|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА  Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра практической и прикладной информатики (ППИ)**

**Реферат**

по дисциплине «Архитектура ВМиС»

**Тема реферата**: «ОС - Linux Mint»

Студент группы ИНБО-03-18 Зубко М.В. (подпись) \_\_\_\_\_\_\_\_

Руководитель Доцент Понаморев А.Н. (подпись) \_\_\_\_\_\_\_\_

Реферат представлен к защите «\_\_»\_\_\_\_\_\_\_\_\_\_ (Подпись) \_\_\_\_\_\_\_\_

**Москва 2020**

**Назначение ОС**

Если надоел Windows, есть отличное решение для вашего компьютера — это программное обеспечение Linux Mint! Его уникальность состоит в легкости и простоте его использования, гораздо проще, чем с Windows 7, 8.х, 10.

На основе Ubuntu Linux (Ubuntu) выпущен софт Linux Mint, используемый в виде источника для пакетов программного обеспечения, в основном для домашнего использования, при этом он является релизом LTS с длительным сроком поддержки. В данном случае пользователю не нужно беспокоиться, что через какой-то период времени вас попросят перейти на новую версию программы.

Linux Mint является абсолютно бесплатным, что в свою очередь дает явное преимущество перед остальным програмным обеспечением, особенно в условиях экономического кризиса. Количество установок софта зависит только от вашего желания и количества компьютеров, которым необходимо программное обеспечение, для этого не существует никаких ограничений.

**История создания**

Самый первый релиз Linux Mint в 2006 году был основан на Kubuntu. Имел версию 1.0 и кодовое имя Ада. Следующий релиз версии 2.0 с кодовым именем Барбара, был первой версией, использующей именно Ubuntu собственные репозитории. У Linux Mint было не так уж много пользователей до третьего релиза 3.0 с именем Кассандра.

В 2008 году Linux Mint перешел на такой же цикл выпуска, как и у Ubuntu и убрали младший номер версии, а для большей совместимости между двумя системами пересмотрели способ построения дистрибутивов. Начиная с версии 6, Фелиссия, каждый релиз использовал полностью кодовую базу последней версии Ubuntu. Он собирался непосредственно на основе Ubuntu и выходил через месяц после релиза Ubuntu.

В 2010 году от Linux Mint ответвилась еще одна версия -  LMDE - Linux Mint Debian Edition. Независимо от версий, основанных на Ubuntu, был основан на Debian и не был привязан ни к пакетам Ubuntu, ни к его графику релизов.

В мае 2015 года команда LinuxMint решила больше не поддерживать исходный код LMDE. Но второго января 2016 вышла новая версия LMDE Second Edition Бетси. Это LTS релиз, основанный на Debian Jessie. После выпуска все пользователи могут обновить программное обеспечение пакета MintTools и версию окружения рабочего стола до самой новой версии раньше чем они появятся в главном издании Linux Mint.

**Структура И Ядро**

Ядро **Linux** бесполезно само по себе — оно является частью одной большой системы, которая целиком является работоспособной. Поэтому, имеет смысле обсудить ядро в контексте всей системы. На изображении ниже отображена схема работы и связи всей системы, в которой участвует ядро (Рис1.)

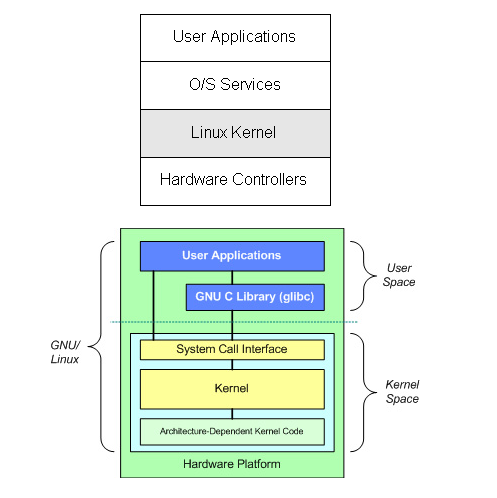


Рис 1. Ядро **Linux**

Операционная система **Linux** (или — [**GNU/Linux**](https://rtfm.co.ua/goto/https:/www.gnu.org/gnu/why-gnu-linux.html)) состоит из четырёх основных частей:

* 1. **User applications** (*пользовательские приложения*) — набор пользовательских приложений может быть разным в различных дистрибутивах **Linux**, но типичные примеры — это приложения обработки текста и веб-браузер;
  2. **O/S services** (службы операционной системы) — службы, которые обычно считаются частью операционной системы, такие как оконный менеджер, командная строка и т.д. Так же, сюда входит программный интерфейс ядра (компилятор/ы и библиотеки);
  3. Ядро **Linux** — основная интересующая нас часть; ядро создаёт абстрактный слой и является «посредником» между первыми двумя частями и hardware-частью компьютера;
  4. **Hardware controllers** (*контроллеры оборудования*) — подсистема, охватывающая все возможные физические устройства, такие как CPU, устройства памяти, жёсткие диски, сетевые карты — все они являются частью этой подсистемы.

Каждая подсистема может взаимодействовать только с двумя соседними, расположенными непосредственно «выше» и «ниже» её уровня. Кроме того, зависимости между этими подсистемами направлены сверху — вниз: слои расположенные выше — зависят от частей ниже, но части, расположенные ниже — не зависят от частей выше их.

**Роль ядра**

Ядро **Linux** представляет собой своего рода виртуальную машину для процессов. Процессы работают без всякой информации про оборудование компьютера — ядро абстрагирует весь уровень оборудования в единый совместимый виртуальный интерфейс. Кроме того, ядро реализует многозадачность прозрачно для всех процессов — каждый процесс «думает», что он является единственным процессом в системе, и имеет полные и эксклюзивные права на память и другие ресурсы оборудования компьютера. Фактически же — ядро выполняет несколько процессов одновременно, и оно ответственно за распределение ресурсов оборудования таким образом, что бы каждый процесс получил достаточный доступ к этим ресурсам.

**Обзор структуры и зависимостей подсистем ядра**

Ядро **Linux** состоит из пяти основных подсистем:

1. **Process Scheduler** (SCHED) — *планировщик процессов*, отвечает за контроль над доступом процессов к CPU. Планировщик обеспечивает такое поведения ядра, при котором все процессы имеют справедливый доступ к центральному процессору.
2. **Memory Manager** (MM) — *менеджер памяти*, обеспечивает различным процессам безопасный доступ к основной памяти системы. Кроме того, MM обеспечивает работу [виртуальной памяти](http://rtfm.co.ua/linux-tipy-pamyati/), которая позволяет процессам использовать больше памяти, чем реально доступно в системе. Выделенная, но неиспользуемая память вытесняется на файловую систему, и при необходимости — возвращается из неё обратно в память (swapping).
3. **Virtual File System**(VFS) — *виртуальная файловая система*, создаёт абстрактный слой, скрывая детали оборудования, предоставляя общий файловый интерфейс для всех устройств. Кроме того, VFS поддерживает несколько форматов файловых систем, которые совместимы с другими операционными системами.
4. **Network Interface** (NET) — *сетевые интерфейсы*, обеспечивает работу с различными сетевыми стандартами и сетевым оборудованием.
5. **Inter-Process Communication** (IPC) — *межпроцессная подсистема*, поддерживающая несколько механизмов для *process-to-process* связей в единой **Linux**-системе.

На рисунке ниже изображены связи всех этих подсистем, где линии исходят от зависимых подсистем к подсистемам, от которых они зависят (Рис2.)

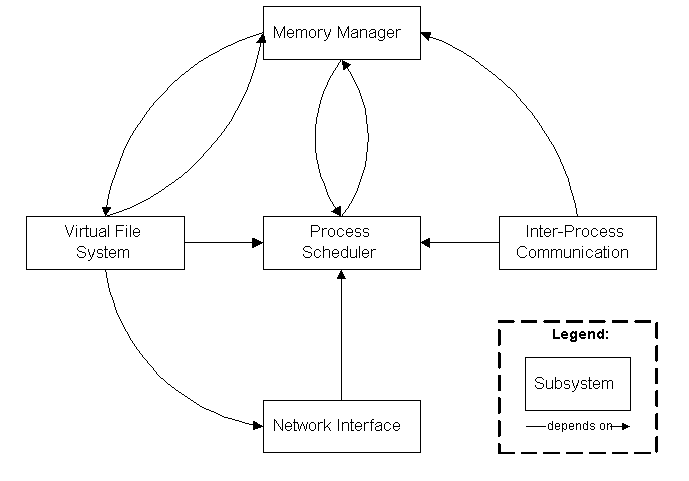


Рис 2.Общая структура и связи подсистем ядра Linux.

Тут явно видно, что основной подсистемой является *планировщик процессов*: все остальные системы зависят от него, так как всем им необходимо приостанавливать и возобновлять выполнение процессов. Как правило, процесс приостанавливается в ожидании каких-либо операций с оборудованием, и возобновляет выполнение, когда операция завершена. Например — когда процесс пытается отправить некое сообщение по сети, сетевой интерфейс может приостановить выполнение процесса, пока сетевое оборудование выполняет отправку сообщения. После того как сообщение отправлено (или — получено сообщение об ошибке) — сетевой интерфейс возобновляет работу процесса, передавая ему код возврата, который указывает на успешное завершение операции, или ошибку её при выполнении. Все остальные подсистемы (*memory manager, virtual file system и inter-process communication*) так же зависят от планировщика процессов по схожим причинам.

Другие зависимости не так явны, но не менее важны:

* *Планировщик процессов* использует менеджер памяти, что бы согласовывать карту памяти для каждого процесса, когда он возобновляет своё выполнение;
* Подсистема IPC зависит от менеджера памяти для поддержания механизма использования общей памяти; этот механизм позволяет двум процессам получить доступ к области общей памяти, в дополнение к их приватной памяти;
* Виртуальная файловая система использует сетевые интерфейсы для обеспечения работы [**NFS**](http://rtfm.co.ua/centos-nastrojka-servera-i-klienta-nfs/) (*Nework File System*), и использует менеджер памяти для обеспечения работы устройства [**RAM drive**](https://rtfm.co.ua/goto/https:/ru.wikipedia.org/wiki/RAM_drive);
* Менеджер памяти использует подсистему виртуальной файловой системы для поддержки swapping-а, и это единственная причина, по которой менеджер памяти зависит от файловой системы; когда процесс запрашивает доступ к участку памяти, который уже был вытеснен на диск (*swapped out*) — менеджер памяти выполняет запрос к файловой системе, что бы получить обратно данные из хранилища (swap-раздела или swap-файла), а выполнение процесса на это время приостанавливается.

Кроме зависимостей, которые видны явно, все подсистемы ядра зависят  от некоторых общих ресурсов, которые не показаны тут. Сюда входят процедуры, которые используют все подсистемы ядра для выделения (*allocate*) им и/или высвобождения ими памяти для использования ядром, процедуры для сообщений об ошибках и проблемах, и прочее.

**Управление памятью**

Базовой единицей в организации памяти для систем UNIX/Linux является страница памяти. Обладающая размером от 4 Кбайт, которому соответствует объём физического пространства в оперативной или виртуальной (область подкачки на диске или другом устройстве хранения) памяти. При запуске процессов, они запрашивают у системы (т. е. у ядра посредством соответствующих системных вызовов) память для своей работы. А в ответ на это ядро выделяет для них достаточное количество страниц памяти. Виртуальная память или как её ещё называют, «резервное ЗУ» (резервное запоминающее устройство) для страниц памяти. Которые содержат, к примеру, исходный текст исполняемого приложения, представляют собой обычные исполняемые файлы на диске. Равно как и для других файлов данных резервным ЗУ являются сами файлы. Информация о том как взаимосвязаны страницы физической и виртуальной памяти хранится в соответствующих таблицах страниц памяти.

Для работы с памятью в Linux (как и в других UNIX-подобных системах) характерно такое явление как «страничный обмен» (paging). Оно заключается в том, что ядро выделяет процессам столько памяти, сколько им необходимо. В том смысле, чтобы её (памяти) всегда хватало. Это достигается за счёт расширения физической памяти за счёт виртуальной, т. е. «подкачки». Поскольку выполнение процессов должно происходить в реальной физической памяти. То ядро постоянно перемещает страницы памяти процессов между физической и виртуальной памятью. Забегая вперёд, следует отметить, что в виртуальной памяти хранятся «неактивные» страницы. Которые не задействованы процессом в данный момент, но необходимые ему для полноценной работы впоследствии.

## Основные принципы управления памятью в Linux

Первое, на что следует обратить внимание, это то, что ядро старается управлять памятью таким образом, чтобы недавно используемые процессом страницы находились в физической памяти. И в свою очередь, «неактивные» или редко используемые страницы перемещаются и хранятся в виртуальной памяти в области «подкачки». Такой механизм распределения памяти называется LRU (least recently used) — замещение наиболее редко используемых страниц.

Вторым важнейшим аспектом в работе памяти является использование кеш-буфера страниц. Это вытекает из работы алгоритма LRU, который довольно сложен в своей реализации. Поскольку следить за всеми обращениями к страницам — это в некоторых случаях, довольно ощутимые потери в производительности системы. Использование же страничного кеш-буфера куда проще в своей реализации при тех же самых результатах. К тому же данный подход имеет огромный модернизационный потенциал (в отличие от LRU) и алгоритмы анализа содержимого кеш-буфера (для определения, какие страницы должны быть перемещены из виртуальной памяти) постоянно совершенствуются. Что заметно сказывается на производительности и эффективности управления памятью.

Когда процессу не хватает памяти, то ядро начинает искать «занятые» страницы. Которые можно использовать для «голодающего» процесса. Обычно такими страницами являются те, что давно не были использованы. Ядро проверяет их на предмет модификации каким-либо процессом. Для этого существуют определённые признаки, при последнем обращении и если изменения были, то такие страницы помечаются ядром как «грязные». Т. е. такие, которые ещё нужны процессам. Для повторного использования памяти такие страницы сначала обязательно переносятся в виртуальную память. Все же остальные страницы являются «чистыми». И поэтому ядро их использует для предоставления другим или «голодающим» процессам.

## Особенности управления памятью в Linux

Когда происходит обращение к страницам памяти, которые некоторое или долгое время не использовалис, т. е. к «неактивным» страницам. То ядро выполняет с ними несколько важных задач:

* возвращает ссылки на эти страницы в соответствующей таблице страниц;
* сбрасывает в нулевое значение время «неиспользования» этих страниц;
* помечает эти страницы как «активные».

Со страницами, находящимися в виртуальной памяти не всё так однозначно. Дело в том, что для того, чтобы «активизировать» такие страницы, они должны быть предварительно прочитаны с диска.

Системное ядро комплектуется специализированными модулями. Которые содержат алгоритмы и даже целые технологии. С помощью которых система довольно эффективно «предсказывает», сколько может потребоваться памяти при разной степени активности и загруженности процессов. Эти алгоритмы имеют своей целью обеспечение процессов свободной памятью с максимальной эффективностью. Т. е. так, чтобы процессам как можно реже приходилось простаивать в «ожидании» выгрузки очередной страницы в свободную память. Таким образом, наблюдая за состоянием страничного обмена во время рабочей нагрузки системы, можно делать выводы о том, нужна ли ей дополнительная память. Если страничный обмен интенсивный — то однозначно следует установить дополнительные модули ОЗУ.

Если же происходит так, что процессам не хватает ни реальной физической, ни виртуальной памяти. Т. е. когда память полностью исчерпана, то система начинает завершать (а точнее уничтожать) целые процессы. Либо запрещает создание новых. Конечно в этом случае в первую очередь уничтожаются наиболее «безболезненные» для системы процессы. Однако в таких случаях даже «на глаз» и по собственным ощущениям видно что она большую часть времени тратит на управление памятью, а не на выполнение рабочих задач.

В Linux можно настроить параметр, который задаёт, насколько быстро ядро должно «отбирать»страницы памяти у процессов. Которым они менее нужны для процессов, которым они на данный момент необходимы. Этот параметр содержится в файле /proc/sys/vm/swappiness и по-умолчанию равен 60. Если задать его меньшим значением (например 0). То ядро будет забирать страницы процесса в самую последнюю очередь. Используя вместо этого любые другие варианты. Если это значение в пределах между 60 и 100. То страницы будут отбираться у процессов с более высокой вероятностью. Вариант с изменением данного параметра на самом деле говорит о том, что необходимо либо снизить нагрузку на систему, адаптировав её для других менее производительных задач, либо увеличить объём ОЗУ.

**Управление вводом и выводом**

Традиционно в ОС UNIX выделяются три типа организации ввода/вывода и, соответственно, три типа драйверов. Блочный ввод/вывод главным образом предназначен для работы с каталогами и обычными файлами файловой системы, которые на базовом уровне имеют блочную структуру. В пп. 2.4.5 и 3.1.2 указывалось, что на пользовательском уровне теперь возможно работать с файлами, прямо отображая их в сегменты виртуальной памяти. Эта возможность рассматривается как верхний уровень блочного ввода/вывода. На нижнем уровне блочный ввод/вывод поддерживается блочными драйверами. Блочный ввод/вывод, кроме того, поддерживается системной буферизацией (см. п. 3.3.1).

Символьный ввод/вывод служит для прямого (без буферизации) выполнения обменов между адресным пространством пользователя и соответствующим устройством. Общей для всех символьных драйверов поддержкой ядра является обеспечение функций пересылки данных между пользовательскими и ядерным адресными пространствами.

Наконец, потоковый ввод/вывод (который мы не будем рассматривать в этом курсе слишком подробно по причине обилия технических деталей) похож на символьный ввод/вывод, но по причине наличия возможности включения в поток промежуточных обрабатывающих модулей обладает существенно большей гибкостью.

**Файловая система**

Операционная система Windows может быть установлена только на файловую систему NTFS, поэтому обычно у пользователей не возникает вопросов какую ФС лучше использовать. Но Linux очень сильно отличается, здесь в ядро системы встроены и могут использоваться несколько файловых систем, каждая из которых оптимизирована для решения определенных задач и лучше подходит именно для них. Чтобы на каждом разделе можно было работать с файлами и каталогами, необходима файловая система. Мы могли бы писать просто содержимое файлов на диск, но нужно еще где-то хранить данные о папках, имена файлов, их размер, адрес на жестком диске, атрибуты доступа. Всем этим занимается файловая система.

От файловой системы зависит очень многое, скорость работы с файлами, скорость записи и даже размер файлов. Также от стабильности файловой системы будет зависеть сохранность ваших файлов.

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. Давайте рассмотрим каждую из них более подробно.

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

В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.

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 сбоев.

ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами.

Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.

XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.

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.

Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

Другие файловые системы, такие как NTFS, FAT, HFS могут использоваться в Linux, но корневая файловая система linux на них не устанавливается, поскольку они для этого не предназначены.

**Особенности**

## Экономное использование памяти

Cinnamon занимает в памяти почти вдвое меньше места, чем Unity. И значительно меньше, чем GNOME. Обладателям мощнейших компьютеров с i7 и 16-гигагабайным ОЗУ об этом заботится не надо. Но если оперативной памяти всего 4 гигабайта, лучше остановиться на Cinnamon.

Linux часто используют в качестве дополнительной системы, устанавливая его на довольно старый компьютер с весьма скромными техническими характеристиками. Разумеется, существуют еще более экономно относящиеся к ресурсами версии, но в случае с Mint речь идет о системе, которая не только бережно относится к ресурсам компьютера, но и обладает великолепным функционалом.

**Достоинства и недостатки**

* **Простота в настройке**

Ubuntu, Fedora и другие дистрибутивы, которые используют окружение Gnome не имеют такого большого количества настроек, как в Cinnamon. Все эти настройки спрятаны и для доступа к ним нужно устанавливать сторонее программное обеспечение. В Linux Mint просто выполнив клик правой кнопкой мышки по любому элементу рабочего стола, вы получите доступ ко множеству его настроек.

Несмотря на то, что Cinnamon основан на Gnome, здесь реализовано намного больше настроек в интерфейсе. Для этой оболочки существует множество тем панелей, приложений и иконок. Вы можете настроить Linux Mint так, как захотите ничего дополнительно не устанавливая.

* **Для LINUX MINT НЕ НУЖЕН МОЩНЫЙ КОМПЬЮТЕР**

Много людей переходят на Linux потому, что новая версия Windows требует больше ресурсов, чем их компьютер может предоставить. Linux менее требователен к ресурсам, однако, некоторым оболочкам всё же необходимо много оперативной памяти и вычислительной мощности процессора. С Linux Mint даже старый компьютер будет хорошо работать.

Это не значит, что в Linux Mint используется устаревшее программное обеспечение. Наоборот, вы будете получать самые свежие версии веб-браузеров, офисных пактов, мультимедийных программ и других продуктов. Вы можете вдохнуть жизнь в старую машину, которую производители уже рекомендуют заменить.

**Недостатки**

* **Консоль и множество команд**

Действительно, новичку для выполнения каких-то действий очень часто придётся использовать терминал. Некоторые могут возразить, что в современных дистрибутивах консоль не нужна, там и так есть графический интерфейс, которого для работы вполне достаточно. К сожалению, я не согласен с таким мнением. В любом случае вы будете использовать команды.

* **Проблема с установкой принтеров**

Эта проблема очень частая для пользователей Linux. Многие модели принтеров попросту не работают, в итоге приходится думать, искать решения, чтобы настроить всё, как положено. Часто возникают проблему с принтерами Canon. Уже 2020 год, а разработчики всё никак не могут сделать оптимальные условия для работы с данным видом устройств.

* **Не работает Wi-Fi**

Точнее сказать, Linux работает не со всеми моделями адаптеров. Со многими возникают проблемы, поэтому приходится изощриться так, чтобы нужный драйвер для устройства встал как надо. В Windows такой проблемы нет, подключил устройство, установил драйвера и ПО из коробки и всё работает.

**Вывод**

Это далеко не вся информация о Linx Mint, и о Linux в целом. Данный доклад был только кратким обзором некоторых аспектов, связанных с этой ОС. Несмомненно, в Linux множество как своих достоинств, так и недостатков. Linx Mint является очень гибкой и нетребовательной системой, что позволяет оптимально её использовать под большое число задач. Однако, что-бы выбрать систему «под себя» идеально - нужно хорошо понимать для каких целей вы собираетесь её использовать, и тогда вы точно сможете выбрать идеальную ОС.