

Лабораторная работа №1

Установка и конфигурация операционной системы на
виртуальную машину

Выполнил:

Куликов Максим Игоревич

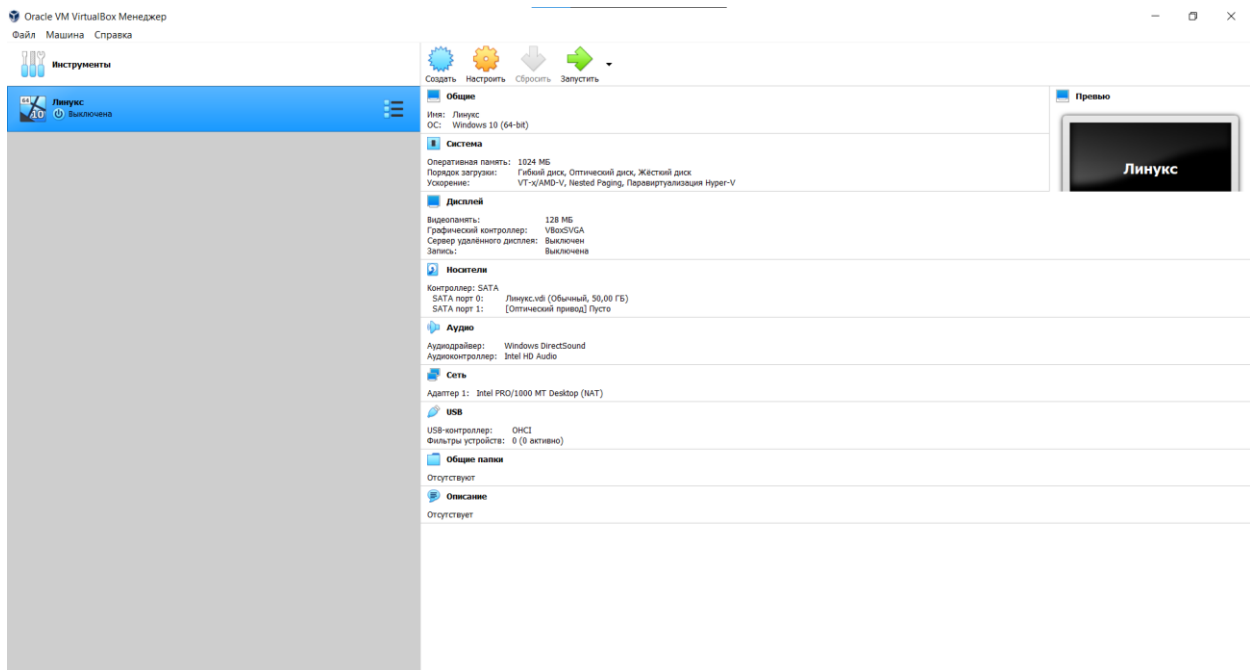
НПМбд-02-20

Москва

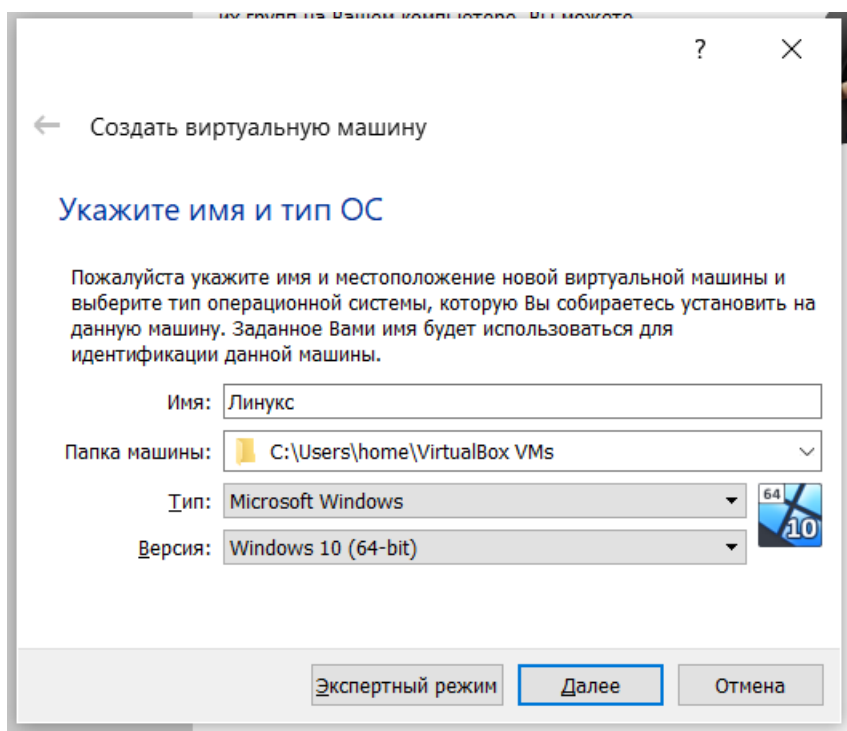
2021 год

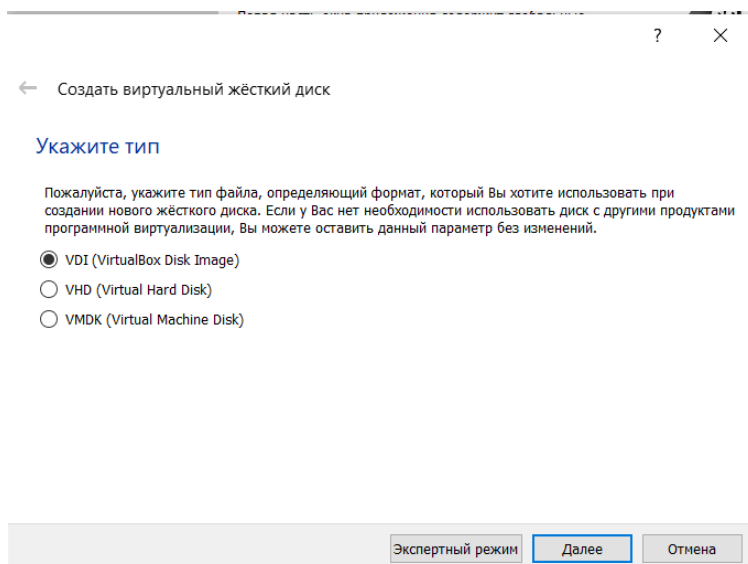
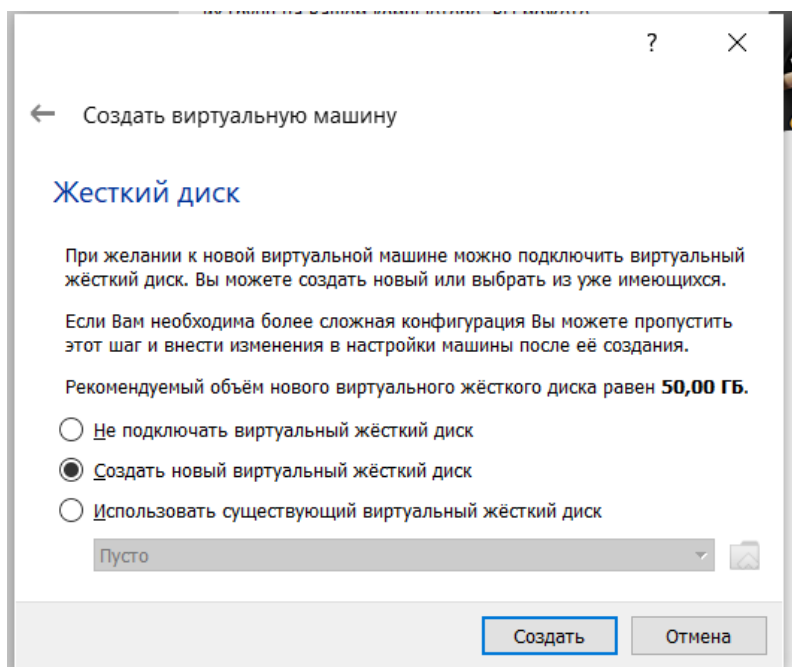
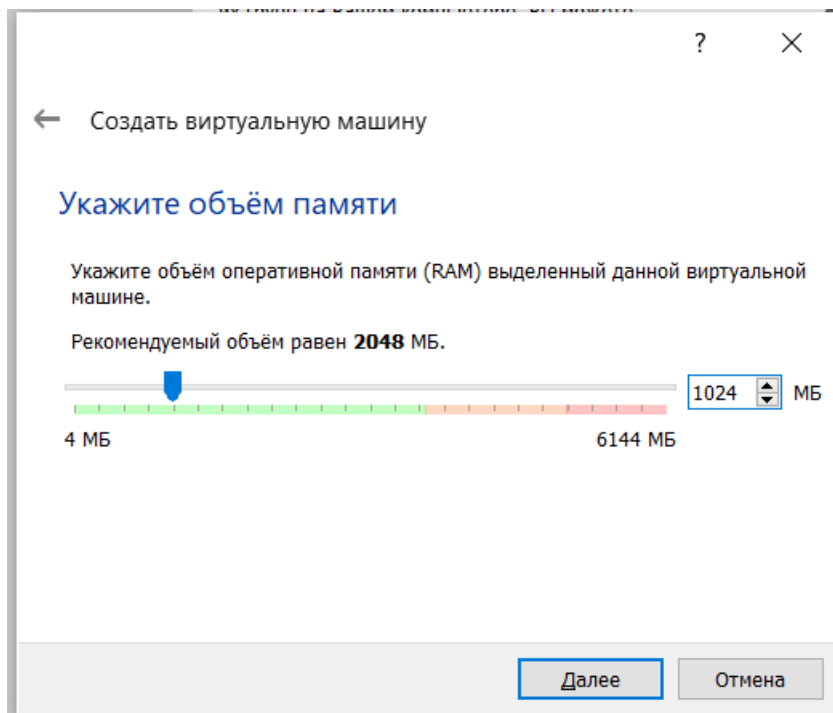
Цель работы: Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов

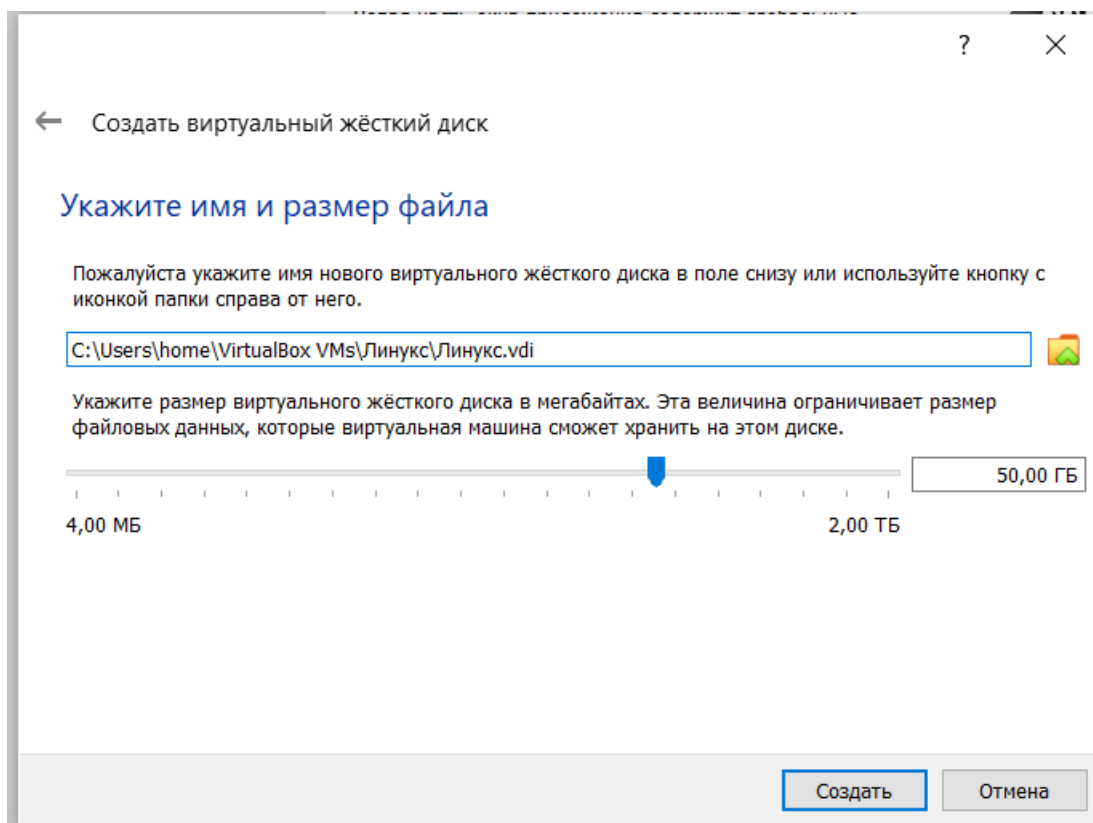
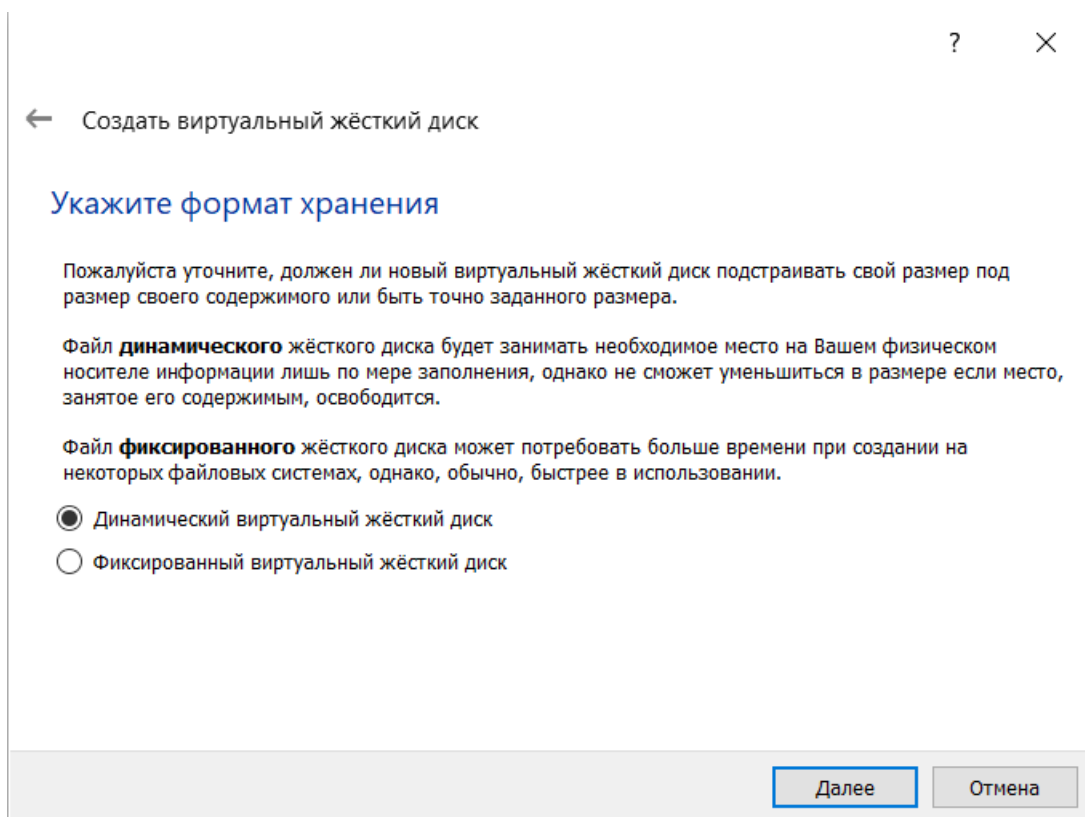
1) Установил VirtualBox на свой персональный компьютер. Начинаю работу с ним



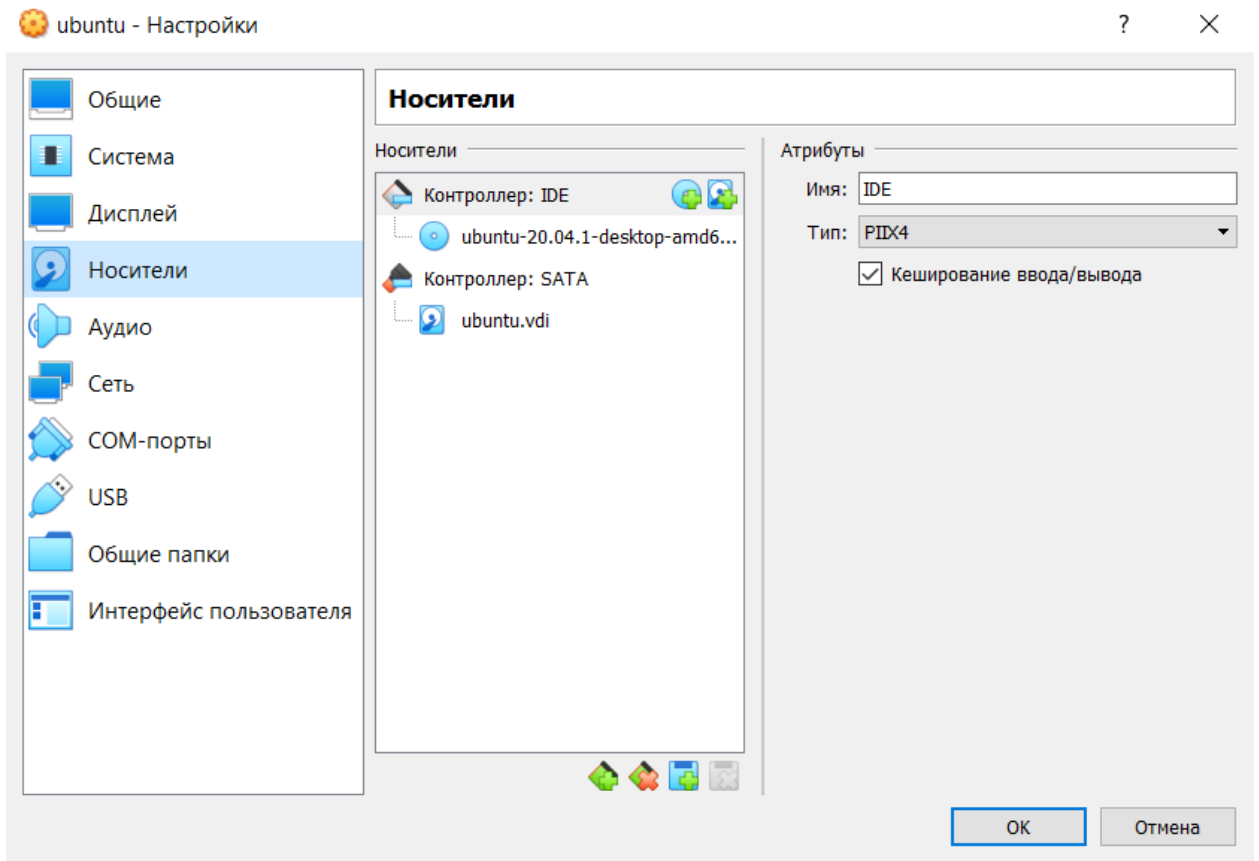
2) Создаю виртуальную машины и вношу нужные данные: название машины; выделяю оперативную память; создаю виртуальный жёсткий диск; выбираю тип диска; выбираю динамический формат хранения; выделяю память для жёсткого диска.



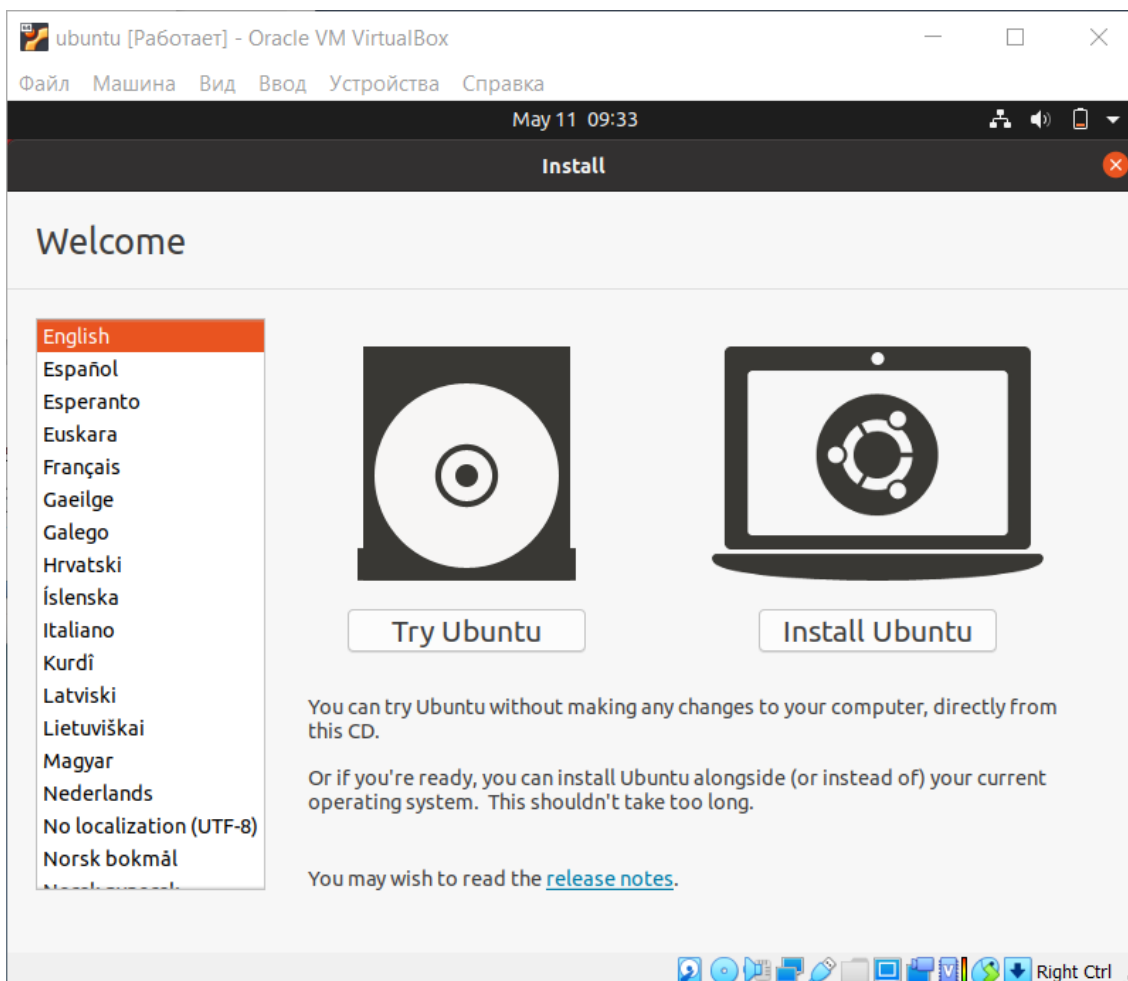




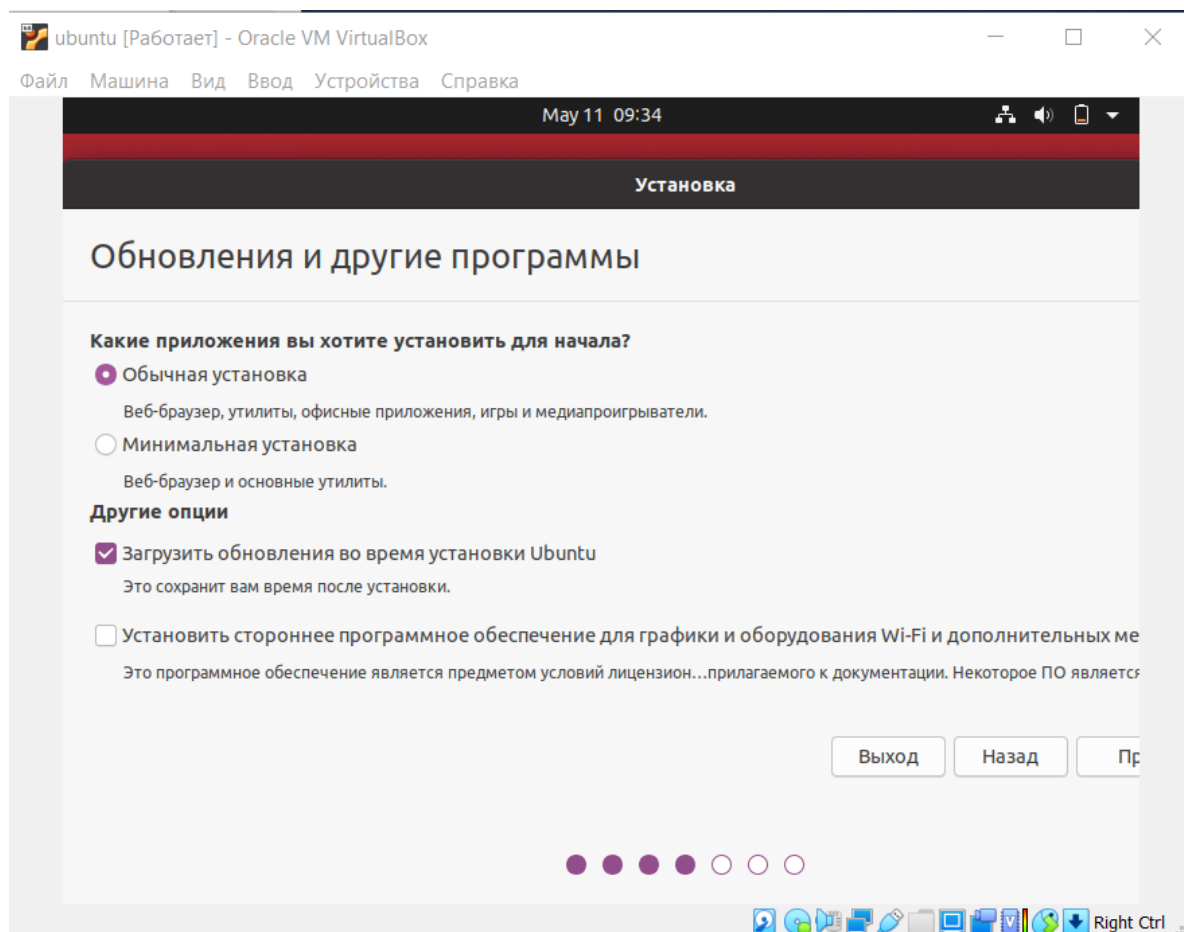
3) Настраиваю носители



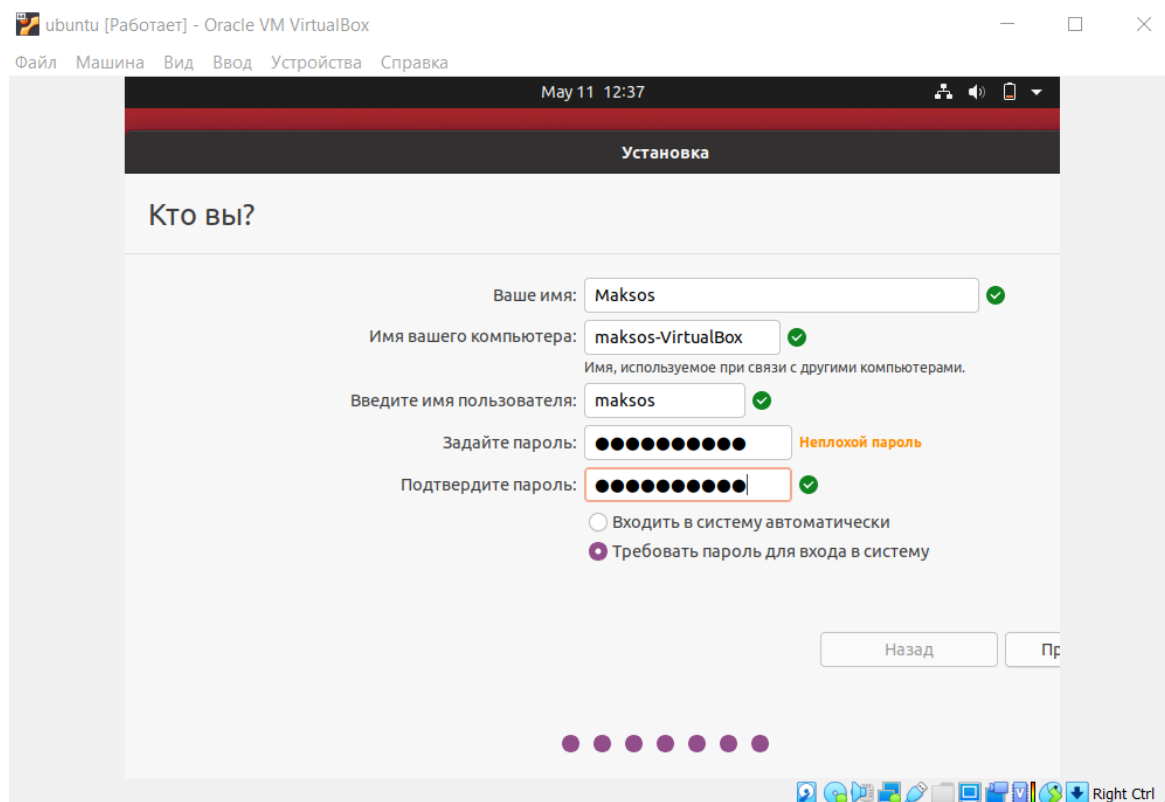
4) Начинаю настройку системы



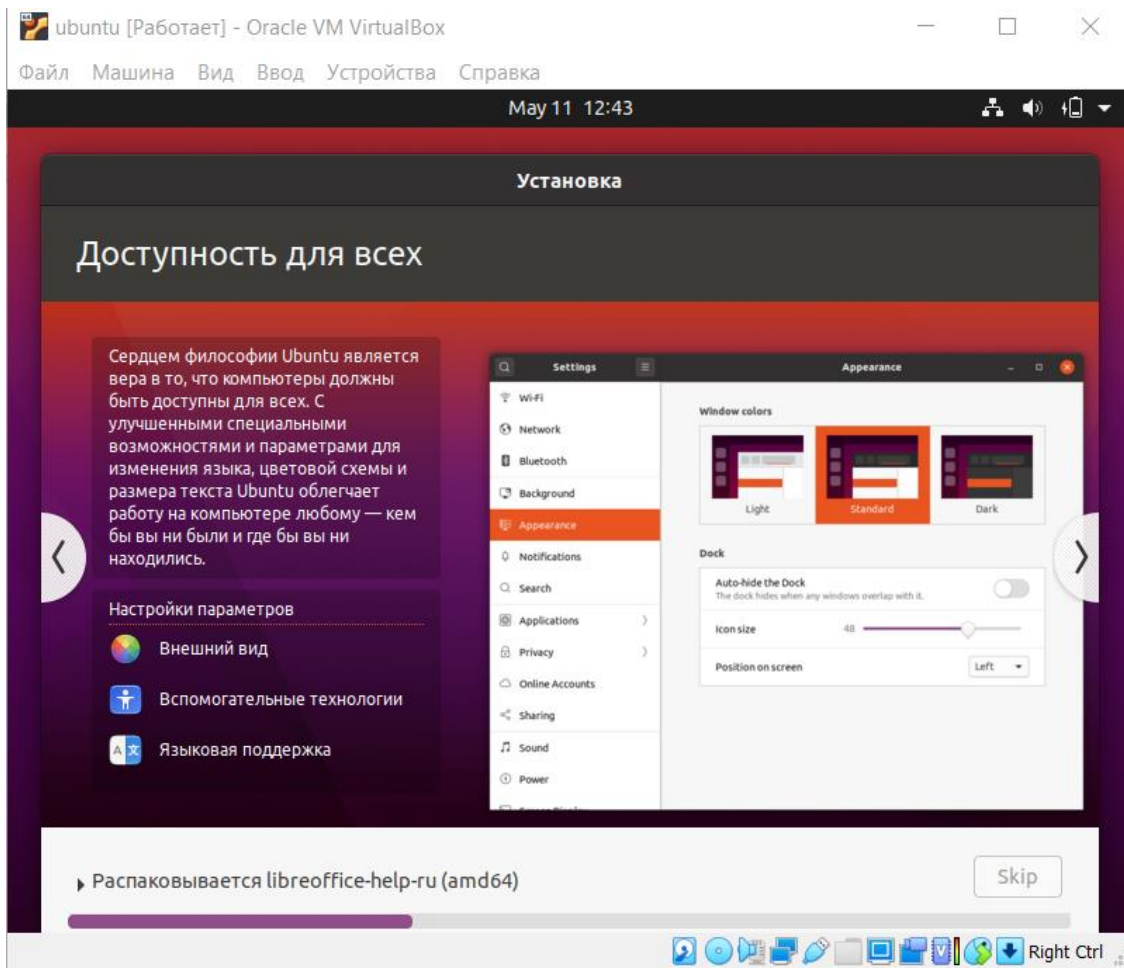
5) Продолжаю установку



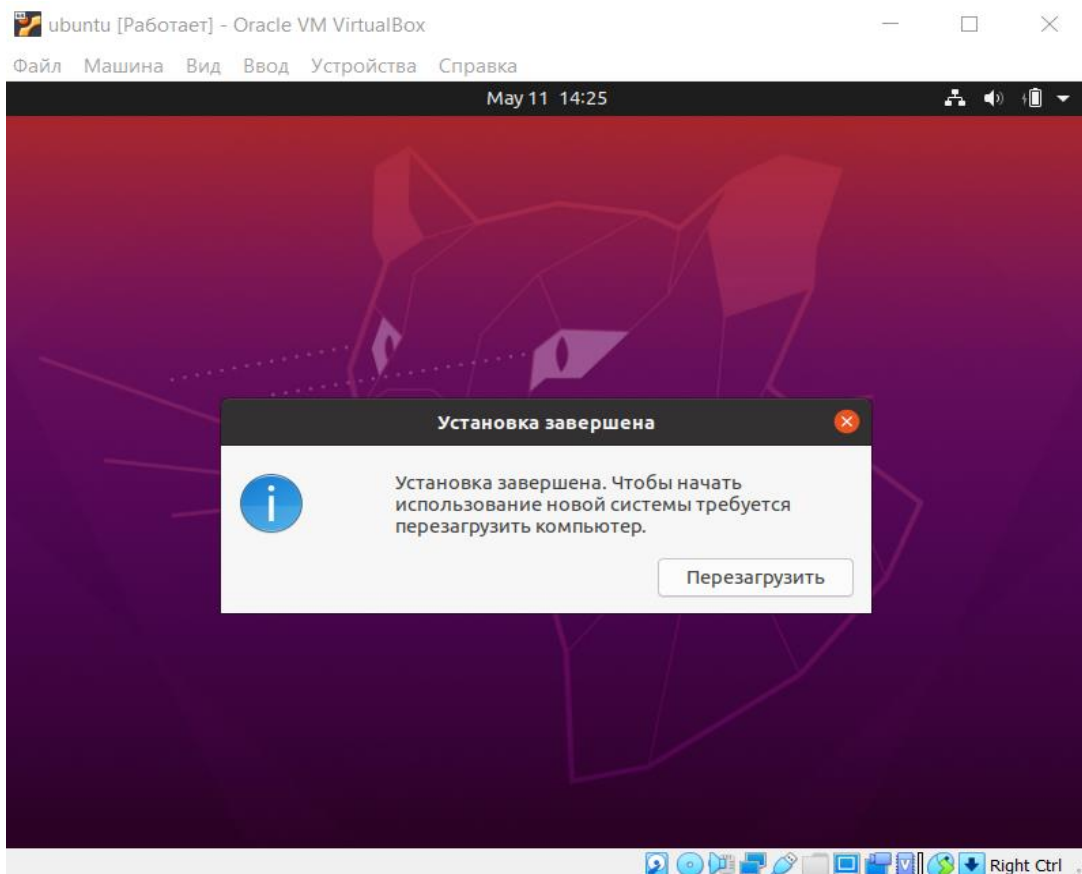
6) Создаю пользователь



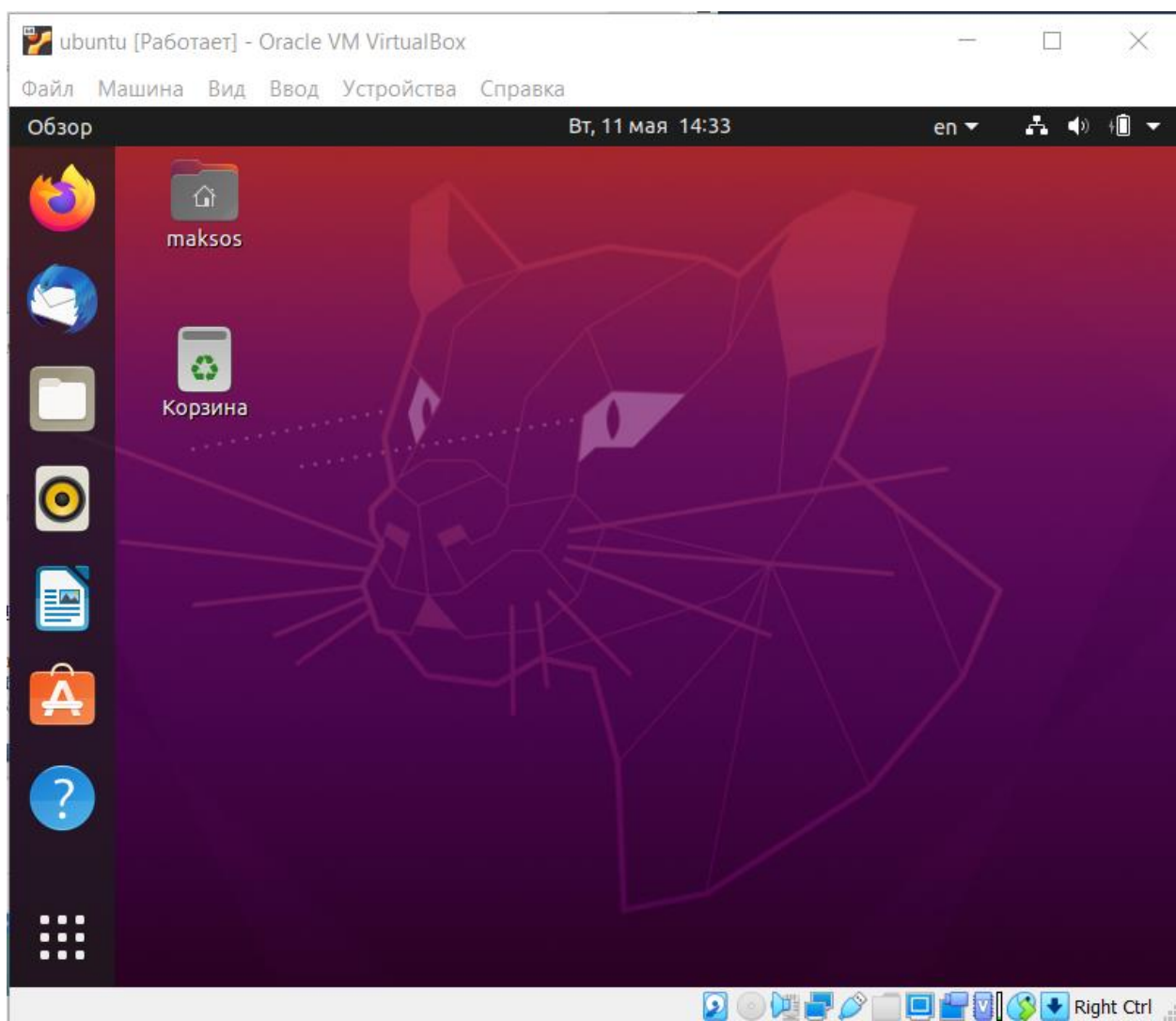
7) Выполняется установка системой



8) Установка выполнена. Перезагружаю систему



9) Система готова к работе



Контрольные вопросы

1. Когда пользователь регистрируется в системе (проходит процедуру авторизации, например, вводя системное имя и пароль), он идентифицируется с учётной записью, в которой система хранит информацию о каждом пользователе: его системное имя и некоторые другие сведения, необходимые для работы с ним. Именно с учётными записями, а не с самими пользователями, и работает система. Таким образом, учетная запись пользователя содержит:
 - 1) Системное имя (user name)
 - 2) Идентификатор пользователя (UID)
 - 3) Идентификатор группы (GID)
 - 4) Полное имя (full name)
 - 5) Домашний каталог (home directory)
 - 6) Начальная оболочка (login shell)
2. Команды терминала:
 - 1) **man** [аргумент] – для получения справки по команде. **Пример:** *man ls* – выведет информацию о команде *ls*;
 - 2) **cd** [путь] – для перемещения по файловой системе. **Пример:** *cd /* – для перехода в корневой раздел;
 - 3) **ls** [опции] – для просмотра содержимого каталога. **Пример:** *ls -alS /* – выведет подробной информации о файлах в корневом каталоге с сортировкой по алфавиту;
 - 4) **du** [опции] [путь] – для определения объёма каталога. **Пример:** *du -h ~/\"Изображения\"* – выведет размер каталога «Изображения» с указанием единицы измерения;

5) **mkdir** [опции] [путь] / **rmdir** [опции] [путь] / **rm** [опции] [путь] – для создания / удаления каталогов / файлов. **Примеры:** *mkdir -pv ~/MyDir ~/MyDir2* – создаст каталоги MyDir и MyDir2 в домашней директории; *rmdir -v ~/MyDir* – удалит каталог MyDir из домашней директории; *rm -rv ~/MyDir2* – удалит каталог MyDir2 из домашней директории;

6) **chmod** [аргументы] [путь] – для задания определённых прав на файл / каталог. **Пример:** *chmod o-w ~/Readme.txt* – отнимет право изменения текстового файла Readme.txt остальным пользователям, кроме владельца и группы владельца.

7) **history** [опции] – для просмотра истории команд. **Пример:** *history -c* – очистит историю команд Linux.

3. **Файловая система** – порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании.

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы. Каждый дистрибутив Linux позволяет использовать одну из этих файловых систем, каждая из них имеет свои преимущества и недостатки:

- Ext2;
- Ext3;
- Ext4;
- JFS;
- ReiserFS;
- XFS;
- Btrfs;
- ZFS;

Все они включены в ядро и могут использоваться в качестве корневой файловой системы.

Ext2, Ext3, Ext4 или Extended Filesystem – это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих. Её кодовая база изменяется очень редко. Кроме того, данная файловая система содержит больше всего функций. Версия **ext2** была разработана уже именно для Linux и получила много улучшений.

JFS или Journaled File System была разработана в IBM для AIX UNIX и

использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоя.

XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.

- Команда *findmnt* используется для поиска примонтированных файловых систем. Она используется для поиска монтированных устройств, а также может монтировать или размонтировать их при необходимости.

Для просмотра всех примонтированных файловых систем использовать команду:
findmnt --all

- Каждый процесс в Linux имеет свой идентификатор, называемый PID. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого воспользуемся командами *ps* и *grep*. Команда *ps* предназначена для вывода списка активных процессов в системе и информации о них. Команда *grep* запускается одновременно с *ps* (в канале) и будет выполнять поиск по результатам команды *ps*.

Есть еще один более простой способ узнать PID процесса — это команда *pidof*, которая принимает в качестве параметра название процесса и выводит его PID.

Когда известен PID процесса, мы можем убить его командой *kill*. Команда *kill* принимает в качестве параметра PID процесса.

Вообще команда *kill* предназначена для отправки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал SIGTERM (от слова termination — завершение). SIGTERM указывает процессу на то, что необходимо завершиться. Каждый сигнал имеет свой номер. SIGTERM имеет номер 15. Список всех сигналов (и их номеров), которые может послать команда *kill*, можно вывести,

выполнив *kill -l*.

Сигнал SIGTERM может и не остановить процесс (например, при перехвате или блокировке сигнала), SIGKILL же выполняет уничтожение процесса всегда, так как его нельзя перехватить или проигнорировать.

Команда *killall* в Linux предназначена для «убийства» всех процессов, имеющих одно и то же имя. Это удобно, так как нам не нужно знать PID процесса.

Вывод: Я приобрел практические навыки установки операционной системы на виртуальную машину и настроил минимально необходимые для дальнейшей работы сервисы. А также узнал необходимую мне информацию о моей виртуальной системе через терминал.

Вывод: Приобрел практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов