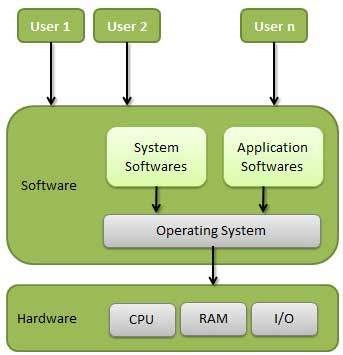
**Зачем нужна операционная система?**



Современная информатика - это множество слоев абстракций. Абстракции позволяют создавать и поддерживать невероятно сложные системы, которые ни один человек не понимает до конца. Работа современного компьютера - это тоже большое количество абстракций. На самом нижнем слое располагаются физические устройства - центральный процессор и различные внешние устройства. На самом верхнем - прикладные программы и пользовательский интерфейс. Если бы при создании прикладных программ нужно было бы напрямую работать с аппаратным обеспечением - создание программ было бы невероятно долгим, скучным и дорогостоящим процессом. Так и было на заре компьютерной техники.

Операционные системы - это средний слой абстракции между пользователем, его программами и “железом”. Любой современный компьютер работает под управлением операционной системы. Здесь под термином “компьютер” мы понимаем и настольный ПК, и сервер и смартфон и даже некоторое носимые и встроенные устройства. Исключение составляют только самые примитивные электронные устройства и встраиваемые компьютеры.

Операционная система - это особая программа, которая загружается и начинает исполняться на компьютере при его включении и позволяет выполнять как бы внутри себя другие программы. Без операционной системы на компьютере могла бы выполняться только одна программа. Так и функционировали компьютеры до появления операционных систем. Современные операционные системы позволяют организовать многозадачность - параллельное выполнение множества программ в пределах одной пользовательской среды. Операционные системы выполняют множество разных задач, облегчающих пользование компьютером для всех категорий пользователей - обычных, работающих в браузере, офисных, работающих с документами, программистов, системных администраторов. Но две функции операционной системы являются базовыми, из которых происходят все остальные.

Операционная система выполняет две главные функции: управление ресурсами и предоставление расширенной машины. Управление ресурсами подразумевает, что именно операционная система управляет доступом всех программ к ресурсам компьютера. Под ресурсами понимается информационное и аппаратное обеспечение. Аппаратное обеспечение компьютера - это в первую очередь центральный процессор и оперативная память. Информационные - это файлы (в широком смысле), расположенные на данном компьютере. Управление доступом к процессору означает, что операционная система выполняет все программы, запущенные на компьютере, переключая их множество раз в секунду, то есть реализует многозадачность. Такая многозадачность может быть даже, если в компьютере присутствует только одно вычислительное ядро. Но чаще таких ядер несколько. И опять же именно ОС распределяет задачи по ядрам, управляет приоритетом и очередью выполнения процессов на компьютере.

Управление доступом к оперативной памяти заключается в том, что при запуске нового процесса операционная система выделяет ему определенный участок в оперативной памяти, куда имеет доступ только этот процесс. Если бы программы могли свободно обращаться к памяти друг друга это было бы очень опасно с точки зрения разделения ответственности. Мы никогда не могли бы быть уверены, что работе программы не помешают другие процессы, злонамеренно или по неосторожности программиста. Управлять таким хаосом было бы невозможно. Именно операционная система изолирует программы друг от друга и таким образом повышает надежность функционирования всей системы в целом.

Точно так же операционная система разделяет между программами доступ ко всем остальным ресурсам - файлам, внешним устройствам, сети. Когда одна программа открывает файл и начинает туда писать она может быть уверена, что посередине этого процесса в запись не вклинится другая программа. Операционная система блокирует файлы при открытии, чтобы с ними мог работать только один процесс одновременно. Если в этот момент другой процесс хочет открыть тот же файл, он должен подождать, когда первый его освободит. За все этим следит определенная подсистема операционной системы.

Для того, чтобы операционная система могла выполнять эту функцию она должна обладать эксклюзивным доступом к устройствам компьютера - от процессора до внешних устройств, таких как жесткий диск или сетевой адаптер. Иначе любая программа могла бы обойти все защитные механизмы ОС и работать с аппаратурой напрямую. Поэтому ядро операционной системы и только оно работает в определенном, привилегированном режиме центрального процессора, когда инструкциям ядра ОС можно выполнять всевозможные действия на компьютере, в том числе для работы с внешними (по отношению к ЦП) устройствами. Остальные программы работают в ограниченном, пользовательском режиме работы. В нем они не имеют доступа к аппаратному обеспечению компьютера, а могут только манипулировать данными внутри данного процесса.

Но как же в таком случае, программы могут открывать файлы (расположенные на жестком диске), посылать документы на печать, общаться по сети и выполнять другие действия? На самом деле, они не могут. Каждый раз, когда программе требуется получить доступ к внешнему ресурсу (аппаратному или информационному), она должна попросить это сделать операционную систему. Для этого в ядре ОС существует набор функций, которые может вызвать любая программа на компьютере. Такие функции называются системными вызовами. Например, существует системный вызов для открытия файла, отправки данных по сети, создания нового процесса и еще множество. Все они образуют так называемый интерфейс системных вызовов операционной системы. Интерфейс системных вызовов определяет, что программы могут делать с использованием этой конкретной операционной системы.

При выполнении системного вызова операционная система в первую очередь проверяет, имеет ли право данная программа выполнить данной действие. Потом - есть ли что-то, что может мешать выполнить его в данный момент (не заблокирован ли нужный файл, например). И только если ОС решит, что все в порядке, она выполнит просьбу программы и вернет ей результат. А если что-то пойдет не так, то ОС вместо результата вернет ошибку. Еще раз подчеркнем, пользовательские программы (а по отношению к операционной системе все программы являются ее пользователями) не могут никак получить доступ к компьютеру, кроме как через механизм системных вызовов.

У каждой операционной системы этот интерфейс может отличаться. Именно поэтому, а еще потому, что формат исполняемых файлов в разных ОС разный, программы пишутся специализированно под конкретную операционную систему. Однако, многие операционные системы следуют стандартному интерфейсу системных вызовов. Этот стандарт называется POSIX. Он регламентирует, какие вызовы должны быть у операционной системы, какие нужно передавать аргументы, как они должны выполняться. Все UNIX-подобные операционные системы, и еще некоторые другие, являются POSIX-совместимыми. Поэтому большинство программ, написанные для одной такой системы могут быть относительно легко перенесены на другую. Именно поэтому мы говорим об этих системах, как о родственных, как о семействе. Отметим, что одно из самых известных семейств несовместимых с POSIX операционных систем - это Windows.

Для своей работы операционные системы оперируют множеством понятий, которые не имеют смысла вне ОС, которые существуют только внутри системы. Мы уже упоминали системные вызовы, процессы внутри ОС. Еще операционные системы организуют информацию в виде привычных уже пользователям файлов и папок, организуют хранение файлов на диске при помощи файловых систем, отслеживают зарегистрированных и авторизованных пользователей, следят за соблюдением пользователями прав доступа. Все это и многое другое - ключевые концепции современных операционных систем.

Выводы:

1. Современная компьютерная техника очень сложна и многообразна.
2. Операционная система - это промежуточный слой между прикладными программами и аппаратным обеспечением.
3. Операционная система - это основная программа компьютера.
4. ОС позволяет запускать другие программы, в том числе - одновременно.
5. ОС распределяет ресурсы - устройства компьютера между программами (процессами).
6. ОС предоставляет универсальный интерфейс для программ к аппаратным устройствам.
7. Только операционная система имеет полный доступ к компьютеру. Программы должны обращаться к ней.
8. Ключевые понятия операционных систем - системные вызовы, процессы, файловые системы, пользователи и права.

**Зачем изучать Linux?**



Сейчас существует большое количество разных операционных систем. Некоторые из них более известны обычному пользователю. Некоторые - используются профессиональными разработчиками, дизайнерами или системными администраторами. Некоторые предназначены для узкоспециализированных задач.

Обычному пользователю более знакомы настольные операционные системы, такие как Windows и MacOS, а также мобильные ОС - Android и iOS. Эти системы в первую очередь предназначены для широкого круга непрофессиональных пользователей, которые выполняют разнообразные задачи - от работы в офисных программах до игр. Но кроме них существуют операционные системы, предназначенные для профессионального использования - для разработки ПО, для организации сетевых сервисов. Чаще всего все операционные системы делят на десктопные и серверные. К ним предъявляются очень разные требования. Десктопные системы предназначены для повседневного использования. Они должны быть в первую очередь, удобные в работе, обладать интуитивным графическим интерфейсом, поддерживать широкий круг пользовательских программ и пользовательских аппаратных средств (принтеров, видеокарт, Wi-Fi адаптеров, мониторов и так далее). Серверные операционные системы должны быть защищенными с точки зрения безопасности, обладать надежной внутренней архитектурой, предсказуемым поведением и интерфейсом, прекрасно работать с сетевыми протоколами, быть многопользовательскими (с сопутствующим разделением доступа). Конечно, это деление условно. Хорошая десктопная система тоже должна быть защищенной, а серверная - удобной. Но исторически такое разделение возникло очень давно и сохраняется до сих пор.

Одной из самых известных профессиональных операционных систем была и остается ОС UNIX. Она была создана одновременно с языком C. У нее было много преимуществ по отношению к тогдашним альтернативам - она была изначально многопользовательской и сетевой и обладала крайне удачной архитектурой. Она имела огромный успех и на ее основе создавались похожие и совместимые с ней другие системы, такие, как например, BSD или Linux. На сегодняшний день, Linux - это одна из самых известных и широко распространенных операционных систем семейства UNIX. Хотя изначально она создавалась для профессионального применения, сейчас ее вполне можно применять и в качестве десктопной.

Но десктопное применение Linux все-таки имеет свои минусы для обычного пользователя. Чаще всего ее применяют именно для серверных нужд. В настоящее время Linux занимает подавляющую долю на рынке серверных ОС. На основе Linux работают большинство серверов в интернете, облачных сервисов, IoT. Если вы в своей профессиональной деятельности столкнетесь с серверным окружением, скорее всего, оно будет работать под управлением какой-то из версий Linux.

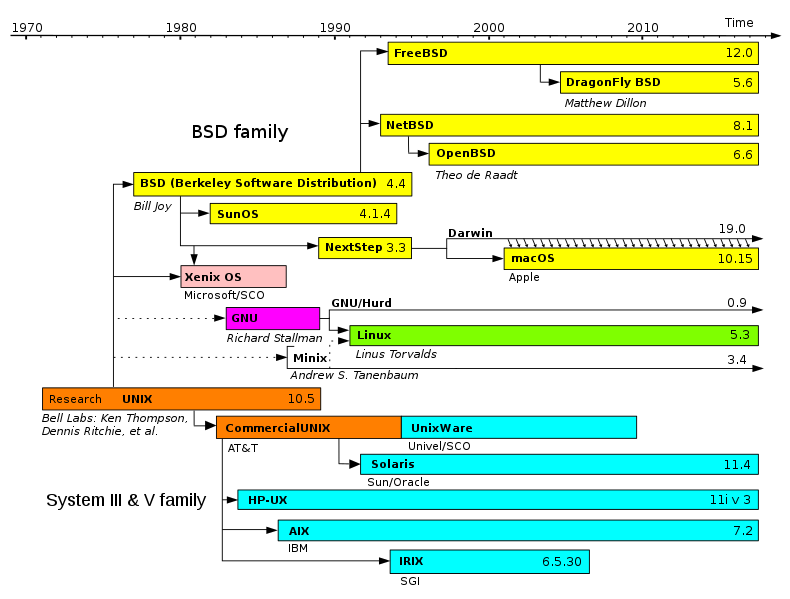
Как потомок UNIX, Linux - POSIX-совместимая операционная система. Вообще, так случилось, что многие концепции, которые изначально были представлены в UNIX, а затем были унаследованы в Linux стали в профессиональной среде стандартными подходами. Например, классическая модель пользователей и разделения прав доступа, стек сетевых технологий (на его основе работает вообще весь Интернет, а для UNIX - он родной), механизм дистрибуции программ, текстовые конфигурационные файлы, стандартный командный интерпретатор и многое другое.

Поэтому почти все профессиональные ИТ-специалисты должны хотя бы в какой-то степени быть знакомыми с этим семейством операционных систем. Знание основ Linux для компьютерщика - один из базовых навыков, неважно, работаете вы программистом, сисадмином, аналитиком данных или девопс-инженером. Кроме непосредственно практической пользы, изучение непривычной операционной системы может принести и косвенные выгоды - вы невольно станете лучше понимать, как функционирует компьютер, как программы выполняются в среде ОС, как работает сама операционная система. А в случае с Linux - вы обязательно улучшите свои навыки автоматизации рутинных задач администрирования, написания скриптов, развертывания информационных окружений, эта система очень располагает к такой деятельности.

Выводы:

1. Существует множество операционных систем. Каждая имеет свое предназначение.
2. Различают десктопные и серверные ОС.
3. Среди операционных систем есть своя генеалогия. Linux относится к UNIX-семейству.
4. Linux - это изначально серверная ОС, которая может использоваться как десктопная.
5. Linux - одна из самых распространенных серверных ОС.
6. Многие механизмы, принятые в Linux стали стандартами в компьютерной технике и сетях.
7. Изучение Linux прокачивает понимание компьютеров и навыки автоматизации.

**Чем Linux отличается от Windows?**



Для начала знакомства с Linux стоит подчеркнуть, в чем ее отличия от самой известной широкому кругу пользователей операционной системы - Windows. Главное отличие - Linux распространяется с открытым исходным кодом. Это, казалось бы, техническое отличие влечет за собой принципиально иную модель распространения ОС и большинства программ для нее.

Многие связывают свободное распространение программ в исходных кодах с бесплатностью, хотя это не связано напрямую. Многие бесплатные программы являются закрытыми, и существуют открытые программы, продающиеся за плату. Но открытые исходные коды подразумевают, что любой пользователь, если он обладает необходимыми навыками, может внести любые изменения в программу и распространять уже свою версию. Это приводит к тому, что большинство свободных программ создаются и распространяются сообществом энтузиастов на некоммерческих основаниях. Кроме того, практически у каждой свободной программы есть так называемые форки - альтернативные варианты, созданные когда любители программы внесли изменения независимо от изначальных разработчиков. Еще это приводит к тому, что для Linux существуют программы практически на все случаи жизни.

Обилие форков не обошло стороной и саму операционную систему. В мире Linux существует большое количество так называемых дистрибутивов. Дело в том, что для прикладного использования ядра операционной системы недостаточно. Кроме него нужны системные и прикладные программы. Каждый любитель может собрать свой собственный набор нужного ему программного обеспечения вместе с ядром ОС и распространять в пакете. Такой пакет и называется дистрибутивом. Этого понятия в принципе не существует в Windows.

Самое заметное обычному пользователю различие между разными операционными системами - несовместимость программного обеспечения. Большинство прикладных программ распространяются в скомпилированном виде - в виде бинарных исполняемых файлов. А компилировать программу нужно под конкретную операционную систему. В большинстве случаев, чтобы программ работала под другой операционной системой нужно вносить существенные изменения в код программы. Поэтому множество программ, особенно прикладных, разрабатывается только для необходимых операционных систем. Поэтому большинство программ, знакомых обычному пользователю по миру Windows в UNIX системах либо не существуют, либо есть только аналоги. В последнее время ситуация сильно исправляется, так как и сама Linux получает все большее распространение, и написание кросс-платформенных программ все больше упрощается.

Кроме того, что набор прикладного ПО существенно отличается, Linux использует централизованную модель дистрибуции программного обеспечения. Это значит, что для установки программы не обязательно искать программу на сайте производителя. В дистрибутив обязательно включается специальная программа - менеджер пакетов. Она устанавливает программы из централизованного источника - репозитория. К системе может быть подключено несколько репозиториев, каждый из которых играет роль каталога программ. Сейчас мы можем повсеместно наблюдать похожую картину в мобильных операционных системах, когда мы пользуемся магазином приложений. И неудивительно, ведь все распространенные мобильные операционные системы - UNIX-подобные.

При такой системе устанавливать распространенные прикладные программы в систему становится гораздо легче. Это происходит, по сути дела, одной командой. Поэтому многие программы установки дистрибутивов включают в себя инструкции по установке определенного количества прикладных программ. Когда вы устанавливаете Linux, вы получаете не просто “голую” систему, как в случае с Windows, а готовую к использованию, включающую большое количество необходимых каждый день программ - офисных пакетов, браузеров, программ для работы с мультимедиа, даже аудио и видео редакторы.

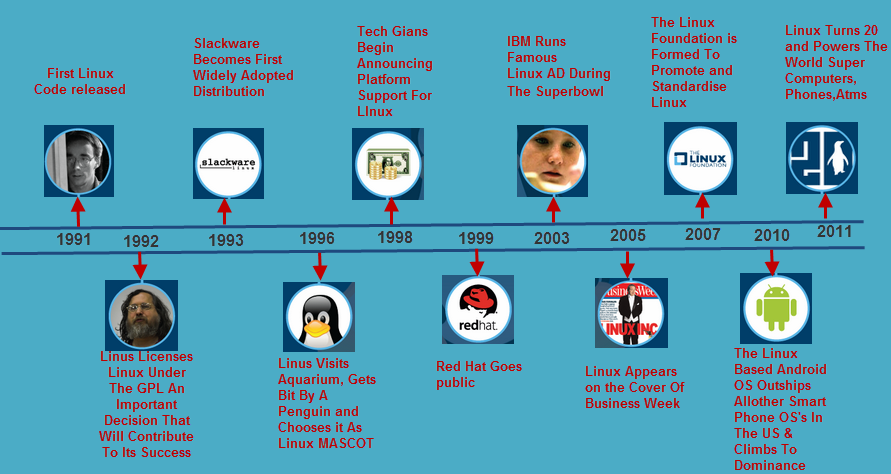
Но Linux отличается от Windows не только набором программ, но и внутренним устройством. Здесь мы не будем вдаваться в детали архитектуры ядра ОС, на эту тему есть большое количество специализированной литературы. Отметим лишь, что на всех уровнях изучения, устройство Linux значительно отличается, ведь эти два семейства разрабатывались полностью (или почти полностью) независимо.

Для пользователя различия во внутреннем устройстве выражаются наиболее четко в одном неожиданном аспекте. Linux - очень текстоцентрична. Она практически полностью ориентирована на работу с текстом. Большинство настроек хранятся в текстовых файлах, а не в бинарном реестре, большинство служебных команд командной строки представляют собой те ил иные манипуляции с текстовыми потоками. Поэтому при изучении Linux так важно научиться работать с текстом и обрабатывать структурированные различными способами текстовые потоки.

Выводы:

1. Linux распространяется свободно, исходный код ОС открыт.
2. Это более важное отличие, чем вы думаете.
3. Существует множество дистрибутивов Linux.
4. У Linux другой набор программного обеспечения.
5. Программы обычно устанавливаются из репозиториев.
6. Дистрибутивы Linux включают много прикладных программ.
7. Внутреннее устройство совершенно другое.
8. Linux ориентирована на работу с текстом.

**Как создавалась Linux?**



История Linux начинается, когда один финский студент, Линус Торвальдс, в конце 80-х годов решил создать собственное ядро операционной системы наподобие UNIX, но бесплатное. В то время система UNIX была наиболее известной и популярной операционной системой для профессионального использования. Главным ее минусом было то, что она распространялась на коммерческой основе и многие, особенно студенты, не могли себе ее позволить.

Торвальдс опубликовал первую версию ядра Linux в 1991 году. Начавшаяся как студенческий проект, Linux быстро завоевала популярность. В то время существовал еще один проект по созданию свободной альтернативы UNIX - проект GNU. Им руководил Ричард Столлман, идейный вдохновитель движения за свободное программное обеспечение. Но они концентрировались на создании набора системного программного обеспечения - интерпретатора командной строки, компилятора языка C, файловых утилит и прочего. Единственное, чего им не хватало до полноценной операционной системы - ядра. Естественно, они довольно быстро объединились с Торвальдсом. Получившаяся в результате система полностью называется GNU/Linux - набор программ GNU, работающий на основе ядра Linux. Но в широких кругах ее называют просто Linux.

С 1991 года Linux как операционная система все больше набирает популярность. Она постоянно поддерживается, до сих пор активно выходят новые версии ядра, добавляющие новые функции и поддерживающие разные компьютерные архитектуры. В разработке ядра Linux принимают участие сотни высококвалифицированных системных программистов со всего мира.

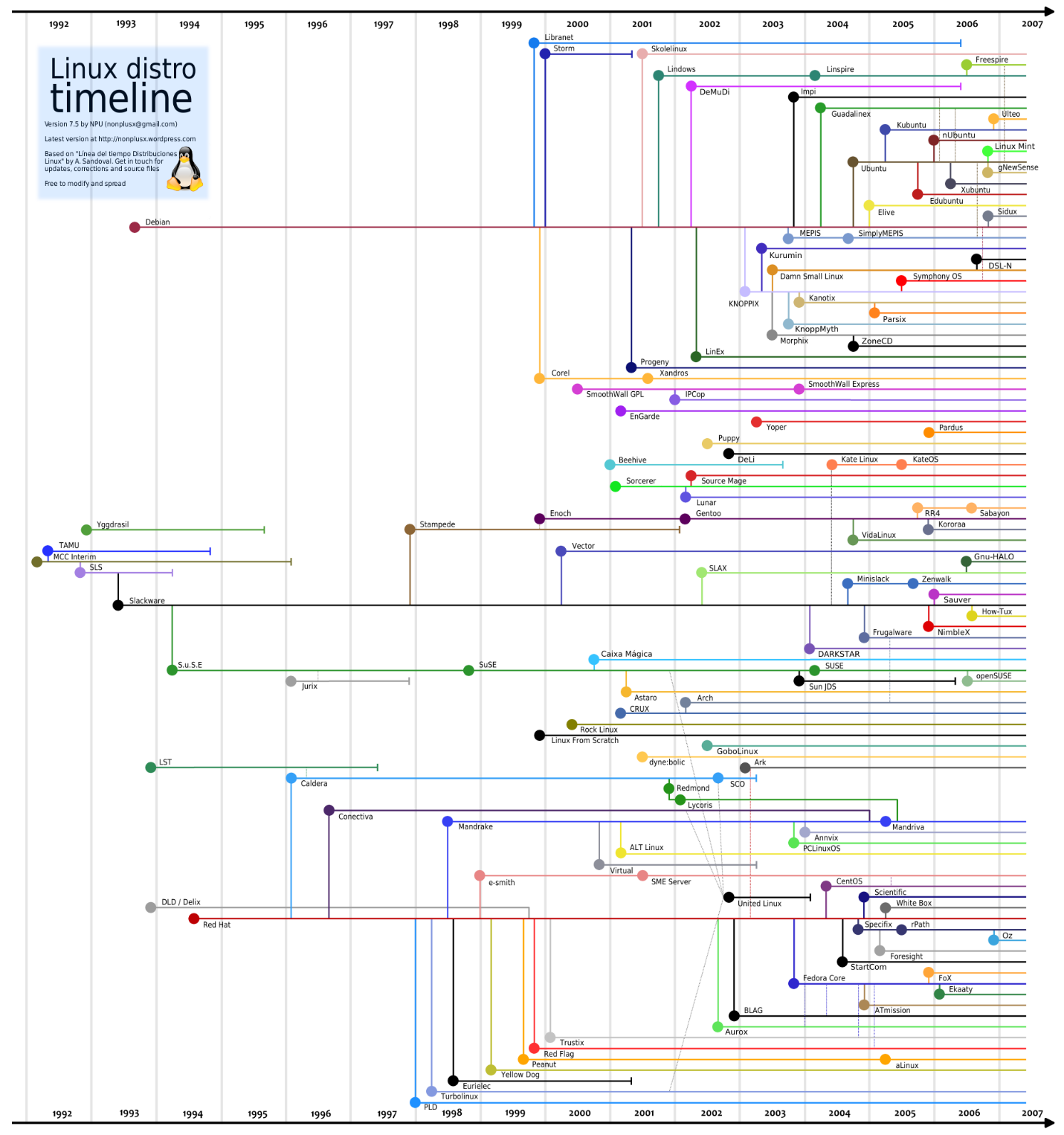
Будучи свободной альтернативой UNIX, при этом полностью с ней совместимой, Linux нашла свою популярность в качестве серверной операционной системы, практически полностью вытеснив с этого рынка саму UNIX и другие подобные системы. Но кроме этого, Linux активно развивается и как десктопная операционная система. Использование графических окружений, расширение набора доступного программного обеспечения, поддержка все большего числа аппаратных средств делают Linux вполне применимой для повседневного использования.

Отдельно отметим, что при этом Linux все еще довольно затруднительно использовать людям, абсолютно не ориентирующимся в работе компьютера. Регулярная нужда в использовании командной строки отпугивает очень многих, поэтому мы не убеждаем вас использовать Linux как замену Windows или MacOS в повседневном режиме. В этом есть как плюсы, так и минусы. И мы изучаем Linux не потому, что считаем ее лучше или хуже других операционных систем. Главная причина изучения Linux для работника сферы IT - широкое ее распространение в профессиональной инфраструктуре, в первую очередь в качестве серверной ОС.

Выводы:

1. Собственно говоря, Linux - это название ядра операционной системы.
2. Кроме ядра в Linux входит программное обеспечения из проекта GNU.
3. Ядро Linux было разработано в 1991 году Линусом Торвальдсом.
4. Linux изначально - это свободная альтернатива UNIX.
5. Сейчас профессионалы используют Linux, а фанатики ждут вендекапец.

**Что такое дистрибутив?**



При знакомстве с Linux пользователь становится перед выбором дистрибутива. В мире Windows такого понятия как дистрибутив не существует, так как эта система разрабатывается и распространяется одной компанией.

Дистрибутив - это совокупность ядра, набора системных и прикладных программ и программы установки, которые скомпонованы, настроены и распространяются в виде, готовом для установки на компьютеры пользователя. По сути, дистрибутив - это что-то вроде версии, или конкретной реализации операционной системы.

Разные дистрибутивы выпускаются разными производителями. Среди них есть как крупные многопрофильные ИТ-компании, такие как Microsoft, IBM, Oracle, так и специализированные разработчики, такие как Canonical, RedHat, FSF, есть даже известные дистрибутивы, разработанные и поддерживаемые энтузиастами и сообществом.

В принципе, выбор дистрибутива - это вопрос удобства и легкого старта. Теоретически, установкой, удалением и настройкой соответствующих компонент можно превратить один дистрибутив в другой. НО мы не рекомендуем этого делать.

Среди множества дистрибутивов легко запутаться. Споры о том, какой дистрибутив лучше для определенных целей - это любимое занятие энтузиастов, но мы не будет об этом говорить. Для того, чтобы иметь какой-то ориентир, нужно понять какой дистрибутив выбрать и по каким критериям.

Первое, что нужно узнать про дистрибутивы - это то, что их можно объединить в семейства. Основные из них - это Debian, RedHat, Gentoo и Arch. Дело в том, что одни дистрибутивы могут разрабатываться на основе других. Это уже упоминавшаяся особенность мира свободного ПО - обилие форков. Если вам удобен какой-то дистрибутив, но хочется в нем поменять что-то, вы вполне можете это сделать и распространять получившийся дистрибутив как производную.

Основное, в чем проявляется родство дистрибутивов - это программные репозитории. Репозиторий по сути - это просто каталог программ, хранящийся на централизованном сервере. Обычно разработчики крупных и основных дистрибутивов составляют и поддерживают такие репозитории специально для своего продукта. Можно провести аналогию с мобильными операционными системами, у Android и iOS свои магазины приложений. Более того, вы не можете подключить AppStore к Android и наоборот, PlayMarket к iPhone. Потому, что у этих систем установка программ, формат пакетов тоже разный. Так и в разных семействах дистрибутивов использующиеся по умолчанию менеджеры пакетов, программы, которые устанавливают пакеты из репозиториев, тоже могут быть разными. Например, во всех дистрибутивах из семейства Debian для установки программ. используется программа apt. В семействе Red Hat используют другую программу - rpm. Поэтому и репозитории этих дистрибутивов разные, и использовать друг друга они не могут.

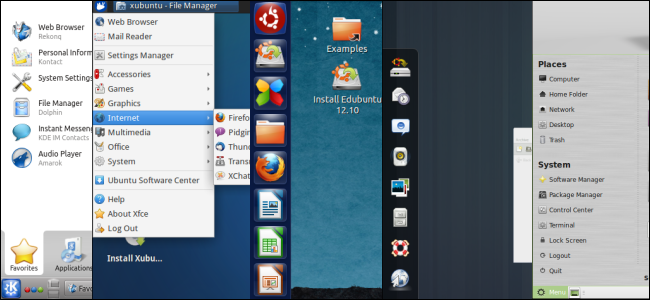
Но даже в пределах одного семейства многие дистрибутивы не имеют своих репозиториев (или имеют маленькие и специализированные), а используют репозитории “старших товарищей”. Так, например, дистрибутив Linux Mint использует набор репозиториев для дистрибутива Ubuntu Linux. Поэтому также можно сказать, что первый в какой-то мере основан на втором.

Но как выбрать из всего этого многообразия дистрибутив для использования так, чтобы потом не приходилось жалеть о сделанном выборе и переделывать много всего? Мы рекомендуем выбирать наиболее известные и популярные дистрибутивы. Так у вас больше шансов получить стабильно работающую систему, с большим коммьюнити, документацие и репозиториями программ. При выборе дистрибутива можно ориентироваться на актуальные рейтинги популярности, которые регулярно обновляются. Практически все популярные дистрибутивы общего назначения подойдут для первоначального освоения этой операционной системы. В дальнейшем в этом курсе мы будет ориентироваться на дистрибутивы из Debian семейства, такие как Ubuntu Linux или Linux Mint.

Выводы::

1. Дистрибутив - это ядро ОС и куча программ в нагрузку.
2. Выбор дистрибутива - это очень сложное, но абсолютно неважное дело.
3. Основные семейства дистрибутивов - Debian, RedHat, Gentoo, Arch, есть еще куча мелких.
4. Новичкам лучше выбрать распространенный и дружелюбный дистрибутив (Ubuntu, Mint).
5. Всегда можно попробовать другой и перейти.

**Что такое графическое окружение?**



Кроме непосредственно дистрибутивов перед пользователями Linux встает проблема выбора графического окружения. Графическое окружение, оболочка рабочего стола - это программа, которая позволяет работать в операционной системе в уже привычной для большинства пользователей оконной среде при помощи мыши и, в меньшей степени, клавиатуры. В мире Windows графическая оболочка является интегрированной и неотъемлемой частью самой операционной системы. А вот для Linux это отдельная программа, которая может и отсутствовать.

Опять же из-за открытого исходного кода программисты могут создавать разные графические окружения. И за десятилетия существования Linux их было создано немало. Некоторые из них более распространены, некоторые - менее. Все дистрибутивы, ориентированные на повседневного пользователя включают в себя одну или даже несколько графических оболочек. Именно они задают вид пользовательского интерфейса - внешний вид окон, меню и панелей, эффекты и анимации, прозрачности и прочие визуальные эффекты. Исторически, наибольшую популярность приобрели две графические среды - KDE и Gnome. Выбор между ними сводится к индивидуальным эстетическим предпочтениям.

Здесь есть одна особенность. При казуальном использовании операционной системы, в отсутствие тяжелых вычислительных процессов, графическая оболочка становится наиболее ресурсоемкой программой. Использование всяческих эффектов приводит к росту требований к оперативной памяти и мощности процессора компьютера. И KDE и Gnome - это сложные полнофункциональные оболочки, которые включают в себя широкие возможности настройки и использования визуальных эффектов. Если ресурсы компьютера ограничены, следует обратить внимание на более легковесные оболочки, которые созданы специально для экономии ресурсов. Они выглядят попроще, но и требований таких не имеют.

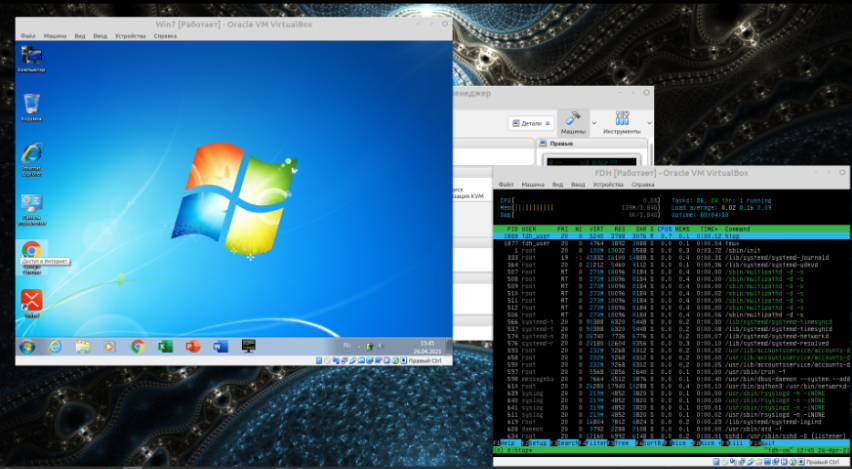
KDE и Gnome - это довольно старые программы, которые активно развиваются и совершенствуются. Кроме непосредственно отображения окон они включают в себя большое количество вспомогательных программ - файловых менеджеров, просмотрщиков мультимедиа файлов, даже собственные офисные пакеты. Так что со временем они превращаются в такие многофункциональные программные комбайны. Другие оболочки лишены этой особенности.

Для профессионального использования вопрос выбора графической оболочки имеет второстепенное значение, так как большую часть задач необходимо будет решать в командной строке, так что как будут выглядеть окна, мелькающие на заднем плане - не очень важно и не влияет на рабочий процесс. В остальном выбор и дистрибутива и графической оболочки остается за вами.

Выводы:

1. Еще одно отличие от Windows - наличие нескольких графических окружений.
2. Существует множество графических оболочек.
3. Самые распространенные - KDE и Gnome.
4. Основное отличие - это внешний вид и наличие эффектов.
5. Хотя, многие оболочки имеют свои собственные пакеты прикладных программ.
6. А еще разные оболочки имеют разные требования к производительность железа.
7. Профессионалы все равно работают в командной строке.

**Зачем пользоваться виртуальной машиной?**



Для работы с операционной системой необходимо ее установить на компьютер. Так как это основная программа компьютера, процесс ее установки сильно отличается от установки обычной прикладной программы. Он связан с переразметкой жесткого диска, освобождением места для новой операционной системы, перезаписью системного загрузчика. И в любом случае, в каждый конкретный момент на компьютере может работать только одна операционная система. Чтобы переключиться в другую, вам нужно перезагрузить компьютер и запустить другую установленную операционную систему. Если вы всего лишь хотите попробовать новую для вас среду, это может быть слишком обязывающим. Более того, в этом процессе легко сделать что-то не так, а ошибки чреваты потерей информации и повреждением существующих на компьютере программ и данных.

Но есть способ избежать лишнего риска и усилий. Существуют специальные программные средства, которые позволяют запускать одну операционную систему внутри другой, как обычную программу. Для этого они эмулируют так называемый виртуальный компьютер (виртуальную машину). А уже с этой виртуальной машиной можно делать все что угодно, в том числе, установить любую операционную систему, без риска навредить основному, физическому компьютеру. Существует несколько таких программ виртуализации одна из самых популярных - бесплатная VirtualBox от Oracle. Ее мы и будем рекомендовать использовать для установки операционной системы Linux.

Виртуальные машины - это очень полезные программы. И не только чтобы проще можно было пробовать новые операционные системы. Многим разработчикам требуется тестировать работу создаваемой программы в разных операционных системах, разных дистрибутивах или версиях. Часто разработчики создают программы для другой операционной системы, нежели установленная на их рабочем компьютере. Поэтому виртуальные машины - незаменимый инструмент разработки и тестирования программного обеспечения. Есть и еще один способ применения. Некоторые программы настолько сложны в установке, что они распространяются в виде готовой виртуальной машины, чтобы их можно было попробовать, или даже развернуть в виде виртуалки в рабочее окружение. Наконец, виртуальные машины - неотъемлемый инструмент облачных технологий. Многие облачные сервисы работают на виртуальных машинах - так гораздо проще создавать новые конфигурации, а при необходимости - удалить машину и все содержимое.

Еще одно преимущество виртуальных машин - вы можете одновременно запустить несколько и переключаться между ними без перезагрузки. Они действительно работают как обычные программы - можно запустить столько процессов, сколько помещается в оперативную память.

Одной из особенностей работы в виртуальной машине является полная изоляция. Все, что происходит в виртуальной машине никак напрямую не влияет на физическую машину (хост). Это очень полезно в плане безопасности. Например, на виртуальной машине можно проверять работу нестабильных или потенциально опасных программ. Кроме того, если вы что-то испортите в виртуальной машине, можно просто начать все заново. А это важно, ведь в процессе обучения мы не должны бояться совершать незнакомые действия.

Но что, если все-таки хочется при работе в виртуальной машине обратиться вовне? Виртуальные машины можно объединять в сеть. Можно даже создать сеть между виртуальной машиной и основной (хостом). Причем на уровне сетевых протоколов не будет видно никакой разницы между работой с виртуальной или физической машиной. Это делает виртуалки незаменимыми для тестирования сетевых приложений или тестирования настроек сети. Конечно, это подразумевает, что вы можете в виртуальной машине иметь доступ в Интернет. Причем, существует много способов подключения виртуальных машин, о которых мы скажем дальше. И можно, например, эмулировать трансляцию сетевых адресов, если вы хотите сделать такое подключение односторонним.

Еще одно преимущество работы с виртуальными машинами по сравнению с реальными - возможность в любой момент сделать резервную копию всей виртуальной машины на диск, вместе со всеми данными и установленными программами. Существуют специальные форматы файлов, в которые можно сохранить полностью виртуальную машину. Конечно, такая копия (образ) машины занимает порядочно места (это сильно зависит, сколько места занято на диске виртуальной машины), но она позволяет еще больше обезопасить себя от каких-либо сбоев. Ведь вы можете создать копию машины в момент ее стабильной работы, а если произошел сбой или ошибка, просто восстановиться из нее и продолжать работать или экспериментировать. Главное, не забывать вовремя делать бекапы.

Но, конечно, у всего есть недостатки. Главный минус виртуальных машин - большое потребление ресурсов компьютера. Чудес не бывает и здесь, и при запуске операционной системы требования по оперативной памяти, процессору и месту на жестком диске никуда не исчезают. Помните, что любая виртуальная машина потребляет ресурсы машины-хоста. Любые программы, выполняющиеся на виртуальной машине в реальности выполняются на физическом процессоре хоста. Так что обмануть процессор и запустить множество тяжелых приложений, которые в нормальном режиме он не тянет, не получится. Кстати, помимо самих программ в виртуальной машине, в этот же момент на процессоре будет выполняться сама основная операционная система, запущенные в ней программы (в том числе сам менеджер виртуальных машин), так что в целом производительность только несколько снижается, но никак не может быть повышена.

То же самое относится и к памяти. Для работы виртуалки вы должны выделить для нее в реальной оперативной памяти необходимый объем. И еще оставить место для основной системы. А если вы хотите запустить сразу несколько виртуалок - то надо, чтобы суммарный объем их памяти был меньше, чем у вас физически в компьютере. То же можно сказать и про объем жесткого диска. Виртуальная машина хранит все свои файлы на реальном жестком диске, в специальном файле (виртуальном жестком диске). Помните, что для комфортной работы операционной системы нужно как минимум несколько десятков гигабайт пространства. И столько свободного места у вас должно быть на физическом диске.

Но в реальности, многочисленные плюсы от использования виртуальных машин перевешивают недостатки. Как и любой инструмент, виртуальные машины нужно использовать для решения определенных задач. Иногда бывает такое, что они не нужны или вредны. Но для того, чтобы протестировать операционную систему без реальной установки, нет способа лучше.

Выводы:

1. Установка операционной системы - это сложный и требовательный процесс.
2. Виртуальные машины позволяют работать с ОС без установки на реальный компьютер.
3. Это сильно упрощает тестирование разных ОС.
4. Можно запускать несколько виртуалок одновременно.
5. Виртуальные машины безопаснее - можно не бояться сделать что-то не так.
6. Между виртуальными машинами можно создавать сети.
7. Можно создавать резервные копии виртуальных машин.
8. Недостаток виртуальных машин - небольшое снижение производительности.

**Что необходимо для установки?**

Давайте рассмотрим процесс установки ОС Linux на виртуальную машину. Для этого на понадобится, во-первых, сам менеджер виртуальных машин. Это программа, которая управляет виртуалками - позволяет создавать, запускать, удалять виртуальные машины, делать их резервные копии и многое другое. Существует несколько альтернативных программ, но мы будем использовать Oracle VirtualBox. Она бесплатная, дружественная к новичками и имеет широкие возможности. Вы ее можете скачать для своей операционной системы с официального сайта.

Во-вторых, нам понадобится дистрибутив операционной системы. В случае с Linux мы тоже можем скачать его с официального сайта. Для обучения мы выбрали дистрибутив Linux Mint. Используем дистрибутив Linux Mint, скачанный заранее, в формате iso.

distro image

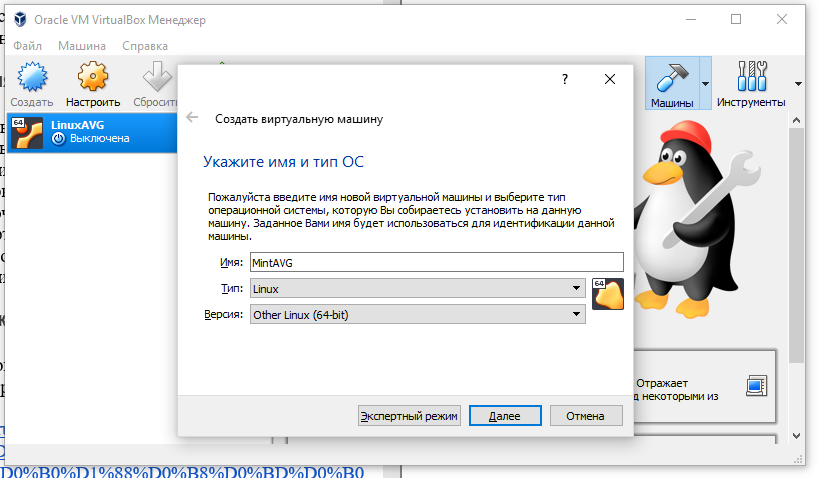
Для работы в виртуальной машине необходимо:

1. Физический компьютер с основной операционной системой (хост).
2. Менеджер виртуальных машин. Рекомендуем использовать Oracle VirtualBox.
3. Дистрибутив нужной (гостевой) операционной системы.
4. Свободное место на физическом жестком диске и оперативная память для работы.
5. Примерно полчаса на установку.

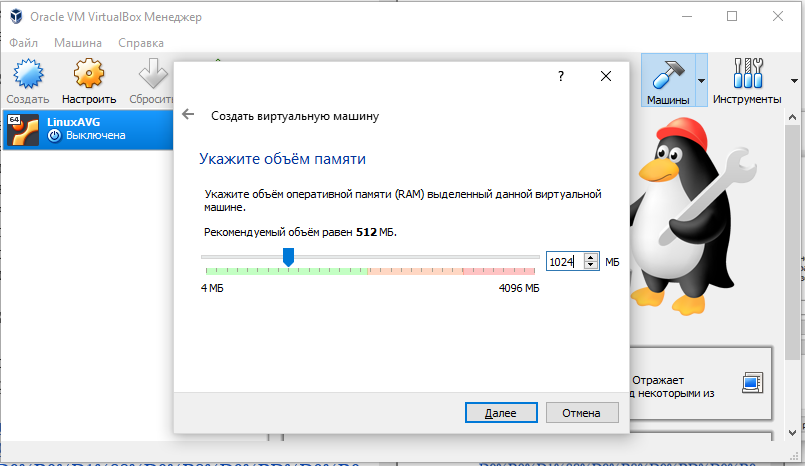
**Как создать виртуальную машину?**

Для начала работы необходимо создать виртуальную машину - то есть программный эмулятор компьютера, в который можно будет установить любую операционную систему и работать с ним изолированно, как с отдельным компьютером.

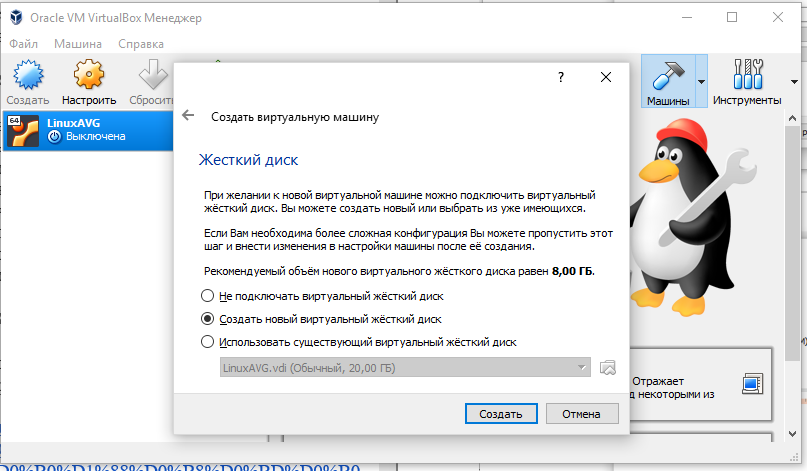
Для этого можно нажать соответствующую кнопку, или выбрать пункт меню. Указываем имя, тип и версию ОС. Имя используется для того, чтобы отличить одну виртуалку от другой, ведь мы можем создать их любое количество.



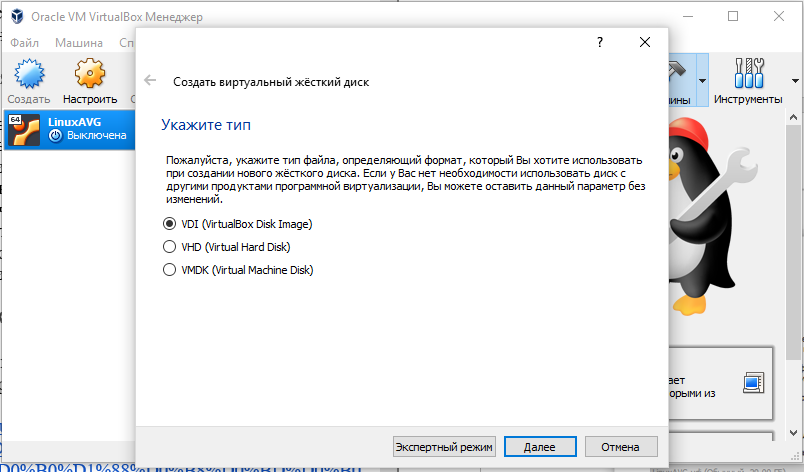
Обратите внимание на версию и разрядность создаваемой ОС. Она должна соответствовать дистрибутиву, который вы собираетесь установить на виртуальную машину. В нашем случае, выбираем Linux в типе. В поле “Версия” обратите особе внимание на разрядность операционной системы. Все современные дистрибутивы являются 64-разрядными. Бывает такое, что в списке вы видите только 32-разрядные версии. В такой виртуалке 64 дистрибутив работать не будет (наоборот, кстати, все работает - вы можете установить 32-битную ОС в 64-разрядную машину). Обычно, это свидетельствует о том, что на вашем компьютере отключена аппаратная виртуализация. Посмотрите, как включить ее в документации к BIOS вашей материнской платы.



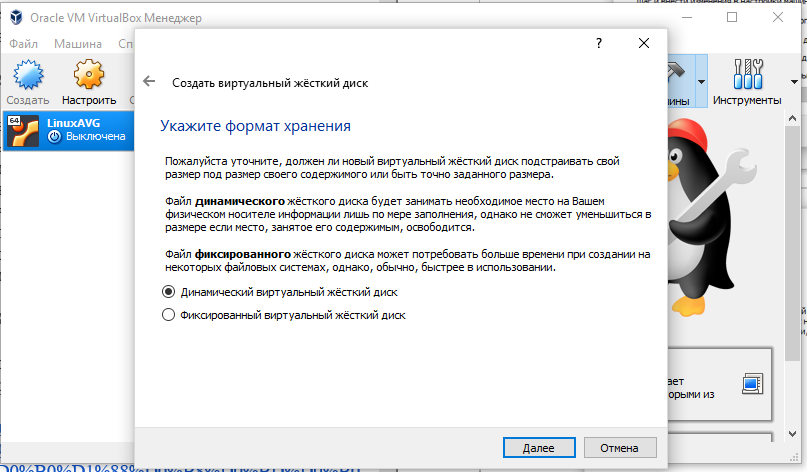
Указываем объем памяти. Рекомендуется использовать не менее 1 ГБ памяти для современных дистрибутивов. Лучше резервировать как можно больше памяти, но не более половины от физического объема реального компьютера. Если вы планируете запускать несколько виртуалок одновременно, резервируйте еще меньше памяти, чтобы они все вместе поместились в существующую оперативку. Некоторые минималистичные дистрибутивы, или системы без графической оболочки требуют гораздо меньше оперативной памяти. Например, серверный Debian 10 можно запустить с 192 МБ памяти.



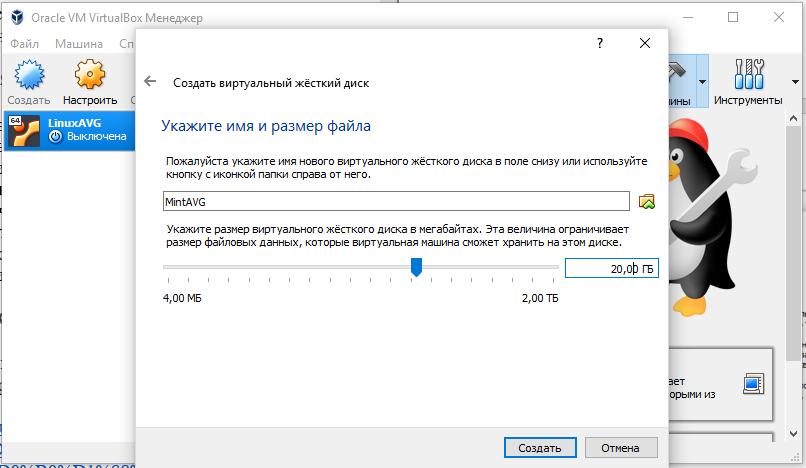
Создаем новый виртуальный жесткий диск. Физически он представляет собой большой файл на вашем жестком диске, а виртуальная машина будет видеть его как отдельный жесткий диск.



Указываем тип жесткого диска. Это определяет формат файла, который будет использоваться для хранения виртуального жесткого диска. Разные программы виртуализации используют разные форматы жестких дисков. Значение по умолчанию можно оставить, если Вы не планируете миграцию ВМ на другие программы виртуализации.

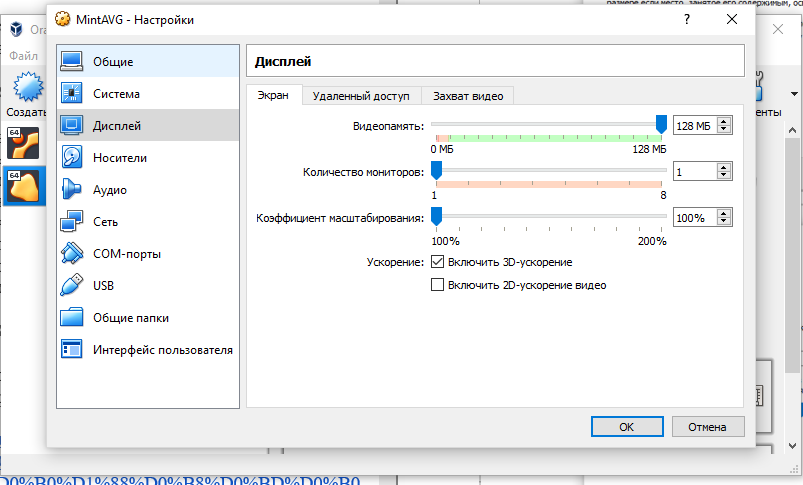


Выбираем формат хранения данных (динамический ВЖД или фиксированный). Фиксированный ВЖД сразу займет выбранный вами объем на диске. Динамический жесткий диск будет занимать столько места, сколько занимают на нем фактически записанные файлы виртуальной машины. Рекомендуется всегда выбирать динамический жесткий диск, так как это сильно экономит место на физической машине.



Указываем размер виртуального жесткого диска и путь его хранения. Если не менять его, то по умолчанию будет выбрана папка, в которой хранятся все виртуальные машины Oracle VM VirtualBox. Обратите внимание на путь к файлу ЖД, если у вас ограничен объем системного раздела. Насчет объема - для современных дистрибутивов достаточно от 20 до 50 ГБ для комфортной работы.

Теперь виртуальная машина создана. Однако, рекомендуется еще выполнить одну операцию. Выделяем созданную виртуалку и находим в интерфейсе пункт “Настройки”. Здесь можно поменять все настройки виртуального аппаратного обеспечения. Нам нужна вкладка “Дисплей”.

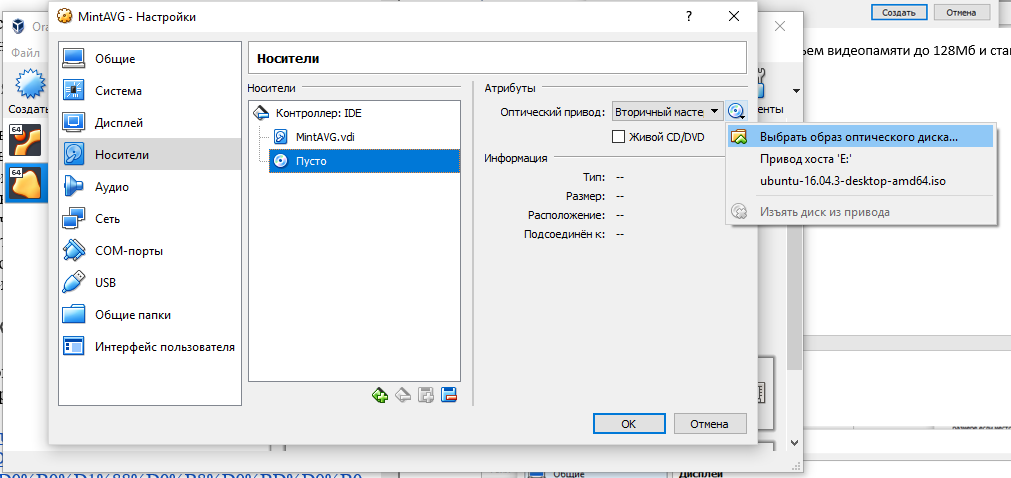


В настройках, в пункте «Дисплей» увеличиваем объем видеопамяти до 128 Мб и ставим галочку на пункте «Включить 3D-ускорение» для лучшей визуализации системы.

**Как установить Linux на виртуальную машину?**

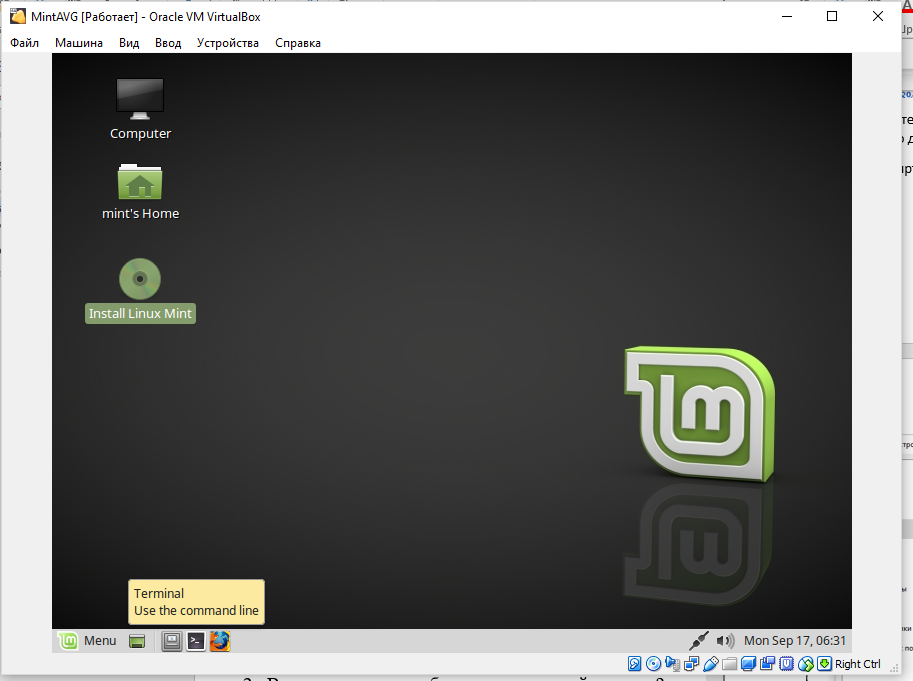
Теперь наша виртуальная машина создана и настроена. Все готово к началу установки в нее операционной системы. Процесс установки полностью аналогичен установке на реальный компьютер, ведь используется тот же дистрибутив и та же установочная программа. Но сначала нужно подключить образ дистрибутива к виртуальной машине.

В пункте «Носители» настроек виртуальной машины выбираем образ оптического диска для подключения образа уже существующего диска. Нам нужно выбрать предварительно скачанный образ дистрибутива в формате iso.



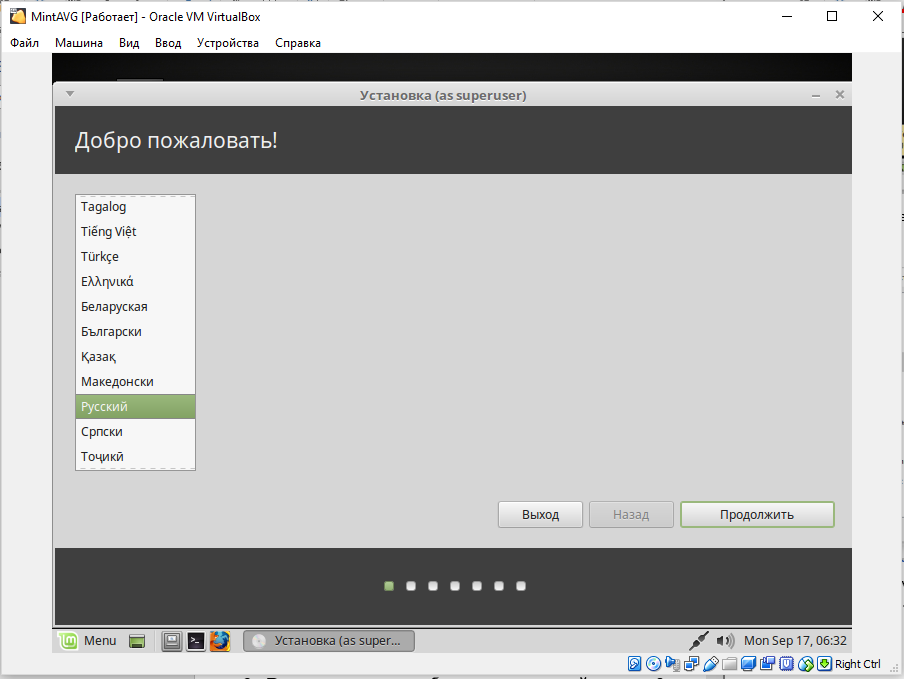
В принципе, это можно сделать и при первом старте виртуалки.

Запускаем виртуальную машину.

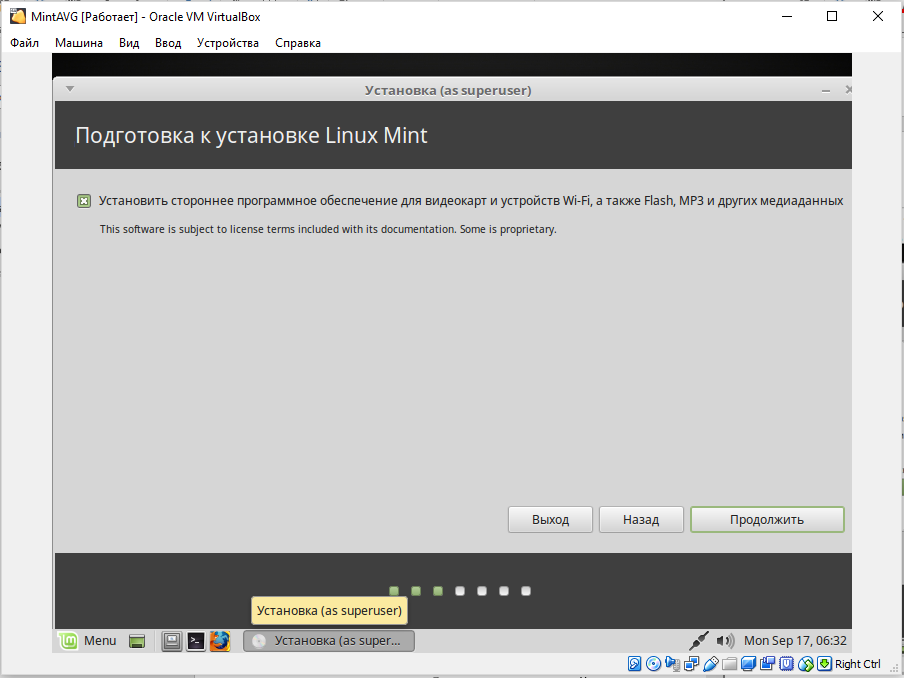


Мы видим окно операционной системы, запущенной в режиме LiveCD. Все современные дистрибутивы позволяют вам опробовать работу системы и без установки, но при первой же перезагрузке все данные, настройки и программы не будут сохранены. Поэтому, на надо именно установить ОС.

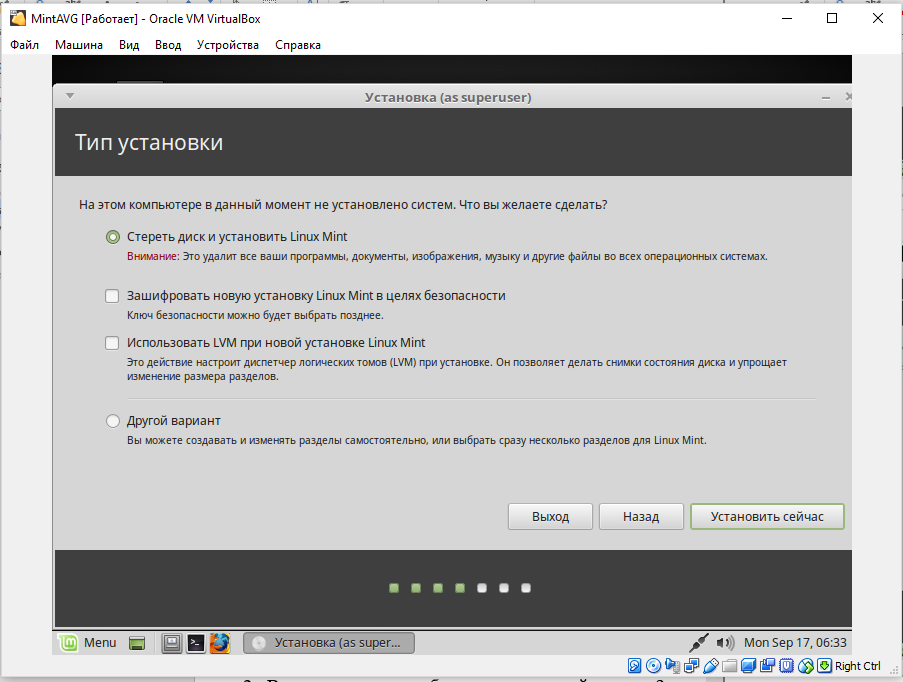
Устанавливаем Linux Mint, выбрав соответствующий ярлык на рабочем столе. Мы увидим программу установки:



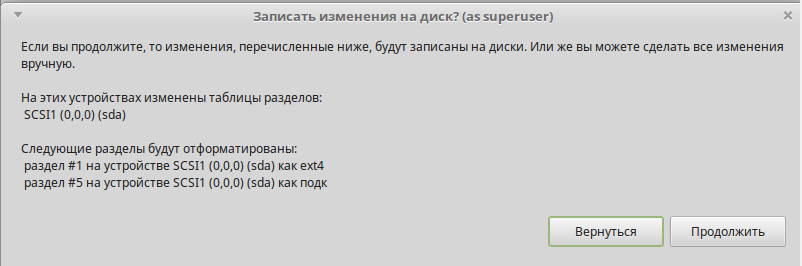
Выбираем язык ОС. Эта настройка влияет и на язык инсталлятора, и на язык системы после установки. Также исходя из выбранного языка будет предложена раскладка клавиатуры.



Устанавливаем стороннее ПО для видеокарты и устройств, для более комфортного использования ОС. Программы установки этой специальной галочкой удостоверяются, что вы хотите установить некоторые полезные компоненты и драйверы (например, MP3), которые не являются свободными программами.

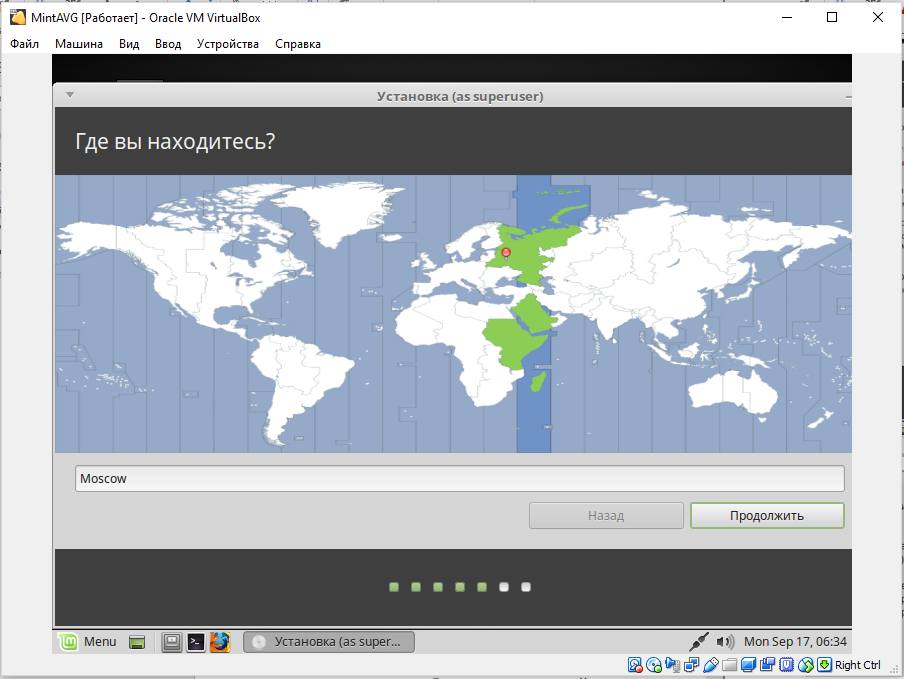


Выбираем, как поступить с жестким диском. Если установщик найдет уже установленные ОС, то предложит установить данную систему рядом с ними. Если вы устанавливаете данный дистрибутив как единственный, можно стереть весь диск.

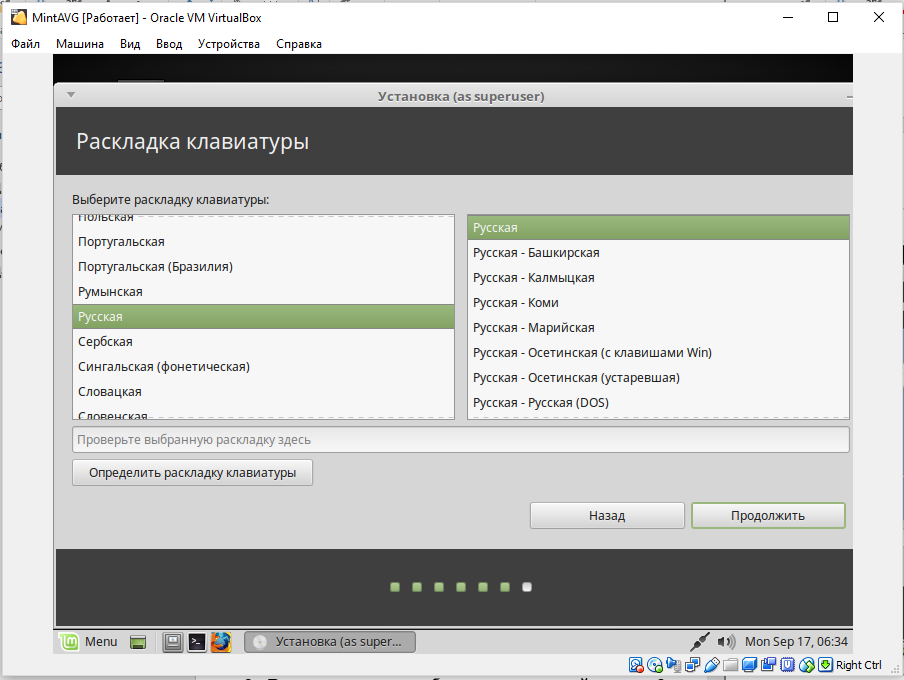


Подтверждаем намерение стереть весь жесткий диск. После этого уже нельзя будет вернуться назад, ведь новая таблица разделов начнет записываться на жесткий диск. Если мы работаем в виртуалке, то можем ничего не бояться, но на реальном компьютере это подразумевает очистку всех данных с этого жесткого диска.

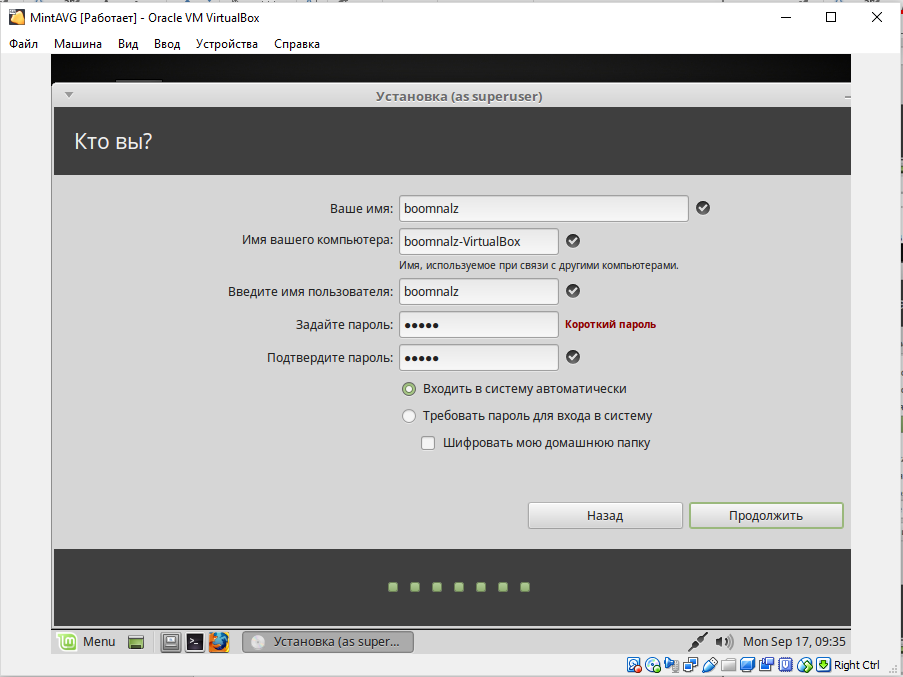
После данного шага программа установки начнет копировать файлы операционной системы на диск. Этот процесс может занять несколько минут. Поэтому в это время программа установки предложит вам внести еще некоторые настройки.



Выбираем часовой пояс в котором мы находимся.



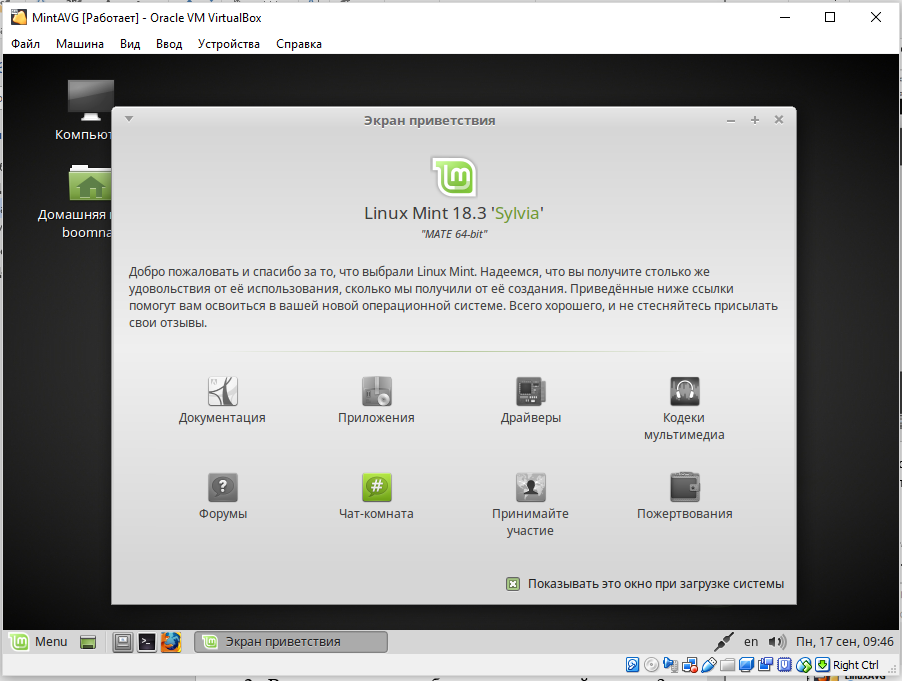
Выбираем необходимую раскладку клавиатуры.



Вносим данные для авторизации как суперпользователь (имя пользователя, пароль), при необходимости отмечаем пункт «Входить в систему автоматически», чтобы не вводить пароль каждый раз. Но это никак не повлияет на безопасность т.к. при выполнении действий, которые могут сильно повлиять на систему, она все равно запросит ввод пароля.

Внимание! Не забудьте пароль суперпользователя! Иначе придется переустанавливать систему с нуля.

После этого система начнет установку, а в конце попросит перезагрузить компьютер для завершения установки.



При включении появится данное окно. Установка успешно завершена!

После этого виртуальная машина перезагрузится и вы сможете начать работу в свежей операционной системе.