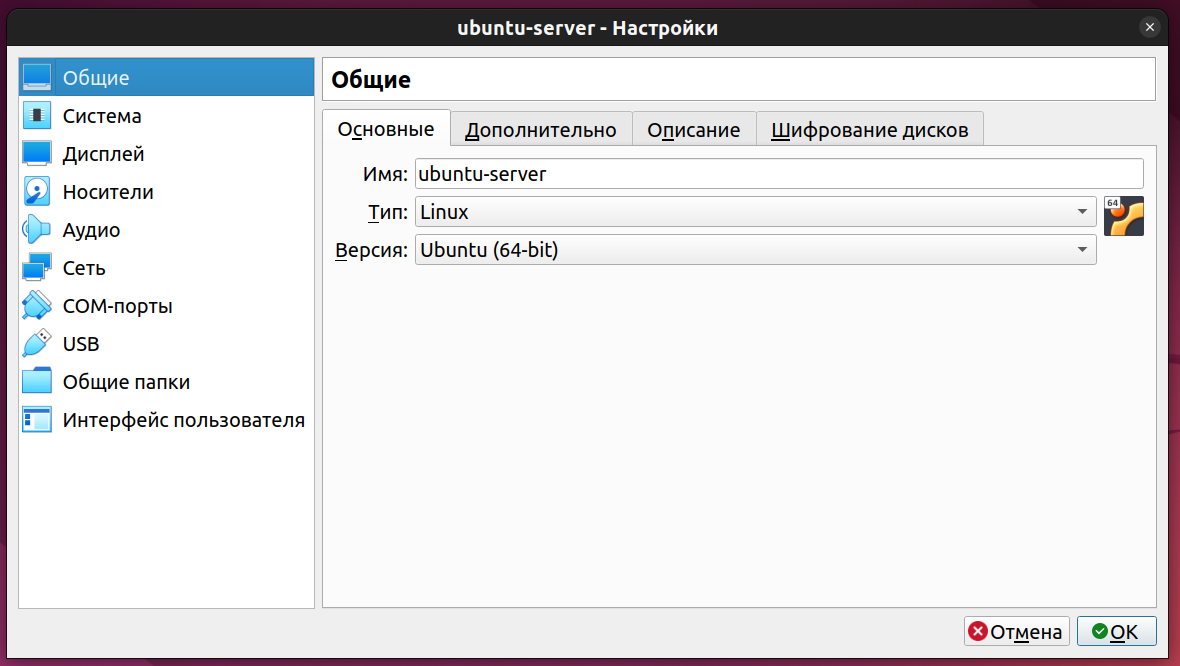


Рис 10. Окно настроек виртуальной машины ubuntu-server.

Перейдем в “Система” -> “Процессор”. Установим рекомендованное количество используемых ядер процессора - 2. Пусть предел загрузки ЦПУ будет 100%.

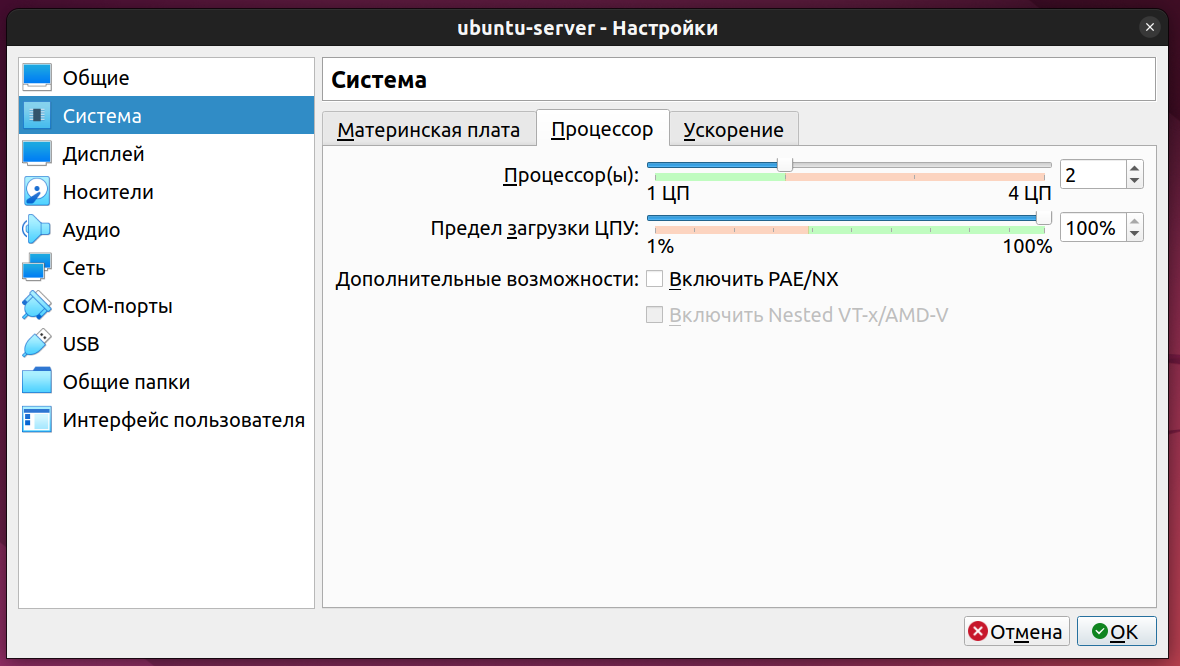


Рис 11. Настройка процессора для ВМ.

Во вкладке “Ускорение” включим пункт “Включить Nested Paging”

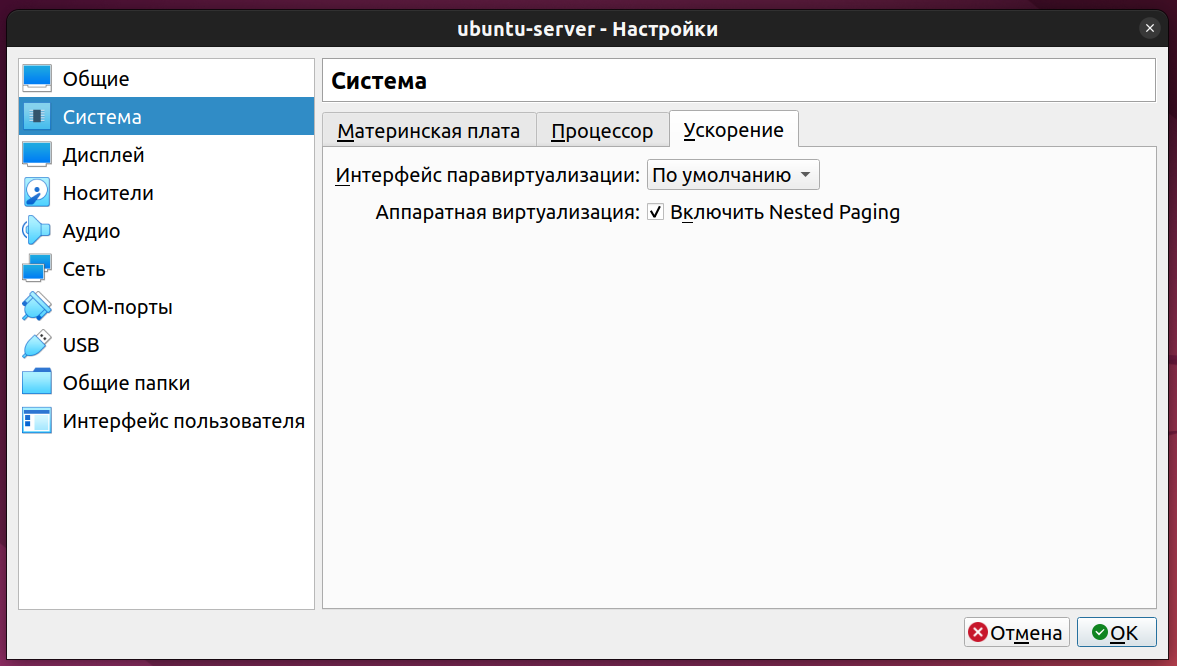


Рис 12. Включаем аппаратную виртуализацию.

Перейдем в “Дисплей” -> “Экран” и выделим 128 мб на видеопамять (вместо 18 мб по умолчанию).

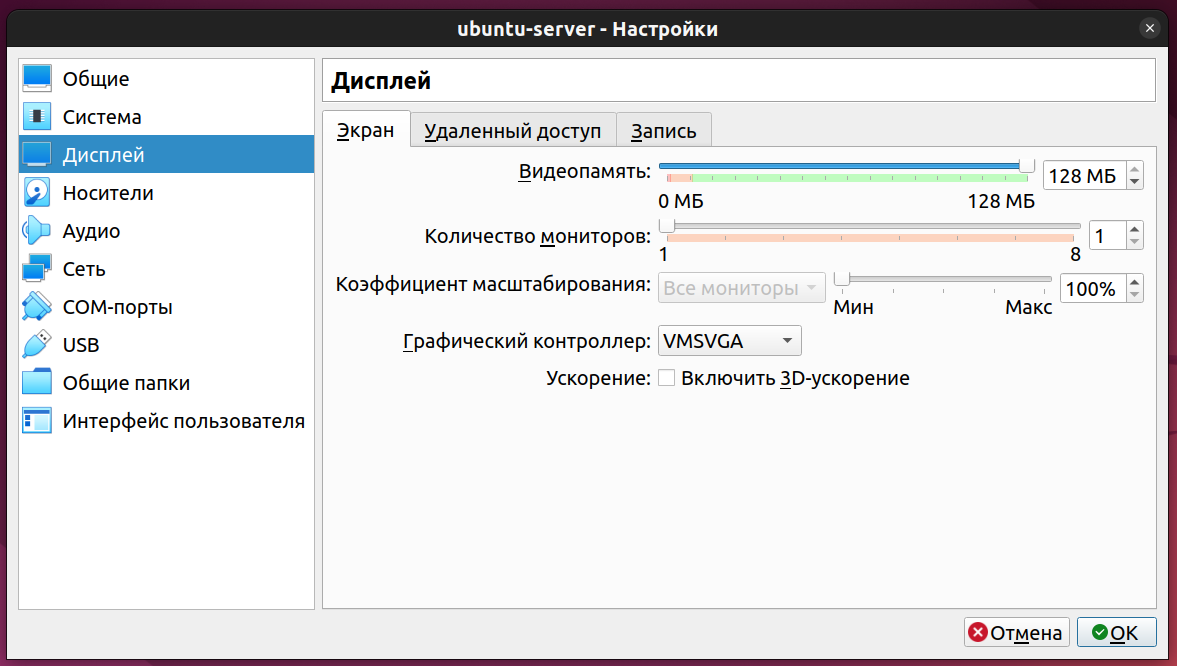


Рис 13. Выделение большего объема видеопамяти для ВМ.

**3. Установка гостевой ОС Linux (Ubuntu Server).**

Перейдем на официальный сайт дистрибутива, в раздел для скачивания Ubuntu Server <https://ubuntu.com/download/server> и скачаем iso-образ дистрибутива Ubuntu Server 22.04.3 LTS

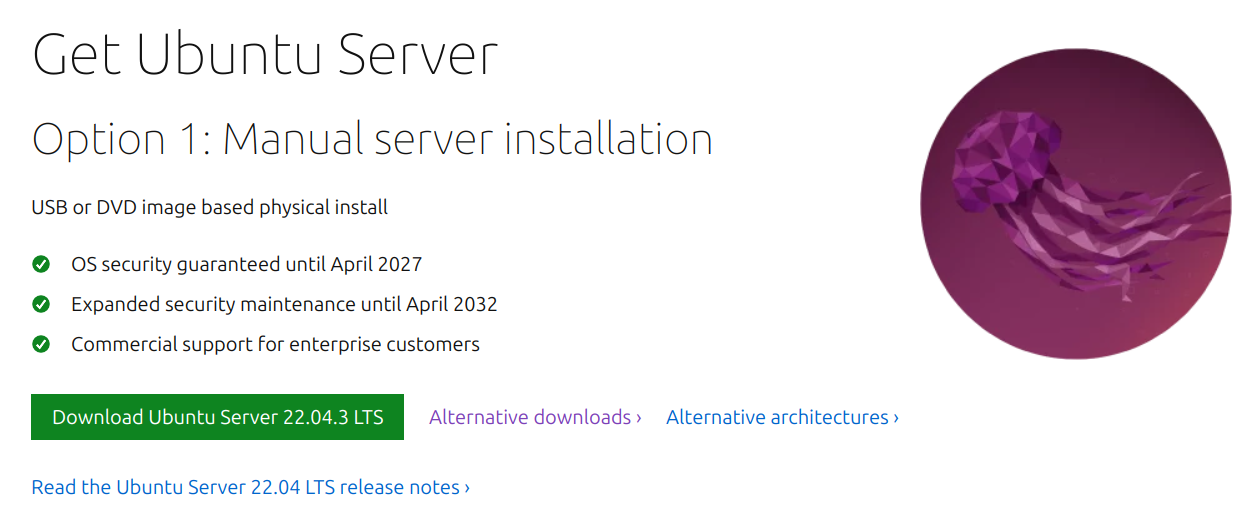


Рис 14. Выбранный дистрибутив.

Запустим созданную ранее виртуальную машину

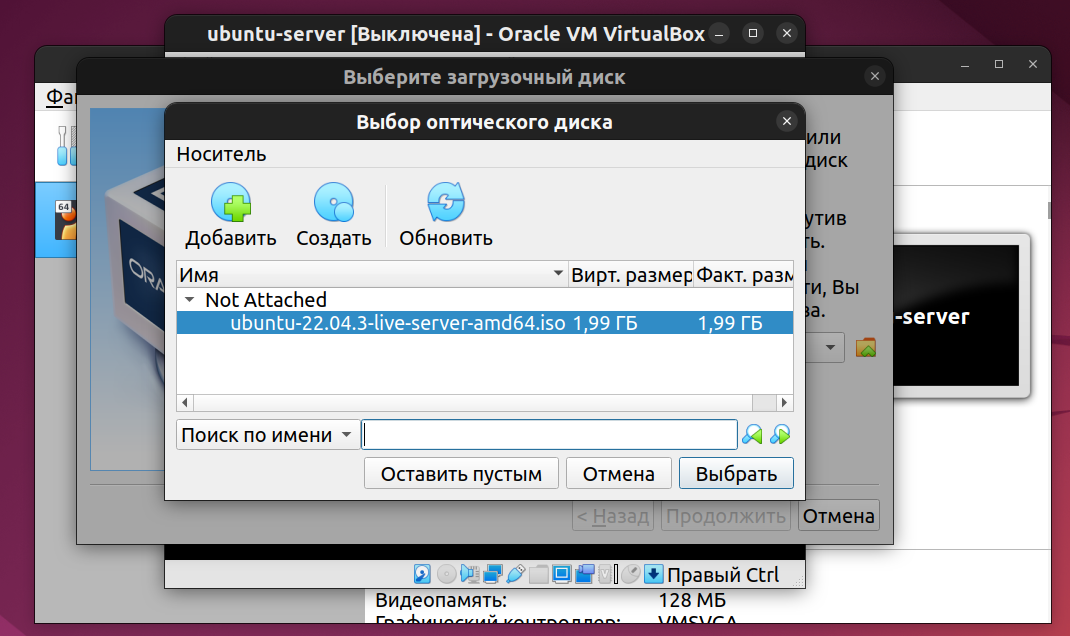


Рис 15. Выбираем ранее установленный iso-образ.

В появившемся Grub-загрузчике выберем Try or install Ubuntu и настроим базовую конфигурацию системы:

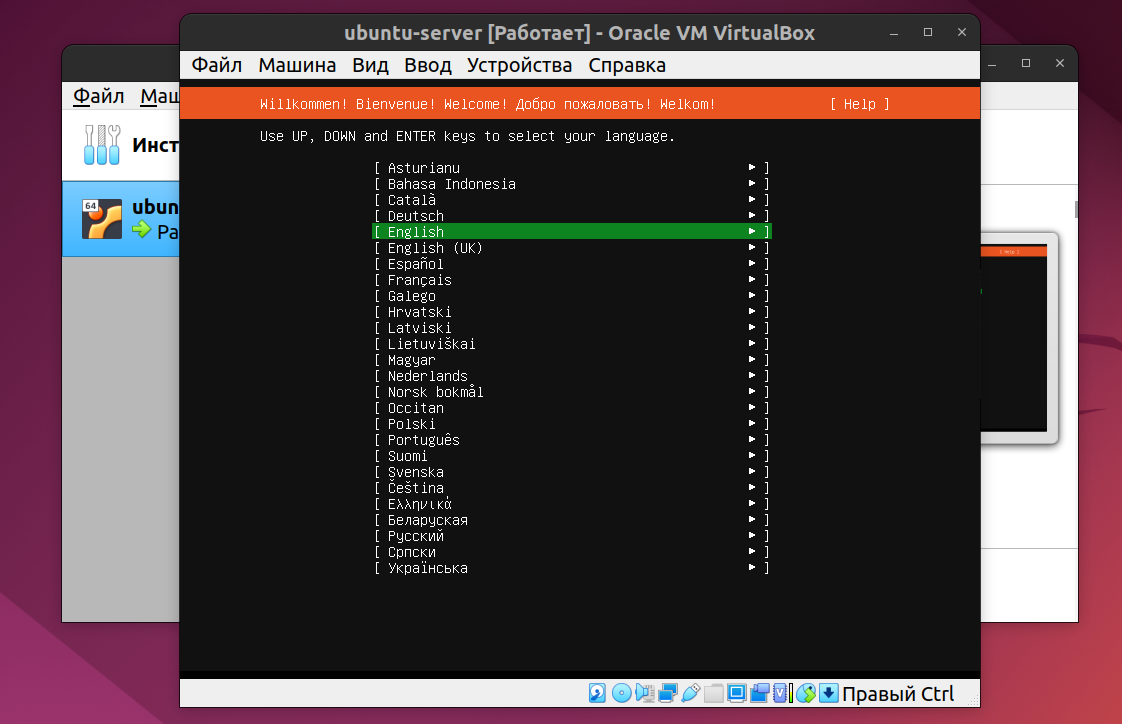


Рис 16. Выбираем английский язык (по умолчанию).

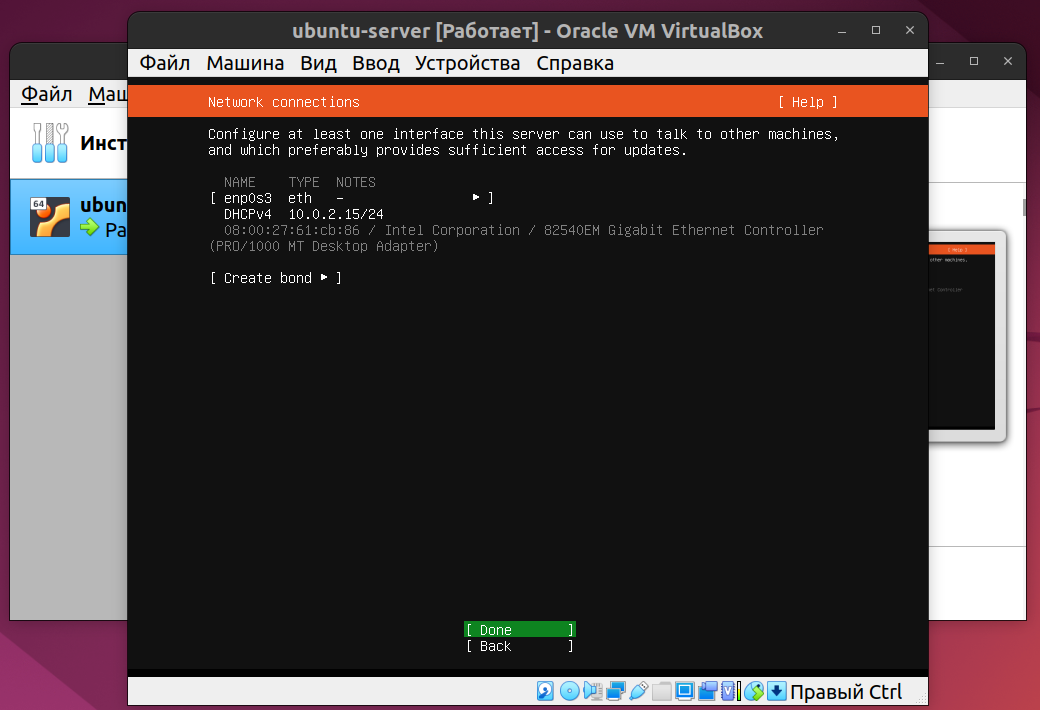


Рис 17. Выбираем сетевое подключение по умолчанию.

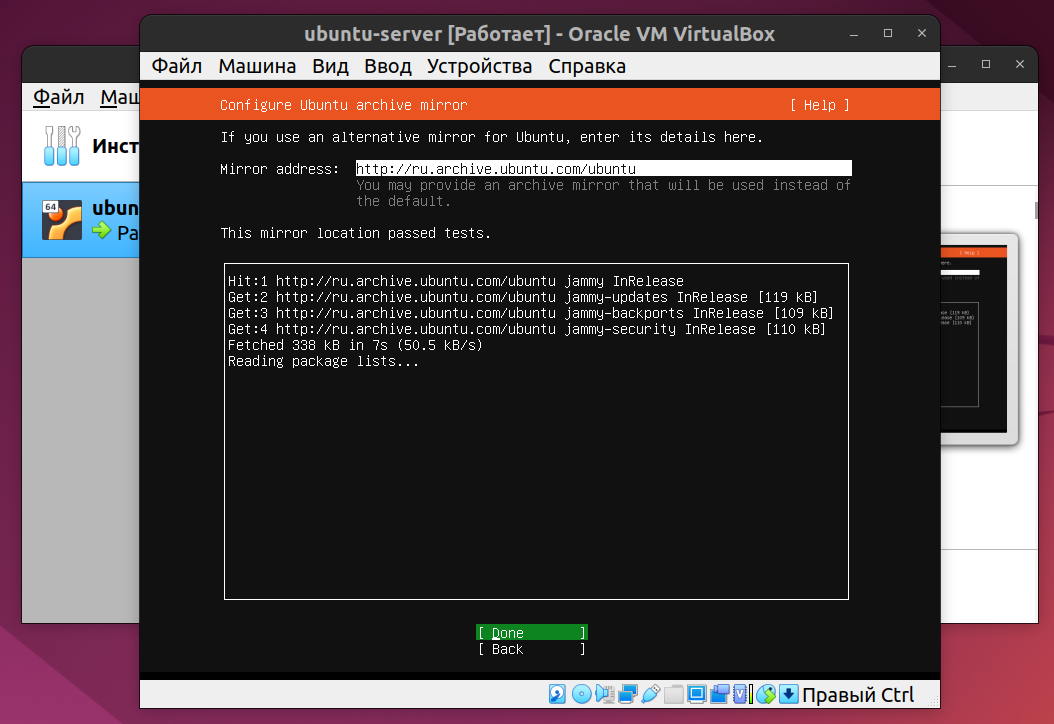


Рис 18. Выбираем с установочными пакетами Ubuntu по умолчанию.

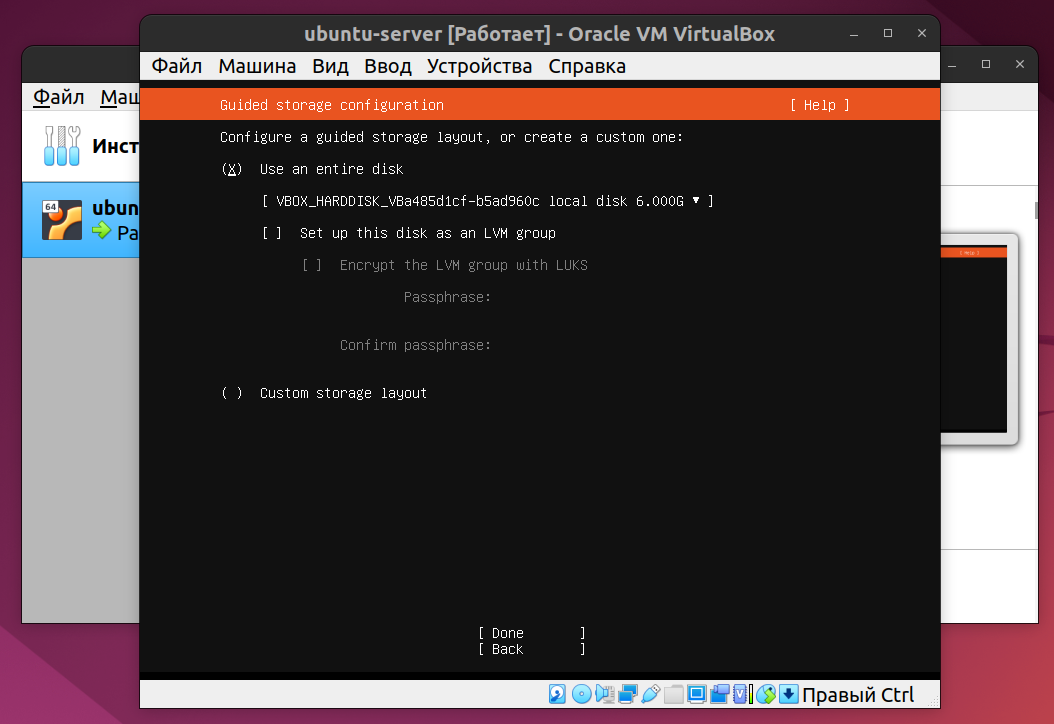


Рис 19. Предоставим установщику весь диск по умолчанию.

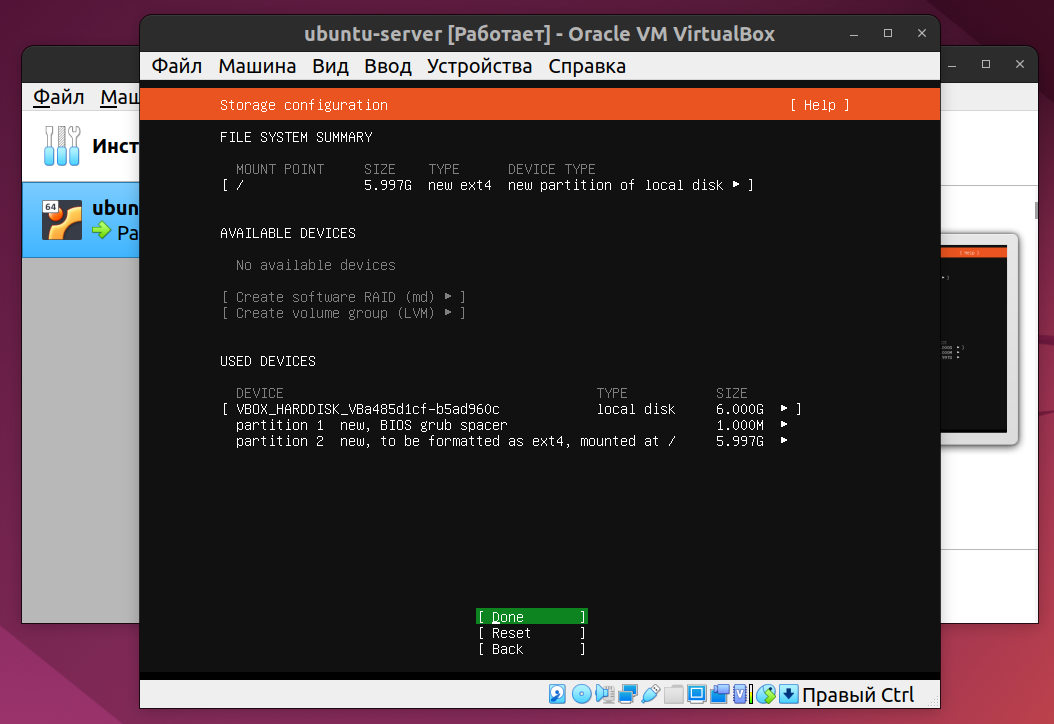


Рис 20. Подтверждаем выбранные настройки.

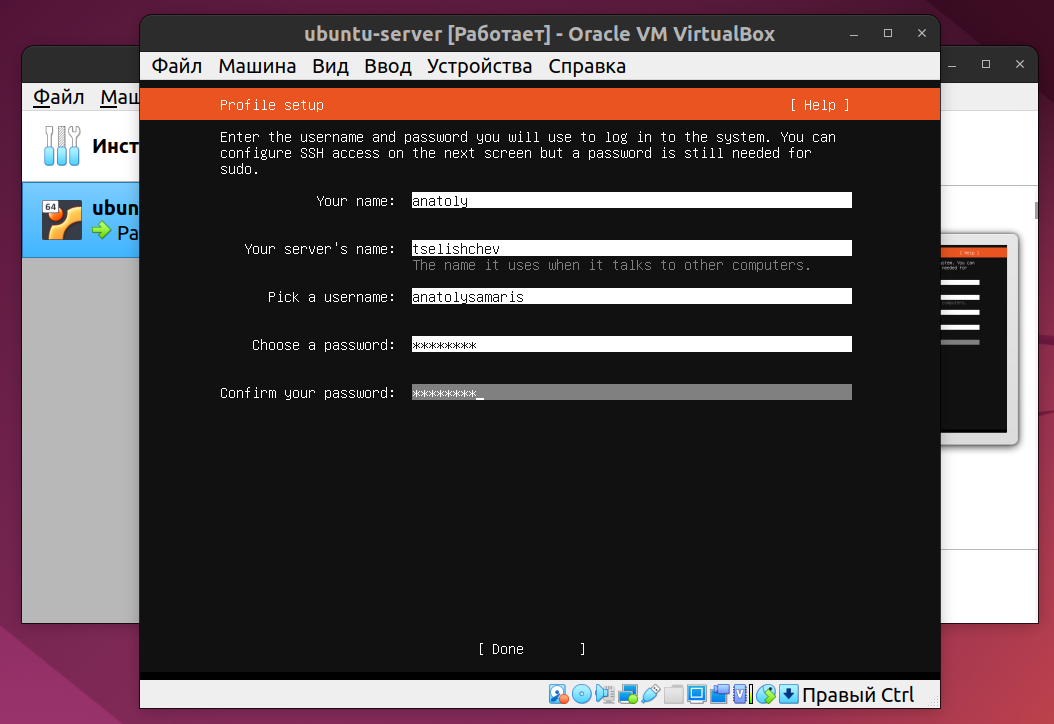


Рис 21. Создаем пользователя.

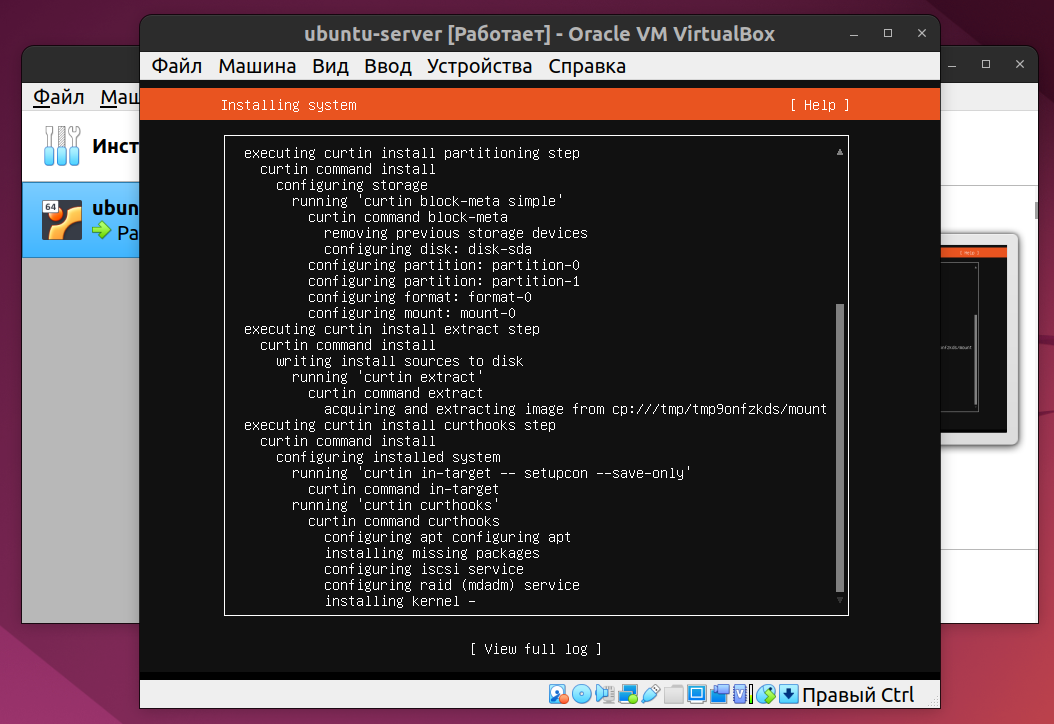


Рис 22. Ждем завершения установки системы.

По завершении установки системы перезагрузим виртуальную машину. После запуска системы введем имя пользователя (в нашем случае anatolysamaris) и пароль пользователя:

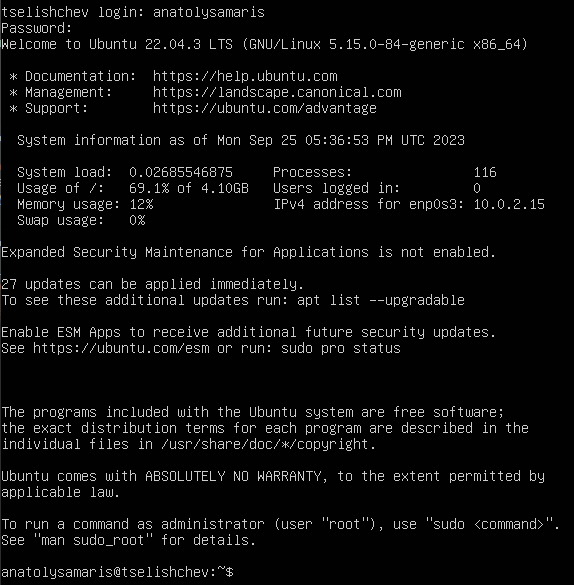


Рис 23. Результат входа в учетную запись.

1. **Разработка на Linux**

Так как при установке мы не выбирали минимальную установку, в системе имеются некоторые утилиты, в том числе Vim. Напишем в терминале

vim code.cpp

Откроется текстовый редактор в командном режиме. Нажмем клавишу i чтобы перейти в режим редактирования (в нижней части редактора появится надпись INSERT). Напишем код, выполняющий задание по варианту, на языке С++:

| Вариант | Задание |
| --- | --- |
| 6 | Дан одномерный массив из N целочисленных элементов (тип int). Задать произвольное число k в интервале 2..N-2. Удалить элемент с этим номером и сдвинуть элементы, находящиеся после него, на эту позицию. |

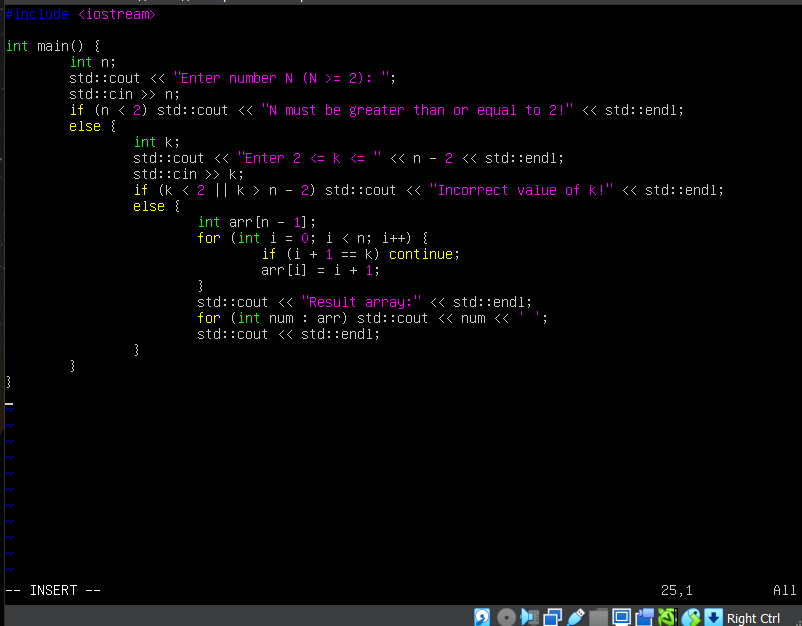


Рис 25. Полный код в файле code.cpp.

1. **Ответы на контрольные вопросы.**

**Что такое IDE?**

IDE это программное обеспечение, объединяющее в едином интерфейсе инструменты разработки и отладки приложений.

**Что такое API?**

Интерфейс программирования приложений: набор готовых процедур, функций, структур, с помощью которых разработчик может взаимодействовать с сервисом в собственных приложениях.

**Что такое библиотека в программировании?**

Это готовый набор функций, классов, типов. Обычно они объединены некоторым общим назначением (например, библиотека для работы с графикой).

**Понятия статической и динамической библиотеки.**

Статическая библиотека - файл с программным кодом, описывающим необходимые функции, классы, который подключается к программе с помощью специальной команды. При запуске программы код библиотеки “вставляется” в программу и становится ее частью.

Динамическая библиотека - файл с кодом, который загружается в программу во время ее выполнения. То есть при компиляции программы динамическая библиотека не становится частью программы, она остается отдельной единицей.