МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Установка ядра операционной системы

Дисциплина: «Операционные системы»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

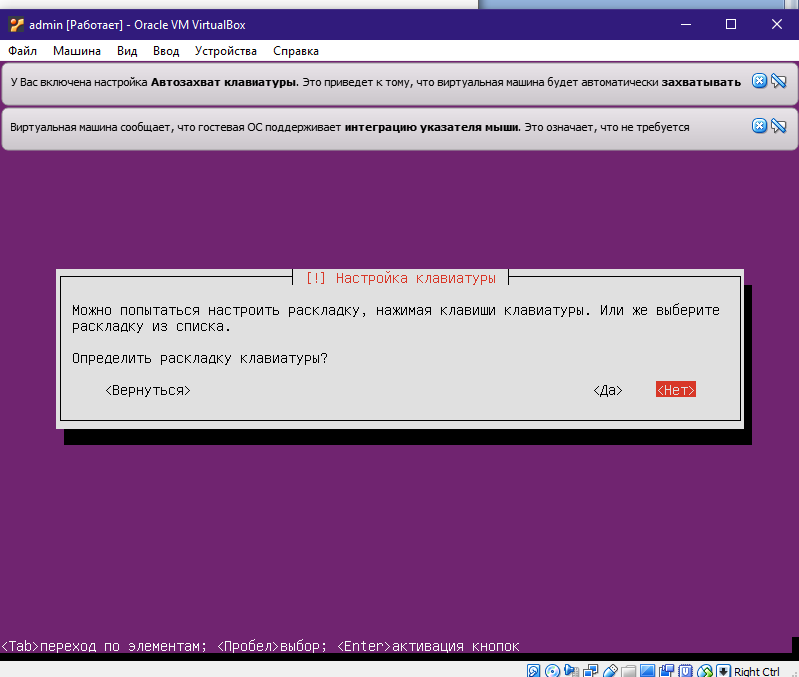
|  |  |
| --- | --- |
| Руководитель | Дианов С.В. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-31 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

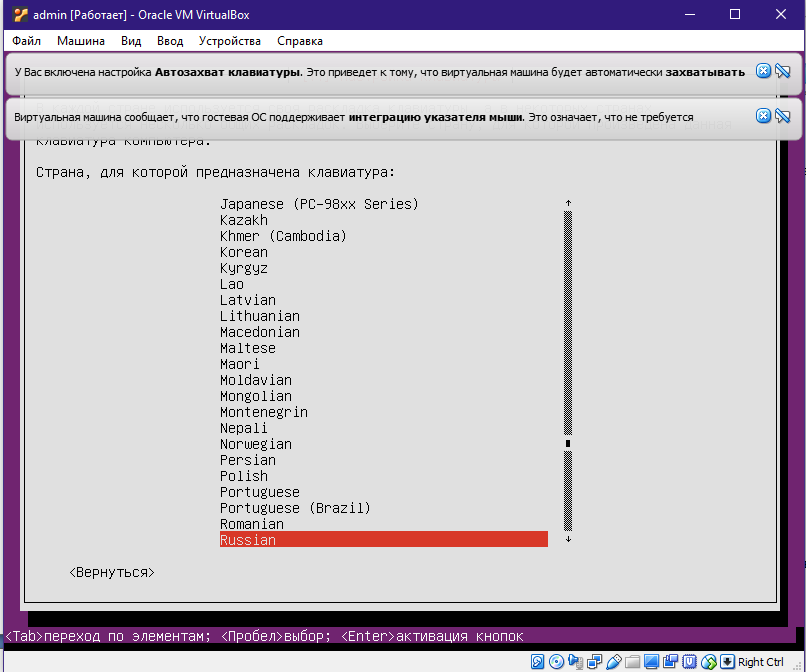
2021 г.

**Цель работы:** получить навыки установки ядра операционной системы Linux.

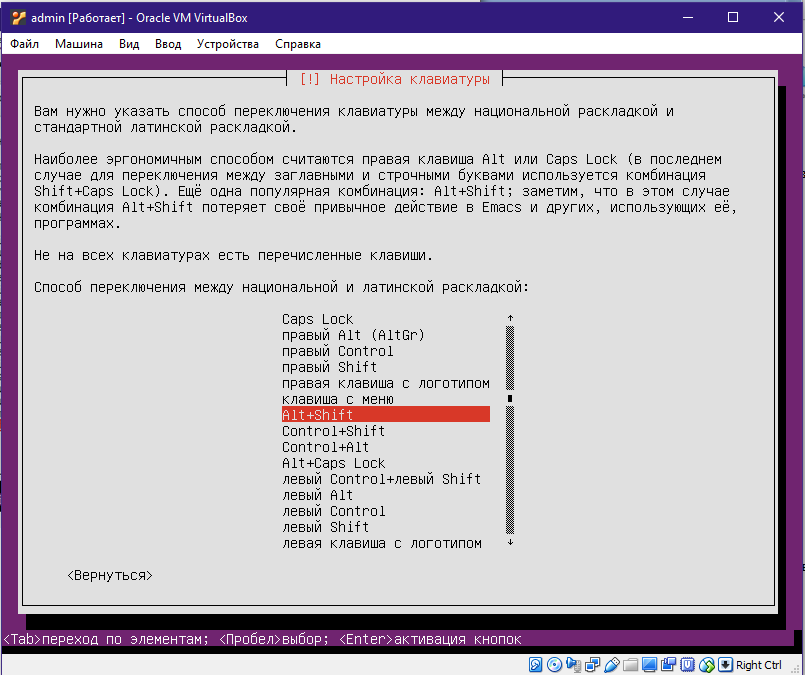
1. Выбираем что нужно определить раскладку клавиатуры



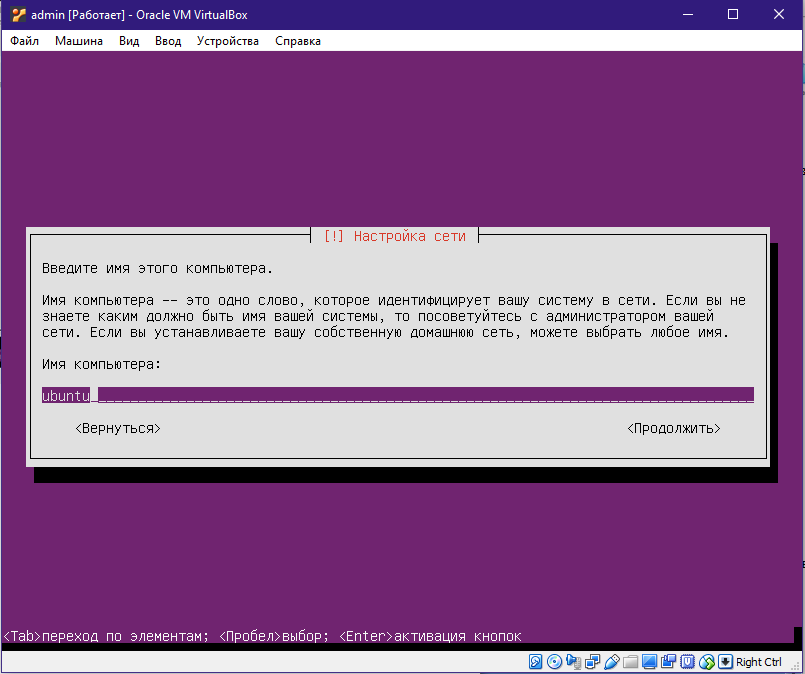
2. Выбираем язык раскладки



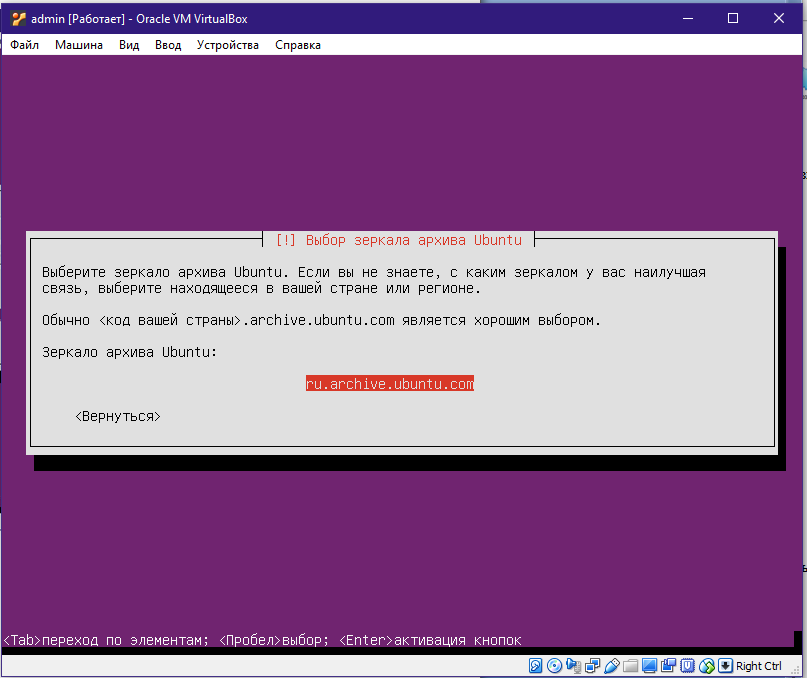
3. Выбираем горячую клавишу для переключения раскладки



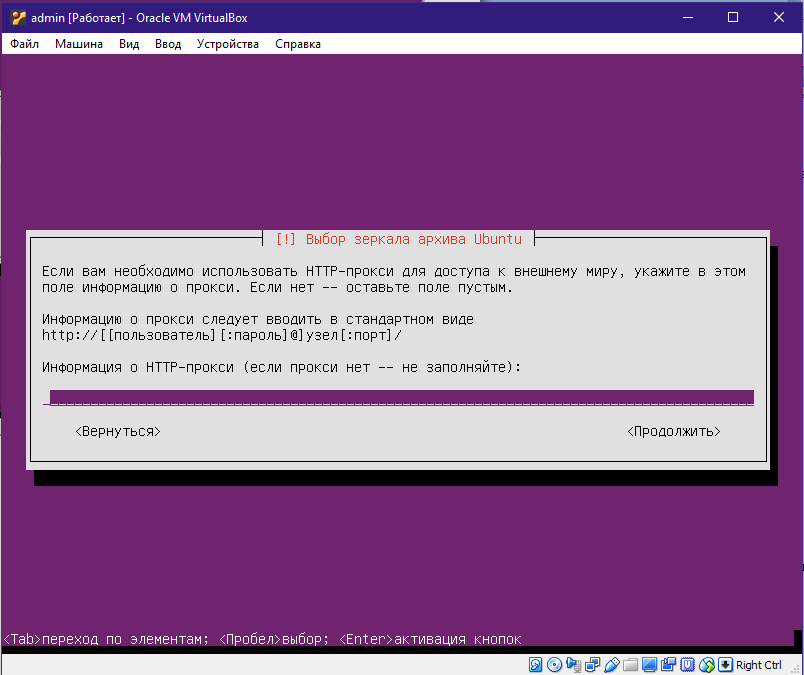
4. Выбираем название пк



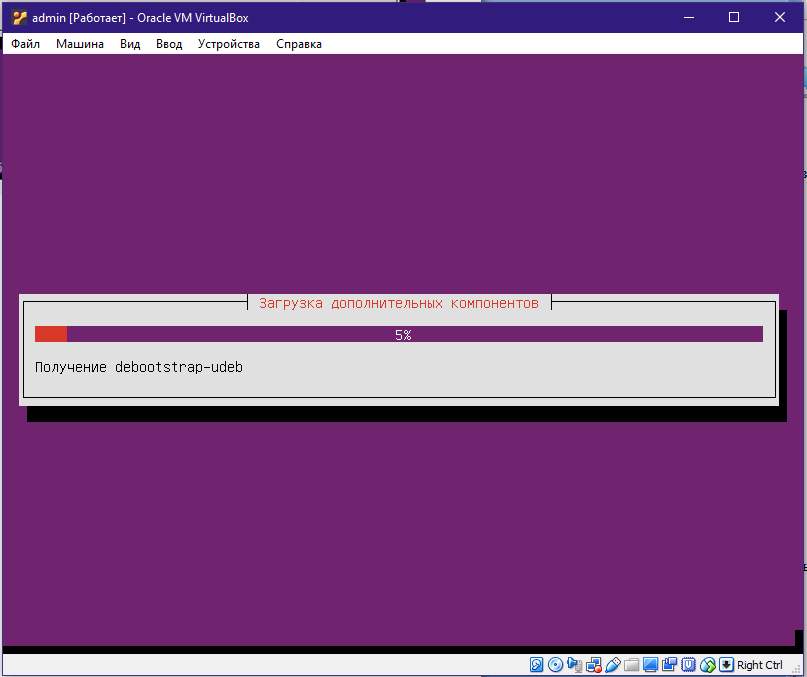
5. Выбираем зеркало для установки



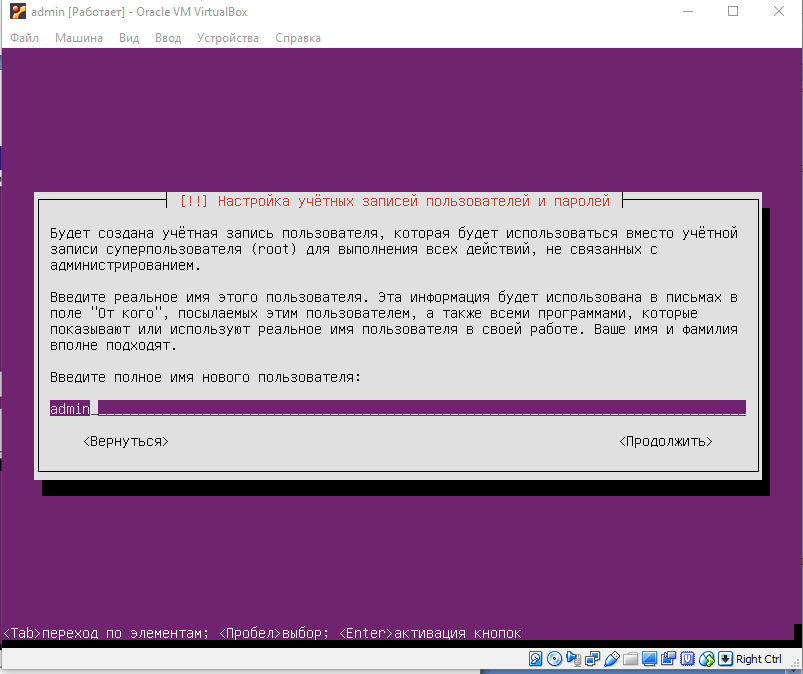
6. Указываем, что не нужно использовать прокси



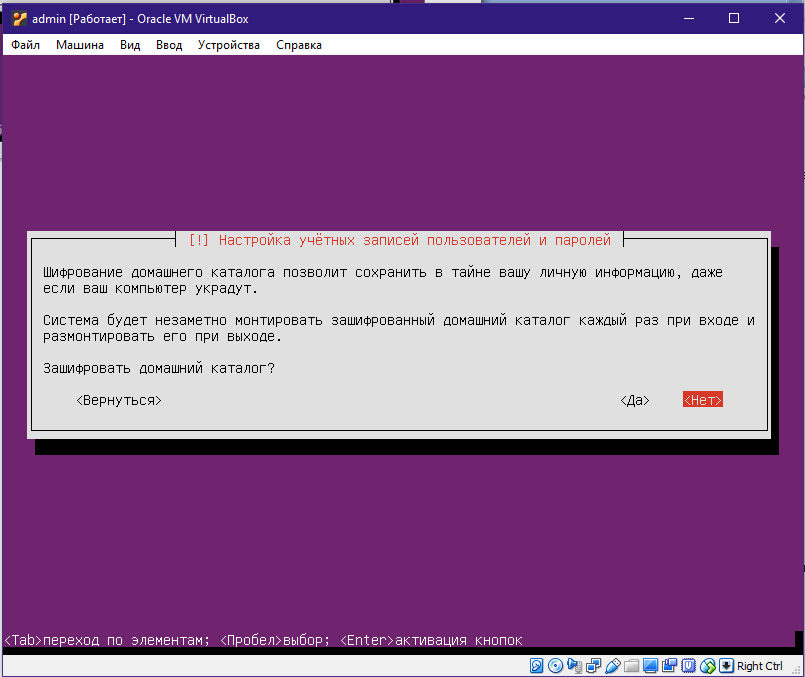
7. Установка компонентов



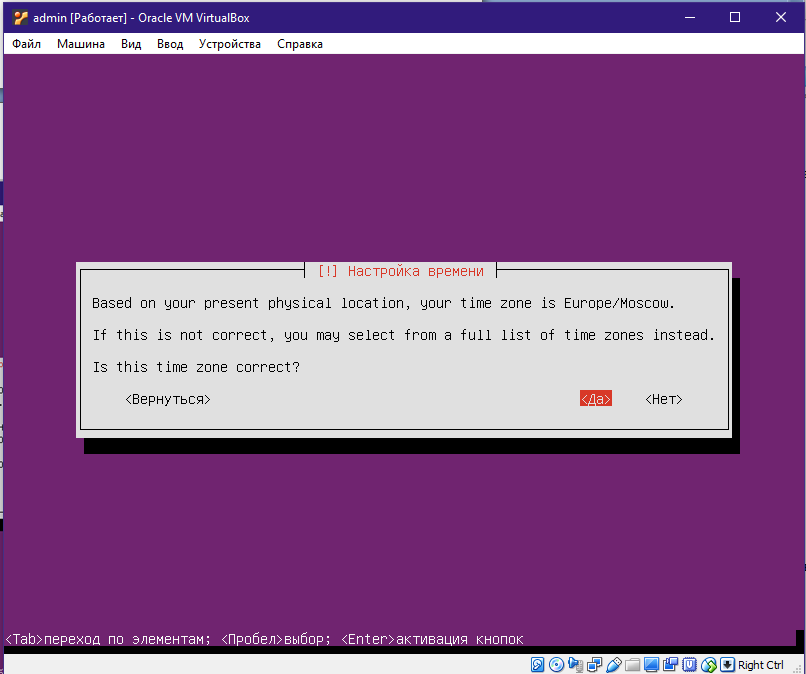
8. Создание root пользователя



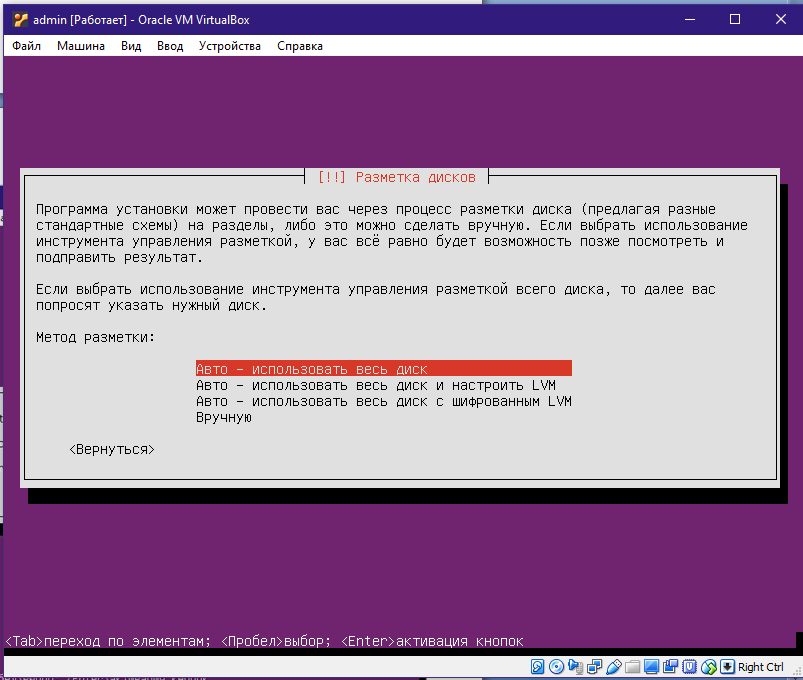
9. Отказываемся от шифрования домашнего каталога



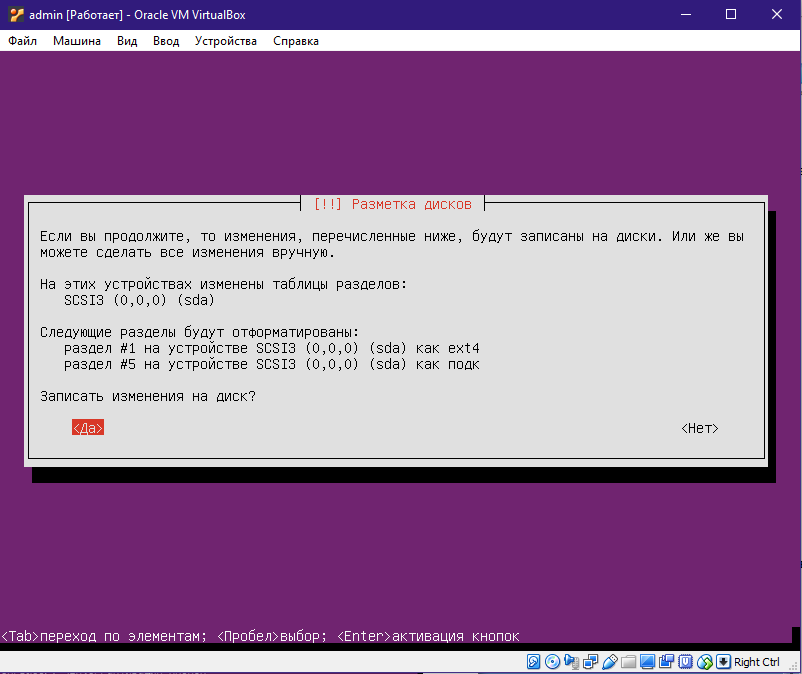
10. Соглашаемся с автоматически определенным регионом



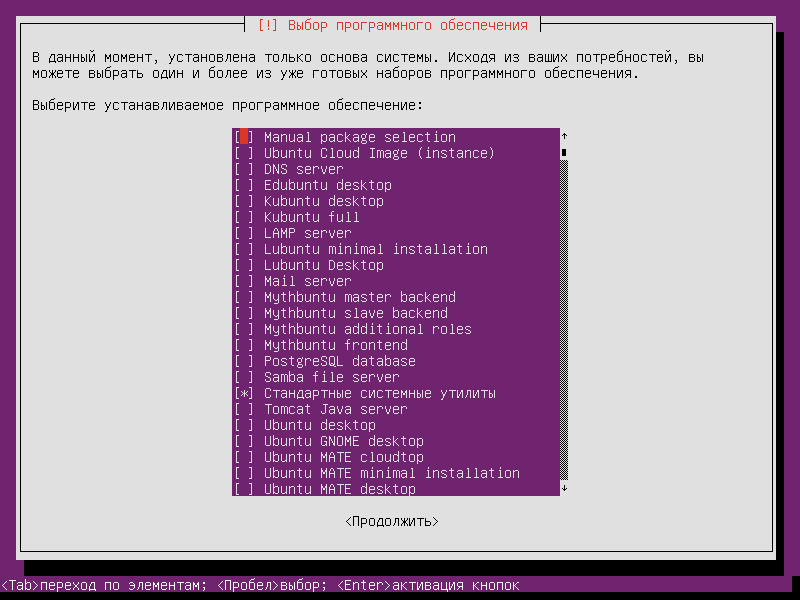
11. Используем автоматическую разметку дисков



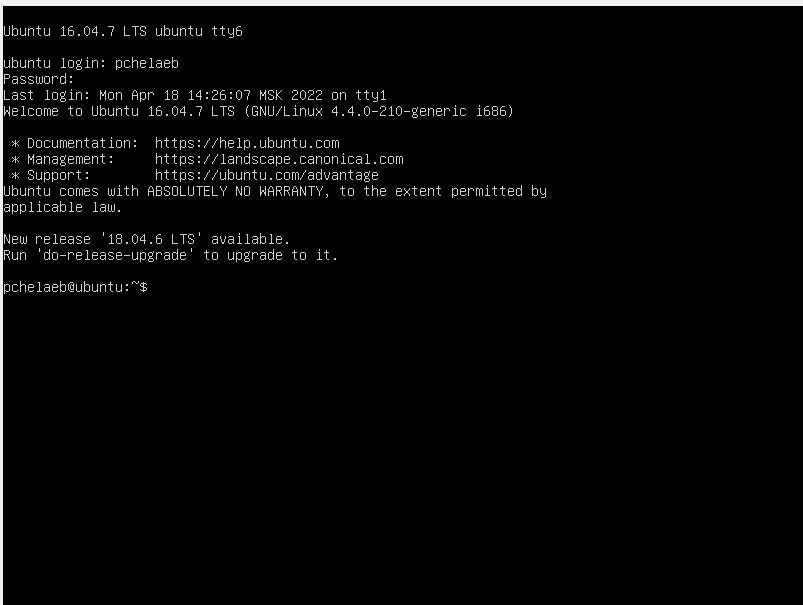
12. Сохраняем изменения по разметке дисков



13. Выбор компонентов системы



14. Авторизуемся под нашим пользователем



**1. Описание функциональных возможностей менеджера виртуальных машин Oracle VM VirtualBox;**

VirtualBox (Oracle VM VirtualBox) — программный продукт [виртуализации](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) для [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [Windows](https://ru.wikipedia.org/wiki/Windows), [Linux](https://ru.wikipedia.org/wiki/Linux), [FreeBSD](https://ru.wikipedia.org/wiki/FreeBSD)[[8]](https://ru.wikipedia.org/wiki/VirtualBox#cite_note-8), [macOS](https://ru.wikipedia.org/wiki/MacOS), [Solaris](https://ru.wikipedia.org/wiki/Solaris)/[OpenSolaris](https://ru.wikipedia.org/wiki/OpenSolaris), [ReactOS](https://ru.wikipedia.org/wiki/ReactOS" \o "ReactOS), [DOS](https://ru.wikipedia.org/wiki/DOS) и других

Кратко перечислим основные возможности VirtualBox:

* **Кроссплатформеность.** VirtualBox выполняется на большом количестве 32-разрядных и 64-битовых операционных систем (Windows, Linux, Mac OS X и Solaris; см. [Section 1.4, “Supported host operating systems”](http://mirspo.narod.ru/vbox/ch01s04.html" \l "hostossupport)). Виртуальные машины могут легко быть импортированы и экспортировали с помощью стандарта Open Virtualization Format (OVF, см.[Section 3.8, “Importing and exporting virtual machines”](http://mirspo.narod.ru/vbox/ch03s08.html)). Так как файл и используемые форматы образов идентичны для всех платформах, что позволяет работать с ними на любых ОС.
* **Прозрачная архитектура; модульность.** VirtualBox является модульным проектом с внутренними интерфейсами программирования и разделением кода сервера и клиента. Это позволяет управлять им несколькими интерфейсами сразу: например, вы можете запустить VM просто, нажимая на кнопку в графическом интерфейсе пользователя VirtualBox, а затем управлять этой машиной из командной строки, или даже удаленно. См.[Chapter 7, Alternative front-ends; remote virtual machines](http://mirspo.narod.ru/vbox/ch07.html) for details.

Благодаря модульной архитектуре, VirtualBox предоставляет полные функциональные возможности и конфигурируемость через набор средств для разработки ПО (SDK), который позволяет вам интегрировать VirtualBox с другими программными системами. См.[Chapter 10, VirtualBox programming interfaces](http://mirspo.narod.ru/vbox/ch10.html) for details.

* **Не требуется аппаратная виртуализация.** Не трубуется никаких аппаратных средств виртуализации. Как было указано в предыдущей главе, в большинстве случаев, VirtualBox не требует возможностей процессора, встроенных в более современные системы, такие как VT-X Intel или AMD-V - в противоположность многим другим подобным решениям,поэтому вы можете использовать VirtualBox даже на старых системах, где эти средства виртуализации не присутствуют.
* **Гостевые дополнения (Guest Additions): общие папки, seamless windows, 3D виртуализация.** Гостевые дополнения VirtualBox - пакет программ, который может быть установлен в поддержываемых гостевых ОС , чтобы улучшить их производительность и обеспечить дополнительную интеграцию и коммуникацию с рабочей системой (с хостом). После установки дополнений виртуальная машина поддержит автоматическую настройку разрешений видео , seamless windows, ускорение трехмерной графики и т.п. Гостевые дополнения одробно описаны в [Chapter 4, Guest Additions](http://mirspo.narod.ru/vbox/ch04.html" \l "guestadditions).

В частности гостевые дополнения дают возможность создавать "общие папки", которые позволяют вам обращаться к файлам системы хоста из гостя. Механизм общих папок описан в [Section 4.6, “Folder sharing”](http://mirspo.narod.ru/vbox/ch04s06.html" \o "4.6. Folder sharing).

* **Большая поддержка аппаратных средств.**Среди прочего, VirtualBox поддерживает:
  + **Полная поддержка ACPI.** Поддержка ACPI полностью реализована VirtualBox, что упрощяет создание образов компьютеров реальных машин или клонирование виртуальных машин сторонних разработчиков в VirtualBox. С уникальной поддержкой состояния питания ACPI VirtualBox может даже сообщить на ACPI-поддерживающие операционные системы гостя о состоянии питания главного компьютера. Для мобильных систем, работающих на батарее, гость может таким образом экономить энергию и уведомляет пользователя о уровне зарядки батарей (например, в полноэкранных режимах).
  + **Поддержка I/O APIC.** VirtualBox имитирует улучшенный программируемый контроллер прерываний ввода/вывода (I/O APIC ),который имеется во многих современных системах.
  + **Поддержка устройств USB.** VirtualBox реализует виртуальный контроллер USB и позволяет вам подключать произвольные устройства USB к своими виртуальными машинами, без необходимости устанавливать драйверы этих устройств на главном компьютере. Поддержка USB не ограничена ни какими видами устройства. Для подробностей, см.[Section 3.7.7.1, “USB settings”](http://mirspo.narod.ru/vbox/ch03s07.html#settings-usb).
  + **Multiscreen resolutions.** VirtualBox virtual machines support screen resolutions many times that of a physical screen, allowing them to be spread over a large number of screens attached to the host system.
  + **Поддержка встроеных iSCSI.** Данная уникальная функция позволяет вам подключать виртуальную машину непосредственно к серверу iSCSI , не используя ресурсы хостовой системы. ВМ обращается к iSCSI непосредственно, без использования дополнительных средств, которые требуется для обеспечения виртуализации жестких дисков в файлах образах VDI. Для подробностей, см.[Section 5.5, “iSCSI servers”](http://mirspo.narod.ru/vbox/ch05s05.html#storage-iscsi).
  + **Сетевая загрузка по PXE.** Виртуальные сетевые карты VirtualBox полностью поддерживают удаленную загрузку через PXE.
* **Множественные снимки системы(Multigeneration snapshots).** VirtualBox может сохранять снимки состояния виртуальной машины. Вы можете вернуть виртуальную машину к любому состоянию ранее сохраненному в снимке. Для подробностей, см. [Section 3.4.4, “Snapshots”](http://mirspo.narod.ru/vbox/ch03s04.html" \l "snapshots" \o "3.4.4. Snapshots).
* **Удаленный доступ по VRDP.** Вы можете работать с любой виртуальной машиной через специальный модуль VirtualBox, который действует как сервер Удаленного рабочего стола (VRDP). Благодоря этой особенности, VirtualBox обеспечивает высокоэффективный удаленный доступ к любой виртуальной машине. Сервер RDP встроен непосредственно на уровене гипервизора и обеспечивает хорошую производительность.

Так же имеется возможность:

* Поддержка виртуализации аудиоустройств (эмуляция [AC’97](https://ru.wikipedia.org/wiki/AC%E2%80%9997) или [Sound Blaster](https://ru.wikipedia.org/wiki/Sound_Blaster) 16 или [Intel HD Audio](https://ru.wikipedia.org/wiki/High_Definition_Audio) на выбор)
* Поддержка различных видов сетевого взаимодействия (NAT, Host Networking via Bridged, Internal)
* Поддержка цепочки сохраненных состояний виртуальной машины (snapshots), к которым может быть произведён откат из любого состояния гостевой системы
* Поддержка Shared Folders для простого обмена файлами между хостовой и гостевой системами (для гостевых систем Windows NT 4.0 и новее, Linux и Solaris)[[21]](https://ru.wikipedia.org/wiki/VirtualBox#cite_note-21)
* Поддержка интеграции рабочих столов (seamless mode) хостовой и гостевой операционной системой
* Поддержка формата [OVF/OVA](https://ru.wikipedia.org/wiki/Open_Virtualization_Format)
* Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс)
* Базовая версия полностью открыта по лицензии [GNU GPL](https://ru.wikipedia.org/wiki/GNU_General_Public_License), соответственно нет ограничений в использовании

**2. Архитектура ядра операционной системы Linux.**

Ядро Linux представляет собой единый блок бинарного кода. Все коды ядра и структуры данных, в том числе драйверы устройств, коды распределения ресурсов и виртуальной памяти, сетевой поддержки, а так же файловая система - хранятся в едином адресном пространстве. Преимуществом такой структуры является то, что не требуется никаких переключений при запросах процессами системных ресурсов или прерываниях от различных устройств.

Общее адресное пространство, однако, не препятствует модульности системы. По мере необходимости Linux подгружает в память либо выгружает из нее указанные модули.  
  
В Linux все ядро работает в привилегированном режиме - режиме ядра. Никакая часть кода не работает в режиме пользователя. Фрагменты поддержки ОС, не требующие запуска в режиме ядра, помещаются в раздел системных библиотек.

**Разделяемые системные библиотеки (system libraries)** содержат стандартный набор функций, используемых приложениями для запросов к системным сервисам ядра. В библиотеках хранятся также код функций отдельных сервисов ядра, исполняемых в обычном режиме без привилегий ядра.



Рис.1 Компоненты Linux

При обращении приложения к системным ресурсам управление от части системы, работающей в пользовательском режиме, передается ядру. Библиотеки осуществляют контроль за корректностью представленного запроса и преобразование параметров/аргументов запроса к требуемому формату.  
  
Под **системными утилитами (system utilities**, программы управления системой) понимают программы, отвечающие за выполнение отдельных специализированных задач управления (*управляющих функций системы*). Одни утилиты запускаются лишь один раз для инициализации и конфигурирования отдельных элементов системы, другие вызываются регулярно, например, утилиты принимающие запросы на регистрацию с терминалов системы, либо утилиты обновляющие файлы регистрации.

Вывод: познакомились с установкой ядра операционной системы Linux