**Установка (на примере любого пакетного менеджера) и запуск приложений.**

Под пакетами в Linux подразумевается программное обеспечение (ПО), которое Вы хотите установить на компьютер

Пакеты бывают двух видов:

* бинарные (архив с расширением .deb. Cодержит исполняемые файлы, файлы настроек, справочные страницы)
* с исходным кодом (архив с расширением .src, содержащий исходный код и описание)

Пакеты хранят в себе исполняемые файлы, документацию, файлы настроек

**Зависимости:**

В Linux часто бывает, что один пакет при установке будет требовать какой-либо другой установленный пакет для его корректной работы. В системе пакетов Debian есть несколько типов «зависимостей» пакетов друг от друга, задуманных для определения степени независимости одной программы (например, А) от наличия в данной системе другой (Б):

* Пакет A **зависит** (Depends) от пакета Б, если Б абсолютно необходим для работы A. В некоторых случаях A не просто зависит от Б, но дополнительно требует определённую версию Б. В этом случае обычно накладывается требование, чтобы версия Б была не ниже заданной.
* Пакет A **рекомендует** (Recommends) пакет Б, если сопровождающий пакета считает, что большинство пользователей не захотят пользоваться A, не имея функциональности, предоставляемой пакетом Б.
* Пакет A **предлагает** (Suggests) пакет Б, если Б содержит файлы, имеющие отношение к функциональности пакета A (и обычно её расширяющие).
* Пакет A **конфликтует** (Conflicts) с пакетом Б, когда A не может работать, если установлен пакет Б. Наиболее часто конфликты возникают, когда A содержит усовершенствованные версии файлов, содержащихся в Б. «Конфликтует» часто задаётся вместе с «заменяет».
* Пакет A **заменяет** (Replaces) пакет Б, когда файлы, установленные из пакета Б, удаляются и (в некоторых случаях) замещаются файлами из A.
* Пакет А **ломает** (Breaks) пакет Б, когда нельзя одновременно настроить оба пакета в системе. Система управления пакетами предотвратит установку одного, если в системе уже установлен и настроен другой.
* Пакет A **предоставляет** (Provides) пакет Б, когда все файлы и функциональность Б имеются в A. Этот механизм позволяет пользователям с ограниченным дисковым пространством получить только ту часть пакета А, которая действительно им нужна.

**Репозитории**

**Репозиторий** - это место в сети интернет, где хранятся какие-либо данные. Репозиторий операционной системы линукс - это место где хранятся программы (пакеты) этой операционной системы. В репозитории содержаться практически все необходимые Вам программы, здесь же содержаться и их зависимости (пакеты, которые необходимы для работы других пакетов).

/etc/apt/sources.list.d – тут хранятся репозитории добавленные с помощью add-apt

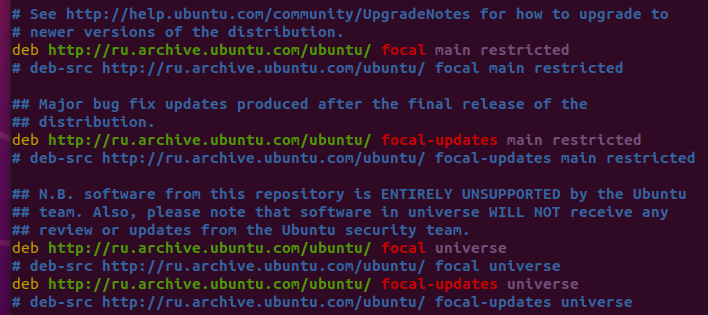
/etc/apt/sources.list – тут хранятся репозитории добавленные мануально

В Ubuntu есть 3 возможности управления репозиториями:

* Через графический интерфейс менеджеров пакетов
* Редактированием файла /etc/apt/sources.list
* Через консоль – используя утилиту add-apt-repository

**1. sources.list**

Файл sources.list расположен в **/etc/apt/sources.list** и может представлять собой следующую структуру:



Каждая отдельная строчка в файле – это указание на репозиторий с архивами.

Формат записи репозитория:

**<type>-<url>-< release >-<component1>-<component2>**

1. **Type**: бывает двух видов – deb и deb-src: **deb** – тип архива хранящий бинарные deb пакеты. **deb-src** – архив пакетов с иходным кодом, который содержит оригинальные исходные коды.
2. **URL** - URL репозитория с которого вы хотите скачивать пакеты.
3. **Release** – кодовое имя релиза.
4. **Component** - В Ubuntu всё программное обеспечение делится на четыре секции, называемые компонентами, чтобы отразить разницу в лицензии и уровне доступной поддержки.

* **Main** – свободное ПО, официально поддерживаемое компанией Canonical.
* **Restricted** – проприетарное ПО (в основном — драйверы устройств), официально поддерживаемое компанией Canonical.
* **Universe** – свободное ПО, официально не поддерживаемое компанией Canonical (но поддерживаемое сообществом пользователей).
* **Multiverse** – проприетарное ПО, неподдерживаемое компанией Canonical.

Существует четыре основных репозитория Ubuntu.

1. **$release** – это пакеты на момент выхода релиза.
2. **$release-security** – пакеты критических обновлений безопасности.
3. **$release-updates** – пакеты обновления системы (т.е. более поздние версии ПО, вышедшие уже после релиза).
4. **$release-backports** – бэкпорты более новых версий некоторого ПО, которое доступно только в нестабильных версиях Ubuntu.
5. **partner** – репозиторий содержищий ПО компаний-партнеров Canonical.

**Настройка sources.list**

**sudo nano /etc/apt/source.list** – открываем файл с репозиториями

deb http://nginx.org/packages/ubuntu/ bionic nginx - добавляем туда APT строку. Чем «выше» (т.е. ближе к началу файла) стоит строка, тем больший приоритет получит добавленный репозиторий

**2. Используя утилиту add-apt-repository**

**Что такое PPA?**

Официальные репозитории Ubuntu хоть и содержат огромное количество пакетов, труднодоступны для обычных разработчиков, в них сложно поддерживать актуальные версии своей программы, да и вообще, не каждый проект имеет шанс туда попасть. Для создания собственного репозитория необходимы ресурсы, как минимум, сервер, и неплохие навыки администратора. Всё это создаёт некоторые препятствия для распространения программ для Ubuntu.

Поэтому для поддержки и развития свободных проектов был создан сервис [launchpad.net](http://launchpad.net/). С помощью этого сервиса можно делать много чего, но одно из его ключевых достоинств для обычного пользователя - возможность любому создать свой репозиторий, который в рамках launchpad называет персональный архив пакетов (Personal Package Arhive - PPA), и использовать его для распространения своих пакетов без лишних технических трудностей.

Утилита **add-apt-repository** позволяет добавить репозиторий в файл**/etc/apt/source.list** или в **/etc/apt/source.list.d/**

**sudo add-apt-repository ppa:**репозиторий/ppa – добавить репозиторий из Launchpad.net

**ls /etc/apt/sources.list.d** – проверяем что репозиторий добавлен

**sudo add-apt-repository –r** **ppa:**репозиторий/ppa – удаление репозитория

**Пакетный менеджер**

В Linux существует различное множество пакетных менеджеров – утилит, предназначенных для управлением пакетами в система. dpkg - это пакетный менеджер для Debian систем. Он может устанавливать, удалять и создавать пакеты, но, в отличие от других систем управления пакетами, он не может автоматически загружать и устанавливать пакеты или их зависимости.

Advanced Packaging Tool – набор утилит для управления программными пакетами в операционных системах основанных на Debian. APT менеджер решает зависимости

Одна из утилит - apt-get. Это утилита управления пакетами. apt-get требует прав суперпользователя для своей работы (для предоставления пользователю привилегий суперпользователя root использовать sudo).

Есть еще rpm, yum пакетные менеджеры

Основные команды APT, DPKG:

**sudo add-apt-repository ppa:**репозиторий/ppa - добавление репозитория

**sudo add-apt-repository -r ppa:**репозиторий/ppa – удаление репозитория

**sudo apt-cache search** имя\_пакета -поиск как называется прилага в репозитории

**sudo apt-get update** – обновляет индексы пакетов. Обновление базы обновлений. Ничего не устанавливает

**sudo apt-get upgrade** – установить обновления

**sudo apt dist-upgrade** – установить обновление с удалением старых версий

**sudo apt-get download** имя\_пакета – загрузка пакета без установки

**sudo apt-get install** имя\_пакета – скачать приложение

**sudo apt-get install** ./файл\_установки – установить приложение из текущей папки

**sudo apt-get remove** имя\_пакета – удаляет приложение, но не его зависимости или конфигурационные файлы

**sudo apt-get purge** имя\_пакета – удаляет приложение, конфигурационные файлы, но не зависимости

**sudo apt-get autoremove** – удаляет неиспользуемые зависимости (делать после удаления приложения)

**dpkg -l** – получение списка пакетов в системе

**dpkg -L** имя\_пакета – список файлов пакета в системе

**dpkg -S** путь\_к\_файлу – какому пакету принадлежит файл

**dpkg -c** пакет – просмотр содержимого пакета

**dpkg –s** имя\_пакета – проверить установлена ли запрашиваемая программа. Dpkg (система управления файлами deb расширения, только в debian)

**dpkg -i** имя\_пакета – установить пакет формата .deb

**dpkg -r** имя\_пакета – удаление пакета

**dpkg --extract** имя\_файла путь\_папки – распаковать .deb файл

**Git clone** (github…) – скачать (клонировать) утилиту с сайта github. Git – система управления версиями, ветвь разработки приложения. Github – всемирная сеть, ветвь, разработчиков, куда они выкладывают свои приложения, утилиты.

Что бы запустить приложение, достаточно ввести его название в терминале

**Snap пакеты**

snap пакеты содержат саму программу, а также все ее зависимости. Это имеет несколько преимуществ перед обычными deb или rpm пакетами, обрабатывающими зависимости. А главное, из них - разработчик может быть уверен что нет никаких регрессий из-за изменений версий библиотек в системе.

**snap list** – список всех установленных snap пакетов (Установленные snap пакеты не отображаются в выводе dpkg -l)

**snap find** имя\_пакета – найти пакет в snapstore

**snap install** имя\_пакета – установка пакета

**snap refresh** имя\_пакета – обновить пакет/пакеты

**snap remove** имя\_пакета – удаление пакета

**Desktop Environment**

desktop environment – настройки, приложения и тд

windows manager – отрисовка рабочего стола